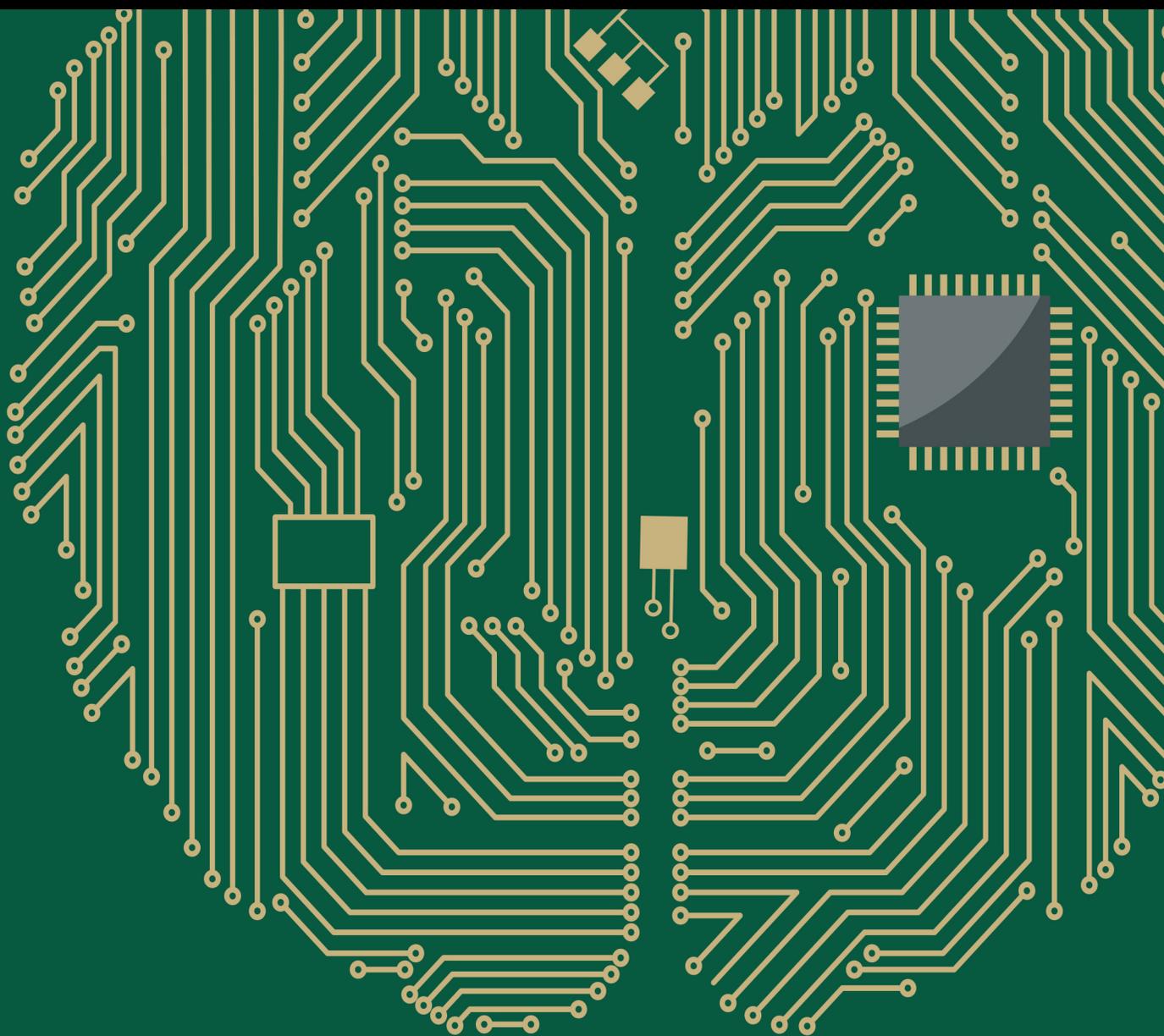


Emerging Trends in Machine Learning for Signal Processing

Lead Guest Editor: George A. Papakostas

Guest Editors: Konstantinos I. Diamantaras and Francesco A. N. Palmieri





Emerging Trends in Machine Learning for Signal Processing

Computational Intelligence and Neuroscience

Emerging Trends in Machine Learning for Signal Processing

Lead Guest Editor: George A. Papakostas

Guest Editors: Konstantinos I. Diamantaras
and Francesco A. N. Palmieri



Copyright © 2017 Hindawi. All rights reserved.

This is a special issue published in “Computational Intelligence and Neuroscience.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editorial Board

Ricardo Aler, Spain
Amparo Alonso-Betanzos, Spain
Diego Andina, Spain
Pietro Aricò, Italy
Hasan Ayaz, USA
Sylvain Baillet, Canada
Theodore W. Berger, USA
Vince D. Calhoun, USA
Francesco Camastra, Italy
Michela Chiappalone, Italy
Andrzej Cichocki, Japan
Jens Christian Claussen, Germany
Silvia Conforto, Italy
Justin Dauwels, Singapore
Christian W. Dawson, UK
Paolo Del Giudice, Italy
Maria Jose del Jesus, Spain
Thomas DeMarse, USA
Piotr Franaszczuk, USA
Leonardo Franco, Spain
Paolo Gastaldo, Italy
Samanwoy Ghosh-Dastidar, USA

Manuel Graña, Spain
Rodolfo H. Guerra, Spain
Pedro Antonio Gutierrez, Spain
Dominic Heger, Germany
Stephen Helms Tillery, USA
J. A. Hernández-Pérez, Mexico
Luis Javier Herrera, Spain
Etienne Hugues, USA
Pasi A. Karjalainen, Finland
Raşit Köker, Turkey
Dean J. Krusienski, USA
Fabio La Foresta, Italy
Mikhail A. Lebedev, USA
Cheng-Jian Lin, Taiwan
Ezequiel López-Rubio, Spain
Reinoud Maex, France
Kezhi Mao, Singapore
José David Martín-Guerrero, Spain
Sergio Martinoia, Italy
Elio Masciari, Italy
Paolo Massobrio, Italy
Michele Migliore, Italy

Klaus Obermayer, Germany
Karim G. Oweiss, USA
Massimo Panella, Italy
Fivos Panetsos, Spain
Sandhya Samarasinghe, New Zealand
Saeid Sanei, UK
Michael Schmuker, UK
Friedhelm Schwenker, Germany
Sergio Solinas, Italy
Toshihisa Tanaka, Japan
Jussi Tohka, Spain
Carlos M. Travieso-González, Spain
Lefteri Tsoukalas, USA
Marc Van Hulle, Belgium
Pablo Varona, Spain
Meel Velliste, USA
Francois B. Vialatte, France
Thomas Villmann, Germany
Ivan Volosyak, Germany
Michal Zochowski, USA
Rodolfo Zunino, Italy

Contents

Emerging Trends in Machine Learning for Signal Processing

George A. Papakostas, Konstantinos I. Diamantaras, and Francesco A. N. Palmieri
Volume 2017, Article ID 6521367, 2 pages

Multimodal Personal Verification Using Likelihood Ratio for the Match Score Fusion

Long Binh Tran and Thai Hoang Le
Volume 2017, Article ID 9345969, 9 pages

Stacked Autoencoders for Outlier Detection in Over-the-Horizon Radar Signals

Eftychios Protopapadakis, Athanasios Voulodimos, Anastasios Doulamis,
Nikolaos Doulamis, Dimitrios Dres, and Matthaïos Bimpas
Volume 2017, Article ID 5891417, 11 pages

Decoding of Human Movements Based on Deep Brain Local Field Potentials Using Ensemble Neural Networks

Mohammad S. Islam, Khondaker A. Mamun, and Hai Deng
Volume 2017, Article ID 5151895, 16 pages

Pathological Brain Detection Using Weiner Filtering, 2D-Discrete Wavelet Transform, Probabilistic PCA, and Random Subspace Ensemble Classifier

Debesh Jha, Ji-In Kim, Moo-Rak Choi, and Goo-Rak Kwon
Volume 2017, Article ID 4205141, 11 pages

A Novel Active Semisupervised Convolutional Neural Network Algorithm for SAR Image Recognition

Fei Gao, Zhenyu Yue, Jun Wang, Jinping Sun, Erfu Yang, and Huiyu Zhou
Volume 2017, Article ID 3105053, 8 pages

Application of the Intuitionistic Fuzzy InterCriteria Analysis Method with Triples to a Neural Network Preprocessing Procedure

Sotir Sotirov, Vassia Atanassova, Evdokia Sotirova, Lyubka Doukovska, Veselina Bureva, Deyan Mavrov,
and Jivko Tomov
Volume 2017, Article ID 2157852, 9 pages

Patch-Based Principal Component Analysis for Face Recognition

Tai-Xiang Jiang, Ting-Zhu Huang, Xi-Le Zhao, and Tian-Hui Ma
Volume 2017, Article ID 5317850, 9 pages

Editorial

Emerging Trends in Machine Learning for Signal Processing

George A. Papakostas,¹ Konstantinos I. Diamantaras,² and Francesco A. N. Palmieri³

¹*Humane-Lab, Eastern Macedonia and Thrace Institute of Technology, Kavala, Greece*

²*TEI of Thessaloniki, Sindos, Greece*

³*Seconda Università di Napoli (SUN), Aversa, Italy*

Correspondence should be addressed to George A. Papakostas; gpapak@teikav.edu.gr

Received 17 October 2017; Accepted 18 October 2017; Published 19 November 2017

Copyright © 2017 George A. Papakostas et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, there is an increasing interest in developing “smart” devices and systems able to interact with their environment, for example, Internet of Things and Human-Machine Interfaces. The term “smart” is used to describe a set of advanced functionalities implemented utilizing sophisticated computational intelligence (CI) algorithms. Machine learning (ML) constitutes an important area of CI dealing with the ability of computers/machines to learn through knowledge representation, processing, and storing. ML offers solutions to difficult engineering problems, in a similar way to the humans’ brain processing. Moreover, considering the large amount and diversity of data (image, video, time series, 1D signals, text, etc.) massively generated and stored by modern “smart” systems, the need for efficient ML algorithms in terms of accuracy and speed become increasingly important. In the light of this rapid development of machine learning tools, this special issue focuses on recent trends in applying ML methodologies for processing signals coming from any source.

One paper of this issue addresses the diagnosis of pathological brain images using well known image processing tools (Wiener filtering, 2D-DWT, and Probabilistic PCA) and machine learning models (Random Subspace Ensemble (RSE), K -nearest neighbors). The proposed methodology was compared with 21 state-of-the-art algorithms, in terms of accuracy, sensitivity, and specificity for four datasets.

Another paper presents a novel active semisupervised Convolutional Neural Network (CNN) algorithm able to recognize SAR images without the need for a large number of labeled samples in the training phase. Initially, the method

applies active learning for the labeled data and then a semisupervised regularization process is designed for the remaining unlabeled data.

One paper is on the application of intuitionistic fuzzy Intercriteria Analysis on reducing the number of input parameters of a Multilayer Perceptron neural network. This will allow the reduction of the weight matrices, as well as the implementation of the neural network in limited hardware, thus this will save time and resources in training.

Another paper deals with the face recognition problem. For this purpose a patch-based Principal Component Analysis (PCA) method is proposed that utilizes the local spatial information enclosed in face image patches. Extensive experiments on two benchmark face datasets showed outperformance of the proposed method against several PCA-based similar algorithms.

One of the papers explores the use of recorded deep brain local field potentials (LFPs) for robust movement decoding of Parkinson’s Disease (PD) and Dystonia patients. A novel ensemble classifier was proposed for accurate prediction of finger movement and its forthcoming laterality. The ensemble classifier consists of three base neural network classifiers, namely, feed-forward, radial-basis, and probabilistic neural networks, while the majority-voting rule was used to fuse the decisions of three base classifiers to generate the final decision.

Another paper proposes a novel personal verification system based on the likelihood ratio test for fusion of match scores from multiple biometric matchers (face, fingerprint, hand shape, and palm print). Zernike moments are used as

multimodal features during the matching phase and after the fusion of the match scores a Gaussian Mixture Model estimates the genuine and impostor densities of match scores for personal verification.

One paper explores the applicability of deep learning techniques for identifying unexpected behavior in ship plot and track patterns, as captured by an Over-The-Horizon (OTH) radar. The proposed methodology exploits the non-linear mapping capabilities of deep stacked autoencoders in combination with density based clustering. A detailed comparative experimental evaluation of the approach shows promising results.

*George A. Papakostas
Konstantinos I. Diamantaras
Francesco A. N. Palmieri*

Research Article

Multimodal Personal Verification Using Likelihood Ratio for the Match Score Fusion

Long Binh Tran¹ and Thai Hoang Le²

¹Computer Science Department, University of Lac Hong, Dong Nai 810000, Vietnam

²Computer Science Department, VNUHCM-University of Science, Ho Chi Minh City 700000, Vietnam

Correspondence should be addressed to Thai Hoang Le; lhthai@fit.hcmus.edu.vn

Received 3 May 2017; Revised 14 August 2017; Accepted 24 September 2017; Published 31 October 2017

Academic Editor: George A. Papakostas

Copyright © 2017 Long Binh Tran and Thai Hoang Le. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, the authors present a novel personal verification system based on the likelihood ratio test for fusion of match scores from multiple biometric matchers (face, fingerprint, hand shape, and palm print). In the proposed system, multimodal features are extracted by Zernike Moment (ZM). After matching, the match scores from multiple biometric matchers are fused based on the likelihood ratio test. A finite Gaussian mixture model (GMM) is used for estimating the genuine and impostor densities of match scores for personal verification. Our approach is also compared to some different famous approaches such as the support vector machine and the sum rule with min-max. The experimental results have confirmed that the proposed system can achieve excellent identification performance for its higher level in accuracy than different famous approaches and thus can be utilized for more application related to person verification.

1. Introduction

It is proven in the literature that personal verification systems using biometric modalities acquire outweighing advantages in terms of security and conveniences. Thus, there are now many biometric systems which are used widely, like face, facial thermograms, fingerprint, hand geometry, hand vein, iris, retinal pattern, signature, voice-print, and so on [1].

Currently, unibiometric systems, the systems working on single biometric traits, are rather popular in use. Despite their significant development, these systems still have some disadvantages that can curb their effectiveness in performance in terms of noise, limited level of freedom, intraclass variability, spoofing attack, unacceptable error rates, and so on. Some of these drawbacks, however, can be handled by systems using multiple biometrics including different sensors, multiple samples of the same biometrics, different feature representations, multiple algorithms, or multimodalities [2–4]. Among these, multimodal systems utilize multiple traits, physiological or behavioural, for enrollment and identification.

Multimodal biometric systems have been accepted by many professionals thanks to (1) their superior performance

and (2) to overcome other limitations of unibiometric systems [3]. This leads to the hypothesis that our employment of multiple modalities (face, fingerprint, palm print, and hand shape) can conquer the limitations of the single modality-based techniques. Multimodal biometrics have many fusion levels [3], such as sensor level, feature level, matching score level, and decision level. With its efficiency and simplicity, fusion at score level becomes a preferable fusion technique [3, 5] although combining scores of different matchers with dissimilar nature and scale is a real challenge because the scores of different matchers can be either distance or dissimilarity measure. Finally, the match scores may follow different probability distributions, may provide quite different accuracies, and may be correlated. Techniques of fusing at score level are put in three groups: transformation-based score fusion [6–8], classifier-based score fusion [9, 10], and density-based score fusion [11, 12]. The last group is based on the likelihood ratio test and it requires explicit estimation of genuine and impostor match score densities. This scores density approach is based on the Neyman-Pearson theorem [13], which has the advantage that it directly achieves optimal performance at any desired operating point, provided the score densities are estimated accurately.

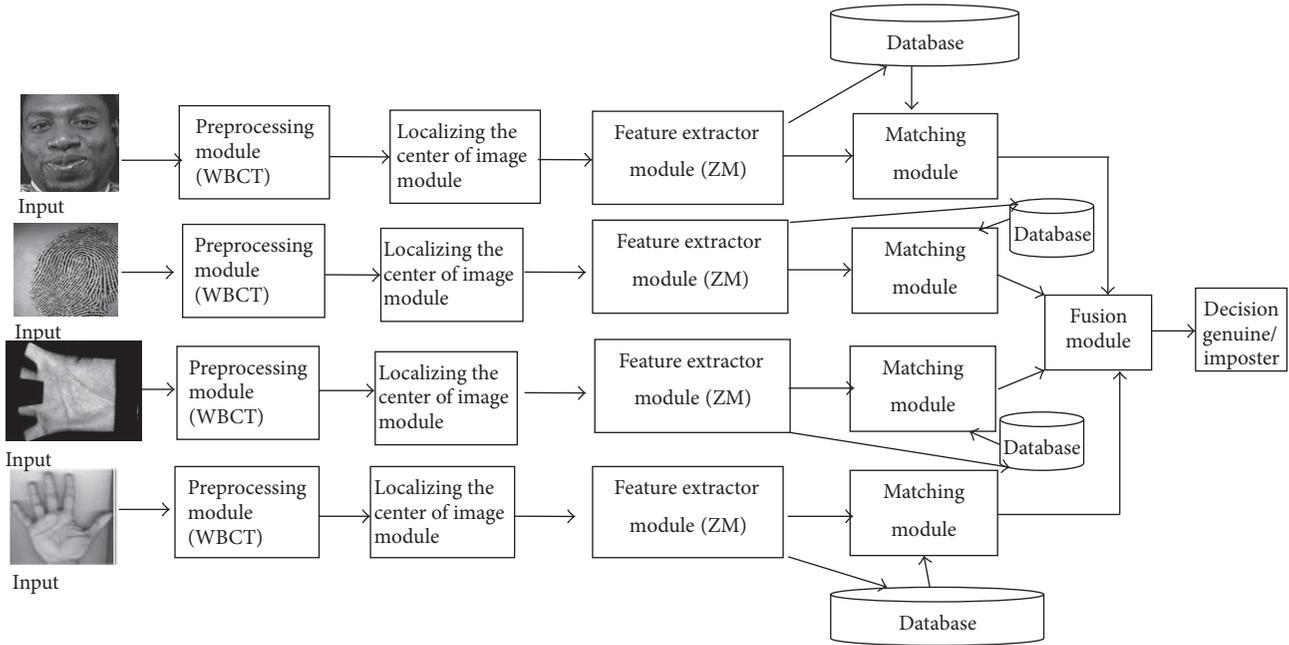


FIGURE 1: The chart of proposed personal verification system.

Our work aims at exploring effective ways to combine extracted multiple biometric features into templates for personal verification. To achieve this aim, we suggest an approach using Zernike Moment (ZM) and score level fusion technique based on likelihood ratio test and the finite Gaussian mixture model (LR-GMM) [14]. In this approach, ZM [15] is used to extract features of multimodal images (face, fingerprint, palm print, and hand shape). In this way, the basis function of ZM is defined on a unit circle and the center of the unit circle is set to coincide with the center of biometric images. This will extract more features, increasing the accuracy of personal verification. After matching, the performance of fusing the match scores using the likelihood ratio (LR) test and a finite Gaussian mixture model (GMM) for estimating the genuine and impostor score densities is examined. Finally, a decision is identified: an individual is genuine or impostor. Our proposed technique is also compared with the famous techniques such as support vector machine (SVM) and the sum rule with min-max and this comparison has shown outstanding results of the proposed technique.

The rest of this paper is about these contents: a depiction of the proposed system in Section 2; a description of the suggested methodology in Section 3; discussions about the experimental results in Section 4; and the paper conclusion in Section 5.

2. Proposed Multimodal System

In our work, a system using multiple biometric traits (face, fingerprint, palm print, and hand shape images) for personal identification (Figure 1) is proposed consisting of two phases: enrollment and verification. Both phases include preprocessing biometric images with Wavelet-Based Contourlet

Transform [16], localizing the center of image, extracting the feature vectors with ZM.

In the enrollment phase, the captured images are normalized and localizing the center of image for later feature extraction. Scores generated from the feature extractions are stored as templates in the database.

In the verification phase, the sets of feature scores obtained after image preprocessing, localizing the center of image and feature extraction, are supplied to the matching module where they are matched with the stored templates achieved in the enrollment phase, generating matching scores. These scores are fused and finally the chosen individual is identified.

Our proposed personal verification system is composed of five modules. In the first module, the image was pre-processed prior to the feature extraction. Our identification system used Wavelet-Based Contourlet Transform [16] to process the image normalization, noise elimination, illumination normalization, and so on. In the second module, Algorithms in [17–22] were used to locate the center of the best-fit ellipse in a face image, the reference point in a fingerprint image, the reference point in a palm print image, and the center of the elliptical model of a palm and each finger, and then the center of the unit circle of ZM is set to coincide with the reference point in a fingerprint image and with the center of the best-fit ellipse in a face image, the reference point in a palm print image, and the center of the elliptical model of a palm and each finger. In the third module, different features were extracted from the derived image normalization (feature domain) in parallel structure. To extract the features from the input images, Zernike Moment (ZM) was used. In the fourth module, the matching was carried out by Euclidean distance, based on the chosen features. The matching was done in each feature domain

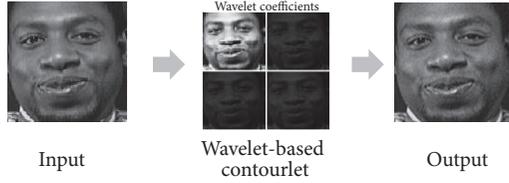


FIGURE 2: An example of the WBCT method.

in parallel as Figure 1. In the last module, the outputs of each matcher were combined to construct the identification. In this paper, match score fusion method was selected for decision strategy and FVC2004 database [23], ORL database [24], PolyU database [25], and IIT Delhi database [26] were used for the experiment.

3. Methodology

In our paper, main modules of the proposed system including image preprocess, localizing the center of image, feature extraction, matching, and a multimodal biometric verification model are described in detail.

3.1. Image Preprocess. Due to the noise in biometric images, the quality of images may be poor and thus the identification cannot be done efficiently; therefore, this module aims at normalizing an image by reducing or eliminating some of its variations. To do it, Wavelet-Based Contourlet Transform (WBCT) [16] is used.

Wavelet-Based Contourlet Transform in [16] is briefly described as follows: this system consists of two stages. In stage 1, an image is disintegrated into components of low frequency and high frequency, creating coefficients of various bands, which are later handled individually. Histogram equalization is applied to the approximation of the coefficients of low frequency. In stage 2, coefficients of high frequency are handled with a directional filter bank for smoothing the image edge. The image is normalized thanks to the coefficients modified by an inverse Wavelet-Based Contourlet Transform. The normalized image is enhanced in its contrast, its edges, and its details, all of which are necessary for further biometric image recognition (Figure 2).

See [16] for a detailed description.

3.2. Localizing the Center of Image. In this phase, we find the center of biometric images after normalization. This will extract more features and increase the accuracy of personal verification.

3.2.1. The Reference Point of Fingerprint. The reference point of a fingerprint is defined as the point of maximum curvature in the most internal crests. Usually, the core point is used as reference point. This point can be located by an algorithm which is briefly described as follows [17]:

- (1) Choose a window with $w \times w$ size for the estimation of the orientation field O . A 7×7 mean filter is used in

our work. The smoothed orientation field O' at (i, j) is computed as follows:

$$O'(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{\Phi'_y(i, j)}{\Phi'_x(i, j)} \right). \quad (1)$$

- (2) Estimate ε , an image with the sine component of O' :

$$\varepsilon(i, j) = \sin(O'(i, j)). \quad (2)$$

- (3) Initialize A , a label image used for reference point indication.

- (4) Identify the highest value in A and assign its coordinate to the core, that is, the reference point (Figure 3).

See [17] for a detailed description.

3.2.2. The Center of Face Image. In face image with frontal view, the face shape is approximate to an ellipse (Figure 5). In the algorithm, to find the best-fit ellipse [18], an ellipse model with five parameters is used; X_0, Y_0 denote the ellipse center; θ is the orientation; α and β are the minor and the major axes of the ellipse individually (Figure 4). Geometric moments are considered for the calculation of those five parameters.

The geometric moments of order p, q of a digital image are specified as

$$M_{pq} = \sum_x \sum_y f(x, y) x^p y^q, \quad (3)$$

where $p, q = 0, 1, 2, \dots$ and $f(x, y)$ denotes the gray scale value of the digital image at x and y locations. The origin is placed at the image center to capture the translation invariant central moments as summarized in the following equation:

$$\mu_{pq} = \sum_x \sum_y f(x - x_0, y - y_0) (x - x_0)^p (y - y_0)^q, \quad (4)$$

where $x_0 = M_{10}/M_{00}$ and $y_0 = M_{01}/M_{00}$ represent the centers of the joined components of which center of gravity indicates the ellipse center. The orientation θ of the ellipse is estimated by the least moment of inertia [19, 20]

$$\theta = \frac{1}{2} \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right), \quad (5)$$

where μ_{pq} is the central moment of joined components (4). By the least and the greatest moment of inertia of an ellipse is defined as

$$I_{\text{Min}} = \sum_x \sum_y [(x - x_0) \cos \theta - (y - y_0) \sin \theta]^2, \quad (6)$$

$$I_{\text{Max}} = \sum_x \sum_y [(x - x_0) \sin \theta - (y - y_0) \cos \theta]^2;$$

the lengths of the major and minor axes are computed as

$$\alpha = \frac{1}{\pi [I_{\text{Max}}^3 / I_{\text{Min}}]^{1/8}}, \quad (7)$$

$$\beta = \frac{1}{\pi [I_{\text{Min}}^3 / I_{\text{Max}}]^{1/8}}.$$

See [19, 20] for a detailed description.

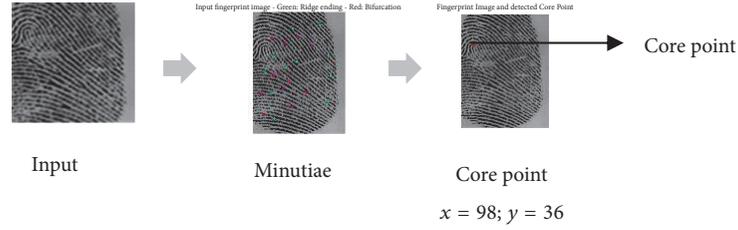


FIGURE 3: The reference point (the core point) on the fingerprint.

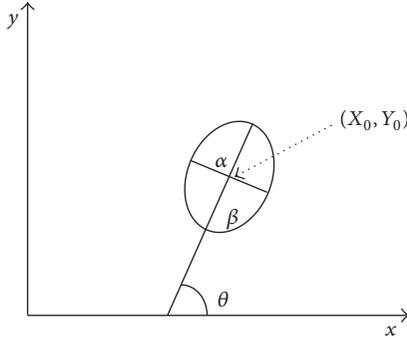


FIGURE 4: Face model based on ellipse model.



FIGURE 5: Localizing faces using best-fit ellipse.

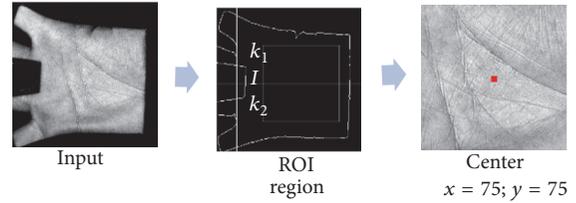


FIGURE 6: Localizing the center of ROI region of palm.

3.2.3. *The ROI of Palm Print.* In this phase, we will find the region of interest (called ROI) in the palm table. The ROI is defined in square shape and it contains sufficient information to represent the palm print for further processing. The outline of the ROI could be obtained as follows [21].

The center part of the palm print image is extracted as it contains prominent features such as wrinkles, ridges, and principal lines. The following are steps involved in ROI extraction:

- (1) Compute the centroid of a palm print and locate the point “I” between the middle finger and ring finger.
- (2) Take a 3×3 8-connectivity matrix by placing the pointer “P” (P is the center of mask) of the matrix at “I” and trace the corner points “ k_1 ” and “ k_2 .”
- (3) Locate the midpoint “mid” between “ k_1 ” and “ k_2 .”
- (4) Move from the point “mid” with the fixed number of pixels toward center of the palm and position the fixed sized square to crop the image and extract the subimage (ROI).
- (5) The center of ROI is the center of the fixed sized square to crop the image (Figure 6).

See [21] for a detailed description.

3.2.4. *Hand Shape.* The segmentation of the hand silhouette is performed without requiring the extraction of any landmark points on the hand and this segmentation can be summarized as follows.

After binarization, the first, the hand silhouette is segmented into six regions corresponding to the palm and

the fingers. Segmentation is performed using an iterative process based on morphological filters [22]. The second is the geometric moment [19, 20] of each component of the hand that is considered for the calculation of five parameters; X_0, Y_0 denote the ellipse center; θ is the orientation; α and β are the minor and the major axes of the ellipse individually (Figure 4). Finally, the center of the best-fit ellipse of each component of the hand has been defined (Figure 7).

See [19, 20, 22] for a detailed description.

3.3. *Feature Extraction with Zernike Moment.* This module aims at extracting feature vectors or image-representing information. Features are extracted by ZM [15]. In our system, the extraction is performed on the derived images in parallel structure. That enables more characteristics of biometric images to be obtained.

3.3.1. *Zernike Moment.* For a 2D image $f(x, y)$, the image is changed from Cartesian coordinate into polar coordinate $f(r, \theta)$, where r and θ are radius and azimuth, respectively. The transformation of the images is done by the following formulae:

$$r = \sqrt{x^2 + y^2}, \quad (8)$$

$$\theta = \arctan\left(\frac{y}{x}\right). \quad (9)$$

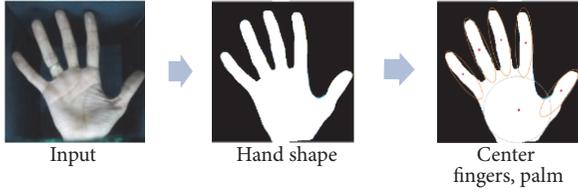


FIGURE 7: Localizing the center of fingers and palm.

The image is specified on the unit circle with $r \leq 1$ and enlarged by the basic functions $V_{nm} = (r, \theta)$.

Zernike Moment with order n and repetition m is defined as

$$M_{nm} = \frac{n+1}{\pi} \int_0^1 \int_0^{2\pi} [V_{nm}(r, \theta)]^* f(r, \theta) r dr d\theta, \quad (10)$$

where $*$ denotes complex conjugate, $n = 0, 1, 2, \dots, \infty$, and m is an integer subject to the constraint that $n - |m|$ is nonnegative and even. $V_{nm}(r, \theta)$, Zernike polynomial, is defined over the unit disk as follows:

$$V_{nm}(r, \theta) = R_{nm}(r) e^{im\theta} \quad (11)$$

with the radial polynomial $R_{nm}(r)$ defined as

$$R_{nm}(r) = \sum_{s=0}^{(n-|m|)/2} \frac{(-1)^s (n-s)! r^{n-2s}}{s! ((n+|m|)/2 - s)! ((n-|m|)/2 - s)!} \quad (12)$$

The kernels of ZMs are a set of orthogonal Zernike polynomials so that any images can be represented by complex ZMs. Given all ZMs of an image, the image can be reconstructed as follows:

$$f(r, \theta) = \sum_n \sum_{(All\ m's)} M_{nm} V_{nm}(r, \theta). \quad (13)$$

The advantages of Zernike moments are translation, rotation, and scaling invariant. The invariant properties of Zernike moments are utilized as pattern sensitive features in recognition applications [27]. A short discussion about their invariant properties should be considered.

(1) Translation invariance can be obtained by converting the original image $f(x, y)$ into the absolute pixel coordinates as follows $f(x + \bar{x}, y + \bar{y})$, where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$ are the centroid coordinates of the original image (with m denoting the geometrical moment).

(2) Scaling invariance can be achieved by normalizing the Zernike Moment with respect to the geometrical moment m_{00} of the image. The improved Zernike moments are derived from the following equation: $Z'_{nm} = Z_{nm}/m_{00}$ with Z_{nm} are the Zernike moments of (10).

(3) Rotation invariance can be considered when $f(x, y)$ is rotated by an angle α ; we have the Zernike Moment Z_{nm} of the rotated image defined as

$$Z'_{nm} = Z_{nm} e^{-jm\alpha}. \quad (14)$$

In this way, the magnitudes of ZMs can be used as features of rotational invariances of an image.



FIGURE 8: Example of ZM used for biometric images feature extraction.

3.3.2. Feature Extraction. In this phase, the center of the unit circle (the basis functions of ZM) in biometric images is determined. The center of the unit circle of ZM is set to coincide with the reference point in a fingerprint image, with the center of the best-fit ellipse in a face image (best-fit ellipse is an ellipse that encloses the facial region in a face image with frontal view), with the center of the circumscribed circle of square region in a palm table which is called region of interest (ROI), with the center of the best-ellipse-fitting of a palm and each finger (Figure 8).

Zernike Moment has shown in literature its ability to perform better than other moments (e.g., Tchebichef moment [30], Krawtchouk moment [31]). In fact, the increase in the orders of ZM will lead to a reduction in the quality of the reconstructed image due to the numerical changeability of ZM. Thus, in our work, the first 10 orders of ZM with 36 feature vector elements were chosen for a better performance of ZM.

3.4. Proposed Matching and Fusion. The sets of feature vectors obtained following image feature extraction are supplied to the matching modules, where they are matched with templates stored in the database. The Euclidean distance metric is applied to calculate similarity between the two feature vectors to generate matching scores.

In this work, we propose a supervised fusion where the classifiers (genuine or impostor) are trained using the match score densities and the parameters of the finite Gaussian mixture model that are used for modelling the genuine and impostor score densities of the training data.

According to the Neyman–Pearson theorem, the optimal test for deciding a score vector \mathbf{x} to the class genuine or impostor is the likelihood ratio test given by

$$L(\mathbf{x}) = \frac{f_{\text{gen}}(\mathbf{x})}{f_{\text{imp}}(\mathbf{x})}, \quad (15)$$

where $f_{\text{gen}}(\mathbf{x})$ and $f_{\text{imp}}(\mathbf{x})$ are the estimated densities from the training data of genuine and impostor match scores, respectively. In this paper, the GMM automatically estimates the number of components and the component parameters using the Expectation-maximization (EM) algorithm [14]

and the minimum message length criterion. The probability distribution for a d -dimensional object \mathbf{x} is given by

$$N(\mathbf{x}) = (2\pi)^{-d/2} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\}, \quad (16)$$

where \mathbf{x} is the match score vector, $\boldsymbol{\mu}$ is the mean vector, and Σ is the covariance matrix of the training set. Assuming that both the genuine class and the impostor class have a mixture of Gaussian distributions, as expressed by

$$f_{\text{gen}}(\mathbf{x}) = \sum_{i=1}^{M_{\text{gen}}} c_{\text{gen},i} N_{\text{gen},i}(\mathbf{x})$$

$$f_{\text{imp}}(\mathbf{x}) = \sum_{i=1}^{M_{\text{imp}}} c_{\text{imp},i} N_{\text{imp},i}(\mathbf{x}), \quad (17)$$

where M_{gen} (M_{imp}) is the number of mixture components of the genuine (impostor) score and $c_{\text{gen},i}$ ($c_{\text{imp},i}$) is the weight assigned to the i th mixture component, $\sum_{i=1}^{M_{\text{gen}}} c_{\text{gen},i} = \sum_{i=1}^{M_{\text{imp}}} c_{\text{imp},i} = 1$.

4. Experimental Results and Discussion

4.1. Experimental Results. Experiments have been conducted on several datasets. Brief information about the four used databases (Figure 9) is presented as follows:

- (1) FVC2004 fingerprint database [23]: FVC2004 DB4 includes 800 fingerprints of 100 fingers (8 images of each finger). Size of each fingerprint image is 288×384 pixels, and its resolution is 500 dpi.
- (2) ORL face database [24]: ORL is comprised of 400 images of 40 people with various facial expressions and facial details. All images were taken on dark background with a size of 92×112 pixels.
- (3) PolyU palm print database [25]: PolyU contains 7752 grayscale images corresponding to 386 different palms. Around 20 images per palm have been collected in two sessions. Size of each image is 384×284 pixels.
- (4) IIT Delhi hand shape database [26]: IIT Delhi has collected left and right hand images from 235 subjects. Each subject contributed at least 5 hand images from each of the hands. Size of each image is 800×600 pixels. From this dataset several biometric characteristics are segmented (palm, fingers, and hand shape). The palm and fingers are segmented using morphological operators proposed in [19, 20, 22].

In Table 1, we report the number of mixture found for the genuine data and for the impostor data in the four datasets used in this work.

In our experiment, the training set used for density estimation was formed with half of the genuine and half



FIGURE 9: Some samples from the dataset used in this work.

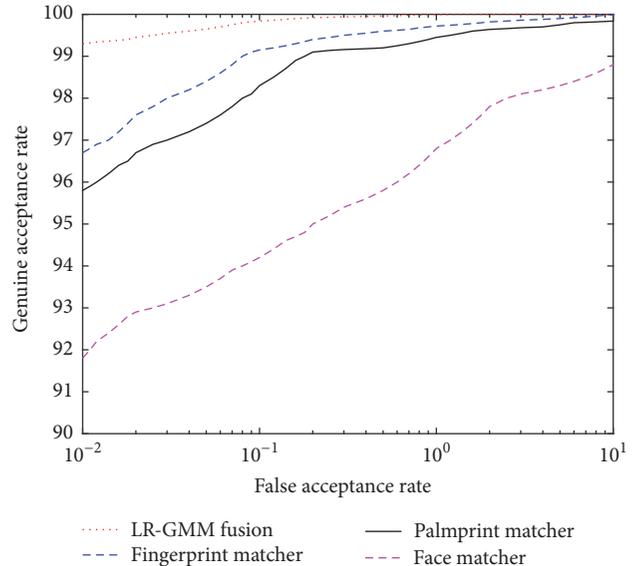


FIGURE 10: The ROC curves of the LR-GMM fusion and individual matchers.

of the impostor match scores chosen randomly, and this division was repeated 10 times. As the achieved experimental results, the receiver operating characteristic (ROC) curves match the mean of genuine accept rate (GAR) values in all 10 tests conducted at different FAR values, and our proposed approach led to average verification accuracies in GAR.

The ROC curves of the LR-GMM fusion rule with four matchers and individual matchers in the FVC2004, the PolyU, and the ORL database are presented in Figure 10.

The performance of the LR-GMM fusion rule is significantly improved in comparison to the best individual modality from the four databases. LR-GMM fusion rule also brings about an increase in GAR with FAR of 0.01% (Table 2). Noticeably, the average verification accuracies presented in Table 2 show that the efficiency of the proposed method remained constant in 10 cross-validation trials and that multibiometric fusion of difference traits (fingerprint scores and palm print scores, fingerprint scores and face scores, and palm print scores and face scores) in the FVC2004, the PolyU, and the ORL databases considerably improved GAR compared to other multibiometric fusions (two fingerprint scores' fusion, two palm print scores' fusion, two face scores' fusion and hand scores' fusion).

The ROCs curves of LR-GMM fusion rule on four databases and LR-GMM fusion rule on each database (two fingerprint scores' fusion, two palm print scores' fusion, two

TABLE 1: The number of mixtures for the genuine data and the impostor data.

	FVC2004-DB4 Fingerprint	ORL Face	PolyU Palm print	IITK hand shape Palm	IITK hand shape Fingers
Genuine	6	8	6	4	4
Impostor	6	8	6	4	4

TABLE 2: Performance achieved.

Database		Single matcher	Mean GAR at 0.01% FAR	
			The same traits	Difference traits
IIT-Delhi	Multimodal		99.4%	99.4%
	Hand shape		99.32%	
FVC2004-DB4	Fingerprint	96.7%	97.8%	99.1% (fingerprint-palm print)
PolyU	Palm print	95.8%	96.3%	98.9% (fingerprint-face)
ORL	Face	91.8%	93.2%	97.3% (palm print-face)

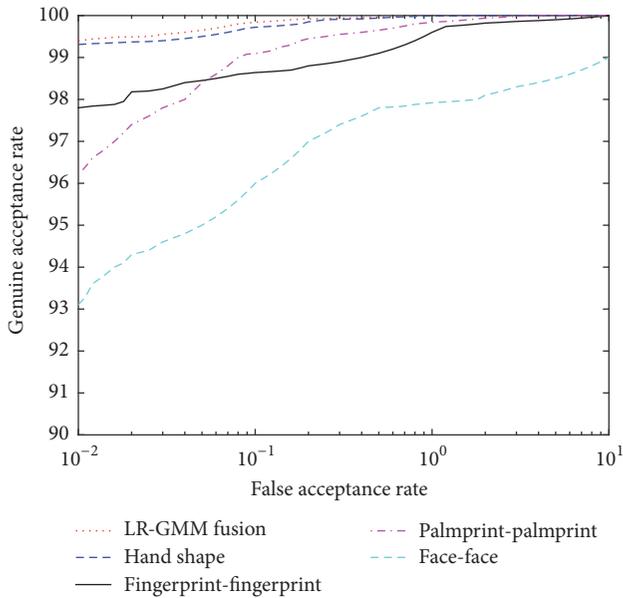


FIGURE 11: The ROC of LR-GMM on each database and on four databases.

face scores' fusion and hand scores' fusion) are presented in Figure 11.

The ROCs curves of LR-GMM fusion rule on four databases and LR-GMM fusion rule of difference traits (fingerprint scores and palm print scores, fingerprint scores and face scores, palm print scores and face scores) in the FVC2004, the PolyU, and the ORL database are shown in Figure 12.

According to our achieved experimental results, LR-GMM fusion can improve the GAR compared to the best individual modality. In particular, at the FAR of 0.01%, the mean GAR of LR-GMM fusion rules is 99.4% while the GAR values of the face, fingerprint, palm print, and hand shape modality are successively 93.2%, 97.8%, 96.3%, and 99.32%.

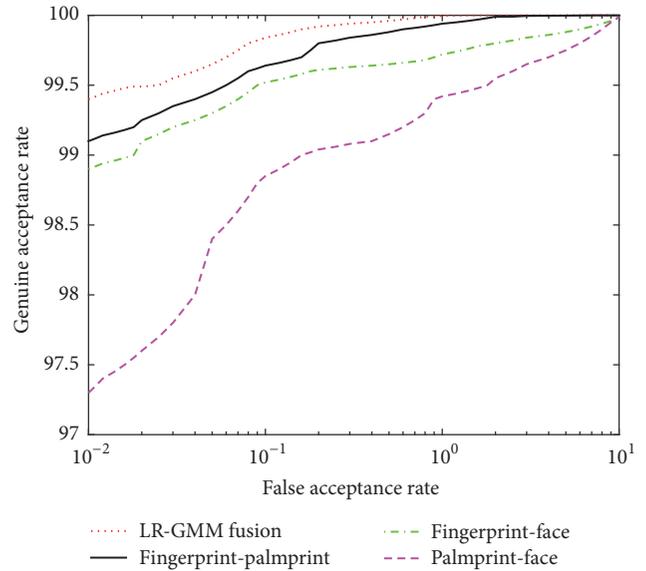


FIGURE 12: The ROC of LR-GMM on four databases and LR-GMM of difference traits.

The performance of LR-GMM fusion rule was also compared with its performance using the support vector machine (SVM) classifier-based fusion, a classifier-based score fusion technique, and the sum of scores fusion method, a transformation-based score fusion technique. To enhance performance, the radial basis function (RBF) was chosen as the kernel function for SVM classifier. To use the sum of scores technique, the min-max normalization method [8] was used. We noted that the sum rule with min-max worked efficiently in our experiments on the chosen datasets. The ROC curves of the LR-GMM fusion rule, SVM classifier, and the sum rule with min-max on the multimodals of FVC2004, ORL, PolyU, and IIT Delhi are shown in Figure 13.

The proposed system was also compared with the other recognition systems, particularly face recognition system

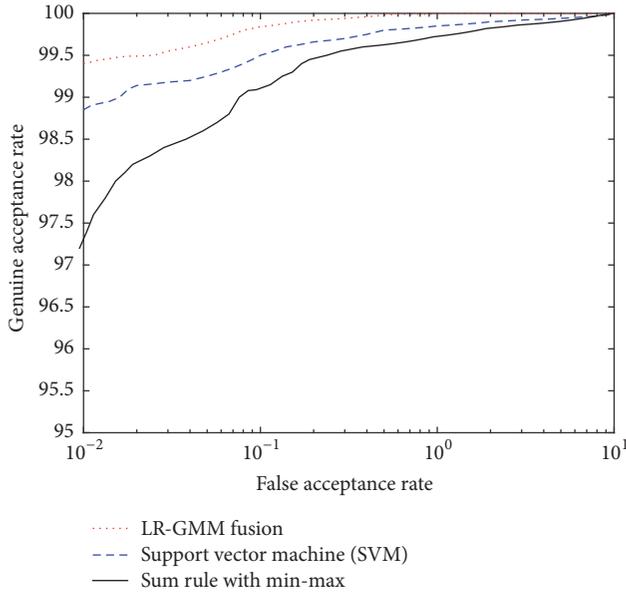


FIGURE 13: The ROC of LR-GMM, SVM, and Sum rule.

TABLE 3: Accuracy rate achieved by different algorithms.

	Ours	Exist
Face	93.2%	92.8% [27]
Fingerprint	97.8%	92.89% [28]
Palm print	96.3%	91.25% [29]
Hand shape	99.32%	99.3% [22]
Proposed LR-GMM fusion	99.4%	

[27], fingerprint recognition system [28], palm print recognition system [29], and hand shape recognition system [22] using Zernike Moment and similar databases. The comparative results in Table 3 prove that the average verification accuracies at 0.01% FAR of our system can perform better than other recognition systems in terms of recognition rate.

4.2. Discussion. From the experimental results, some significant features of the proposed system using ZM-LR-GMM can be seen as below.

- (1) Determining the center of the biometric images will extract more features and increase the accuracy of the personal identification.
- (2) ZM is invariant to rotation, scale, and translation. Also, the feature extraction using Zernike Moment can provide feature sets with similar coefficients for easy computation.
- (3) The fusion rule using LR-GMM achieved high verification rate as well as easy implementation.
- (4) Our proposed method can work well on more databases.

Typically, there is a tradeoff between the additional cost and the improvement in performance of a multibiometric

system. The cost could be the number of sensors deployed, the time required for acquisition and processing, performance gain (reduction in FAR/FRR), storage and computational requirements, and perceived convenience to the user.

5. Conclusion

In this paper, the authors have presented a novel feature extraction approach for the fusion of match scores in a multibiometrics system based on the likelihood ratio test and the finite Gaussian mixture model, in which biometric images are extracted by Zernike Moment to obtain comparable feature vectors. The proposed ZM-LR-GMM approach was tested on the publicly available databases such as FVC2004, ORL, PolyU, and IIT Delhi. It can be noted from the experiment that the fusion of comparable feature vectors contains more information about biometric images and thus can improve the verification rate. Practically, the highest verification rates GAR of 99.4% and FAR of 0.01% are achieved; this represents the outstanding performance of this proposed system. With its advantages, the proposed ZM-LR-GMM system can minimize lack of information and increase verification rate.

Disclosure

This manuscript is an extended version of the paper entitled Personal Authentication Using Relevance Vector Machine (RVM) for Biometric Match Score Fusion at the 2015 seventh International Conference on Knowledge and Systems Engineering (KSE) [10].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] A. K. Jain, P. Flynn, and A. A. Ross, *Handbook of Biometrics*, Springer, 2008.
- [2] D. Jagadiswary and D. Saraswady, "Biometric authentication using multimodal biometric," in *Proceedings of the International Conference on Computational Modelling and Security, CMS 2016*, pp. 109–116, India, February 2016.
- [3] A. Ross and A. Jain, "Information fusion in biometrics," *Pattern Recognition Letters*, vol. 24, no. 13, pp. 2115–2125, 2003.
- [4] N. Hezil and A. Boukrouche, "Multimodal biometric recognition using human ear and palmprint," *IET Biometrics Journal*, vol. 6, no. 5, pp. 351–359, 2017.
- [5] R. Parkavi, K. R. C. Babu, and J. A. Kumar, "Multimodal biometrics for user authentication," in *Proceedings of the 2017 11th International Conference on Intelligent Systems and Control, ISCO 2017*, pp. 501–505, India, January 2017.
- [6] L. Nanni, A. Lumini, M. Ferrara, and R. Cappelli, "Combining biometric matchers by means of machine learning and statistical approaches," *Neurocomputing*, vol. 149, pp. 526–535, 2015.
- [7] A. Lumini and L. Nanni, "Overview of the combination of biometric matchers," *Information Fusion*, vol. 33, pp. 71–85, 2017.

- [8] A. Jaina, K. Nandakumara, and A. Rossb, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.
- [9] Y. Ma, B. Cukic, and H. Singh, "A classification approach to multibiometric score fusion," in *Proceedings of Fifth International Conference on AVBPA*, pp. 484–493, New York, NY, USA, 2005.
- [10] L. B. Tran and T. H. Le, "Personal Authentication Using Relevance Vector Machine (RVM) for Biometric Match Score Fusion," in *Proceedings of the 7th IEEE International Conference on Knowledge and Systems Engineering, KSE 2015*, pp. 7–12, Vietnam, October 2015.
- [11] S. C. Dass, K. Nandakumar, and A. K. Jain, "A principled approach to score level fusion in multimodal biometric systems," in *Proceedings of the Audio- and Video-Based Biometric Person Authentication, AVBPA '05*, pp. 1049–1058, 2005.
- [12] K. Nandakumar, *Multibiometric Systems: Fusion Strategies and Template Security*, Phd Thesis, Michigan State University, Department of Computer Science and Engineering, 2008.
- [13] E. L. Lehmann and J. P. Romano, *Testing Statistical Hypotheses*, Springer, 2005.
- [14] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.
- [15] F. Zernike, *Physica*, 1934.
- [16] L. B. Tran and T. H. Le, "Using wavelet-based contourlet transform illumination normalization for face recognition," *International Journal Modern Education and Computer Science*, vol. 7, no. 1, pp. 16–22, 2015.
- [17] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 846–859, 2000.
- [18] J. Haddadnia, K. Faez, and M. Ahmadi, "An efficient human face recognition system using Pseudo Zernike Moment Invariant and radial basis function neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 1, pp. 41–62, 2003.
- [19] J. Haddadnia and K. Faez, "Human face recognition based on shape information and pseudo zernike moment," *5th Int. Fall Workshop Vision, Modeling and Visualization*, pp. 113–118, 2000.
- [20] J. Haddadnia, M. Ahmadi, and K. Faez, "An efficient method for recognition of human faces using higher orders Pseudo Zernike Moment Invariant," in *Proceedings of the 5th IEEE International Conference on Automatic Face Gesture Recognition, FGR 2002*, pp. 330–335, May 2002.
- [21] R. Raghavendra, A. Rao, and G. Hemantha Kumar, "A novel three stage process for palmprint verification," in *Proceedings of the International Conference on Advances in Computing, Control and Telecommunication Technologies, ACT 2009*, pp. 88–92, India, December 2009.
- [22] G. Amayeh, G. Bebis, and M. Hussain, "A comparative study of hand recognition systems," in *Proceedings of the 1st International Workshop on Emerging Techniques and Challenges for Hand-Based Biometrics, ETCHB 2010*, Istanbul, Turkey, August 2010.
- [23] FVC (2004). Finger print verification contest 2004. <http://bias.csr.unibo.it/fvc2004/download.asp>.
- [24] ORL, 1992. The ORL face database at the AT and T (Olivetti) Research Laboratory, <http://www.cl.cam.ac.uk/research/dtg/at-tarchive/facedatabase.html>.
- [25] The PolyU palmprint database. <http://www.comp.polyu.edu.hk/biometrics>.
- [26] IIT Delhi Touchless Palmprint Database, http://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Palm.htm.
- [27] S. M. Lajvardi and Z. M. Hussain, "Higher order orthogonal moments for invariant facial expression recognition," *Digital Signal Processing*, vol. 20, no. 6, pp. 1771–1779, 2010.
- [28] A. Q. Hasan, R. R. Abdul, and S. Al-Haddad, "Fingerprint recognition using zernike moments," *The International Arab Journal of Information Technology*, vol. 4, no. 4, pp. 372–376, 2007.
- [29] S. Karar and R. Parekh, "Palm print recognition using zernike moments," *International Journal of Computer Applications*, vol. 55, no. 16, pp. 15–19, 2012.
- [30] R. Mukundan, S. H. Ong, and P. A. Lee, "Image analysis by Tchebichef moments," *IEEE Transactions on Image Processing*, vol. 10, no. 9, pp. 1357–1364, 2001.
- [31] P.-T. Yap, R. Paramesran, and S.-H. Ong, "Image analysis by Krawtchouk moments," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1367–1377, 2003.

Research Article

Stacked Autoencoders for Outlier Detection in Over-the-Horizon Radar Signals

Eftychios Protopapadakis,¹ Athanasios Voulodimos,^{1,2} Anastasios Doulamis,¹
Nikolaos Doulamis,¹ Dimitrios Dres,³ and Matthaïos Bimpas³

¹National Technical University of Athens, 15780 Athens, Greece

²Department of Informatics, Technological Educational Institute of Athens, 12243 Athens, Greece

³Telesto Technologies, 15561 Cholargos, Greece

Correspondence should be addressed to Athanasios Voulodimos; thanosv@mail.ntua.gr

Received 16 June 2017; Accepted 14 September 2017; Published 23 October 2017

Academic Editor: George A. Papakostas

Copyright © 2017 Eftychios Protopapadakis et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Detection of outliers in radar signals is a considerable challenge in maritime surveillance applications. High-Frequency Surface-Wave (HFSW) radars have attracted significant interest as potential tools for long-range target identification and outlier detection at over-the-horizon (OTH) distances. However, a number of disadvantages, such as their low spatial resolution and presence of clutter, have a negative impact on their accuracy. In this paper, we explore the applicability of deep learning techniques for detecting deviations from the norm in behavioral patterns of vessels (outliers) as they are tracked from an OTH radar. The proposed methodology exploits the nonlinear mapping capabilities of deep stacked autoencoders in combination with density-based clustering. A comparative experimental evaluation of the approach shows promising results in terms of the proposed methodology's performance.

1. Introduction

Detection of targets and outliers in radar signals is a research issue that has gained significant attention in the academic and industrial research community, mainly because of the important associated impact of relevant applications in surveying of large areas. High-Frequency Surface-Wave (HFSW) radars are a category of radars that operate at the frequency band 3–30 MHz and, in contrast with other radars, use ground wave or sky wave propagation and ionospheric reflections of the electromagnetic waves for target detection, which allows for achieving longer ranges, where microwave radars cannot perform [1], but to the detriment of the attained accuracy. For many years, HFSW radars, or over-the-horizon (OTH) radars, as they are commonly known, have been used to remotely measure oceanographic parameters, providing information about surface currents, wave spectra, wind direction and intensity, and so on [2]. Their extraordinary range (up to 200 nautical miles) combined with their continuous

mode of operation make for an ideal candidate tool for long-range oceanic surveillance. However, many associated weaknesses, for example, low spatial resolution, high nonlinearity, and important presence of clutter, negatively impact their performance as early-warning tools for detection, tracking, and identification of vessels.

The promising capabilities of OTH radars have attracted significant interest from the research community and have already resulted in various approaches (e.g., [3, 4]). Nevertheless, related research issues continue to present significant challenges, which can be attributed to few reasons, briefly described below:

- (i) Different targets may present similar dielectric and frequency properties thus making it hard to make a clear distinction among them.
- (ii) Given multipath propagation effects of rough surfaces, scattering from some objects tends to overwhelm the weak backscattering of targets.

- (iii) Due to the changes in atmosphere and ground conditions, noise is added which can confuse the analysis of a radar signal.
- (iv) Ocean and ionospheric clutter generate noise especially for HFSW radars.

On a different note, the surge of deep learning and the great results it has produced in other signal analysis domains, such as computer vision, speech recognition, and natural language processing, create certain expectations regarding its potential efficacy in radar signal analysis applications. Deep learning allows computational models of multiple processing layers to learn and represent data with multiple levels of abstraction mimicking how the brain perceives and processes multimodal information, thereby implicitly capturing intricate structures of large-scale data. Complex abstractions are learnt at a given level based on relatively simpler abstractions formulated in the preceding layer in the hierarchy.

The goal of this paper is to present a framework for detecting deviations from the norm in behavioral patterns of vessels (henceforth called *outliers*), as they are tracked from an OTH radar. The proposed methodology exploits the nonlinear mapping capabilities of deep stacked autoencoders (SAs) [5] in combination with density-based clustering. Stacked autoencoders are used in an unsupervised way to map the track history of any vessel into a compact and informative feature vector. Then, at any moment all tracked ships are projected into a new feature space and clustered using density-based algorithms, such as OPTICS [6]. The outcome of the clustering stage then indicates possible outliers.

The remainder of this paper is structured as follows: Section 2 presents an overview of the related work. In Section 3 we describe in detail the proposed methodology for outlier detection in OTH radar signals, which is followed by the experimental evaluation of the methodology in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

In the literature, several signal processing and machine learning methods have been investigated and proposed to acquire more reliable data with lower noise and extract semantic information from radar signals. Kouemou and Opitz [7] introduced a wavelet-based feature analysis combined with Hidden Markov Models (HMM) to classify real radar signals into predefined categories. Spectral analysis [8] is used by Garbanzo-Salas and Hocking [9] for detecting small objects from harmonic pulse radar data. The use of online bootstrapping machine learning tools to improve target detection rate of radar signals is also one major research area [10]. Radar data can be analyzed using the concepts of transfer learning since often we have only a small number of labelled data available while the majority of signals captured are unlabelled (nonannotated) [11]. Other works focus on modeling of ionospheric disturbances on spaceborne interferometric synthetic aperture radar (SAR) via Echo-State Networks [12, 13] or ensemble classifiers [14].

Denosing techniques for radar signals include low level processing such as the median filter or other nonlinear

convolution schemes [15]. Other approaches spatially or temporally decompose radar signals by wavelet transforms [16, 17]. This way, we can find patterns distributed on space and time domain to improve targets detection efficiency. These methodologies can be extended to the analysis of synthetic aperture radar (SAR) images [11], or by incorporating sparsity-based signal analysis [18]. A neural network based scheme for detecting salient objects in SAR images is recently presented [19]. The goal is to identify changes in SAR content. A similar approach for detecting changes using nonlinear stacked restricted Boltzmann machines is given in the work of Liu et al. [20], while multilayered feature learning to improve detection accuracy of SAR images is described by Xie et al. [21]. Furthermore, low-power HF surface-wave (HFSW) radars have demonstrated being a cost-effective long-range early-warning sensor for ship detection and tracking [22, 23]. A detailed description of various ways in which HFSW radar technology can be used for maritime surveillance is provided by Braca et al. [24].

Regarding deep versus “shallow” learning schemes, traditional machine learning techniques exploit shallow architectures; that is, they use a single layer for data/feature transformation, even in a highly nonlinear space. Shallowness refers here to the simplicity of these architectures that use only one (or few) layer(s) of processing, responsible for transforming the raw input signals or features into the problem-specific feature space. Instead, in a deep learning paradigm, the architectures are composed of many (deep) nonlinear processing stages [25]. Deep learning has been extensively applied in many fields, such as computer vision [26] (e.g., behavior recognition [27] and human tracking [28]) and speech recognition [29]. However, its applicability in radar signal processing had not been investigated until very recently [30]. Even so, most of the proposed works pertain to object detection in SAR image data [31], essentially resembling visual analysis approaches.

3. The Proposed Methodology

The proposed methodology exploits the nonlinear mapping abilities of stacked autoencoders (SAs) [5] in combination with density-based clustering, to identify irregular occurrences, using over-the-horizon radar data. Such an approach is based on two main assumptions:

- (1) The history of a naval vessel, in terms of speed, position, course, signal frequency, or other related data, provided by a ground radar, suffices to extract meaningful features.
- (2) Unexpected deviation from the norm is observed for a few ships, denoted henceforth as outliers.

The approach is relatively straightforward: Given a set of OTH data entries, SAs are used in an unsupervised way to map the track history of any vessel into a compact and informative feature vector. Then, at any moment all tracked ships are projected into a new feature space and clustered using OPTICS [6], a widely used density-based algorithm. The clustering outcome informs about possible

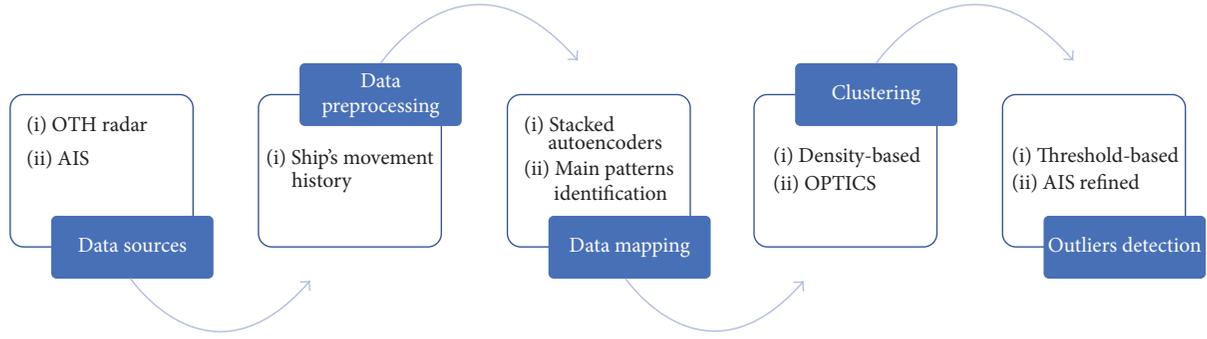


FIGURE 1: Proposed approach flowchart.

outliers. In the following subsections, the different stages of the methodology are presented, after a brief description of the data involved. Figure 1 provides a high-level view of the proposed approach.

3.1. OTH and AIS Data. Heterogeneous data, such as automatic identification system (AIS) data, high-frequency surface wave (HFSW) radar data, and synthetic aperture radar (SAR) data, have been exploited in research for maritime surveillance purposes [32]. In our case, two sources of information were fused to support the outlier detection process: OTH radar and AIS data.

The OTH radar data used for the setting and evaluation of the presented work was acquired by the HFSW STRADI-VARIUS radar by Diginext [33]. OTH radar detection (plot) and tracking (track) data are the output of the OTH radar for a given period. The plot and track data provided include estimated position coordinates, velocity, course, Doppler frequency, global and local noise levels, azimuth, and other parameters, appropriately timestamped.

On a different note, AIS is an automatic tracking system used for collision avoidance on ships and by vessel traffic services. AIS information supplements marine radar, which continues to be the primary method of collision avoidance for water transport. Vessels equipped with AIS transceivers can be tracked by AIS base stations located along coast lines. The International Maritime Organization's International Convention for the Safety of Life at Sea requires AIS to be present aboard international voyaging ships with gross tonnage of 300 or more and all passenger ships regardless of size [34]. AIS reports contain both dynamic information (e.g., latitude, longitude, course over ground, speed over ground, and time) and static information (e.g., vessel type and dimension information).

3.2. Density-Based Clustering as a Basis for Outlier Detection. Clustering refers to the task of identifying groups or clusters in a dataset. In density-based clustering, a cluster is a set of data objects spread in the data space over a contiguous region of high density of objects. Density-based clusters are separated from each other by contiguous regions of low density of objects. Data objects located in low-density regions are typically considered noise or outliers [35]. OPTICS algorithm [6], as one among various approaches for hierarchical

density-based clustering, includes ordering points to identify the clustering structure. OPTICS is based on DBSCAN [36] and the work of Stuetzle [37].

OPTICS computes a Minimum Spanning Tree (MST) of the data, where edge weights represent pairwise distances. These distances are smoothed by a density estimator, called core distance. The core distance of a point \mathbf{x}_i is the smallest threshold r such that \mathbf{x}_i is still considered a core object by the DBSCAN algorithm; that is, \mathbf{x}_i has at least k objects in its neighborhood within radius r . The resulting distance, which is used to construct the MST, is called reachability distance (RD). Taking k as input parameter for smoothing the density estimation, the reachability distance of point \mathbf{x}_i is defined relative to a reference object \mathbf{y} as the minimum of the core distance of \mathbf{y} and the actual distance between \mathbf{x}_i and \mathbf{y} . The outcome of the algorithm can provide us information about the clustering of the objects (see Section 3.4).

3.3. Using Stacked Autoencoders for Data Representation. Density-based algorithms, traditionally, use the Euclidian distance metric [38]. Such distance metrics are prone to high dimensionality related problems. If we have a feature space of many dimensions, that is, the tracked course of a ship, clustering performance decreases.

Let \mathbf{n} and \mathbf{m} be points drawn from a d -dimensional Gaussian distribution, so that $\mathbf{n} \sim N(\mu_1, \sigma_1^2 \cdot \mathbf{I})$ and $\mathbf{m} \sim N(\mu_2, \sigma_2^2 \cdot \mathbf{I})$. Then their expected distance satisfies [39]

$$\begin{aligned} E \{ \|\mathbf{n} - \mathbf{m}\|^2 \} &= E \left\{ \sum_{i=1}^d |n_i - m_i|^2 \right\} \\ &= \sum_{i=1}^d \{ \text{Var}(n_i - m_i) + E \{ n_i - m_i \}^2 \} \\ &= d \cdot (\sigma_1^2 + \sigma_2^2) + \|\mu_1 - \mu_2\|^2. \end{aligned} \quad (1)$$

Thus, the term $d \cdot (\sigma_1^2 + \sigma_2^2)$, where d is a scalar denoting the dimensions of the Gaussian distribution, overshadows the informative term $\|\mu_1 - \mu_2\|^2$. At this point, the need of robust low-dimension features becomes apparent. In such cases the use of autoencoders is advised [5].

An autoencoder is a neural network that is trained to attempt to copy its input to its output. Internally, it has a

hidden layer h that describes a code used to represent the input. The network may be viewed as consisting of two parts: an encoder function $\mathbf{h} = f(\mathbf{x})$ and a decoder that produces a reconstruction $\mathbf{r} = g(\mathbf{h})$. Autoencoders are designed to be unable to learn to copy perfectly, since they are trained such that $g(f(\mathbf{x})) \approx \mathbf{x}$ instead of $(f(\mathbf{x})) = \mathbf{x}$. The model often learns useful properties of the data, because it is forced to prioritize which aspects of the input should be copied.

Usually, training the autoencoder to perform the input copying task will result in h taking on useful properties, constraining h to have smaller dimension than \mathbf{x} . An autoencoder whose code dimension is less than the input dimension is called undercomplete. Learning an undercomplete representation forces the autoencoder to capture the most salient features of the training data

The learning process is described simply as minimizing a loss function, for example, $L(\mathbf{x}, g(f(\mathbf{x})))$, where L is a loss function penalizing $g(f(\mathbf{x}))$ or being dissimilar from \mathbf{x} , such as the mean squared error. When the decoder is linear and L is the mean squared error, an undercomplete autoencoder learns to span the same subspace as PCA. In this case, an autoencoder trained to perform the copying task has learnt the principal subspace of the training data as a side effect

A sparse autoencoder is simply an autoencoder whose training criterion involves a sparsity penalty $\Omega(\mathbf{h})$ on the code layer \mathbf{h} , in addition to the reconstruction error, that is, $L(\mathbf{x}, g(f(\mathbf{x}))) + \Omega(\mathbf{h})$. Sparse autoencoders are typically used to learn features for another task such as classification. An autoencoder that has been regularized to be sparse must respond to unique statistical features of the dataset it has been trained on, rather than simply acting as an identity function.

The core idea of our work lies in using stacked autoencoders to capture a representation of the main patterns present in the data. By doing so, any outlier in data samples will not be explained well using that representation. In other words, outliers will have significant variations from the rest of the data.

3.4. Identifying Outliers. The outlier detection is a combinatorial threshold-based approach built on the interquartile range rule, as in [40], OPTICS output (see Section 3.2), and AIS/OTH matched data (see Section 3.5).

OPTICS outputs (i.e., reachability distances of the ordered ships) are treated as a continuous signal, over which we identify the peaks. Peaks correspond to significant changes between the closest compared vehicles. As such, anything that varies from the norm has a peak, allowing the easy identification of a possible outlier. Then, we calculate a threshold value $\text{ths}^{(t)}$ defined as $\text{ths}^{(t)} = (1/m) \sum_i^m \mathbf{RD}_o(i)$, $m = \lceil 0.1 \cdot n_t \rceil$, where n_t denotes the number of ships at a time t and \mathbf{RD}_o is the reachability distances vector, in a descending order.

In case that an outlier provides AIS data, the detection regarding that ship is ignored. At first, for a specific time instance, ships are ordered in a density-reachable way (Figure 2). Points close to each other should belong to the same cluster, unless there is a significant change in RD value. Then, the outlier RD value threshold is defined over 10% of highest RDs.

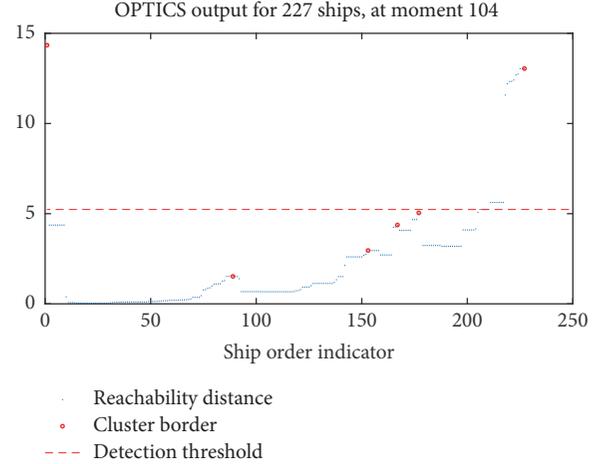


FIGURE 2: (Best viewed in color) illustration of an instance of the outlier detection mechanism at a specific time moment.

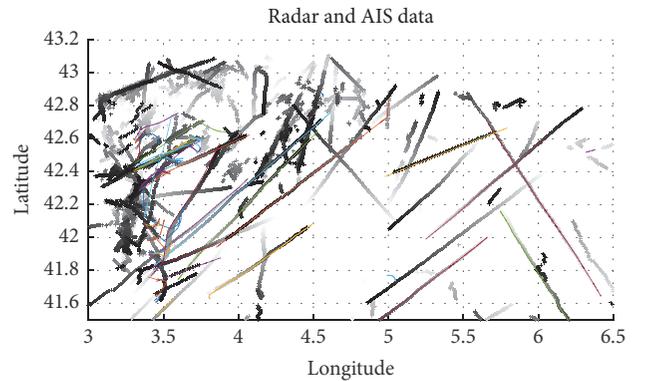


FIGURE 3: (Best viewed in color) an illustration of the investigated ship trajectories. Ground radar trajectories are plotted in grayscale. The fading colors correspond to past times.

3.5. Matching OTH Data to AIS. As explained in Section 3.1, AIS data contain, among others, ships' trajectory points. These coordinates are compared to the radar ones, to identify the similarity among the trajectories. Let us denote as $\mathbf{T}_R^{(v_i)} = [t_1, \dots, t_p]$ the available discrete time instances, created from the ground radar for ship v_i , $i = 1, \dots, n$. The equivalent case for AIS data is $\mathbf{T}_{\text{AIS}}^{(v_j)} = [t_1, \dots, t_q]$ for any ship v_j , $j = 1, \dots, l$, that provides AIS data.

Figure 3 illustrates the available trajectories over a specified area for both radar and AIS data. At this point, we should note that trajectories are calculated for various time intervals, which do not, usually, coincide among the two systems. Typically, for the same ship $p > q$, in a ratio of four radar time instances to one AIS time instance. Also, note that $l < n$, so that a 1-to-1 match among radar and AIS tracked ships is not feasible. Therefore, we should consider both the temporal and the spatial information, to find the matches. The algorithm (presented in pseudocode in Algorithm 1) performs the vessel matching (Figure 4), given OTH and AIS information recorded at the same time (for further details about used data see Section 4.2).

```

Input: OTH and AIS trajectories and other provided information for a set of past time instances  $\{t\}$ 
Output:  $n \times m \times t$  matrix of the closest AIS entries to each OTH entry for a set of past time instances  $\{t\}$ 
FOR each time instance  $t$ 
  FOR each tracked ship  $v_i, i = 1, \dots, n_t$ 
    FOR each AIS transmitting ship  $v_j$ 
      Check difference in track time and AIS transmission time  $T_{diff}^{v_i}$ 
      Find corresponding time instances  $c_t$ 
      IF  $c_t = \emptyset$ 
        No AIS entry is matched to  $v_i$ 
      ELSE
        Run knn search using coordinates among  $v_i$  and  $\{v_j\}_{c_t}$ 
        Maintain 4 closest entries
      END
    END
  END
END
Input:  $n \times m \times t$  matrix of the closest AIS entries to each OTH entry for a set of past time instances  $\{t\}$ 
Output:  $n \times 1$  array with indices of the matched AIS to each one of the  $n$  OTH tracked vehicles
Initialize  $n \times m$  matrix FOR the votes  $V_m$ 
FOR each OTH tracked ship  $v_i$ 
  FOR each time instance  $t$ 
    IF ship  $n_i$  had ship  $m_j$  in the vicinity
       $V_m[i][j] = V_m[i][j] + 1$ 
    END
  END
END
WHILE AIS vessels remain unmatched
  MatchIdxAIS2OTH = arg maxrow( $V_m$ )
  IF length(MatchIdxAIS2OTH) > 1
    KEEP AIS entry closest to OTH vehicle
  END
  Update  $V_m$ 
END

```

ALGORITHM 1: The proposed algorithm in pseudocode.

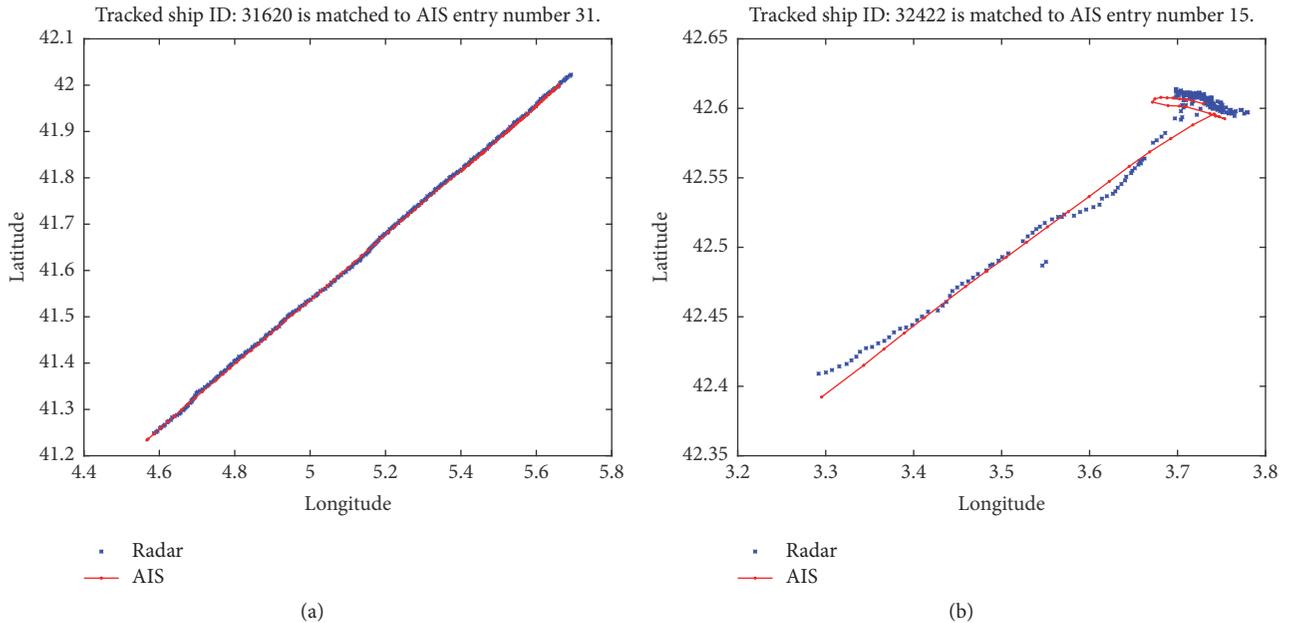


FIGURE 4: (Best viewed in color) illustration of matched trajectories between ground radar and AIS data (a) and matched trajectories despite the noise, due to minor course deviations (b).

TABLE 1: Computational complexity of the different processing steps.

Processing step	Data preprocessing	Data mapping	Data clustering	OTH and AIS matching
Complexity	$O(n)$	$O(n)$	$O(n \cdot \log(n))$	$O(n \cdot m)$

The matching process is based on a voting mechanism. For each of the radar tracked ships v_i , at a time instance t_w , $w = 1, \dots, p$, we calculate the k closest ships v_i , according to their AIS position at the specific time. In order to identify the corresponding (closest) AIS time instance of ship v_i to radar entry v_i , at a time t_w^R , we calculate the time difference $\mathbf{T}_{\text{diff}}^{v_i} = [t_1^{\text{AIS}} - t_w^R, \dots, t_q^{\text{AIS}} - t_w^R]$; then the corresponding time instance is given as $c_t = \arg \min_{\mathbf{T}_{\text{diff}}^{v_i}(m) \geq 0} \{\mathbf{T}_{\text{diff}}^{v_i}(1), \dots, \mathbf{T}_{\text{diff}}^{v_i}(q)\}$. In case that $c_t = \emptyset$, AIS entry v_j is not matched to v_i at time t_w . Then, once we have a set of matched ship instances $M_{v_i}^{t_j} = \{v_r\}_{r=1}^R$, $R < q$, we find k closest entries to v_i according to their position (i.e., longitude, latitude), so that $M_{v_i}^{t_w} = \{v_r\}_{r=1}^k$, $k \ll q$.

4. Experimental Results

In the following subsections, we describe the dataset utilized for the experiments, the performance evaluation metrics employed, and the system setup details, before presenting the experimental evaluation of the proposed framework.

4.1. Computational Complexity. Data preprocessing creates a set of ℓ OTH data related entries, for a predefined set of past moments, for each one of the n_t tracked ships, at a moment t . Since both ℓ and m are constants defined by the user, the required runtime is $O(n)$. The mapping process of a trained SA is $O(1)$ per datum, since SAs are neural networks with a defined number of neurons. OPTICS processes each point once and performs one ϵ -neighborhood query during this processing. Given a spatial index that grants a neighborhood query in $O(\log(n))$ runtime, an overall runtime of $O(n \cdot \log(n))$ is obtained. The matching process between AIS and OTH entries requires an overall runtime of $O(n \cdot m)$, $m_t \ll n_t$, since we compare each of the n_t OTH tracked ships to each of the m_t ships equipped with AIS. Table 1 displays the computational complexity of the different processing steps.

4.2. Utilized Dataset. The utilized dataset pertains to approximately 6 hours of data captured from the Mediterranean coast of France by Diginext in July 2016 in the context of the RANGER EU Horizon 2020 project. AIS data for the same period were also obtained for use as ground truth.

A total of 556 ship entries were in this 6-hour dataset. The following data provided entries are used:

- (1) Longitude and latitude: position values provided in degrees. The typical range is $[-180, 180]$ and $[-90, 90]$, respectively
- (2) Course and speed: course is calculated in degrees, typically in the range $[-180, 180]$, and speed in m/s
- (3) Doppler frequency: it is calculated in Hz, typically in the range $[-0.5, 0.5]$

- (4) Raw Rx azimuth: azimuth angle from the Rx site in the raw spatial grid (equivalent to the reception beam), typically in the range $[110, 230]$
- (5) Local noise: noise level in the surrounding of the plot. It is calculated in dBm, in the range $[-120, -40]$
- (6) Global noise: background noise level of all range-Doppler map. It is calculated in dBm, in the range $[-120, -80]$.

4.3. Performance Metrics. Formally, a cluster analysis can be described as the partitioning a number of N classification objects in K groups or clusters $\{C_k\}$, $k = 1, \dots, K$. Given N objects $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, where \mathbf{x}_j denotes the j th element of \mathbf{x}_i . The grouping of all objects \mathbf{x}_i , $i = 1, \dots, N$, in K clusters can be defined as follows:

$$w_{ki} = \begin{cases} 1, & \text{iff } \mathbf{x}_i \in C_k \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The above formulation ensures that the association of each object to a cluster is unique. A unique association is a valid case for both hierarchical and partitioning cluster analysis. Given matrix \mathbf{W} , various internal quality indices have been calculated, to determine an optimal clustering.

4.3.1. Calinski-Harabasz Index. The Calinski-Harabasz index (CHI) [41] is defined according to the following equation:

$$\text{CHI}(k) = \frac{T_{\mathbf{B}} / (K - 1)}{T_{\mathbf{W}} / (N - K)}, \quad (3)$$

where $T_{\mathbf{B}}$ is defined as

$$T_{\mathbf{B}} = \sum_{k=1}^k |\bar{\mathbf{C}}_k| \|\mathbf{C}_k - \bar{\mathbf{x}}\| \quad (4)$$

and $T_{\mathbf{W}}$ is defined as

$$T_{\mathbf{W}} = \sum_{k=1}^k \sum_{i=1}^N w_{ki} \|\mathbf{x}_i - \bar{\mathbf{C}}_k\|^2. \quad (5)$$

$T_{\mathbf{W}}$ starts at a comparably large value. With increasing number of clusters k , approaching the optimal clustering solution in K^* groups, the value should significantly decrease due to increasing compactness of each cluster. As soon as the optimal solution is exceeded an increase in compactness and thereby a decrease in value might still occur. However, any decrease in value should be notably smaller.

Calculated for each possible cluster solution, the maximum CHI value indicates the best cluster partitioning of the data.

4.3.2. *Davies–Bouldin Index.* The Davies–Bouldin index (DBI) [42] is an internal evaluation scheme, where the validation of how well the clustering has been done is made using quantities and features inherent to the dataset. DBI is defined as follows:

$$DB(k) = \frac{1}{K} \sum_{k=1}^K R_k, \quad (6)$$

where R_k is defined as

$$R_k = \max \left(\frac{\mathcal{S}_k + \mathcal{S}_j}{d_{kj}} \right), \quad j = 1, \dots, K, j \neq k. \quad (7)$$

d_{kj} is a distance function, defined as $d_{kj} = \|\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_j\|$, and \mathcal{S}_k is defined as

$$\mathcal{S}_k = \frac{1}{\sum_{i=1}^N w_{ki}} \sum_{i=1}^N w_{ki} \|\mathbf{x}_i - \bar{\mathbf{x}}_k\|. \quad (8)$$

All the above equations assume that $k \in [1, K]$.

For each cluster C_k an utmost similar cluster—regarding their intracluster error sum of squares—is searched, leading to R_k . The index then defines the average over these values. In this case, the minimum index value corresponds to the best cluster solution.

4.3.3. *Silhouette.* The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from -1 to 1 , where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.

For each datum \mathbf{x}_i , let $\alpha(\mathbf{x}_i)$ be the average dissimilarity (distance) of \mathbf{x}_i with all other data within the same cluster C_k . Let $b(\mathbf{x}_i)$ be the lowest average dissimilarity of \mathbf{x}_i to any other cluster C_l , $l \neq k$, of which \mathbf{x}_i is not a member. We now define a silhouette as

$$s(\mathbf{x}_i) = \frac{b(\mathbf{x}_i) - \alpha(\mathbf{x}_i)}{\max \{\alpha(\mathbf{x}_i), b(\mathbf{x}_i)\}}; \quad (9)$$

thus, $s(\mathbf{x}_i) \in [-1, 1]$. Values close to one indicate that the datum \mathbf{x}_i is appropriately clustered at C_k . The average silhouette value over all data, that is, $\bar{s} = (1/n) \sum_{i=1}^n s(\mathbf{x}_i)$, is another measurement for the quality of the generated clusters.

4.4. *Experimental Setup.* The first step should be the definition of the feature space on which radar data are mapped. As a starting point, we investigated the dimensional space provided by PCA, maintaining 99.1% of the original variation. The adopted stacked autoencoder approach consists of three layers or four layers, depending on the PCA outcome. The loss function was the well-known mean square error [43] with L2 and sparsity regularizers [44].

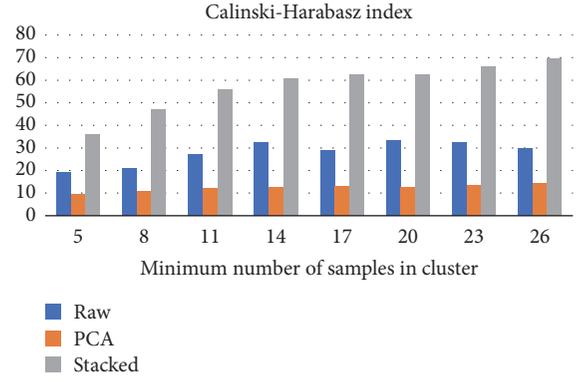


FIGURE 5: The impact of minimum cluster size (OPTICS input parameter) on Calinski–Harabasz index average score. Stacked autoencoders CHI scores are better in all the investigated cases, compared to PCA and raw data based clusters.

Ships track history is composed of 9 consecutive frames, each containing all data as described in Section 4.2. Data are normalized using min-max approach, prior to mapping or clustering approach. The system ignores ships with a narrow appearance span. Any ship that has no enough sufficient entries, that is, 3/4 of past moments tracks, is not taken under consideration.

4.5. *Evaluation of Results.* OPTICS algorithm outcomes depend on the selection of minimum cluster size. We have investigated the clustering outputs assuming at least 2, 5, 8, 11, 14, 17, 20, 23, and 26 members in each cluster. Clustering over SA mapped data performed better than using raw or PCA mapped data, for most of investigated cases.

According to CHI (Figure 5), highest scores are achieved when using 26 ships per cluster. It is intriguing that cluster performance scores over raw data outperform PCA mapped data scores. There is an increasing trend on the CHI as the minimum cluster size increases. The trend is clearly illustrated for SAs, less for raw data, and slightly for PCA projected data.

The next step was the investigation of DBI scores for the same minimum cluster size setup (Figure 6). This time, the best scores are achieved using 14 or 20 as the cluster size. SA mapping provides better clustering scores in five out of seven investigated cases. Regardless of the mapping method, CHI scores, over SA mapped data, improve as the number of clusters rises, but not in a monotonic way.

The last cluster performance metric was the average silhouette distance (Figure 7). Results suggest that accepting two ships as minimum cluster size is the best possible setup, for PCA mapped data. On the other hand, if we use SA for data mapping, the minimum cluster size should be set as 20.

Another significant performance metric is the average reachability distance itself. The smaller the reachability distance of a point is, the higher the density is around it. The core idea of the proposed approach is that only outliers should vary significantly from the norm, on the projected feature space. Thus, all the ships, minus the outliers, should have

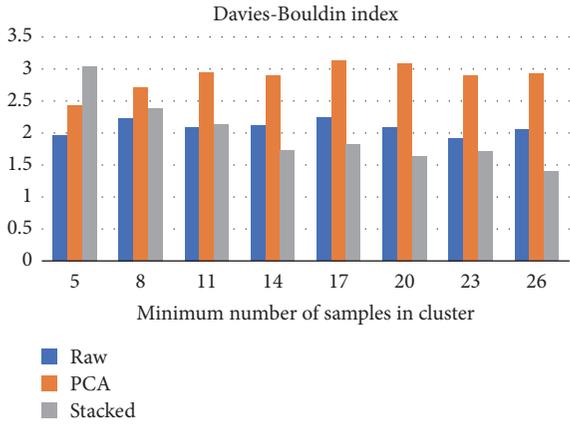


FIGURE 6: The impact of minimum cluster size (OPTICS input parameter) on Davies–Bouldin index average score. Stacked autoencoders CHI scores are better in six out of eight investigated cases, compared to PCA based clusters, and five out of eight cases compared to raw data.

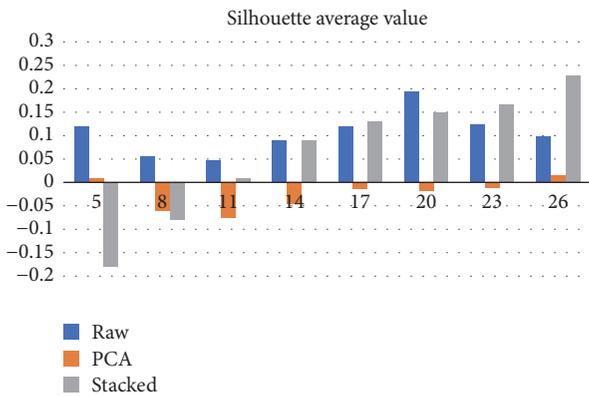


FIGURE 7: Impact of minimum cluster size (OPTICS input parameter) on silhouette average values. Stacked autoencoders silhouette scores are better in five out of eight investigated cases, compared to raw based clusters.

similar feature values, which results in reduced reachability distances.

Providing more training data allows SA to adjust the mapping process to the norm. As illustrated in Figure 8 the average reachability distance tends to one, at a slow pace, while increasing the number of training samples. The variance of the RD is, also, reduced when using more time instances for training, as shown in Figure 9. Furthermore, SA mapping allows for the creation of more clusters compared to PCA or raw data clustering (Figure 10).

Regardless of the adopted feature mapping approach, OPTICS outputs are at least four times less in value, compared to calculated RDs using raw data (see Figures 11(a) and 11(b), top). Additionally, SAs result in more clusters, in most of the cases (see Figures 11(a) and 11(b), bottom). Increasing the number of minimum ships per cluster, close objects have almost identical reachability distances, resulting in almost linear subregions, within RD curve.

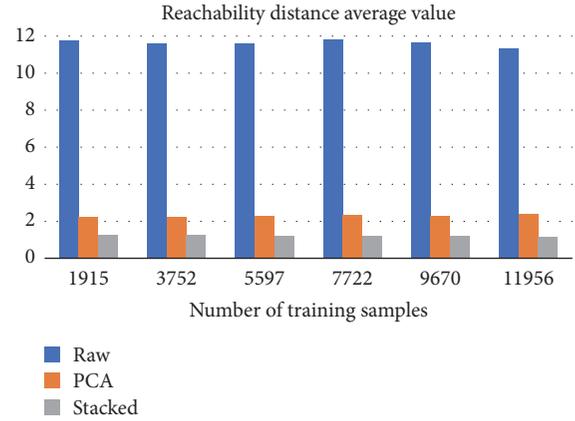


FIGURE 8: An illustration of how the number of training paradigms affects the average reachability distances (OPTICS outputs). Raw data average RD value exceeds 10, in each of the cases.

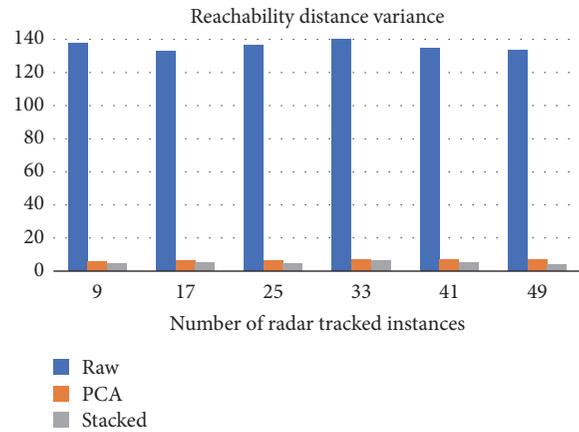


FIGURE 9: Illustration of the training period span effect on the variance in reachability distances. Raw data RD variance exceeds 40, in each case.

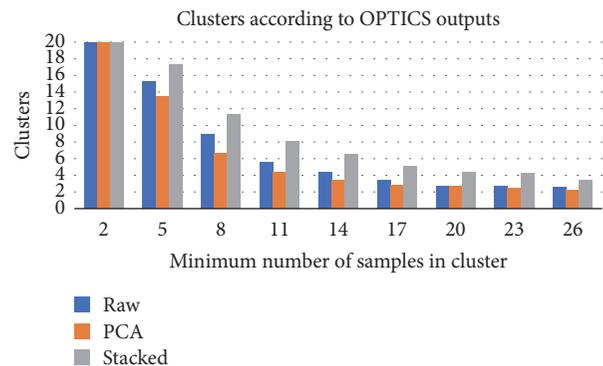


FIGURE 10: Average number of generated clusters given various mapping approaches. In all of the investigated cases (i.e., different minimum cluster size), SAs provide more clusters.

The last step of the performance analysis provides empirical findings. In most of the cases, SAs mapped data results in detection of more outliers compared to the other approaches (Figure 12). The maximum number of detected outliers was three. PCA resulted in no detection at any time.

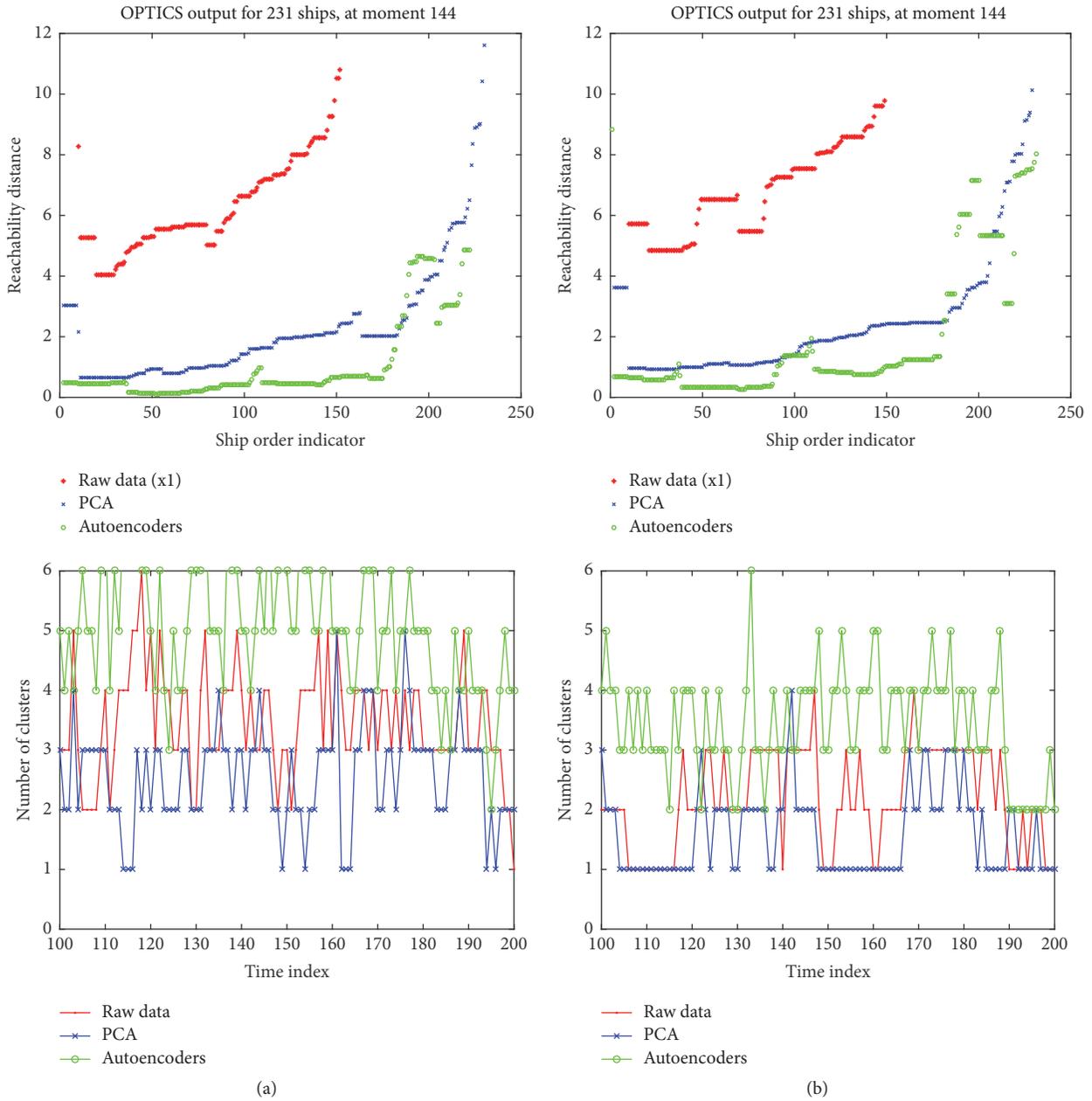


FIGURE 11: (Best viewed in color) comparison of OPTICS outputs over the same time instance, setting as minimum cluster size (a) 20 and (b) 26 ships. Stacked autoencoders result in more clusters than PCA or raw data that implies more peaks in the signal, which leads to more outliers' detection.

There was the possibility of unwanted outlier identification. In particular, ships providing AIS data were considered, a few times, possible outliers. Figure 13 illustrates the case. Typically, using SAs resulted in few possible outliers, which however were not accepted as valid detection, as explained in Section 3.4.

5. Conclusions

In our article, a novel approach that identifies unexpected behavior in ship plot and track patterns, as captured by

an OTH radar, has been presented. The core idea is the unsupervised development of a mapping process, which can project the raw data in a compact, lower feature space. Outliers projected to the same space should have significantly different values. Stacked autoencoders and PCA were used for the mapping process and compared against the exploitation of raw data, for the identification of unusual ship behavior. Density-based clustering algorithms (OPTICS) were employed for clustering-based outlier detection. Experimental results suggest that the approach based on SAs outperforms the other approaches in both generated cluster quality and outliers' identification.

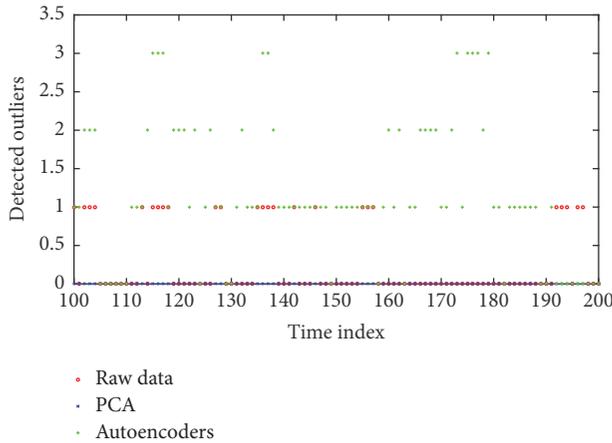


FIGURE 12: (Best viewed in color) illustration of the detected outliers through time. Using SAs' mapped data results in more outliers compared to the other approaches. Some of the selected outliers correspond to ships equipped with AIS transmitters.

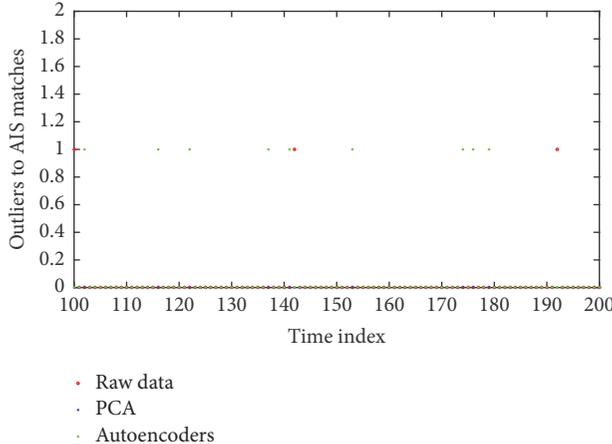


FIGURE 13: (Best viewed in color) illustration of the ships identified as possible outliers, while providing AIS data. Such cases are not considered as outliers.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The research leading to these results has received funding from the European Commission's H2020 Research and Innovation Programme, under Grant Agreement no. 700478 (RANGER project). The authors would like to thank all project partners for their collaboration and especially their partners from Diginext for the provision of the OTH and AIS data and documentation.

References

- [1] L. Sevgi, A. Ponsford, and H. C. Chan, "An integrated maritime surveillance system based on high-frequency surface-wave radars, Part 1. Theoretical background and numerical simulations," *IEEE Antennas and Propagation Magazine*, vol. 43, no. 4, pp. 28–43, 2001.
- [2] S. Grosdidier, A. Baussard, and A. Khenchaf, "HFSW radar model: Simulation and measurement," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 9, pp. 3539–3549, 2010.
- [3] D. Nikolio, Z. Popovic, M. Borenovio et al., "Multi-radar multi-target tracking algorithm for maritime surveillance at OTH distances," in *Proceedings of the 17th International Radar Symposium, IRS 2016*, pp. 1–6, May 2016.
- [4] G. Vivone, P. Braca, and J. Horstmann, "Knowledge-based multitarget ship tracking for HF surface wave radar systems," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 7, pp. 3931–3949, 2015.
- [5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press.
- [6] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '99)*, pp. 49–60, New York, NY, USA, 1999.
- [7] G. Kouemou and F. Opitz, "Impact of Wavelet based signal processing methods in radar classification systems using Hidden Markov Models," in *Proceedings of the 2008 International Radar Symposium, IRS*, pp. 1–4, May 2008.
- [8] M. Garbanzo-Salas and W. K. Hocking, "Spectral analysis comparisons of Fourier-theory-based methods and minimum variance (Capon) methods," *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 132, pp. 92–100, 2015.
- [9] P.-H. Jau, Z.-M. Tsai, N.-C. Kuo et al., "Signal processing for harmonic pulse radar based on spread spectrum technology," *IET Radar, Sonar & Navigation*, vol. 8, no. 3, pp. 242–250, 2014.
- [10] L. Liu, D. Fu, Y. Zhu, D. Su, and M. Diao, "Applications of Bootstrap in Radar Signal Processing," in *Intelligent Computation in Big Data Era*, vol. 503, pp. 474–479, 2015.
- [11] D. Xu, L. Du, H. Liu et al., "Compressive sensing of stepped-frequency radar based on transfer learning," *IEEE Transactions on Signal Processing*, vol. 63, no. 12, pp. 3076–3087, 2015.
- [12] B. A. Massinas, A. Doulamis, N. Doulamis, and D. Paradissis, "An echo state network for ionospheric disturbances behavior modeling on Spaceborne Interferometric Synthetic Aperture Radar," in *Proceedings of the AIAA SPACE 2013 Conference and Exposition*, September 2013.
- [13] B. A. Massinas, A. Doulamis, N. Doulamis, and D. Paradissis, "Applied optimal estimation for ionospheric disturbances behavior on spaceborne interferometric synthetic aperture radar systems," in *Proceedings of the AIAA SPACE Conference and Exposition, 2015*, American Institute of Aeronautics and Astronautics, September 2015.
- [14] B. A. Massinas, A. Doulamis, N. Doulamis, and D. Paradissis, "Ensemble classifiers in optimal estimation for ionospheric disturbances behavior on spaceborne interferometric SAR systems," in *Proceedings of the AIAA Space and Astronautics Forum and Exposition, SPACE 2016*, September 2016.
- [15] M. Cao, S. Li, R. Wang, and N. Li, "Interferometric phase denoising by median patch-based locally optimal wiener filter," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 8, pp. 1730–1734, 2015.
- [16] X. Meng, Z. He, G. Feng, and B. Xiao, "An improved wavelet denoising algorithm for wideband radar targets detection," *Circuits, Systems and Signal Processing*, vol. 32, no. 4, pp. 2003–2026, 2013.

- [17] R. Ahmed, N. Maheshwari, and P. Lalla, "Wavelet based iterative thresholding for denoising of remotely sensed optical and synthetic aperture radar images," in *Proceedings of the IEEE International Conference on Advanced Communication, Control and Computing Technologies (ICACCCT '14)*, pp. 1331–1335, May 2014.
- [18] M. Astry, R. Narayanan, and M. Rangaswamy, "Sparsity-based signal processing for noise radar imaging," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 314–325, 2015.
- [19] M. Gong, J. Zhao, J. Liu, Q. Miao, and L. Jiao, "Change detection in synthetic aperture radar images based on deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 1, pp. 125–138, 2016.
- [20] J. Liu, M. Gong, J. Zhao, H. Li, and L. Jiao, "Difference representation learning using stacked restricted Boltzmann machines for change detection in SAR images," *Soft Computing*, vol. 20, no. 12, pp. 4645–4657, 2016.
- [21] H. Xie, S. Wang, K. Liu, S. Lin, and B. Hou, "Multilayer feature learning for polarimetric synthetic radar data classification," in *Proceedings of the Joint 2014 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2014 and the 35th Canadian Symposium on Remote Sensing, CSRS 2014*, pp. 2818–2821, July 2014.
- [22] S. Maresca, P. Braca, J. Horstmann, and R. Grasso, "A network of HF surface wave radars for maritime surveillance: Preliminary results in the German Bight," in *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2014*, pp. 6077–6081, May 2014.
- [23] S. Maresca, P. Braca, J. Horstmann, and R. Grasso, "Maritime surveillance using multiple high-frequency surface-wave radars," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 8, pp. 5056–5071, 2014.
- [24] P. Braca, S. Maresca, R. Grasso, K. Bryan, and J. Horstmann, "Maritime surveillance with multiple over-the-horizon HFSW radars: An overview of recent experimentation," *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 12, pp. 4–18, 2015.
- [25] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [26] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: a review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.
- [27] K. Makantasis, A. Doulamis, N. Doulamis, and K. Psychas, "Deep learning based human behavior recognition in industrial workflows," in *Proceedings of the 23rd IEEE International Conference on Image Processing, ICIP 2016*, pp. 1609–1613, September 2016.
- [28] N. Doulamis and A. Voulodimos, "FAST-MDL: Fast Adaptive Supervised Training of multi-layered deep learning models for consistent object tracking and classification," in *Proceedings of the 2016 IEEE International Conference on Imaging Systems and Techniques, IST 2016*, pp. 318–323, October 2016.
- [29] G. Hinton, L. Deng, D. Yu et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [30] E. Mason, B. Yonel, and B. Yazici, "Deep learning for radar," in *Proceedings of the 2017 IEEE Radar Conference (RadarConf17)*, pp. 1703–1708, Seattle, WA, USA, May 2017.
- [31] S. Deng, L. Du, C. Li, J. Ding, and H. Liu, "SAR automatic target recognition based on euclidean distance restricted autoencoder," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, no. 99, pp. 1–11, 2017.
- [32] G. Vivone, L. M. Millefiori, P. Braca, and P. Willett, "Performance assessment of vessel dynamic models for long-term prediction using heterogeneous data," *IEEE Transactions on Geoscience and Remote Sensing*, no. 99, pp. 1–14, 2017.
- [33] "HF Surface Wave Radar," [Accessed: 16-Jun-2017] Available: <http://www.diginext.fr/en/offer/critical-operation-support-systems/hf-surface-wave-radar>.
- [34] "Automatic Identification Systems (AIS)," [Accessed: 04-Jun-2017] Available: <http://www.imo.org/en/OurWork/safety/navigation/pages/ais.aspx>.
- [35] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231–240, 2011.
- [36] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD '96)*, pp. 226–231, 1996.
- [37] W. Stuetzle, "Estimating the cluster type of a density by analyzing the minimal spanning tree of a sample," *Journal of Classification*, vol. 20, no. 1, pp. 25–47, 2003.
- [38] E. Protopapadakis and A. Doulamis, "Semi-supervised image meta-filtering using relevance feedback in cultural heritage applications," *International Journal of Heritage in the Digital Era*, vol. 3, no. 4, pp. 613–627, 2014.
- [39] M. Hein, J.-Y. Audibert, and U. von Luxburg, "From graphs to manifolds – weak and strong pointwise consistency of graph Laplacians," in *Learning Theory*, P. Auer and R. Meir, Eds., vol. 3559, pp. 470–485, Springer, Berlin, Germany, 2005.
- [40] F. Xiao and C. Fan, "Data mining in building automation system for improving building operational performance," *Energy and Buildings*, vol. 75, pp. 109–118, 2014.
- [41] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics*, vol. 3, no. 1, pp. 1–27, 1974.
- [42] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1978.
- [43] E. Protopapadakis, M. Schauer, E. Pierri et al., "A genetically optimized neural classifier applied to numerical pile integrity tests considering concrete piles," *Computers & Structures*, vol. 162, pp. 68–79, 2016.
- [44] D. W. Winters, B. D. Van Veen, and S. . Hagness, "A sparsity regularization approach to the electromagnetic inverse scattering problem," *Institute of Electrical and Electronics Engineers. Transactions on Antennas and Propagation*, vol. 58, no. 1, pp. 145–154, 2010.

Research Article

Decoding of Human Movements Based on Deep Brain Local Field Potentials Using Ensemble Neural Networks

Mohammad S. Islam,¹ Khondaker A. Mamun,² and Hai Deng^{1,3}

¹Department of Electrical and Computer Engineering, Florida International University, Miami, FL 33174, USA

²AIMS Lab, Department of Computer Science and Engineering, United International University, Dhaka, Bangladesh

³Nanjing University of Aeronautics and Astronautics, Nanjing, China

Correspondence should be addressed to Khondaker A. Mamun; mamun@cse.uui.ac

Received 19 January 2017; Revised 12 June 2017; Accepted 11 July 2017; Published 19 October 2017

Academic Editor: George A. Papakostas

Copyright © 2017 Mohammad S. Islam et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Decoding neural activities related to voluntary and involuntary movements is fundamental to understanding human brain motor circuits and neuromotor disorders and can lead to the development of neuromotor prosthetic devices for neurorehabilitation. This study explores using recorded deep brain local field potentials (LFPs) for robust movement decoding of Parkinson's disease (PD) and Dystonia patients. The LFP data from voluntary movement activities such as left and right hand index finger clicking were recorded from patients who underwent surgeries for implantation of deep brain stimulation electrodes. Movement-related LFP signal features were extracted by computing instantaneous power related to motor response in different neural frequency bands. An innovative neural network ensemble classifier has been proposed and developed for accurate prediction of finger movement and its forthcoming laterality. The ensemble classifier contains three base neural network classifiers, namely, feedforward, radial basis, and probabilistic neural networks. The majority voting rule is used to fuse the decisions of the three base classifiers to generate the final decision of the ensemble classifier. The overall decoding performance reaches a level of agreement (kappa value) at about 0.729 ± 0.16 for decoding movement from the resting state and about 0.671 ± 0.14 for decoding left and right visually cued movements.

1. Introduction

A fundamental function of the brain-machine interfaces (BMI) is to decode and interpret the recorded neural potentials to classify the patient's intentions or intended behaviors. Such information allows for a better understanding of neuronal circuit mechanisms and enables possible development of treatment methods for neurodegenerative disorders [1].

Deep brain stimulation (DBS) [2–4] is a functional neurosurgical procedure of implanting a miniature medical device to send electronic signals to certain parts of the brain such as subthalamic nucleus (STN) or globus pallidus interna (GPi) in Basal Ganglia (BG) for treatment of movement disorders such as Parkinson's disease (PD) or Dystonia. At the same time, DBS devices can be considered for BMI design and they are able to record the neurosignals called local field potentials (LFPs) [5–7] for body movement prediction

or interpretation. Deep brain LFPs represent the aggregation activities of a large population of local synchronous neurons [5] and can provide neuronal information with better quality (i.e., high SNR) and greater stability over time compared with single-unit activity (SUA). The acquired LFPs from implanted DBS macroelectrodes can be used by researchers and clinicians to investigate on functioning of the Basal Ganglia in motor control [8] for better understanding and more effective treatments of movement disorders [9]. Deep brain LFPs reflect synchronized, subthreshold currents generated in the somata and dendrites of local neuronal elements [10] and they can be subdivided into a number of frequency bands including delta (0–3 Hz), theta (4–7 Hz), alpha (8–12 Hz), beta (13–32 Hz), gamma (31–200 Hz), and high-frequency (>200 Hz) [9] bands. During human body movements, the frequency of the LFP signals can be as high as 300 Hz [7] and is likely to vary due to a varied degree of

behavioral and disease correlation. For example, in case when self-paced (voluntary), externally cued movements or any specified action is intended to be performed, the frequency-dependent event-related synchronization (ERS) and event-related desynchronization (ERD) can be found in various LFP bands recorded in bilateral STNs and/or GPIs [5, 10], which suggests that these oscillations may be related to the preparations of motor response.

With the analysis of intra-operative LFP recordings, it has been found that the frequencies of the synchronized oscillatory activities generally belong to one of two different bands for PD patients withdrawn from dopaminergic therapy [10]. The first band contains activity frequencies (3–12 Hz) of Parkinsonian rest and action tremor, but the signal in this band is neither consistent nor a strong feature of LFPs. However, the second band, called beta band (13–32 Hz), is the frequency range representative of LFP oscillations. This band is antikinetic in nature and is manifested in single-unit activity [10]. Furthermore, for PD patients, the improvement in bradykinesia and rigidity with the subsequent dopaminergic therapy was shown to be correlated to the signal magnitude change in the beta band [9]. However, for PD patients, the oscillatory characteristics of beta frequency band are augmented to such an extent that they dominate over motor commands used for initializing voluntary movements, leading to movement disorders [13]. The most consistent of beta band activities can be found in the untreated, hypodopaminergic Parkinsonian state [14–16]. Recent study also substantiated that the strong signal components in beta frequency band were observed in LFPs recorded from the GPI of PD patients, whereas, for Dystonia patients, the signal in the same frequency band was much less salient [9]. For Essential Tremor (ET), the tremor signals are consistently in the frequency range of 8–27 Hz. For cervical Dystonia, the frequency ranges of 4–10, 11–30, and 65–85 Hz of LFPs are highly correlated to sternocleidomastoid muscle EMG signal frequencies [9]. In addition, ERD in beta band (10–24 Hz) was observed during human movement initiation process and ERS during cessation of movement [9]. At rest and during “OFF” medication Parkinsonian state, alpha (8–12) Hz and beta (13–32) Hz oscillatory activities dominate in the LFP frequency spectra, while they are drastically reduced during “ON” medication state [7]. Moreover, during “OFF” levodopa, the activity in gamma band increases bilaterally during active movement [9] and high-frequency oscillations (HFO) (300–350 Hz) may heighten. In addition, it was also reported that, during “ON” and “OFF” medication states in PD, the extent of power in the frequency band of 4–10 Hz is lower in contrast to Dystonia patients [9]. Although the oscillations in gamma band (>70 Hz) in LFPs that is correlated to human movement (prokinetic) were suppressed [13] or absent in PD patients, during the “ON” medication state, the synchronized oscillatory activity may occur in the STNs and GPIs. Although the evidence suggests that these frequency activities would increase when the body changes from rest to movement, the activities above 65 Hz appear to be an unreliable LFP feature for PD patients [10].

Basal Ganglia STNs activity can be modulated, while patient intends to perform a specified action or watches visual

images of movements [17]. Such intended movements are responsible for generating ERS and ERD in Basal Ganglia which are similar in frequency and time to those during actual voluntary movement [1]. Although the differences in the midst of contra- and ipsi-lateral movement-related oscillatory changes in the STNs have been unknown, some studies suggest that there may not be substantial differences. However, it was also reported recently that, during wrist movement tasks, both contra- and ipsi-lateral ERS were observed in the gamma frequency band [7] but event-related desynchronization (ERD) was found in the low-beta frequency band (~10–24 Hz) [9].

Therefore, multiple frequency-dependent oscillations in motor cortex and BG are directly related to the process of action making, preparations, executions, and imaginations of movements [7]. Recent experimental results showed that, based on distinct oscillations of LFPs, self-paced hand movements can be predicted using a pattern recognition algorithm [18]. The result indicates that LFP activity is directly or indirectly involved in the process of motor preparation. In addition, it is found that the LFPs can be used to infer substantial information about specific types of arm movement parameters such as distance, speed, and directions for motor disorder patients [19, 20]. A recent study showed that movement in eight directions can be decoded with the best recognition rate of up to 92% using the spatial patterns of LFPs in premotor and primary motor areas [19].

Some studies have been conducted to find the coherence and causality between cortex and hand movement. In one study, it was found that noteworthy coherence only exists between the human sensorimotor cortex and contralateral hand and forearm muscles. However, no existence of coherence was found in sensorimotor cortices or any ipsi-lateral hand and forearm muscle [21]. In another study, it was shown that voluntary movement can be decoded up to $76.0 \pm 3.1\%$ using causal strength of LFP signal features computed on neural synchronization of bilateral STNs or GPIs and utilizing bivariate Granger Causality [1]. Additionally, it was found that left and right hand movements are associated with different spatiotemporal patterns of movement-related synchronization and de-synchronization [22]. Therefore, motor control or bilateral coordination can be predicted by decoding movement intention from Basal Ganglia neural activities for left and right hands [1, 7, 12]. These research findings have further demonstrated that LFPs during onset of movement contain supportive information that may advance our knowledge towards reliable movement decoding strategies for neuro-prosthetic device developments, diagnostic assessments, and possible treatment of some chronic neurological disorders. For instance, early prediction of onset of tremor of PD patients may provide the possibility of constructing an adaptive therapeutic intervention mechanism in using DBS for optimal neuromodulation effects [3].

Hence, the prediction and classification of human body movements can be achieved by decoding the recorded BG LFP signals using pattern recognition algorithms. In this paper, we have developed an innovative neural network (NN) based ensemble classifier for effectively decoding the LFP signals recorded from sequential occurrence of movements

and identifying whether the movement is left- or right-sided visually cued in an automated and systematic fashion.

Artificial neural networks (ANNs) [23] are one of the most effective and commonly used machine learning algorithms. However, different types of ANN algorithms possess various advantages and disadvantages in classification. For instance, the FBANN, that is, multi-layer perceptron (MLP), is relatively efficient in optimization or classification with limited training data but tends to be stuck in the local minima and provides less satisfactory classification results [24]. On the other hand, RBFNN could find the global minimum [25] but requires much larger dataset to train. Alternatively, PNN, derived from the Bayes rule and kernel Fisher discriminant, is more accurate than MLP networks and insensitive to outliers in training data [26]. However, PNN needs more training data and is slower than MLP networks in classification. Therefore, it is highly preferable if we can design an ensemble classifier that uses all of the neural networks as the base classifiers for their collective advantages. The ensemble classifier would contain all the advantages of the above-mentioned networks for better activity decoding and classification using LFP dataset. Also, to get robust and consistent movement in decoding performance, we develop a decision fusion algorithm based on the majority voting strategy to combine the classification results from three individual neural networks. The majority voting is simple, intuitive, and effective ensemble approach for improving classification performance [27, 28]. Recently, it has been shown that when seven base classifiers were used in five different ensemble strategies, including majority voting, Bayesian, logistic regression, fuzzy integral, and neural network, the majority voting strategy proved to be as effective as any other algorithm in improving overall classification performance for the dataset provided [28]. We believe that identifying visually cued voluntary movements by decoding oscillatory characteristics of LFP activity may provide ways of developing more advanced neural interface systems such as BCIs and BMIs to enhance our understandings of the underlying process of movements and its important implications in STNs or GPis for controlling movement activities.

2. Experimental Framework and Data Acquisition (DAQ) System

The LFP datasets used in training and testing for movement recognition were recorded through the DBS devices from the patients with Parkinson’s diseases (PD) or Dystonia. The circumstances of the data acquisition are described in detail in this section.

2.1. Patient Details. In this work, a total of twelve Parkinson’s disease or Dystonia patients (7 males and 5 females) with their ages ranging between 23 and 72 years (49.6 ± 13.9 , mean \pm 1SD) were recruited. Each patient underwent bilateral implantation of deep brain stimulation (DBS) electrodes in the STN or GPI for therapeutic stimulation to provide the LFP signals for recording. Their disease-suffering durations were between 3 and 38 years (14.8 ± 10.3 , mean \pm 1SD). The

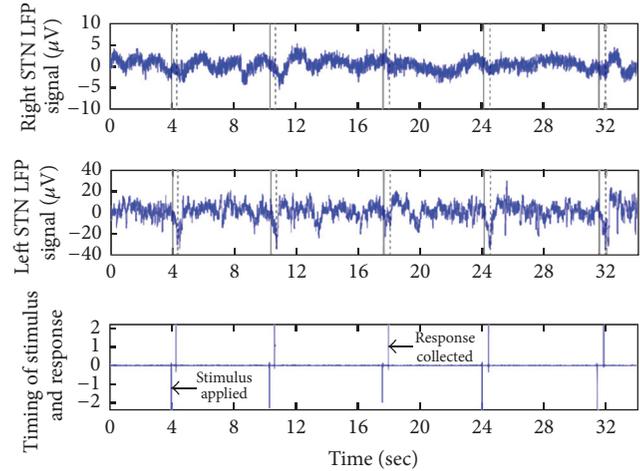


FIGURE 1: Recorded LFP signal from bilateral STNs with chronological visual stimulus applied to the patient. Time of stimulus is presented in solid lines and subsequent motor activity is presented in dotted lines.

corresponding demographics are summarized in Table 1. The LFP data collection was approved by the local research ethics board at Oxford University. All participants provided written consent prior to this study.

2.2. Deep Brain Stimulation (DBS) Electrode Setup. The DBS macroelectrode (model: 3387, manufacturer: Medtronic Neurological Division, Minneapolis, USA) was implanted bilaterally in the left and right STNs or GPis for treatment of the patients with Parkinson’s disease or Dystonia. The macroelectrode consists of four platinum-iridium cylindrical surfaces (diameter: 1.27 mm, length: 1.5 mm, and center to center spacing: 2 mm; contact-0 is the most caudal and contact-3 is the most rostral). Macroelectrodes were inserted after STN and had been identified by using ventriculography and pre-operative magnetic resonance imaging (MRI). Stimulation spots were chosen as the electrode positions, where lessening in Parkinsonian symptoms occurred during intra-operative electrical stimulation and the matching is confirmed by examining the post-operative MRI scan or the fused images of pre-implantation MRI with post-implantation CT.

2.3. Movement Activities of the Patients. During LFP recording from STNs (Figure 1) or GPis, all subjects were instructed to do a finger pressing task in a random order with a short resting period between tasks. Each subject was seated 60 cm (approx.) away from the experimental computer screen. After that, prior to each motor task, they were instructed to keep their left or right index fingers on the distinct keys on the left or right standard keyboard. In addition, all the patients were asked to look at a 10 mm cross that was repetitively displayed in the center of the screen and letter A (height: 8 mm; width: 7 mm) on the screen for the duration of 400 ms instantly to the left or right central cross. It was the indication signal to the patients to move the finger. The interval of cues and laterality were provided randomly in the experiment.

TABLE I: Recording and clinical details of patients.

Patient #	Age	Sex	Years in disease	PD or Dystonia	Elec. placed	Electrode pair used
1	58	F	10	PD	STN	L23/R12
2	63	F	3	PD	STN	L12/R12
3	59	M	7	PD	STN	L01/R01
4	60	M	13	PD	STN	L12/R01
5	72	F	21	PD	GPI	L01/R01
6	55	M	10	PD	STN	L12/R01
7	36	M	14	Dystonia	GPI	L12/R12
8	53	M	5	Dystonia	GPI	L01/R01
9	23	M	7	Dystonia	GPI	L12/R01
10	54	F	38	Dystonia	GPI	L01/R01
11	40	M	25	Dystonia	GPI	L01/R01
12	32	F	24	Dystonia	GPI	L12/R23

2.4. LFP Signal Acquisition from Patients. The LFP signals of twelve patients were recorded at STNs and GPIs for 4–6 days via externalized electrode leads post-operatively after all the patients had been kept “OFF” medication overnight and high-frequency stimulation pulses were completely turned “OFF.” Using MRI, the DBS lead contacts at STNs or GPIs to record LFP signals on both sides were confirmed. Three adjacent pairs consisting of 4 contacts named 0, 1, 2, and 3, respectively (pair positions are 0-1, 1-2, and 2-3), were used to record LFPs in the bipolar signal form and bilaterally. Usually, the bipolar configuration was used to provide “common mode rejection” to far-field activity signals against common mode noise contamination. If DBS stimulation and activity recording are conducted simultaneously, the LFP signal recording can be interfered by the DBS stimulation pulses, leading to inaccurate recording and decoding results. In this experimental setup, we recorded the LFP signals well before the stimulation started to avoid any possible interference of the stimulation pulses to activity recording. DBS macroelectrode pairs were chosen for better therapeutic effects and anatomical structures. After that, the segments of the recorded signal containing erroneous, premature, or no responses were deliberately discarded from the datasets. The number of trials had to be kept at minimum to minimize the stress during the experiment imposed on the PD/Dystonia patients. In the experimental session, 114 ± 43.6 trials (mean \pm 1SD) consisting of minimum 56 and maximum 202 trials across all subjects were employed in the movement decoding process. In addition, for most of the patients, the number of trials is unbalanced for each class. The average number of trials of each class is 58.2 ± 23.6 (mean \pm 1SD) with a minimum of 25 trials and maximum of 113 trials and the average difference between the classes across all the subjects is $14.2\% \pm 19.0$ trials (with a minimum of 1.2% and a maximum of 57.6%). The DBS surgery was only warranted if the patient had exhibited motion-related dysfunction in postural control, gait, and locomotion in addition to usual motor symptoms such as tremor, rigidity, and bradykinesia. Under these circumstances, there will be always challenges with the amount of data with sufficient neuronal information to be collected; therefore to develop

an analysis method that does not rely on a large number of trials is of paramount importance. However, for avoiding rapid repetitive movements and obtaining valid ranges of inter-movement data, the LFP signals obtained outside the time range between 1 s and 5 s during a movement were excluded from the datasets. The contact pair (from bipolar mode: 0-1, 1-2, and 2-3) in the Basal Ganglia were chosen for analysis and showed greatest percentage of beta (β) band (13–32 Hz) modulation due to the movement in contrast to the amplitude of β modulation during the baseline activity period occurring 1-2 seconds before the onset of motor response. The LFP information obtained from the available contact pairs of each electrode would be highly correlated and therefore only one contact pair of each electrode was used for data recording and analysis. In the recording scheme, CED 1902 amplifiers ($\times 10,000$) were employed to amplify the initial signals recorded at the DBS contacts. With tripolar configurations (active-common-reference), surface EMGs were recorded using disposable adhesive Ag/AgCl electrodes (H27P, Kendall-LTP, MA, USA). Based on the recorded EMGs from the index finger, the onset of motor response and other voluntary and involuntary movements were determined by timing of the key presses as registration of motor response. The movement-related artifacts due to equipment lead were carefully identified and the recordings containing excessive noises were excluded from analysis. Contaminated trials with artifact were also removed. In addition, noise of the recorded data related to patients’ movement were avoided as much as possible by instructing patients to stay in steady condition during each session of recordings. In the recorded EMGs, rest and movement conditions were defined as follows: “rest” is defined as no or little hypertonic bursts, “voluntary movements” are defined as regular pulses with a duration of tens of milliseconds, and “uncontrolled contractions” are defined as phasic spasm over seconds. The initial signals were amplified using isolated CED 1902 amplifiers ($\times 10,000$ for LFPs and $\times 1000$ for EMGs), low-pass filtered with a cut-off frequency of 500 Hz, and then digitized using 12-bit CED 1401 mark II with a sampling rate of 2000 Hz. Subsequently, a custom written program in SPIKE 2 (Cambridge Electronic Design (CED), Cambridge,

UK) software was used for recording, online monitoring, and storing the digitized data in the hard drive. Variations of instantaneous magnitude and frequency for both LFPs and EMGs were compared to find correlations between them during movement activities.

2.5. Preprocessing of STN's LFP Signals. For removing high-frequency noise and artifacts, a low-pass type-I Chebyshev filter (zero phase shifting and cut-off frequency 90 Hz) was applied to the STN's LFP signals. A notch filter at 50 Hz was further applied to the processed signals to remove the single-frequency noise associated with the power supplies. Then the LFP datasets were digitally resampled at 256 Hz prior to feature extraction and classification processing.

3. Methodology of Feature Extraction of LFP Signals Using Wavelet Packet Transform (WPT) and Hilbert Transform (HT)

To carry out the identification of finger movements from the LFP data, we used wavelet packet transform (WPT) and Hilbert Transform (HT) to extract the LFP signal features from different frequency bands in the frequency range from 0 to 90 Hz. For non-stationary biosignals such as LFPs, WPT is a better alternative as a data analysis tool than STFT or standard DWT in extracting relevant signal features for pattern recognition in the time-frequency domain [29].

WPT can decompose both approximation and detail spaces into further subbands with functionally distinct scales in a balanced binary tree and has ability to localize any specific information of interest as compared to DWT [30, 31]. In carrying out the WPT at decomposition scale of 5, the discrete Meyer wavelet (demy) was selected and applied to the LFP data to generate different multi-resolution coefficients. The WPT coefficients are obtained by recursively filtering out the coefficients generated in the previous stage with lower resolutions to compute the WPT coefficients at current scale.

After completion of the WPT processing, we segmented a 4-second time window from each frequency band for LFP's left and right clicking event tasks at each motor response registration (Figures 2(a) and 2(b)). Likewise, we can segment the resting activity into a total of 2-second time windows during each stimulus registration. The signal envelope in each frequency band of the reconstructed signal was computed by using the Hilbert Transform (HT) [32] and the signal features were extracted based on the power of each frequency band. From Figure 2(c), it can be seen that event-related synchronization and desynchronization happened in all frequency bands but visible amplitude decrement was found in β band at the left and right STNs or GPIs. However, at the event onset, the signal amplitude in the δ band was quite large compared to those in other bands.

For generating the classification features, instantaneous power was computed by averaging the amplitudes of the defined windows in each frequency band. The window length was either 100 ms or 50 ms and its center locations were varied from -500 ms to +500 ms. Ultimately, based on the left and

right visually cued movements and the oscillatory characteristics of STN's or GPI's LFP signal due to mean energy increment (synchronization) or reduction (desynchronization), the average amplitudes of five consecutive windows (from -150 ms to 350 ms) of length of 100 ms were chosen as the desired period of interest for feature extraction (Figure 2(c)). Similarly, feature extractions were conducted for the resting state. The five windows with a window size of 100 ms from -750 ms to -250 ms were selected to extract features for resting condition (prior to the stimulus applied) of the patients.

Finally, for each patient in each frequency band, vectors of total seventy bilateral features (2 sides \times 7 bands \times 5 points in time) at contra- and ipsi-lateral STNs or GPIs were extracted for decoding voluntary movement and resting activity.

4. Design of the Neural Network Based Ensemble Classifiers for LFP Data Recognition

The objective of the work is firstly to detect if finger movement has happened by decoding deep brain-recorded LFP signals and, if so, subsequently to determine the laterality of that movement. The decoding process, which is actually a two-step three-class classification, consists of LFP data acquisition and preprocessing part, the signal feature extraction part using WPT and HT, and the ensemble classifier that includes three base neural network classifiers and a fusion decision system. The structure of the ensemble classifier for the decoding process is shown in Figure 3.

The proposed overall decoding process using the ensemble classifier in Figure 3 is illustrated in the state diagram in Figure 4. The three base neural network classifiers used in the ensemble classifier will be briefly reviewed and the decision fusion rules and the performance evaluation approaches will be introduced in the rest of the section.

4.1. Three Base Neural Network Classifiers. Three different neural networks that will be used as the base classifiers to form the proposed ensemble classifier will be discussed very briefly in this section.

4.2. Feedforward Backpropagation Artificial Neural Network (FBANN)/Radial Basis Function Neural Network (RBFNN)/Probabilistic Neural Network (PNN). The FBANN [33, 34] was originally designed and trained based on the steepest descent training algorithm. The FBANN network's overall output, θ , with an input vector Xq is computed based on the following equation:

$$\theta = f \left(\sum_{p=1}^n w_p^2 f \left(\sum_{q=1}^m W_{pq}^1 X_q + b_p^1 \right) + b^2 \right), \quad (1)$$

where W_{pq} ($q = 1, 2, \dots, m; p = 1, 2, \dots, n$) are the connection weights, n is the total number of hidden nodes, and m is the total number of the input nodes used to fully connect with the hidden layers. Also, f is the nonlinear activation function. On the other hand, unlike FBANN, the RBFNN consists of an input layer, a hidden layer embedded with a nonlinear RBN activation function, and an output layer [35]. The PNN [26]

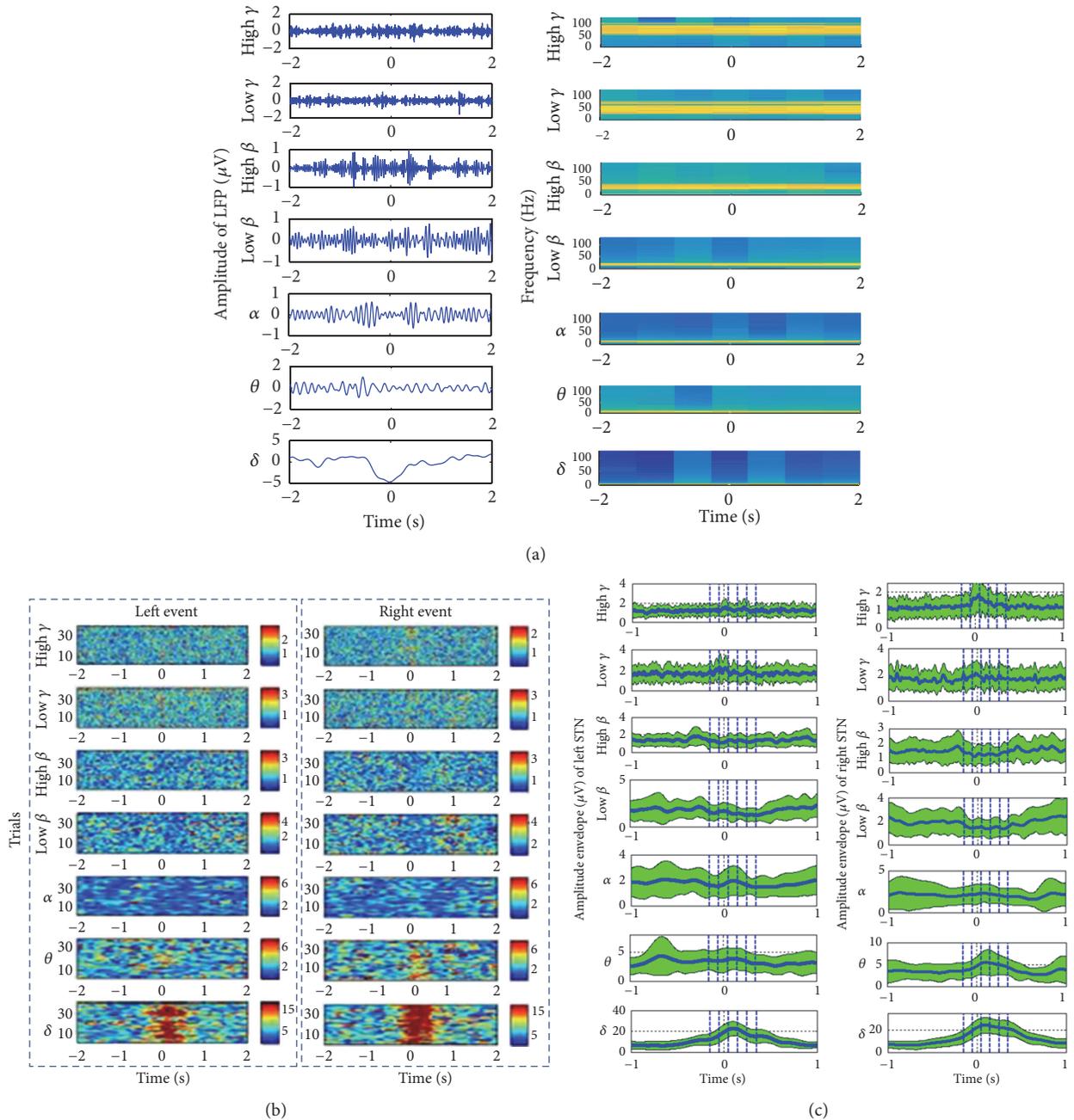


FIGURE 2: (a) Instantaneous amplitude (left) and spectrogram (right) of the right-sided STNs LFP for all frequency bands of patient #1 in a 4-second window centered at the time of response and visually cued left finger clicking events; the extracted frequency bands of LFP signal are delta (0–4 Hz), θ (4–8 Hz), α (8–12 Hz), low β (13–20 Hz), high β (20–32 Hz), low γ (32–60 Hz), and high γ (60–90 Hz), respectively, where high γ band [11] is not the same as the conventional high gamma band (80–200 Hz). (b) The instantaneous magnitude of different bands computed using Hilbert Transform (HT) for all trials of patient #1 during left and right finger visual cued clicking events obtained from deep brain’s left STN LFPs (motor responses situated at the center of each time slice) [12]. (c) The average instantaneous magnitude (blue line) and standard deviation (SD) (green shadow area) acquired from STN LFPs of each component for patient #1 and visual cued left and right finger clicking events within 2 s time window. For each frequency band, LFP signal features were defined with average amplitude in five segments (area covered by dotted line.)

consisting of input, pattern, and decision layers is capable of performing classification tasks for multi-class problems. The decision layer classifies the patterns of the output of the summation layer according to Bayes optimal decision rule.

4.3. Decision Fusion Rule. To obtain an unbiased decision on movement identification, we will use the majority voting-based ensemble classifier for decision fusion processing. For ensemble classifier, the decisions of the base classifiers are

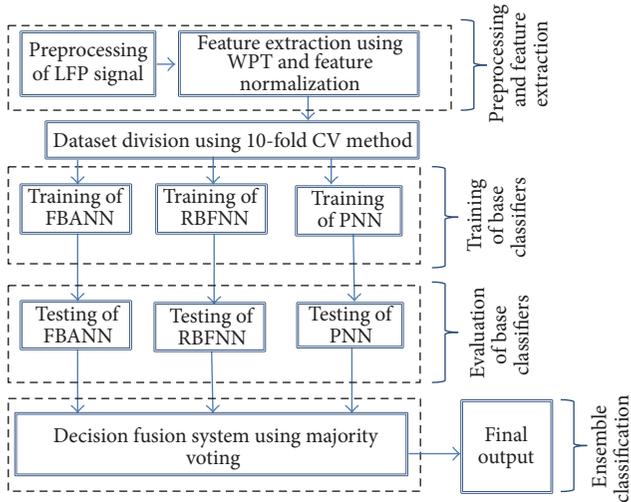


FIGURE 3: Proposed architecture of the ensemble classifier for training, testing, and evaluation.

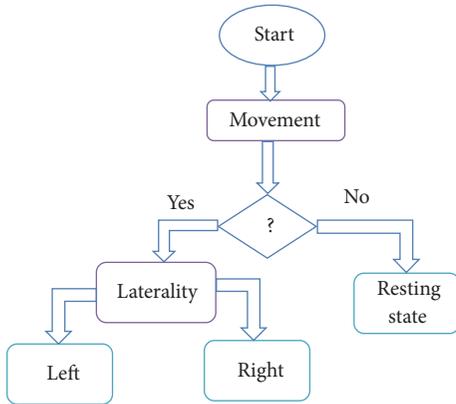


FIGURE 4: Movement detection and its subsequent laterality decoding process using bilateral deep brain's (STNs or GPis) LFP signal.

assumed to be autonomous and the final decisions are derived from a mixture of all base system's decisions [36]. Inherently, in the plurality voting strategy, the ensemble decision picks class w_j , if there is

$$\sum_{t=1}^T d_{t,j} = \max_{j=1}^C \sum_{t=1}^T d_{t,j}, \quad (2)$$

where $d_{t,j}$ is the decision taken by t th base classifier ($t = 1, 2, \dots, T$ and $j = 1, \dots, C$); C is the number of classes and T is the total number of base classifiers used. For plurality voting, if t th classifier predicts class of w_j , then $d_{t,j} = 1$ or 0 for other cases.

In this work, because three base classifiers FBANN, RBFNN, and PNN are used, the majority rule dictates that any two or three base classifiers with the same decision would decide on the acceptance or rejection of the input data as the final decision.

4.4. Performance Evaluation. For classification purpose, a maximum of 28,280 data points from the patients were used

TABLE 2: The distribution of trials used for each patient.

Patient	Number of trials used		
	Left finger movement	Right finger movement	Total
1	52	41	93
2	31	37	68
3	71	84	155
4	31	82	113
5	56	54	110
6	25	31	56
7	61	62	123
8	73	72	145
9	34	28	62
10	59	48	107
11	113	89	202
12	80	76	156

for decoding movement versus rest activities. On the other hand, a maximum of 21,210 data points were employed to decode left- and right-sided visually cued movement activities. In our work, we used bootstrap resampling technique (i.e., random samples were chosen with replacement) in selecting movement and resting datasets of the patients. The corresponding number of trials for movement from each patient is shown in Table 2. Bootstrap is a useful statistical method widely used for classification performance assessment [37]. For a class w_j , if the training set is $X_{N_i}^j = \{x_1^j, x_2^j, x_3^j, \dots, x_{N_i}^j\}$, one can construct the bootstrap samples as follows. Firstly, one sample, $x_{k_0}^j$ from $X_{N_i}^j$, is randomly selected, and the r nearest neighbor samples ($x_{k_1}^j, x_{k_2}^j, \dots, x_{k_r}^j$) from $x_{k_0}^j$ are found based on the Euclidean distance. Then, the bootstrap samples are generated using $x_{k_0}^b = \sum_{i=0}^r w_i x_{k_i}^j$, where $w_i = c_i / \sum_{d=1}^r c_d$; $\sum_r w_i = 1$; and $r \geq c_d \geq 0$ [38]. Gaussian distribution (GD) used to choose c_d and the whole process is repeated until the whole N_i are selected.

To evaluate the overall classification performance of the proposed ensemble classifier, we used 10-fold cross-validation (CV) method to carry out the evaluation. For each design set, CV error was computed according to the following formula:

$$\text{CV error} = \frac{1}{N} \sum_{p=0}^N [d_p(n) - y_p(n)]^2, \quad (3)$$

where N denotes the total number of samples and $d_p(n)$ is the desired output; $y_p(n)$ is the classifier's output for each test set and n denotes the number of conducted epochs. The design sets with the lowest error were considered for the base classifier learning and training. The threshold selection methods for all three base classifiers and the ensemble classifier are summarized in Table 3. Also, the pseudocode for proposed decision fusion algorithm for classification of movement or resting and left or right finger movement activities are listed in Algorithm 1.

```

Threshold value = (FBANN = 1 and RBFNN = 1) or (FBANN = 1 and PNN = 1) or (RBFNN = 1
and PNN = 1);
for  $i = 1$  to number of test vector sets do
if sum of  $\text{sim}(i)$  in (FBANN and RBFNN) or (FBANN and PNN) or (RBFNN and PNN)
 $\geq$  Threshold value
event activity or left finger movement = 1;
prediction( $i$ ) = event activity or left finger movement;
else if sum of  $\text{sim}(i)$  in (FBANN and RBFNN) or (FBANN and PNN) or (RBFNN and
PNN)  $<$  Threshold value
resting activity or right finger movement = 0;
Prediction( $i$ ) = resting activity or right finger movement;
if end
for end
{Find performance metrics of movement and its forthcoming laterality activity from
test data.}

```

ALGORITHM 1: Decision-based pseudocode for decoding event or resting and left or right finger movement activities.

TABLE 3: Threshold settings for individual classifier while detecting the movement and resting activity.

Class	Base classifier threshold setting	Classifier final output after threshold setting
Event condition	≥ 0.5	1
Resting condition	< 0.5	0
Left movement	≥ 0.5	1
Right movement	< 0.5	0

The performance of the proposed ensemble classifier for movement detection and classification was evaluated by using several standard metrics such as cross-validated classification accuracy (CVCA), detection rate (DR), specificity (Table 4), F -measure, TPR, FPR, FNR, kappa, and AUC values. These performance metrics are derived from the standard contingency table based on four commonly used measures (TP/FP/TN/FN) that are commonly adopted in evaluating medical decision systems.

In the contingency table, true positive (TP) is the correct classification rate of the LFP signal generated from the movement state or left movement. True negative (TN) is the correct classification rate of the LFP signal generated from the resting state or right movement. However, false positive (FP) represents the classification rate of the LFP signals as movement or left movement, while they are actually resting state or right movement, respectively. False negative (FN) is the classification rate of the LFP signals as the resting state or right movement when the actual state is movement or left movement.

To obtain highest degree of desirability among the base classifiers to detect movement and resting activity, we have computed unified desirability measures using the following:

$$\text{Desirability}_1 = \frac{\text{mean}_{\text{precision}}}{\text{std}_{\text{precision}}} \times \frac{\text{mean}_{\text{sensitivity}}}{\text{std}_{\text{sensitivity}}} \times \frac{\text{mean}_{\text{specificity}}}{\text{std}_{\text{specificity}}},$$

$$\text{Desirability}_2 = \frac{\text{mean}_{g_{\text{mean-1}}}}{\text{std}_{g_{\text{mean-1}}}} \times \frac{\text{mean}_{g_{\text{mean-2}}}}{\text{std}_{g_{\text{mean-2}}}} \times \frac{\text{mean}_{F\text{-measure}}}{\text{std}_{F\text{-Measure}}},$$

$$\text{Desirability} = \sqrt[6]{\text{Desirability}_1 \times \text{Desirability}_2}. \quad (4)$$

Furthermore, to gauge the correctness of the classifier, we computed Mathew's correlation coefficient (MCC), as shown in (5). MCC in essence is a correlation coefficient between the observed and the predicted binary classification outcomes.

$$\text{MCC} = \frac{((\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN}))}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}. \quad (5)$$

A value of +1 in (5) represents a perfect prediction; a value of 0 represents no better than random prediction and -1 indicates a total disagreement between the prediction and the truth. AUC is the area under the receiver operating characteristic (ROC) curve which is a useful measure in evaluating the performances of binary classification methods [39]. The AUC is defined as follows:

$$\text{AUC} = \frac{1}{2} \left[\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}} \right]. \quad (6)$$

For the sake of convenience and simplicity in comparison, we can compute the AUC values only instead of generating ROC curves, since they would be relatively tedious with the large number of datasets.

Alternatively, to obtain the inflated and more intuitive measure of the performance from the unbalanced datasets of the PD and Dystonia patients, we can use the following balanced accuracy (BACC) [40]:

$$\text{BACC} = \frac{1}{2} (\text{TPR} + \text{TNR}). \quad (7)$$

TABLE 4: Statistical performance measures for decoding of movement and its laterality activity.

Overall accuracy	Sensitivity or DR	Specificity	Overall error rate (OER)
$\frac{TP + TN}{TP + TN + FP + FN}$	$\frac{TP}{TP + FN}$	$\frac{TN}{TN + FP}$	$\frac{FP + FN}{TP + TN + FP + FN}$

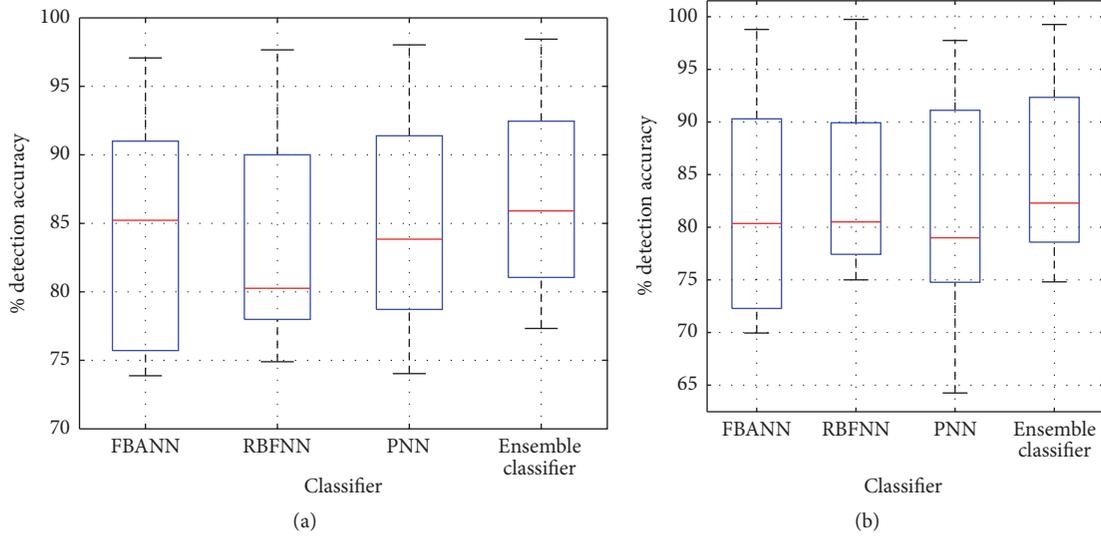


FIGURE 5: (a) Overall accuracy during detection of resting versus movement for all patients. (b) Classification accuracy while decoding left and right movement activity using base and ensemble classifiers.

To further measure the agreement between the predicted and desired classification results in the presence of unbalanced datasets, one can use Cohen's kappa coefficient as the agreement metric [41].

The kappa coefficient (κ) is estimated using the following equation:

$$\kappa = \frac{p_0 - p_e}{1 - p_e}, \quad (8)$$

where p_0 and p_e denote the classification accuracy and the expected agreement of chance, respectively, and these parameters can be calculated from the confusion matrix obtained from the proposed classifier.

If all values of κ within the 95% confidence interval (CI) around the mean are above 0 ($\kappa \pm 1.96 \times \varphi(\kappa) > 0$, where $\varphi(\kappa)$ is the standard error), then the average kappa value is above the chance value. The standard error function, $\varphi(\kappa)$, which is measuring the disagreement, is defined as

$$\varphi(\kappa) = \frac{\sqrt{P_e + P_e^2 - \sum_i [n_{i+} n_{+i} (n_{i+} + n_{+i})]}}{(1 - p_e) \sqrt{N}}, \quad (9)$$

where n_{+i} and n_{i+} are the marginal column and rows sums, respectively, and N is the total number of trials.

5. Experimental Results

Comprehensive computations and simulations of the proposed ensemble classifier have been conducted using the

extracted features for detection of finger movement and subsequent classification of the moving directions. The computer simulations were performed using MATLAB 2012b environment on a PC with 64-bit Intel Core i7-2600 CPU @ 3.40 GHz.

Figures 5(a), 6(a), and 6(b) show the average percentage accuracy, sensitivity, and specificity of the movement decoding for individual patients, respectively. The obtained performance parameters for three base neural networks (mean \pm ISD) are (a) $84.31\% \pm 8.56$ in accuracy, $84.69\% \pm 8.60$ in sensitivity, and $84.77\% \pm 8.11$ in specificity with FBANN; (b) $83.94\% \pm 7.99$ in accuracy, $84.77\% \pm 9.00$ in sensitivity, and $86.25\% \pm 8.71$ in specificity with RBFNN; and (c) $85.03\% \pm 8.30$ (mean \pm ISD) in accuracy, $84.38\% \pm 8.59$ in sensitivity, and $86.16\% \pm 8.65$ in specificity with PNN. With the ensemble classifier, we achieved $87.07\% \pm 7.54$ in accuracy, $87.19\% \pm 7.14$ in sensitivity, and $87.54\% \pm 8.19$ in specificity. These results are about 2–4% better compared to individual base classifier. In addition, from the results in Figures 5(b), 6(c), and 6(d), for laterality of movement decoding, FBANN achieved $82.20\% \pm 10.25$ in overall accuracy, $82.19\% \pm 11.63$ in sensitivity, and $82.80\% \pm 8.00$ in specificity; RBFNN achieved $83.51\% \pm 7.84$ in overall accuracy, $84.78\% \pm 8.93$ in sensitivity, and $87.25\% \pm 9.41$ in specificity; and PNN achieved $81.62\% \pm 11.45$ in overall accuracy, $81.87\% \pm 11.82$ in sensitivity, and $83.95\% \pm 9.70$ in specificity.

The ensemble classifier fused the outputs of three base classifiers (i.e., accuracy in FBANN: 83.042%; in RBFNN: 83.658%; and in PNN: 82.98%) together and achieved 86.073% in detection accuracy, while patients were in resting

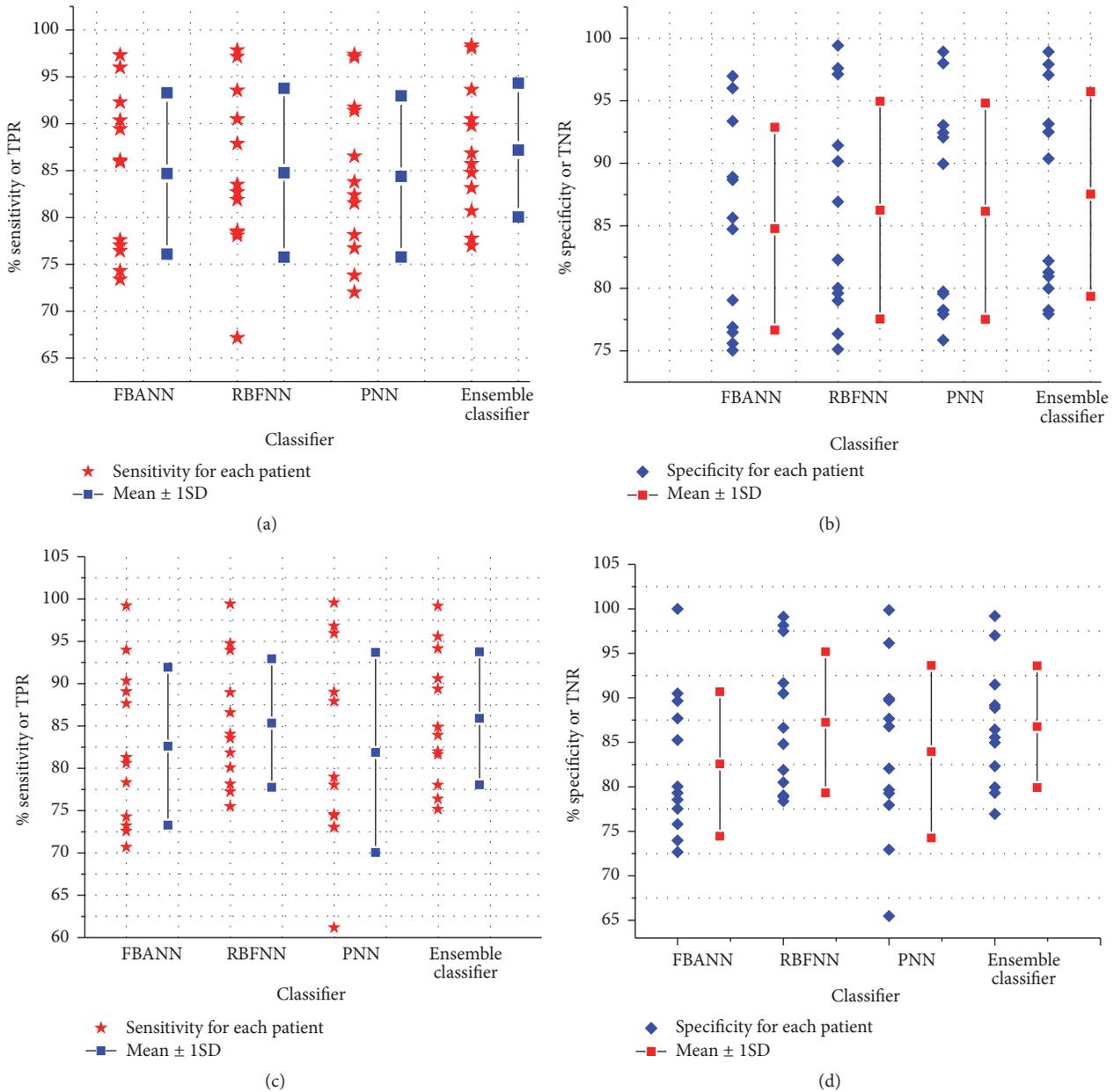


FIGURE 6: (a) Sensitivity during movement versus resting classification obtained from base and ensemble classifier. (b) Specificity of each base and ensemble classifier and detection of movement and resting activity. (c) Sensitivity during classification of left and right finger clicking events obtained from base and ensemble classifier. (d) Specificity obtained from both base and ensemble classifiers during decoding laterality of movement.

state and left or right finger movement activity. Therefore, the overall improvement in detection accuracy of resting from left/right finger movement reached about 3.0% (Figure 9(a)) and the overall error rate (OER) of the detection decreased notably. During movement decoding, RBFNN performed better than PNN and FBANN classifiers in terms of accuracy ($83.94\% \pm 7.99$ versus $87.07\% \pm 7.54$ ($t(22) = -0.9866$, $p < 0.05$)), sensitivity ($84.77\% \pm 9.00$ versus $87.19\% \pm 7.14$ ($t(22) = -0.7300$, $p < 0.05$)), and specificity ($86.25\% \pm 8.71$ versus $87.54\% \pm 8.19$ ($t(22) = -0.3728$, $p < 0.05$)). For laterality

decoding, RBFNN still managed to achieve better performance than the other two base classifiers in accuracy, sensitivity, and specificity.

Essentially, with various feature set sizes, all the classifiers managed high degree of classification accuracy. RBFNN achieved a lower false positive rate (FPR) but has lower detection rate than PNN classifier in decoding movement. On the other hand, in movement laterality decoding, RBFNN classifier maintained less intra-subject variability in performance than the other two base classifiers. Overall, RBFNN achieved

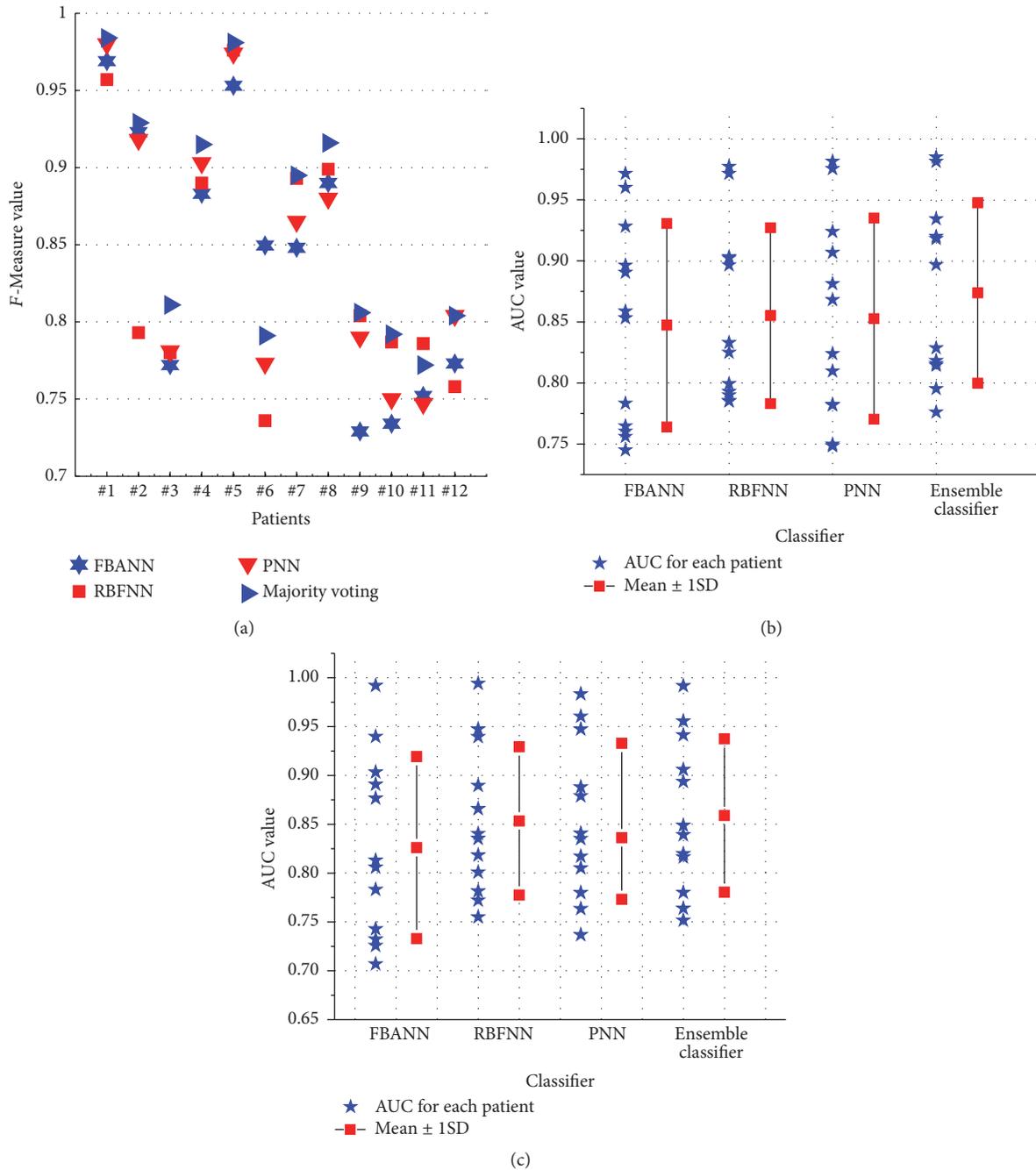


FIGURE 7: (a) *F*-Measure plot of base and ensemble classifier during movement versus resting classification. (b) Area under the ROC curve (AUC) for each patient with mean \pm 1SD obtained from each base and ensemble classifier during movement versus resting classification. (c) Area under the ROC curve (AUC) for each patient with mean \pm 1SD obtained from each base and ensemble classifier during left and right finger movement classification.

the highest classification rate as well as highest specificity among the three base classifiers. It also performs advantageously in comparison to PNN and FBANN in terms of balanced accuracy (Table 6). Although RBFNN classifier has achieved lowest FPR for both movement and laterality classifications compared to others, it did have higher value of FNR compared to FBANN; more importantly it achieved higher TPR and TNR values than PNN algorithm.

To show the impact of the imbalanced classes on the performance, we obtained the AUC values for each classifier in movement and laterality decoding as shown in Figures 7(b) and 7(c). It is found that the average value of AUC (0.873) with the ensemble classifier is greater than those with any individual classifier. Similarly, in laterality of movement decoding, the ensemble classifier achieved better AUC (0.859) values than any base classifier.

TABLE 5: Standardized 1st-order moment of evaluation measures.

Classifiers evaluated	F -Measure	g mean-1	g mean-2	Desirability value
During decoding movement and resting				
FBANN	9.77	9.92	10.17	9.13
RBFNN	10.41	11.52	11.74	10.25
PNN	10.08	10.57	10.32	10.02
Ensemble	11.08	11.78	11.82	11.22
For decoding left and right finger movement				
FBANN	7.27	8.09	8.62	8.25
RBFNN	8.37	11.30	11.90	9.39
PNN	6.88	7.96	8.15	7.58
Ensemble	8.46	9.58	10.98	9.67

Although all the base classifiers performed well in detecting movement and its forthcoming laterality, the ensemble classifier based on the majority voting algorithm performed better than the base classifiers in detection, especially in terms of FNR. The FNR rate of the ensemble classifier is improved by 2.98% compared with that of any of the base classifiers in decoding movement versus resting of the patients.

Moreover, we computed other distinctive performance indicators such as F -measure (Figure 7(a)), g mean-1, and g mean-2. Obviously, larger F -measure values indicate finer precision and higher sensitivity. g -mean value measures the balanced performance of the classifiers between sensitivity, specificity, and precision.

Standardized 1st-order moments of F -measure and g -mean values for this work are tabulated in Table 5. It is observed from Table 5 that RBFNN performs the best in terms of F -measures. In decoding resting versus movement and its forthcoming laterality, RBFNN also shows the highest degree of desirability, since it achieves the highest desirability value among other base classifiers.

From Table 6, it can be seen that PNN and RBFNN classifiers demonstrated better MCC results, showing better agreement between the prediction and actual results in detecting movement and classifying laterality of movement. RBFNN classifier achieves higher BACC value than other base classifiers. However, with the ensemble classifier, the BACC value is improved by at least 2.63% compared to any base classifier in movement and resting classification.

The data from all the patients demonstrated good kappa coefficient values using each classifier while classifying their LFP patterns of movement and laterality (overall values shown in Table 6). The experimental results also showed that the highest kappa value (0.692 ± 0.17) (mean \pm ISD) is obtained using PNN classifier in discriminating movement from resting activity. For movement laterality classification, it managed to have a value of 0.590 ± 0.28 (mean \pm ISD), which indicates a good agreement between actual and predicted identifications.

Figure 8 shows the kappa values using the ensemble classifier for the datasets generated from all twelve patients. Individual kappa value suitably exceeded 0.4, which is equivalent to an accuracy of $>70\%$. An accuracy of 70% is considered necessary for meaningful communication with a 2-class

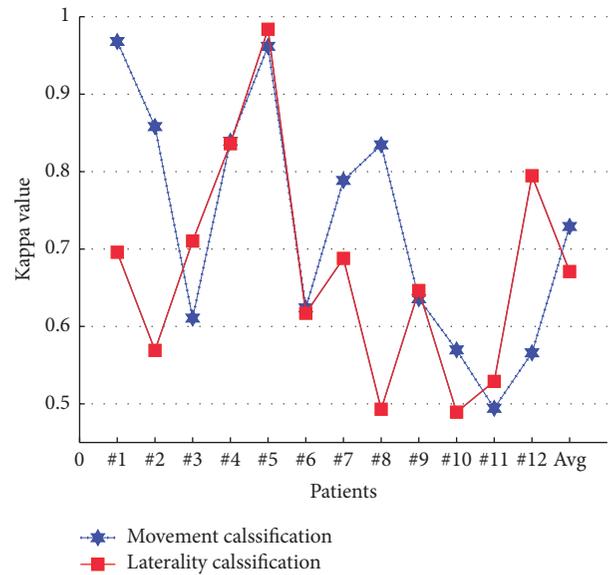


FIGURE 8: Plot of overall kappa value for movement and its laterality classifications using the ensemble classifier for all participating patients.

BCI [42]. Additionally, a very good agreement between the intended and predicted selections (kappa > 0.8 with the peak at 0.96 equivalent to decoding accuracy $> 95\%$) was achieved for two participants while detecting movement and its forthcoming laterality.

Since we do not have enough information such as patient disease severity and handedness, it is difficult to do correlation analysis between movement decoding performance and disease situations. However, based on demographic data, as shown in Table 1, we have computed movement decoding performances, as shown in Table 7. It can be seen that, according to disease types, the patients with Parkinson's disease (PWP) exhibited much higher movement decoding rate than the Dystonia patients. Similarly, the decoding activity of LFP signals recorded through DBS electrodes from STNs achieved higher average accuracy than GPis.

To show further robustness of three-class (resting and left or right hand finger movements) classification performance

TABLE 6: Statistical significance measures of the classifiers while decoding movement from the resting.

Classifiers evaluated	MCC value	Overall kappa value, $\bar{k} \pm \varphi(k)$	BACC
For decoding movement versus resting			
FBANN	0.691	0.607 ± 0.25	84.73
RBFNN	0.691	0.599 ± 0.27	85.51
PNN	0.701	0.692 ± 0.17	85.27
Ensemble	0.737	0.729 ± 0.16	87.36
For decoding left versus right finger movement			
FBANN	0.647	0.634 ± 0.20	82.49
RBFNN	0.665	0.563 ± 0.23	85.96
PNN	0.636	0.590 ± 0.28	82.91
Ensemble	0.712	0.671 ± 0.14	85.96

TABLE 7: Comparison of detection performances of the ensemble classifier for disease conditions and groups of patients.

Patient groups/LFP signal collection methods	Overall accuracy (%)	TPR (%)	TNR (%)
Patients with PD	88.79	88.17	92.33
Patients with Dystonia	82.03	81.55	81.81
LFPs from STNs	89.14	86.56	89.87
LFPs from GPIs	84.62	83.95	84.55

TABLE 8: Statistical performance measures of base and ensemble classifiers to classify resting and left or right finger movement.

Classifiers	<i>F</i> -Measure	AUC	Kappa	False positive rate (FPR)
FBANN	0.8277	83.67	0.6154	16.28
PNN	0.8433	85.29	0.6687	13.95
RBFNN	0.8365	85.42	0.5905	13.26
<i>Ensemble</i>	<i>0.8574</i>	<i>86.63</i>	<i>0.7067</i>	<i>12.88</i>

using the ensemble neural network (NN), we have computed numerical performance metrics based on available datasets from 12 PD and Dystonia patients. The results are presented in Figure 9. It can be seen that the ensemble classifier has better performances than any of the base classifiers (accuracy ~3% better than individual classifier). Furthermore, the majority voting also showed greater sensitivity and specificity (Figures 9(b) and 9(c)) as compared to the base classifiers. Other performance measures for both the ensemble and the base classifiers are shown in Table 8.

6. Further Discussions

This work investigated the potential advantages of neural network ensemble classifiers for decoding of human finger movements or resting activity using deep brain local potential signals. The aforementioned testing results show that the average decoding performance during movement and its laterality decoding process using the proposed ensemble classifier is very promising and this methodological framework may lead to the development of more effective BMI applications. With various feature set sizes, it was demonstrated that RBFNN has been proven to be better decoder

by managing impressive overall classification rate (CR) and PNN has shown the worst performance among the three weak learners. The RBFNN classifier performs advantageously over PNN and FBANN in terms of balanced accuracy with the lowest false detection rate. However, a few factors could have degraded the classifiers' performance; they are the unbalanced number of trials in the dataset, the unbalanced variability within the classes, the higher redundancy, and the unbalanced variation in the feature sets. Further additional factors need to be considered such as magnitude variation among the frequency bands as limited or less expertise of the participants to execute action according to stimulus applied, motivation and concentration to respond, and patient insensitiveness due to fatigue, age, and patient's depth of diseases.

(i) The total number of trials for the clicking events taken from each patient showed potential variation of decoding performance [43]. In the experimental session, obtained LFP datasets were limited in size. 114 ± 43.6 trials (mean \pm 1SD) consisted of minimum 56 and maximum 202 trials across all subjects employed in the movement decoding process. For most of the patients, the number of trials is unbalanced for each class. The average number of trials of each class is 58.2 ± 23.6 (mean \pm 1SD) with a minimum of 25 trials and maximum of 113 trials. This unbalancedness of the trials may contribute to the increase of the overall error rate (OER) for some participants in decoding. However, recent researches also suggest that a larger number of trials are needed to more accurately and robustly assess the predictive model [42].

(ii) It can be seen from Figures 6(c) and 6(d) that patients rapidly and efficiently responded during visually cued right hand finger clicking events compared to left finger clicking events for both STN's and GPI's LFP signal (overall specificity: $87.07\% \pm 7.21$; overall sensitivity: $84.86\% \pm 9.54$ (mean \pm 1SD)). Although we had no abundant information

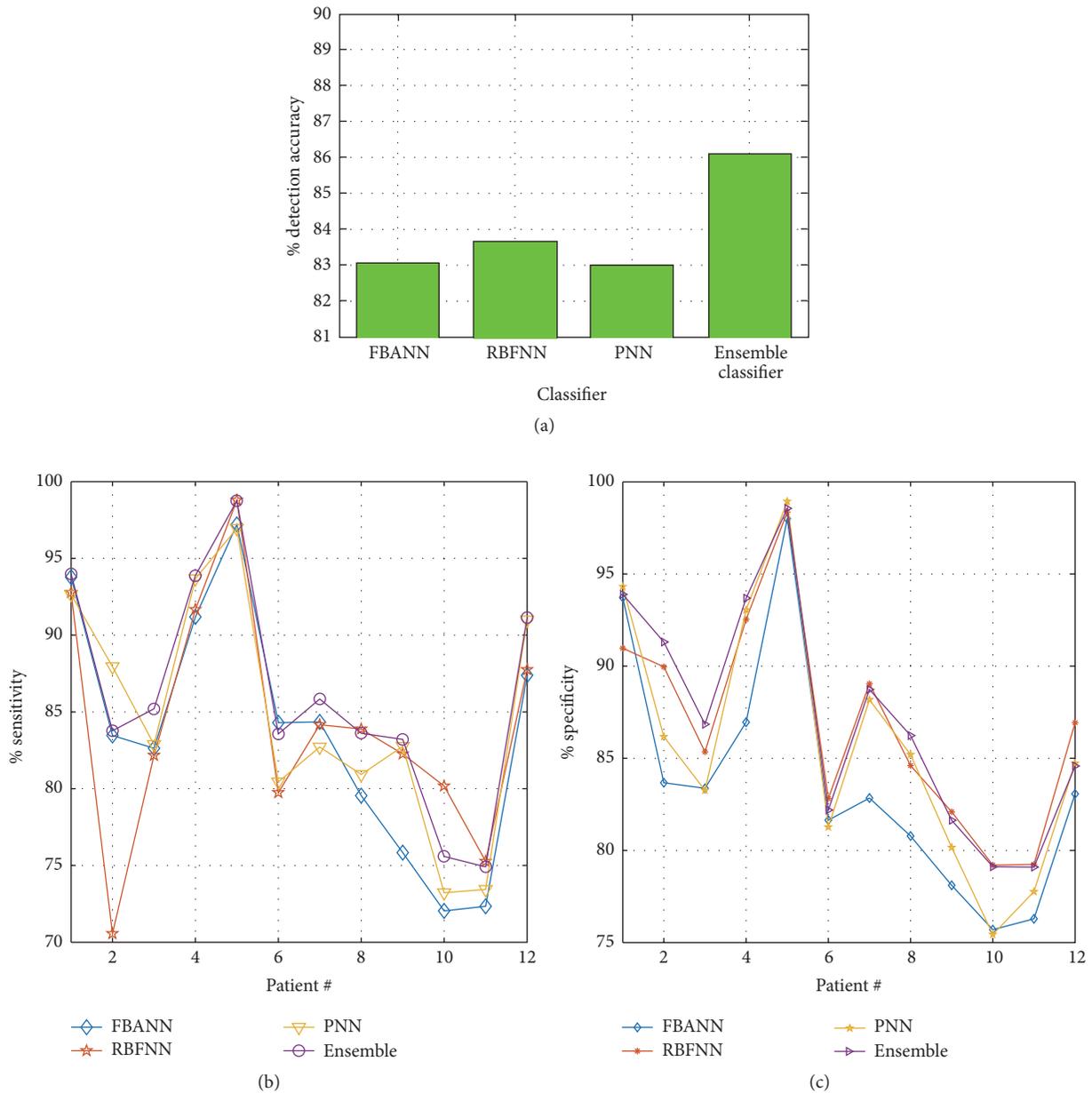


FIGURE 9: (a) Overall detection accuracy, (b) sensitivity, and (c) specificity of resting, left, and right finger movement activities obtained from base and ensemble classifiers.

about handedness of the patients, it can be considered that right handed patients were better trained than left handed patients due to generality of right handedness among human inhabitants.

(iii) LFPs obtained from the patients are more stable than single-neuron activity or noninvasive EEG; nonetheless it can contaminate with conspicuous motion artifact and patient insensitiveness due to fatigue, which ultimately deteriorates LFP signal momentarily during onset of movement event. The proposed decoder system has shown its effectiveness by addressing the aforementioned limitation to a greater extent by using different types and range of patients.

(iv) Although two-session recordings were obtained from four of the participants, LFP recordings from each patient used in this study are involved in a single session only. As a future work, several sessions will be recorded and single-session features will be enforced as a test set, while remaining sessions will be used to train the intelligent classifier to decide substantial, stable, and trustworthy decoding outcome. However, we will carry out further research on early prediction of movement conducted by normal and abnormal people in a controlled and distraction-free environment that is applicable in widespread neuro-interface scenario. With consideration of the above limitations, the performance of the proposed

ensemble classifier in LFP movement detection and classification is very encouraging. To the best of our knowledge, these results achieved in this work are better than those reported previously in the literature in terms of detection rate and the number of patients [1]. Theoretically, the proposed two-layer two-class classifier could be replaced with a three-class classifier. However, our results showed that two-class classifiers are more robust for the datasets used in this work.

7. Conclusion

This study explores an innovative neural network ensemble classifier for effective identification of voluntary movements extracted from oscillatory activity of LFP signals recorded bilaterally in the STN or GPI of twelve Parkinson's disease and Dystonia patients. A majority voting algorithm is used in the ensemble classifier to fuse the results from three individual neural network classifiers. The experimental results demonstrate that decoding rate of clicking events is greater than its laterality of clicking ($87.07\% \pm 7.54$ versus $85.41\% \pm 8.68$ (mean \pm ISD)) using the ensemble neural network classifier. The performances of movement decoding for each base classifier were investigated and evaluated and it is found that the ensemble classifier is consistently better than the base classifiers or other similar classifiers in terms of convergence rate as well as classification accuracy. The results also demonstrated that PNN achieves the best detection accuracy (DA) ($85.03\% \pm 8.30$ (mean \pm ISD)) among those three base classifiers in identifying event. In predicting sequential clicking events, RBFNN ($83.51\% \pm 7.84$ (mean \pm ISD)) outperforms FBANN and PNN. The proposed optimal classifying system may provide a channel for developing wearable and wireless smart stimulation devices that can predict involuntary movements (such as tremor) and adaptively respond to the onset of abnormal neurological events. With three different neural networks as the base classifiers, the classification performance improvement of the ensemble classifier appeared to be modest and yet noticeable. However, ensemble classifier was demonstrated to be an effective approach to improving human finger movement decoding and interpretation performance. It should be pointed out that the real-time convergence is a very important issue for any classification algorithm; however, the investigation of the proposed ensemble classifier is limited to offline analysis at this stage. Our future work in this area includes improvement of better feature extraction algorithms and the optimization of the base classifiers for the ensemble classifier.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors are very grateful to Oxford Functional Neurosurgery Group at University of Oxford in UK for providing the datasets necessary for the successful completion of this research work. The authors are also grateful to

United International University (UIU), Bangladesh (UIU-RG-161010), for supporting this research work.

References

- [1] K. A. Mamun, M. N. Huda, M. MacE et al., "Pattern classification of deep brain local field potentials for brain computer interfaces," in *Proceedings of the 15th International Conference on Computer and Information Technology, ICCIT 2012*, pp. 518–523, bgd, December 2012.
- [2] C. O. Oluigbo, A. Salma, and A. R. Rezaei, "Deep brain stimulation for neurological disorders," *IEEE Reviews in Biomedical Engineering*, vol. 5, pp. 88–99, 2012.
- [3] M. L. Kringelbach, N. Jenkinson, S. L. F. Owen, and T. Z. Aziz, "Translational principles of deep brain stimulation," *Nature Reviews Neuroscience*, vol. 8, no. 8, pp. 623–635, 2007.
- [4] B. Rosin, M. Slovik, R. Mitelman et al., "Closed-loop deep brain stimulation is superior in ameliorating parkinsonism," *Neuron*, vol. 72, no. 2, pp. 370–384, 2011.
- [5] P. Brown and D. Williams, "Basal ganglia local field potential activity: Character and functional significance in the human," *Clinical Neurophysiology*, vol. 116, no. 11, pp. 2510–2519, 2005.
- [6] A. Mazzoni, N. K. Logothetis, and S. Panzeri, *The Information Content of Local Field Potentials: Experiments and Models*, 2012.
- [7] M. Alegre, F. Alonso-Frech, M. C. Rodríguez-Oroz et al., "Movement-related changes in oscillatory activity in the human subthalamic nucleus: Ipsilateral vs. contralateral movements," *European Journal of Neuroscience*, vol. 22, no. 9, pp. 2315–2324, 2005.
- [8] N. F. Ince, R. Gupta, S. Arica, A. H. Tewfik, J. Ashe, and G. Pellizzer, "High accuracy decoding of movement target direction in non-human primates based on common spatial patterns of local field potentials," *PLoS ONE*, vol. 5, no. 12, Article ID e14384, 2010.
- [9] J. A. Thompson, D. Lanctin, N. F. Ince, and A. Abosch, "Clinical implications of local field potentials for understanding and treating movement disorders," *Stereotactic and Functional Neurosurgery*, vol. 92, no. 4, pp. 251–263, 2014.
- [10] T. Boraud, P. Brown, J. A. Goldberg, A. M. Graybiel, and P. J. Magill, "Oscillations in the Basal Ganglia: The good, the bad, and the unexpected," *Advances in Behavioral Biology*, vol. 56, p. 24, 2005.
- [11] S. Ray and J. H. R. Maunsell, "Different origins of gamma rhythm and high-gamma activity in macaque visual cortex," *PLoS Biology*, vol. 9, no. 4, Article ID e1000610, 2011.
- [12] K. A. Mamun, M. MacE, M. E. Lutman et al., "A robust strategy for decoding movements from deep brain local field potentials to facilitate brain machine interfaces," in *Proceedings of the 2012 4th IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechanics, BioRob 2012*, pp. 320–325, ita, June 2012.
- [13] W. D. Hutchison, J. O. Dostrovsky, J. R. Walters et al., "Neuronal oscillations in the basal ganglia and movement disorders: Evidence from whole animal and human recordings," *Journal of Neuroscience*, vol. 24, no. 42, pp. 9240–9243, 2004.
- [14] P. Brown, A. Oliviero, P. Mazzone, A. Insola, P. Tonali, and V. Di Lazzaro, "Dopamine dependency of oscillations between subthalamic nucleus and pallidum in Parkinson's disease," *Journal of Neuroscience*, vol. 21, no. 3, pp. 1033–1038, 2001.
- [15] M. Cassidy, P. Mazzone, A. Oliviero et al., "Movement-related changes in synchronization in the human basal ganglia," *Brain*, vol. 125, no. 6, pp. 1235–1246, 2002.

- [16] A. Priori, G. Foffani, A. Pesenti et al., "Movement-related modulation of neural activity in human basal ganglia and its L-DOPA dependency: Recordings from deep brain stimulation electrodes in patients with Parkinson's disease," *Neurological Sciences*, vol. 23, no. 2, pp. S101–S102, 2002.
- [17] A. A. Kühn, L. Doyle, A. Pogosyan et al., "Modulation of beta oscillations in the subthalamic area during motor imagery in Parkinson's disease," *Brain*, vol. 129, no. 3, pp. 695–706, 2006.
- [18] C. Loukas and P. Brown, "Online prediction of self-paced hand-movements from subthalamic activity using neural networks in Parkinson's disease," *Journal of Neuroscience Methods*, vol. 137, no. 2, pp. 193–205, 2004.
- [19] N. F. Ince, R. Gupta, S. Arica, A. H. Tewfik, J. Ashe, and G. Pellizzer, "Movement direction decoding with spatial patterns of local field potentials," in *Proceedings of the 2009 4th International IEEE/EMBS Conference on Neural Engineering, NER '09*, pp. 291–294, tur, May 2009.
- [20] C. Mehring, M. P. Nawrot, S. C. De Oliveira et al., "Comparing information about arm movement direction in single channels of local and epicortical field potentials from monkey and human motor cortex," *Journal of Physiology Paris*, vol. 98, no. 4-6, pp. 498–506, 2004.
- [21] J. M. Kilner, S. Salenius, S. N. Baker, A. Jackson, R. Hari, and R. N. Lemon, "Task-dependent modulations of cortical oscillatory activity in human subjects during a bimanual precision grip task," *NeuroImage*, vol. 18, no. 1, pp. 67–73, 2003.
- [22] O. Bai, Z. Mari, S. Vorbach, and M. Hallett, "Asymmetric spatio-temporal patterns of event-related desynchronization preceding voluntary sequential finger movements: A high-resolution EEG study," *Clinical Neurophysiology*, vol. 116, no. 5, pp. 1213–1221, 2005.
- [23] T. Wu, B. Yang, and H. Sun, "EEG classification based on artificial neural network in brain computer interface," *Communications in Computer and Information Science*, vol. 97, no. 1, pp. 154–162, 2010.
- [24] K. Fukumizu and S. Amari, "Local minima and plateaus in hierarchical structures of multilayer perceptrons," *Neural Networks*, vol. 13, no. 3, pp. 317–327, 2000.
- [25] T. Xie, H. Yu, and B. Wilamowski, "Comparison between traditional neural networks and radial basis function networks," in *Proceedings of the 2011 IEEE International Symposium on Industrial Electronics, ISIE 2011*, pp. 1194–1199, pol, June 2011.
- [26] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, no. 1, pp. 109–118, 1990.
- [27] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.
- [28] L. Lam and C. Y. Suen, "Application of majority voting to pattern recognition: an analysis of its behavior and performance," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 27, no. 5, pp. 553–568, 1997.
- [29] M. Mace, N. Yousif, M. Naushahi et al., "An automated approach towards detecting complex behaviours in deep brain oscillations," *Journal of Neuroscience Methods*, vol. 224, pp. 66–78, 2014.
- [30] A. T. Walden, "Wavelet analysis of discrete time series," in *Proceedings of the 3rd European Congress of Mathematics*, vol. 202, pp. 627–641, 2001.
- [31] V. J. Samar, A. Bopardikar, R. Rao, and K. Swartz, "Wavelet analysis of neuroelectric waveforms: a conceptual tutorial," *Brain and Language*, vol. 66, no. 1, pp. 7–60, 1999.
- [32] S. Lawrence Marple Jr., "Computing the discrete-time analytic signal via fft," *IEEE Transactions on Signal Processing*, vol. 47, no. 9, pp. 2600–2603, 1999.
- [33] S. O. Haykin et al., *Neural Networks and Learning Machines*, vol. 3, Upper Saddle River: Pearson Education, UpperSaddleRiver, 2009.
- [34] S. Pan, K. Warwick, J. Burgess, M. N. Gasson, S. Y. Wang, and T. Z. Aziz, "Prediction of Parkinson's disease tremor onset with artificial neural networks," in *Proceedings of the International Conference on Biomedical Engineering*, pp. 341–345, 2007.
- [35] J. Park and I. W. Sandberg, "Universal approximation using radial basis function networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991.
- [36] M. Mace, K. Abdullah-Al-Mamun, A. A. Naeem, L. Gupta, S. Wang, and R. Vaidyanathan, "A heterogeneous framework for real-time decoding of bioacoustic signals: Applications to assistive interfaces and prosthesis control," *Expert Systems with Applications*, vol. 40, no. 13, pp. 5049–5060, 2013.
- [37] K. Cho, P. Meer, and J. Cabrera, "Performance assessment through bootstrap," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 11, pp. 1185–1198, 1997.
- [38] F. Wang and B. Li, "A new method for modulation classification based on bootstrap technique," in *Proceedings of the 2008 International Symposium on Computer Science and Computational Technology*, pp. 11–14, Shanghai, China, December 2008.
- [39] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [40] Q. Wei and R. L. Dunbrack Jr., "The role of balanced training and testing data sets for binary classifiers in bioinformatics," *PLoS ONE*, vol. 8, no. 7, Article ID e67863, 2013.
- [41] M. Feuerman and A. R. Miller, "The kappa statistic as a function of sensitivity and specificity," *International Journal of Mathematical Education in Science and Technology*, vol. 36, no. 5, pp. 517–527, 2005.
- [42] F. Nijboer, A. Furdea, I. Gunst et al., "An auditory brain-computer interface (BCI)," *Journal of Neuroscience Methods*, vol. 167, no. 1, pp. 43–50, 2008.
- [43] K. A. Mamun, M. Mace, M. E. Lutman et al., "Movement decoding using neural synchronization and inter-hemispheric connectivity from deep brain local field potentials," *Journal of Neural Engineering*, vol. 12, no. 5, p. 056011, 2015.

Research Article

Pathological Brain Detection Using Wiener Filtering, 2D-Discrete Wavelet Transform, Probabilistic PCA, and Random Subspace Ensemble Classifier

Debesh Jha,¹ Ji-In Kim,¹ Moo-Rak Choi,² and Goo-Rak Kwon¹

¹Department of Information and Communication Engineering, Chosun University, 309 Pilmun-Daero, Dong-Gu, Gwangju 61452, Republic of Korea

²School of Electrical Engineering, Korea University, 145 Anam-ro, Sungbuk-gu, Seoul 02841, Republic of Korea

Correspondence should be addressed to Goo-Rak Kwon; grkwon@chosun.ac.kr

Received 10 May 2017; Revised 7 August 2017; Accepted 23 August 2017; Published 3 October 2017

Academic Editor: George A. Papakostas

Copyright © 2017 Debesh Jha et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Accurate diagnosis of pathological brain images is important for patient care, particularly in the early phase of the disease. Although numerous studies have used machine-learning techniques for the computer-aided diagnosis (CAD) of pathological brain, previous methods encountered challenges in terms of the diagnostic efficiency owing to deficiencies in the choice of proper filtering techniques, neuroimaging biomarkers, and limited learning models. Magnetic resonance imaging (MRI) is capable of providing enhanced information regarding the soft tissues, and therefore MR images are included in the proposed approach. In this study, we propose a new model that includes Wiener filtering for noise reduction, 2D-discrete wavelet transform (2D-DWT) for feature extraction, probabilistic principal component analysis (PPCA) for dimensionality reduction, and a random subspace ensemble (RSE) classifier along with the K -nearest neighbors (KNN) algorithm as a base classifier to classify brain images as pathological or normal ones. The proposed methods provide a significant improvement in classification results when compared to other studies. Based on 5×5 cross-validation (CV), the proposed method outperforms 21 state-of-the-art algorithms in terms of classification accuracy, sensitivity, and specificity for all four datasets used in the study.

1. Introduction

Magnetic resonance imaging (MRI) of the brain provides comprehensive diagnostic information for diagnosis [1]. It is essential because it is noninvasive and safe and yields a higher resolution that cannot be obtained by other techniques. MRI is mainly utilized to diagnose different types of disorders such as strokes, tumors, bleeding, injury, blood-vessel diseases or infections, and multiple sclerosis (MS). The early diagnosis of pathological brain disease and its prodromal stage are critical and can decrease or halt the progression of the disease [2]. Therefore, the classification of normal/pathological brain status from MRIs is essential in clinical medicine as it focuses on soft tissue anatomy and generates a large and detailed dataset about the subject's brain. However, the use of a large database makes manual interpretation of the brain images

tedious, time consuming, and costly. The major drawback of the manual approach is its irreducibility. Therefore, there is a need for automated image analysis tools such as computer-aided diagnosis (CAD) systems [3].

Considerable research has been carried out to develop automatic tools for the classification of MR images to distinguish between normal and pathological brains. El-Dahshan et al. [4] utilized a three-level discrete wavelet transform, accompanied by principal component analysis (PCA), to decrease features. A good success rate was obtained by using feedforward backpropagation neural networks (BPNNs) and the K -nearest neighbor (KNN). Zhang and Wu [5] recommended the application of a kernel support vector machine (KSVM) and presented three new kernels: homogenous polynomial, inhomogeneous polynomial, and Gaussian radial basis for distinguishing between normal

and abnormal images. Patnaik et al. [6] employed DWT to obtain the approximation coefficients. Later, a support vector machine (SVM) was utilized to perform the classification. Zhang et al. [7] recommended a training feedforward neural network (FNN) with a unique scaled conjugate gradient (SCG) technique. Kundu et al. [8] proposed combining the Ripplet transform (RT) for feature extraction, PCA for dimensionality reduction, and the least-square SVM (LS-SVM) for classification, and the 5×5 stratified cross-validation (SCV) offered high classification accuracies. El-Dahshan et al. [9] utilized the feedback pulse-coupled neural network for the preprocessing of MR images, the DWT for feature extraction, PCA for features reduction, and the FBPNN for the classification of pathological and normal brains. Damodharan and Raghavan [10] used wavelet entropy as the feature space, and they then used the traditional naïve-Bayes classifier classification method. Wang et al. [11] utilized the stationary wavelet transform (SWT) to substitute for DWT. Likewise, they proposed a hybridization of particle swarm optimization (PSO) and the artificial bee colony (HPA) method to obtain the optimal weights and biases of FNN. Nazir et al. [12] applied denoising at the beginning, and they achieved an overall classification accuracy of 91.8%. Harikumar and Vinoth Kumar [13] used wavelet-energy and SVM. Padma and Sukanesh [14] used the combined wavelet statistical feature to segment and classify Alzheimer's disease (AD) as well as benign and malignant tumor slices. Zhang et al. [15] utilized Hu moment invariants (HMI) and generalized eigenvalue proximal SVM (GEPSSVM) for the detection of pathological brain in MRI scanning and obtained an accuracy of 98.89%, sensitivity of 99.29%, and specificity of 92.00%. Later on, Zhang et al. [16] used multilayer perceptron (MLP) for classification, where two pruning techniques like dynamic pruning (DP) and Bayesian detection boundaries (BDB) were used to find the optimal hidden neurons and an adaptive real coded BBO (ARCBBO) method was implemented to determine the optimal weights and obtained an accuracy of 98.12% and 98.24%, respectively. Nayak et al. [17] used 2D-DWT, PCA, and Adaboost algorithm with random forest as its base classifier and obtained an accuracy of 98.44% for classification of pathological brain MR image with Dataset-255. Later on, Nayak et al. [18] utilized two-dimensional stationary wavelet transform (SWT), symmetric uncertainty ranking (SUR) filter, and Adaboost with SVM classifier for the detection of pathological brain MR images and obtained an accuracy of 98.43% with Dataset-255. Wang et al. [19] employed Pseudo Zernike moment and linear regression classifier for classification of Alzheimer's disease and yielded an accuracy of 97.51%, sensitivity of 96.71%, and specificity of 97.73%. Alam et al. [20] utilized dual-tree complex wavelet transform (DTCWT), principal component analysis (PCA), and twin support vector machine (TSVM) for the detection of Alzheimer's disease classification and obtained an accuracy of 95.46 ± 1.26 .

Scholars have proposed different methods to extract features for the pathological brain disease [21]. After analyzing the above methods, we found that all of the methods achieved promising results which indicated that 2D-DWT is effective in feature extraction for pathological brain detection.

However, there are two problems. (1) Most of them utilize traditional PCA for feature extraction which is computational-intensive for large datasets with a higher dimensions. (2) The classification performance can be further improved, because the feature vector contains excessive features, which required more memory and increased computational complexity. Moreover, it required too much time to train the classifiers.

To address the above-mentioned problems, we proposed a new pathological brain detection system based on brain MR images which has the potential improvements over the other schemes. Weiner filter is used for the preprocessing of the images. The proposed method uses 2D DWT for the extraction of features because of its ability to analyze images at different scales. PPCA is used in place of PCA for the reduction of features which has the advantages of computing the efficient dimension reduction in terms of the distribution of latent variables, maximum-likelihood estimates, probability model, dealing with the missing data, and a combination of multiple PCA as probabilistic mixture. A relatively new classifier known as random subspace ensemble (RSE) classifier is employed which has the advantage of low computational burden over the traditional classifiers. Hence, the novelty of the proposed method lies in the application of PPCA features and RSE classifier.

The article is organized as follows: Section 2 presents details about the materials and methods. Section 3 describes the experimental results, evaluation procedure, and discussions. Finally, Section 4 presents the conclusion and future research.

2. Materials and Methods

2.1. Materials. At present, there are four benchmark datasets (DS) as DS-66, DS-90, DS-160, and DS-255, of different sizes of 66, 90, 160, and 255 images, respectively. All the datasets (DS) contain axial, T2-weighted, 256×256 -pixel MR images downloaded from medical school of Harvard University (Boston, MA, USA) (URL: <http://www.med.harvard.edu/aalib/home.html>) website. T2-weighted images are selected as input image because T2-weighted (spin-spin) relaxation gives better image contrast that is helpful to show different anatomical structure clearly. Also, they are better in detecting lesions than T1 weighted images.

We selected five slices from each subject. The selection criterion is that, for healthy subjects, these slices were selected at random. For pathological subjects, the slices should contain the lesions by confirmation of these radiologists with ten years of experiences. A sample of diseased slices is shown in Figure 2. In this investigation, all diseases are treated as pathological, and our task is a binary classification problem, that is, to distinguish pathological brain from healthy brains. Here, the whole brain is considered as the input image. We did not select local characteristics like point and edge, and we extract global image characteristics that are further learned by the new cascade model. Let us keep in mind that our procedure is different from the way neuroradiologists do. They usually take the local features and compare with standard template to check whether focuses exist, such as shrink, expansion, bleeding, and inflammation. While our technique

is like AlphaGO, the computer researcher gives the machine sufficient data, and then the machine can learn how to make classification naturally. Including patients' information (age, gender, handedness, memory test, education, etc.) can add additional information and thus may assist us to improve the classification performance. Nevertheless, this new model proposed in our research is only dependent on the imaging data. Besides, the imaging data from the website does not contain the subjects' information.

The cost of predicting pathological to normal types is severe, because the subjects may be told that she/he is normal and thus avoids the mild symptoms displayed. The treatments of patients may be postponed. Nevertheless, the cost of misclassification of healthy to pathological types is low, since correct treatment can be given by other diagnosis means. The cost-sensitivity (CS) problem was resolved by changing the class distribution at the beginning state, since original data was accessible. That means we purposely picked up more pathological brains than healthy ones into the dataset, with the goal of making the classifier biased to pathological brains, to solve the CS problem. The overfitting problem was supervised by cross-validation technique.

In our experiment, DS-66 and DS-160 are extensively employed for brain MR image classifications that consist of normal brain images as well as abnormal brain images from seven types of diseases, namely, glioma, meningioma, Alzheimer's disease, Alzheimer's disease plus visual agnosia, Pick's disease, sarcoma, and Huntington's disease. DS-90 contains MR brain images of a healthy brain, AIDS dementia, Alzheimer's disease plus visual agnosia, Alzheimer's disease, cerebral calcinosis, cerebral toxoplasmosis, Creutzfeldt-Jakob disease, glioma, herpes encephalitis, Huntington's disease, Lyme encephalopathy, meningioma, metastatic adenocarcinoma, metastatic bronchogenic carcinoma, motor neuron disease, MS, Pick's disease, and sarcoma.

The third dataset, DS-255, includes images of four new types of diseases embedded with the above seven types of diseased images and normal brain images. The four additional diseases are chronic subdural hematoma, cerebral toxoplasmosis, herpes encephalitis, and MS.

2.2. Proposed Methodology. The proposed method comprises four vital stages, namely, image preprocessing, feature extraction using 2D-DWT, feature reduction utilizing PPCA, and classification using the RSE classifier. In order to enhance the quality of the MR images, Wiener filter is employed, followed by the extraction of approximation coefficients from MR images utilizing a 2D-DWT with three-level decomposition. Then, we saved these obtained features as our primary features. Thereafter, then we employ PPCA for obtaining uncorrelated discriminant set of features. Finally, we classified the reduced features using the RSE classifier with KNN as a base classifier. The complete block diagram of the proposed system is shown in Figure 1. A brief description about all these four stages is shown below.

2.2.1. Preprocessing Using Wiener Filter. The gif images were downloaded individually from the website of the Harvard Medical School. Then, each of the gif images was converted

into JPG format manually. The images were in RGB format, and they were then converted into grayscale intensity images. Next, the intensity image is converted to double precision. Acquired brain MR images require preprocessing to improve the quality, enabling us to obtain better features. In our study, we used the popular Wiener filter method.

The Wiener filter is used to replace the finite impulse response (FIR) filter in order to decrease noise in signals [22]. When an image is blurred by a familiar low-pass filter (LPF), we can recover the image by inverse filtering. However, inverse filtering is extremely sensitive to additive noise. Wiener filtering accomplishes an optimal trade-off between inverse filtering and noise smoothing in that it eliminates the additive noise and inverts the blurring simultaneously. In addition, it reduces the overall mean-square error during the course of inverse filtering plus noise smoothing. The Wiener filtering method generates a linear approximation of the original image and is based on the stochastic framework. The orthogonality principle indicates that the Wiener filter in the Fourier domain can be articulated as follows:

$$W(f_1, f_2) = \frac{H^*(f_1, f_2) S_{xx}(f_1, f_2)}{\|H(f_1, f_2)\|^2 S_{xx} + S_{mm}(f_1, f_2)}. \quad (1)$$

Here, $S_{xx}(f_1, f_2)$ is the power spectrum of the original image, $S_{mm}(f_1, f_2)$ is the adaptive noise, and $H(f_1, f_2)$ is the blurring filter.

2.3. 2D-DWT

2.3.1. Advantage of Wavelet Transform. The FT is the most commonly used tool for the analysis of signals, and it breaks down a time-domain signal into constituent sinusoids of various frequencies, thus changing the signal from the time domain to the frequency domain. Nevertheless, the FT has a serious disadvantage as it removes the time information from the signal. For instance, an investigator cannot determine when a specific event took place based on a Fourier spectrum. Therefore, the classification accuracy decreases as the time information is lost.

Gabor modified the FT to examine only a small part of the signal at a time. This approach is known as windowing or the short-time FT (STFT) [23]. It accumulates a window of appropriate shape to the signal. STFT can be considered as a compromise between the time information and frequency information. Nevertheless, the precision of the information is limited by the window size.

The wavelet transform (WT) constitutes the next logical step. It uses a windowing method with variable size, and the progress of the signal analysis is shown in Figure 3. Another benefit of the WT is that it selects a "scale" in place of the traditional "frequency"; that is, it does not generate a time-frequency view of a specific signal but a time-scale view. The time-scale view is another way of visualizing data and is more commonly used and effective.

2.3.2. DWT. This is an effective implementation of the WT, and it utilizes the dyadic scales and positions [24]. The

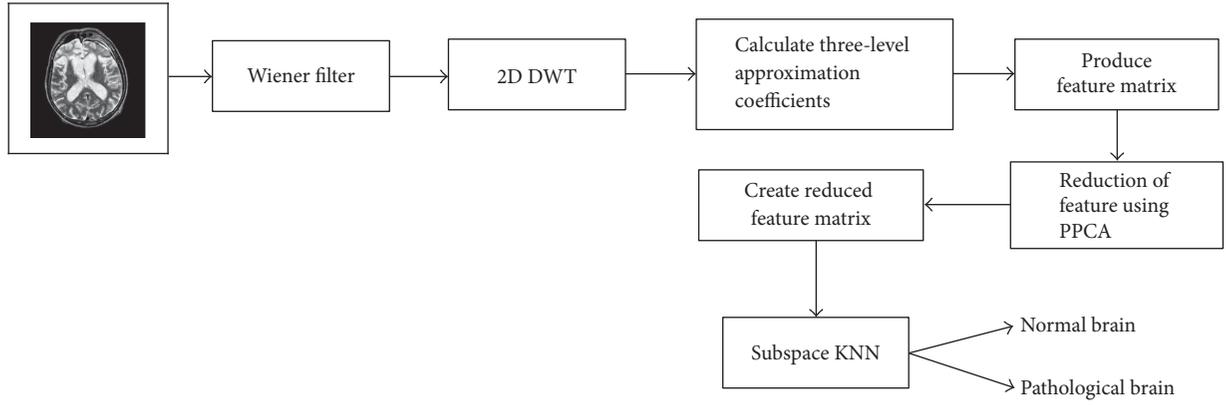


FIGURE 1: Block diagram of the proposed system.

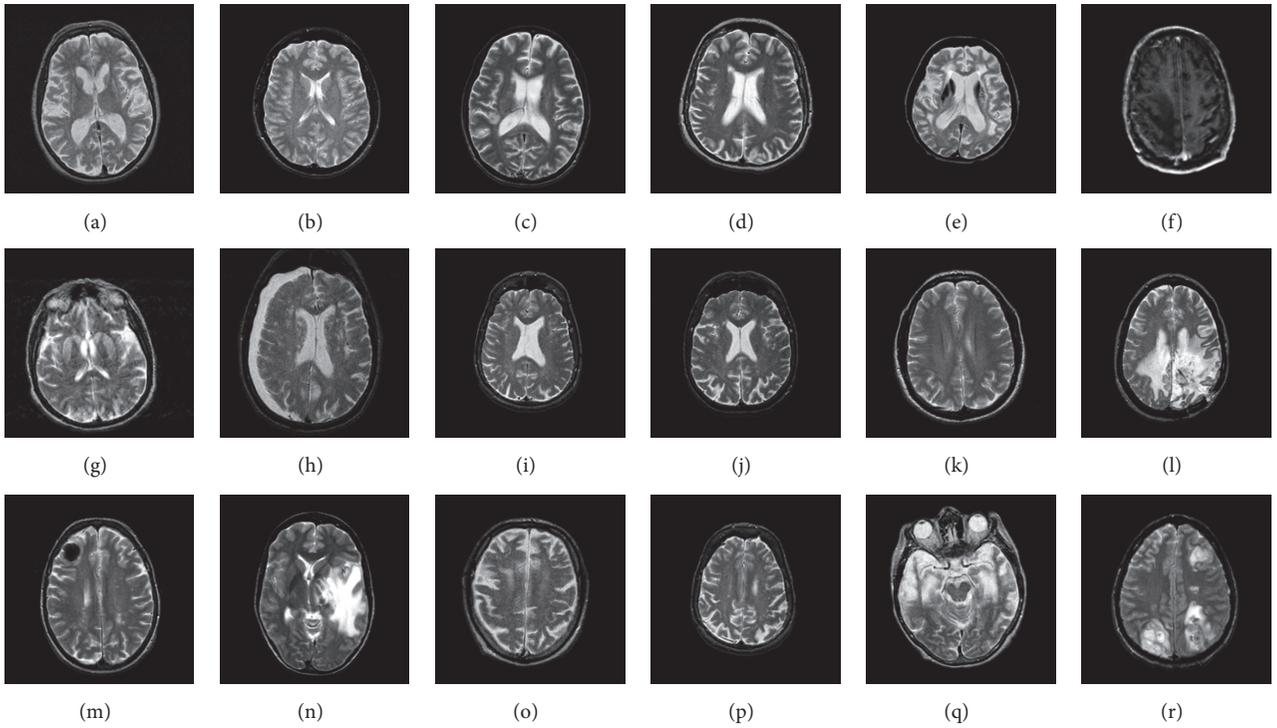


FIGURE 2: Brain MR images: (a) healthy brain; (b) AIDS dementia; (c) Alzheimer's disease plus visual agnosia; (d) Alzheimer's disease; (e) cerebral calcinosis; (f) cerebral toxoplasmosis; (g) Creutzfeldt-Jakob disease; (h) glioma, (i) herpes encephalitis; (j) Huntington's disease; (k) Lyme encephalopathy; (l) meningioma; (m) metastatic adenocarcinoma; (n) metastatic bronchogenic carcinoma; (o) motor neuron disease; (p) multiple sclerosis; (q) Pick's disease; and (r) sarcoma.

fundamentals of the DWT are as follows. Let $x(t)$ be a square-integral function. The continuous WT of the signal $x(t)$ relative to a real-valued wavelet $\psi(t)$ is defined as

$$W(a, \tau) = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{a}} \psi * \left(\frac{t - \tau}{a} \right) dt, \quad (2)$$

where $W(a, \tau)$ is the WT, τ indicates the function across $x(t)$, and the variable a is the dilation factor (both real and positive numbers). Here, the asterisk (*) indicates the complex conjugate.

Equation (1) can be discretized by restraining a and τ to a discrete lattice ($a = 2^j$ and $\tau = 2^j k$) to provide the DWT, which is given as follows:

$$\begin{aligned} cA_{j,k}(n) &= DS \left[\sum_n x(n) l_j^*(n - 2^j k) \right], \\ cD_{j,k}(n) &= DS \left[\sum_n x(n) h_j^*(n - 2^j k) \right]. \end{aligned} \quad (3)$$

Here, $cA_{j,k}$ and $cD_{j,k}$ refer to the coefficients of the approximation components and detailed components,

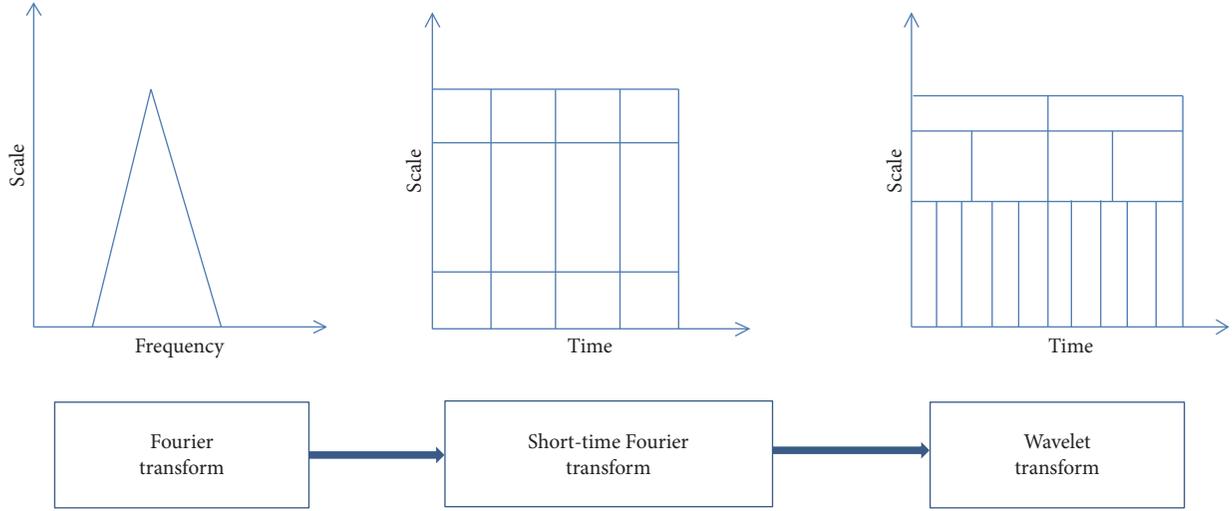


FIGURE 3: Progress of signal analysis.

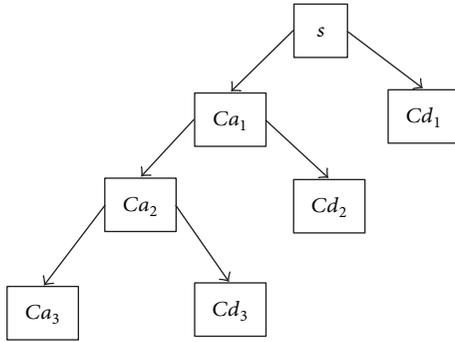


FIGURE 4: Three-level wavelet decomposition tree.

respectively. $l(n)$ and $h(n)$ represent the LPF and high-pass filter (HPF), respectively. j and k represent the wavelet scale and translation factors, respectively. The DS operator represents downsampling. The approximation component has low-frequency components of the image, whereas the detailed components contain high-frequency components. Figure 4 shows a three-level decomposition tree.

2.3.3. 2D-DWT. In a case involving 2D images, the DWT is employed in each dimension separately. A sample of a pathological brain MR image with its three-level wavelet decomposition is shown in Figure 5. Consequently, there are four subband images (LL, LH, HH, and HL) at each scale. The subband LL is utilized for the other 2D-DWT and can be considered as the approximation component of the image, whereas the LH, HL, and HH subbands can be considered as the detailed components of the image. As the level of the decomposition is increased, a more compact, but coarser approximation component is accessed. Thus, wavelets give a simple hierarchical foundation for clarifying the image information.

There are various types of wavelets, for example, Daubechies, symlets 1, coiflets 1, and biorthogonal wavelets and

reverse biorthogonal 1.1. We tested our result with each type of the wavelet family as shown in Table 2. In our research, the approximation coefficient of three-level wavelet decomposition along with a Haar wavelet yields promising results when compared to others in the wavelet family. Hence, Haar wavelet was selected in the experiment. It is also the simplest and most significant wavelet of the wavelet family. Moreover, it is very fast and can be used to extract basic structural information from an image. All the features are present for all the images, and a feature matrix is generated.

2.4. Probabilistic Principal Component Analysis. The PPCA algorithm proposed by Tipping et al. [36–38] is based on the estimation of the principal axes when any input vector has one or more missing values. The PPCA reduces the high-dimensional data to a lower-dimensional representation by relating a p -dimensional observation vector y to a k -dimensional latent (or unobserved) variable x that is regarded as normal with zero mean and covariance $I(k)$. Moreover, PPCA depends on an isotropic error model. The relationship can be established as

$$y^T = W * x^T + \mu + \varepsilon, \quad (4)$$

where y denotes the row vector of the observed variable, ε denotes the isotropic error term, and x is the row vector of latent variables. The error term, ε , is Gaussian with zero mean and covariance $\nu * I(k)$, where ν is the residual variance. To make the residual variance greater than 0, the value of k should be smaller than the rank. A standard principal component where ν equals 0 is the limiting condition of PPCA. The observed variables, y , are conditionally independent for the given values of the latent variables x . Therefore, the correlation between the observation variables is explained by the latent variables, and the error justifies the variability unique to y_i . The dimension of the matrix W is $p \times k$, and it relates both latent and observation variables. The vector μ allows the model to acquire a nonzero mean. PPCA considers

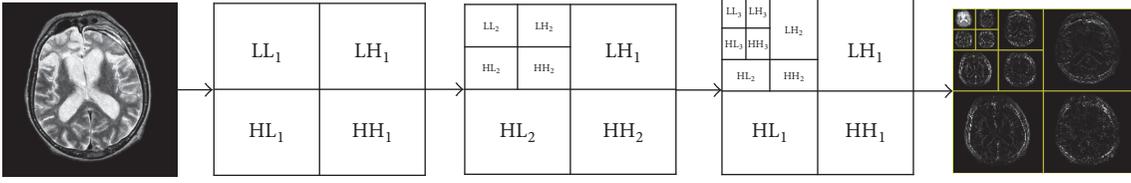


FIGURE 5: Pathological brain image and its wavelet coefficient at three-level decomposition.

the values to be missing and arbitrary over the dataset. From this model,

$$y \sim N(\mu, W * W^T + \nu * I(k)). \quad (5)$$

Given that the solution of W and ν cannot be determined analytically, we used the expectation-maximization (EM) algorithm for the iterative maximization of the corresponding log-likelihood function. The EM algorithm considers missing values as additional latent variables. At convergence, the columns of W span the solution subspace. PPCA then yields the orthonormal coefficients.

With respect to our research, the size of the image is 256×256 . After three-level decomposition, the vector feature becomes $32 \times 32 = 1024$. Here, all the features are not relevant for the classification. Because of the high computational cost, we utilized PPCA for the dimensionality reduction. The advantage of PPCA over PCA is its computational efficiency.

2.5. RSE Classifier. Ensemble classification includes combining multiple classifiers to obtain more accurate predictions than those obtained utilizing individual models. In addition, ensemble learning techniques are considered very useful for upgrading prediction accuracy. Nevertheless, base classifiers must be as precise and diverse as possible to increase the generalization capability of an ensemble model.

For the classification of normal and pathological brain MRI images, we used a random subspace classifier that uses KNN as a base classifier. The main idea behind the success of ensemble classification is the diversification in the classification that makes the ensemble classifier. With the ensemble classification approach, each classifier provides a different error for different instant. Therefore, we can develop a strong classifier that can decrease the error. The random subspace classifier is a machine-learning classifier that divides the entire feature space into subspaces. Each subspace randomly selects features from the original feature space. It must be guaranteed that the boundaries of the particular base classifier are significantly different. To realize this, an unstable or weaker classifier is utilized as base classifier because they create sufficiently varied decision boundaries, even for small disturbances in the training data parameters.

We used the majority voting method to obtain the final decision of the class membership. In the proposed algorithm, we used KNN as the base classifier owing to its simplicity. After selecting a random subspace, a new set of KNNs is estimated. The majority voting method was utilized to combine the output of each base classifier for the decision preparing test class.

TABLE 1: Confusion matrix for a binary classifier to discriminate between two classes (A_1 and A_2).

True class	Predicted class	
	A_1 (patients)	A_2 (controls)
A_1 (patients)	TP	FN
A_2 (controls)	FP	TN

Here, TP (true positive): correctly categorized as positive cases, TN (true negative): correctly categorized as negative cases, FP (false positive): incorrectly categorized as negative cases, FN (false negative): incorrectly categorized as positive cases.

TABLE 2: Comparison of different wavelet families.

Wavelet family	Accuracy
Haar	99.20%
Daubechies 2	98.60%
Coiflets 1	96.98%
Symlets 1	99.01%
Biorthogonal 1.1	98.64%

2.6. Pseudocode of Proposed System. Our proposed system can be outlined in four major stages. The steps involved are depicted in Pseudocode 1.

2.7. Performance Measures. Various techniques are used to evaluate the classifier's efficiency. The performance is determined based on the final confusion matrix. The confusion matrix holds correct and incorrect classification results. Table 1 illustrates a confusion matrix for binary classification, where TP, TN, FP, and FN depict true positive, true negative, false positive, and false negative, respectively.

Here, pathological brains are assumed to hold the value "true," and normal control (NC) ones are assumed to hold the value "false" following normal convention. Now, we calculate the performance of the proposed approach on the basis of sensitivity, specificity, accuracy, and precision as follows.

(i) Sensitivity (true positive rate): this is the tendency or ability to determine that the diagnostic test is positive when the person has the disease:

$$\text{Sensitivity} = \frac{TP}{TP + FN}. \quad (6)$$

(ii) Specificity (true negative rate): this is the tendency or ability to determine that the diagnostic test is negative when the person does not have the disease:

$$\text{Specificity} = \frac{TN}{TN + FP}. \quad (7)$$

```

Input: T2-weighted MR brain images.
Parameter:  $N$ , total number of images
Step 1 (weiner filter)
for  $i = 1 : N$ 
Read the images and apply wiener filter
end
Step 2 (2D-DWT)
For  $i = 1 : N$ 
Read in the image file
Apply the DWT using for the 3rd level using "Haar" wavelet to extract the wavelet coefficients.
A matrix  $X [M \times N]$  is employed to store all the coefficients.
End
Step 3. Reduce the features from the coefficients using PPCA
for  $j = 1 : N$ 
Apply PPCA transformation on the obtained wavelet coefficients.
Put the new dataset in a matrix  $Y$ .
End
Step 4 (RSE classification using  $5 \times 5$  cross-validation)
Divide the input data  $I$  and target data  $T$  into 5 different groups randomly
For  $k = 1 : 5$ 
Use the  $k$ th group for test, and other 4 groups to train the RSE algorithm.
Classify test image
End
Calculate average specificity, sensitivity, and accuracy.

```

PSEUDOCODE 1: Pseudocode of the proposed system.

Investigation 1					
Investigation 2					
Investigation 3					
Investigation 4					
Investigation 5					

Training
 Validation

FIGURE 6: Illustration of k -fold cross-validation.

(iii) Accuracy: this is a measure of how many diagnostic tests are correctly performed:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (8)$$

(iv) The precision and the recall are formulated by

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (9)$$

2.8. *Cross-Validation*. Cross-validation (CV) is a model-assessment method that is used to evaluate the performance of a machine-learning algorithm prediction on a new DS on which it has not been trained. It helps to solve the overfitting

problems. Each cross-validation round involves randomly portioning the original DS into a training set and a validation set. The illustration of the k -fold CV is shown in Figure 6. The training set is used to train a supervised learning algorithm, while a test set is used to evaluate its performance.

To make the RSE classifier more reliable and generalize to independent datasets, a 5×6 -fold stratified cross-validation (SCV) and 5×5 -fold SCV are employed. A 5×6 -fold SCV is employed for DS-66 and 5×5 -fold SCV is used for DS-90, DS-160, and DS-255. For DS-66, 55 MR images are used for training whereas 75, 128, and 204 images are used for DS-90, DS-160, and DS-255 respectively. The validation images for DS-66, DS-90, DS-160, and DS-255 are 11, 15, 32, and 51, respectively.

TABLE 3: Comparison result of the proposed method.

Proposed method	Feature	DS-66	DS-90	DS-160	DS- 255
Logistic regression	13	100.00	100.00	100.00	92.50
QDA	13	100.00	98.90	98.90	96.50
KNN	13	100.00	100.00	100.00	97.30
RSE classifier	13	100.00	100.00	100.00	99.20

TABLE 4: Classification comparison with DS-90.

Existing methods	Success cases	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)
DWT + PCA + BPNN [25]	388	88.00	56.00	97.14	86.22
DWT + PCA + RBF-NN [25]	411	92.47	72.00	98.25	91.33
DWT + PCA + PSO-KSVM [25]	440	98.12	92.00	99.52	97.78
WE + BPNN [26]	390	88.47	56.00	97.16	86.67
WE + KSVM [27]	413	93.18	68.00	98.02	91.78
DWT + PCA + GA-KSVM [28]	439	97.88	92.00	99.52	97.56
WE + PSO-KSVM [29]	437	97.65	88.00	99.28	97.11
WE + BBO-KSVM [29]	440	98.12	92.00	99.52	97.78
WE + QPSO-KSVM [30]	442	98.59	92.00	99.52	98.22
WFRFT + PCA + GEPSVM [31]	446	99.53	92.00	99.53	99.11
HMI + SEPSVM [15]	445	99.06	96.00		98.89
HMI + TSVM [15]	445	99.29	92.00		98.89
<i>Proposed</i>					
2D- DWT + PPCA + RSE (proposed)	450	100.00	100.00	100.00	100.00

3. Results and Discussion

In this study, we implemented a new machine-learning framework using MATLAB 2016a on an Intel computer with a Core-i5 processor and 16 GB RAM running under the Windows 7 operating system. This program can be tested or run on any computer platform where MATLAB is available.

3.1. Feature Extraction and Optimum Wavelet. In the proposed system, the three-level 2D-DWT of the Haar wavelet breaks down the input image into 10 subbands, as illustrated in Figure 5. The top left corner of the wavelet coefficient image (Figure 5) represents the approximation coefficients of the three-level decomposition of the image, whose size is only $32 \times 32 = 1024$. These obtained features are the initial features. The size of these features is still large, and the matrix size needs to be reduced. Now, these reduced features are sent as the input to the PPCA.

3.2. Feature Reduction. The use of PPCA as a dimension-reduction tool reduces the feature size to its desired size. Here, we can take the feature as desired. It is better that the desired number of features should at least preserve more than 90% of the variance. However, in this study, we did not take 95% of the variance because it may lead to a higher computational cost. Researchers have considered different numbers of features. In our case, we first used a small number of features, but the accuracy was poor. However, the result with 13 principal components was excellent. Hence,

the proposed method uses 13 principal components to earn higher classification accuracy.

3.3. Classification Results. The reduced features were sent to the classifier, and the results obtained with the different classifier are promising. From the experiment, it is seen that the proposed method works well for all four DSs using 13 principal components. The performances obtained with logistic regression, quadratic discriminant analysis, KNN, and RSE classifier with KNN as a base classifier are shown in Table 3. From the table, we see that the proposed method outperforms other methods. We utilized a 5-fold CV for DS-90, DS-160, and DS-255, whereas we utilized a 6-fold CV for DS-66. The RSE classifier obtained an accuracy of 100.00%, 100.00%, 100.00%, and 99.20%, with DS-66, DS-90, DS-160, and DS-255, respectively. The result obtained with the cubic SVM is the same as the RSE classifier for the dataset beside DS-66, where it could only achieve 98.50%.

3.4. Comparison with Existing Schemes. To further demonstrate the effectiveness of the proposed approach, we compared 21 existing algorithms. The algorithms and their corresponding results are listed in Tables 4 and 5. Table 4 shows the comparison result with DS-90. It is evident from Table 4 that our proposed method correctly matched all cases with 100% sensitivity, 100% specificity, 100% precision, and 100% accuracy. A comparison of the obtained results shows that our algorithm is superior to the others. This shows the effectiveness of the preprocessing technique combined with

TABLE 5: Classification comparison (DS-66, DS-160, and DS-255).

Approaches	Feature	Run	Accuracy (%)		
DWT + SVM + POLY [24]	4761		<i>DS-66</i>	<i>DS-160</i>	<i>DS-255</i>
DWT + SVM + RBF [24]	4761	5	98.00	97.15	96.37
DWT + PCA + k -NN [4]	7	5	98.00	97.33	96.18
DWT + PCA + FNN + ACPSO [32]	19	5	98.00	97.54	96.79
DWT + PCA + FNN + SCABC [33]	19	5	100.00	98.75	97.38
DWT + PCA + BPNN + SCG [7]	19	5	100.00	98.93	97.81
DWT + PCA + KSVM [5]	19	5	100.00	98.29	97.14
RT + PCA + LS-SVM [34]	9	5	100.00	99.38	98.82
SWT + PCA + IABAP-FNN [11]	7	10	100.00	98.88	98.43
WT + PCA + ABC-SPSO-FNN [11]	7	10	100.00	99.44	99.18
WE + NBC [35]	7	10	92.58	99.62	99.02
DWT + PCA + ADBRF [17]	13	5	100.00	99.30	98.44
DWT + SUR + ADBSVM [18]	7	5	100.00	99.22	98.43
FRFE + DP-MLP + ARCBBO [16]	12	10	100.00	99.19	98.24
FRFE + BDP-MLP + ARCBBO [16]	12	10	100.00	99.31	98.12
DWT + PCA + RSE	13	5	100.00	99.57	98.90
DWT + PPCA + RSE (proposed)	13	5	100.00	100.00	99.20

features extracted using the WT and PPCA. Table 4 shows the result of 5 runs of the proposed system. Table 5 demonstrates the comparison results over the three DSs in terms of the number of features, number of runs, and average accuracy. Here, some of the recent schemes were run 10 times, while others were run five times. From Tables 4 and 5, we see that most of the techniques achieved excellent classification when subjected to DS-66 as it is smaller in size. However, none of the algorithms achieved 100.00% with DS-90 and DS-160 because DS-255 is larger in size and includes more types of diseased brains; therefore, no current CAD system can earn a perfect classification.

Finally, this proposed “DWT + PPCA + RSE” achieved an accuracy of 100% for DS-66, DS-90, and DS-160 and an accuracy of 99.20% for DS-255, which is comparable with other recent studies and greater than the entire algorithm presented in Table 5. The improvement realized by the recommended scheme appears to be marginal compared with other schemes, but we obtained this result based on a careful statistical analysis (five repetitions of k -fold CV). Thus, this improvement is reliable and robust.

4. Conclusion

This paper proposed a new cascade model of “2D-DWT + PPCA + RSE” for the detection of pathological brains. The experiments validated its effectiveness as it achieved an accuracy of 99.20%. Our contributions lie in three points. First, we introduced the Wiener filter and showed its effectiveness. Besides this we introduced the PPCA and RSE classifier and proved it gives the better performance when compared with other state-of-the-art algorithms. In this work, we transformed the PBD problem to a binary classification task. We presented a novel method that replaced PCA and introduced RSE classifier. The experiment showed the superiority of our methods to existing approaches.

The proposed algorithm can also be employed in other fields, for example, face recognition, breast cancer detection, and fault detection. Moreover, this method has been validated on the publically available datasets which are limited in size. Also, in the selected dataset, the images are collected during the late and middle stage of diseases; however, the images with disease at early stages need to be considered.

In future research, we may consider images from other modalities like MRSI, PET, and CT to increase robustness to our scheme. The proposed method can be validated on a larger clinical dataset utilizing modern machine-learning techniques like deep learning, extreme learning, and so on, after collecting the enough brain images from the medical institutes. Internet of things can be another promising research field to embed this PBDS.

Nomenclature

MR(I):	Magnetic resonance (imaging)
DWT:	Discrete wavelet transform
PPCA:	Probabilistic principal component analysis
KNN:	k -nearest neighbor
CV:	Cross-validation
BPNN:	Backpropagation neural network
KSVM:	Kernel support vector machine
SCG:	Scale conjugate gradient
LS-SVM:	Least-square support vector machine
FBPNN:	Feedforward backpropagation neural network
SWT:	Stationary wavelet transform
PSO:	Particle swarm optimization
CAD:	Computer-aided diagnosis
STFT:	Short-time Fourier transform
QDA:	Quadratic discriminant analysis
SUR:	Symmetric uncertainty ranking

PZM:	Pseudo Zernike moment
SWT:	Stationary wavelet transform
DTCWT:	Dual-tree complex wavelet transform
RBFNN:	Radial basis function neural network
CT:	Computed tomography
TSVM:	Twin support vector machine
HMI:	Hu moment invariants
MLP:	Multilayer perceptron
ARCBBO:	Adaptive real coded biogeography-based optimization
DP:	Dynamic pruning.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Brain Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT & Future Planning (NRF-2014M3C7A1046050). And this work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2017RIA2B4006533).

References

- [1] D. Jha and G.-R. Kwon, "Alzheimers disease detection in MRI using curvelet transform with K-NN," *Journal of KIIT*, vol. 14, no. 8, 2016.
- [2] S. Alam, M. Kang, J.-Y. Pyun, and G.-R. Kwon, "Performance of classification based on PCA, linear SVM, and Multi-kernel SVM," in *Proceedings of the 8th International Conference on Ubiquitous and Future Networks, ICUFN 2016*, pp. 987–989, Vienna, Austria, July 2016.
- [3] F. Thorsen, B. Fite, L. M. Mahakian et al., "Multimodal imaging enables early detection and characterization of changes in tumor permeability of brain metastases," *Journal of Controlled Release*, vol. 172, no. 3, pp. 812–822, 2013.
- [4] E.-S. A. El-Dahshan, T. Hosny, and A.-B. M. Salem, "Hybrid intelligent techniques for MRI brain images classification," *Digital Signal Processing*, vol. 20, no. 2, pp. 433–441, 2010.
- [5] Y. Zhang and L. Wu, "An MR brain images classifier via principal component analysis and kernel support vector machine," *Progress in Electromagnetics Research*, vol. 130, pp. 369–388, 2012.
- [6] L. M. Patnaik, S. Chaplot, and N. R. Jagannathan, "Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network," *Biomedical Signal Processing and Control*, vol. 1, no. 1, pp. 86–92, 2006.
- [7] Y. Zhang, Z. Dong, L. Wu, and S. Wang, "A hybrid method for mri brain image classification," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10049–10053, 2001.
- [8] M. K. Kundu, M. Chowdhury, and S. Das, "Brain MR image classification using multi-scale geometric analysis of ripplelet," *Progress in Electromagnetics Research*, vol. 137, pp. 1–17, 2013.
- [9] E. A.-S. El-Dahshan, H. M. Mohsen, K. Revett, and A.-B. M. Salem, "Computer-aided diagnosis of human brain tumor through MRI: A survey and a new algorithm," *Expert Systems with Applications*, vol. 41, no. 11, pp. 5526–5545, 2014.
- [10] S. Damodharan and D. Raghavan, "Combining tissue segmentation and neural network for brain tumor detection," *The International Arab Journal of Information Technology*, vol. 12, no. 1, 2015.
- [11] S. Wang, Y. Zhang, Z. Dong et al., "Feed-forward neural network optimized by hybridization of PSO and ABC for abnormal brain detection," *International Journal of Imaging Systems and Technology*, vol. 25, no. 2, pp. 153–164, 2015.
- [12] M. Nazir, F. Wahid, and S. A. Khan, "A simple and intelligent approach for brain MRI classification," *Journal of Intelligent & Fuzzy Systems. Applications in Engineering and Technology*, vol. 28, no. 3, pp. 1127–1135, 2015.
- [13] R. Harikumar and B. Vinoth Kumar, "Performance analysis of neural networks for classification of medical images with wavelets as a feature extractor," *International Journal of Imaging Systems and Technology*, vol. 25, no. 1, pp. 33–40, 2015.
- [14] A. Padma and R. Sukanesh, "Segementation and classification of brain CT images using combined wavelet statistical texture features," *Arabian Journal for Science Engineering*, vol. 39, no. 2, 2014.
- [15] Y. Zhang, J. Yang, S. Wang, Z. Dong, and P. Phillips, "Pathological brain detection in MRI scanning via Hu moment invariants and machine learning," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 29, no. 2, pp. 299–312, 2017.
- [16] Y. Zhang, Y. Sun, P. Phillips, G. Liu, X. Zhou, and S. Wang, "A multilayer perceptron based smart pathological brain detection system by fractional fourier entropy," *Journal of Medical Systems*, vol. 40, no. 7, article 173, 2016.
- [17] D. R. Nayak, R. Dash, and B. Majhi, "Brain MR image classification using two-dimensional discrete wavelet transform and AdaBoost with random forests," *Neurocomputing*, vol. 177, pp. 188–197, 2016.
- [18] D. R. Nayak, R. Dash, and B. Majhi, "Stationary wavelet transform and AdaBoost with SVM based pathological brain detection in MRI scanning," *CNS and Neurological Disorders - Drug Targets*, vol. 16, no. 2, pp. 137–149, 2017.
- [19] S.-H. Wang, S. Du, Y. Zhang et al., "Alzheimers disease detection by pseudo zernike moment and linear regression classifier," *CNS & Neurologia Disorders*, vol. 16, no. 1, pp. 11–15, 2017.
- [20] S. Alam, M. Kang, and G. Kwon, "Alzheimer disease classification based on TSVM and Kernel SVM," in *Proceedings of the 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 565–567, July 2017.
- [21] D. Jha, J. Kim, and G. Kwon, "Diagnosis of Alzheimer's disease using dual-tree complex wavelet transform, PCA, and feed-forward neural network," *Journal of Healthcare Engineering*, vol. 2017, Article ID 9060124, 13 pages, 2017.
- [22] H. Naimi, A. B. H. Adamou-Mitiche, and L. Mitiche, "Medical image denoising using dual tree complex thresholding wavelet transform and Wiener filter," *Journal of King Saud University - Computer and Information Sciences*, vol. 27, no. 1, pp. 40–45, 2015.
- [23] L. Durak, "Shift-invariance of short-time FOUrier transform in fractional FOUrier domains," *Journal of the Franklin Institute. Engineering and Applied Mathematics*, vol. 346, no. 2, pp. 136–146, 2009.
- [24] S. Chaplot, L. M. Patnaik, and N. R. Jagannathan, "Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network," *Biomedical Signal Processing and Control*, vol. 1, no. 1, pp. 86–92, 2006.

- [25] Y. Zhang, S. Wang, G. Ji, and Z. Dong, "An MR brain images classifier system via particle swarm optimization and kernel support vector machine," *The Scientific World Journal*, vol. 2013, Article ID 130134, 9 pages, 2013.
- [26] R. Choudhary, S. Mahesh, J. Paliwal, and D. S. Jayas, "Identification of wheat classes using wavelet features from near infrared hyperspectral images of bulk samples," *Biosystems Engineering*, vol. 102, no. 2, pp. 115–127, 2009.
- [27] M. R. K. Mookiah, U. Rajendra Acharya, C. M. Lim, A. Petznick, and J. S. Suri, "Data mining technique for automated diagnosis of glaucoma using higher order spectra and wavelet energy features," *Knowledge-Based Systems*, vol. 33, pp. 73–82, 2012.
- [28] S. Wang, G. Ji, P. Phillips, and Z. Dong, "Application of genetic algorithm and kernel support vector machine to pathological brain detection in MRI Scanning," in *Proceedings of the 2nd National Conference Information Technology Comp. Science*, pp. 450–456, Shanghai, China, 2015.
- [29] G. Yang, Y. Zhang, J. Yang et al., "Automated classification of brain images using wavelet-energy and biogeography-based optimization," *Multimedia Tools and Applications*, vol. 75, no. 23, pp. 1–17, 2015.
- [30] Y. Zhang, G. Ji, J. Yang et al., "Preliminary research on abnormal brain detection by wavelet-energy and quantum-behaved PSO," *Technology and Health Care*, vol. 24, pp. S641–S649, 2016.
- [31] Y.-D. Zhang, S. Chen, S.-H. Wang, J.-F. Yang, and P. Phillips, "Magnetic resonance brain image classification based on weighted-type fractional Fourier transform and nonparallel support vector machine," *International Journal of Imaging Systems and Technology*, vol. 25, no. 4, pp. 317–327, 2015.
- [32] Y. Zhang, S. Wang, and L. Wu, "A novel method for magnetic resonance brain image classification based on adaptive chaotic PSO," *Progress in Electromagnetics Research*, vol. 109, pp. 325–343, 2010.
- [33] Y. Zhang, L. Wu, and S. Wang, "Magnetic resonance brain image classification by an improved artificial bee colony algorithm," *Progress In Electromagnetics Research*, vol. 130, pp. 369–388, 2012.
- [34] S. Das, M. Chowdhury, and M. K. Kundu, "Brain MR image classification using multi-scale geometric analysis of ripplelet," *Progress in Electromagnetics Research*, vol. 137, pp. 1–17, 2013.
- [35] X. Zhou, S. Wang, W. Xu et al., "Detection of pathological brain in MRI scanning based wavelet-entropy and naïve Bayes classifier," in *Proceedings of the International Conference on Bioinformatics and Biomedical Engineering*, pp. 201–209, 2015.
- [36] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, vol. 61, no. 3, pp. 611–622, 1999.
- [37] S. Roweis, "EM algorithms for PCA and SPCA," in *Advance in Neural Information Processing System*, vol. 10, pp. 626–632, MIT Press, Cambridge, MA, USA, 1998.
- [38] A. Ilin and T. Raiko, "Practical approaches to principal component analysis in the presence of missing values," *Journal of Machine Learning Research*, vol. 11, pp. 1957–2000, 2010.

Research Article

A Novel Active Semisupervised Convolutional Neural Network Algorithm for SAR Image Recognition

Fei Gao,¹ Zhenyu Yue,¹ Jun Wang,¹ Jinping Sun,¹ Erfu Yang,² and Huiyu Zhou³

¹Electronic Information Engineering, Beihang University, Beijing 100191, China

²Space Mechatronic Systems Technology Laboratory, Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow G1 1XJ, UK

³School of Electronics, Electrical Engineering and Computer Science, Queen's University, Belfast BT7 1NN, UK

Correspondence should be addressed to Jun Wang; wangj203@buaa.edu.cn

Received 5 May 2017; Revised 9 August 2017; Accepted 23 August 2017; Published 1 October 2017

Academic Editor: George A. Papakostas

Copyright © 2017 Fei Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Convolutional neural network (CNN) can be applied in synthetic aperture radar (SAR) object recognition for achieving good performance. However, it requires a large number of the labelled samples in its training phase, and therefore its performance could decrease dramatically when the labelled samples are insufficient. To solve this problem, in this paper, we present a novel active semisupervised CNN algorithm. First, the active learning is used to query the most informative and reliable samples in the unlabelled samples to extend the initial training dataset. Next, a semisupervised method is developed by adding a new regularization term into the loss function of CNN. As a result, the class probability information contained in the unlabelled samples can be maximally utilized. The experimental results on the MSTAR database demonstrate the effectiveness of the proposed algorithm despite the lack of the initial labelled samples.

1. Introduction

Synthetic aperture radar (SAR) has wide applications in both military and civilian fields due to its merits, such as strong penetrating ability and adaption to severe weathers. SAR automatic target recognition technology (SAR-ATR) aims at automatically recognizing the targets from SAR images [1]. With an increasing amount of the data acquired by a SAR imaging system, the SAR-ATR has become one of research hotspots.

Traditional machine learning methods for the SAR-ATR include Support Vector Machine (SVM) [2], local texture feature [3, 4], dictionary learning [5, 6], and sparse representation [7]. These methods have produced some promising results, but they heavily rely on the hand-crafted feature extraction [8]. Because of the imaging nature, clutters and speckling noise exist in the SAR images, which increases the difficulty of feature extraction despite the fact that experts are involved.

In recent years, with the development of deep learning techniques, CNN has received a great attention in object

recognition [9–11]. It can automatically extract the target features without experts' intervention. Compared with the traditional machine learning methods, the CNN is more effective and robust and has been successfully applied to SAR image recognition. In [12], a CNN method was proposed for improving the SAR image classification accuracy. The experimental result showed that the CNN method outperforms the Gabor feature extraction-based SVM method, which demonstrated a great potential of the CNN for SAR image recognition. A convolutional network was designed in [13] to automatically extract the features for SAR target recognition. Using the learned convolutional features, the accuracy of 84.7% was achieved on the 10 types of targets in the MSTAR dataset. Zhou et al. [14] studied the application of the Deep Convolutional Neural Networks (DCNN) in the polarimetric SAR image classification, in which the hierarchical spatial features of images could be automatically learned by the DCNN and the classification accuracy was improved significantly.

As can be found, the CNN has made a great breakthrough in the SAR image recognition. However, the sample

labelling for SAR image is still time-consuming, and the accuracy of the image recognition decreases quickly when the labelled samples are insufficient. Active learning (AL) can be effective by adding the most informative and reliable unlabelled samples into the labelled training set. As a result, it would be a promising way to solve the above-mentioned problem. Wang et al. proposed an AL method for the SAR image classification based on a SVM classifier [15]. The most uncertain samples were chosen according to the confidence value, and the experimental results showed that the AL-based method can effectively improve the classification accuracy when the labelled samples are insufficient. Babaei et al. presented an active learning method by employing a low-rank classifier as the training model. This method selects the samples whose labels are predicted wrong but the classifier is highly certain about them, namely, the first certain wrong labelled (FCWL) selection criteria [16]. Samat et al. reported an active extreme learning machine (AELM) method for the polarimetric SAR image classification. In this method, the class supports based on the posterior probability are utilized as the selecting criterion. According to the experimental results, the proposed method was faster than the existing techniques in the both learning and classification phases [17].

Active learning method effectively adds the most informative and reliable unlabelled samples into the training set. The remaining samples may be less informative and the use of active learning may cause too much computational complexity. However, the information contained in them can be used to improve the generalization ability of the classification algorithm. Semisupervised learning (SSL) is an effective way to utilize the information contained in the unlabelled samples. The commonly used SSL methods include semisupervised SVM [18], label propagation [19], and semisupervised clustering. Recently, SSL has been successfully applied in the SAR image recognition. Duan et al. introduced a semisupervised classification method incorporating the likelihood space approach in the training and testing processes so that the unlabelled samples can be effectively used to improve the classification performance [20]. To overcome the complexity of data and the difficulty of creating a fine-level ground truth, a semisupervised method for ice-water classification based on self-training was presented in [21]. By integrating the spatial context model, the region merging, and the self-training technique, the proposed algorithm is capable of accurately distinguishing ice and open water in SAR images using a very small number of labelled samples. In [22], the unlabelled samples were analysed by an unsupervised clustering algorithm under the usage of all the available information. Besides, each sample was classified by a supervised method using the available information at the current phase of clustering. The experimental results on the SAR image showed that the proposed semisupervised method leads to promising classification results.

Recently, inspired by the superiority of CNN, AL, and SSL, the combination of the three methods has become a research tendency. For example, a deep active learning method and a semisupervised CNN were constructed [23–25]. The experimental results demonstrated the effectiveness of these methods for hyperspectral or optical image

recognition. However, we have seen a number of problems from SAR images, for example, difficult feature extraction, time-consuming sample labelling, and insufficient labelled samples. The developed techniques are rarely applied to SAR image recognition. In this paper, a novel active semisupervised CNN algorithm for SAR image recognition is proposed. First, the most informative and reliable samples selected by the active learning method are labelled using an information entropy criterion. We believe that the information entropy can be used to effectively measure the reliability of the unlabelled samples, and it can be calculated based on the output of the CNN framework. Then, the class probability information of the remaining unlabelled samples is obtained from the output of the softmax layer of the CNN. Afterwards, the class probability information is designed as the regularization term and added to the loss function of the CNN for the retraining purpose. Since the class probability information can effectively control the impact of the unlabelled samples in the training process, the unlabelled samples are well utilized at this stage.

The rest of this paper is arranged as follows. In Section 2, the convolutional neural network is briefly introduced. Section 3 describes the proposed method in detail. Then experiments are performed in Section 4. Finally, we summarize this paper in Section 5.

2. Convolutional Neural Network

As a multilayer neural network structure, CNN is mainly composed of an input layer, a convolution layer, a pooling layer, and an output layer, where both the convolution and pooling layers are hidden. The input layer is used to receive the pixel values from the original image. The convolution layer extracts the image features by utilizing the convolution kernel. The pooling layer uses local image correlation to reduce the amount of data to be processed. The output layer maps the extracted features to the corresponding labels. The training of the CNN is composed of two ways: forward and backward propagation.

2.1. Forward Propagation. The mapping process of an image in the CNN is a forward propagation process, where the output of a previous layer is taken as the input of the current layer. In order to provide a full version of the linear model, a nonlinear activation function is added to the neurons of each layer in the mapping process. Since the first layer only receives pixel values from the image, there are no activation functions. From the second layer of the CNN, the nonlinear activation functions are employed. The output of each layer can be expressed as follows:

$$\begin{aligned} z^l &= W^l x^{l-1} + b^l, \\ a^l &= \sigma(z^l), \end{aligned} \tag{1}$$

where l denotes the l th layer. If $l = 2$, $x^{2-1} = x^1$ is the pixel value matrix of the image. If $l > 2$, x^{l-1} represents the feature map matrix a^{l-1} , which is extracted from the $(l - 1)$ th layer, that is, $x^{l-1} = a^{l-1} = \sigma(z^{l-1})$. W^l , b^l , and z^l represent the

weight matrix, the bias matrix, and the weighted input of the l th layer, respectively; σ is the nonlinear activation function, and a rectified linear unit (Relu) is selected in this paper. Suppose $l = L$; the L th layer is the output layer, and a^l denotes the final output vector.

2.2. Backpropagation. The standard backward propagation (BP) algorithm is used to update the parameters W^l and b^l of the CNN [10]. The BP algorithm is a supervised learning method which firstly constructs a cost function based on the actual and the expected outputs, and then a gradient descent method (GD) is used to update W^l and b^l along the gradient descent direction of the cost function. In detail, we suppose E_0 represents the cost function of the CNN structure. The error vector of the output layer can be expressed as follows:

$$\delta^L = \frac{\partial E_0}{\partial z^L}. \quad (2)$$

In the process of backward propagation, the error vector δ^{L-1} of the $(L-1)$ th layer can be derived from the error vector δ^L of the output layer. Thus, the error vector δ^l for each layer can be computed by the Chain Rule as follows:

$$\delta^l = W^{l+1} \delta^{l+1} \circ \sigma'(z^l), \quad (3)$$

where the symbolic \circ is the Hadamard product (or Schur product) which denotes the element-wise product of the two vectors. The gradients of W^l and b^l are denoted by $\partial E_0 / \partial W^l$ and $\partial E_0 / \partial b^l$, respectively. The partial derivative of E_0 to W^l and b^l can be calculated using (1) and (3):

$$\begin{aligned} \frac{\partial E_0}{\partial W^l} &= \frac{\partial E_0}{\partial a^l} \circ \frac{\partial a^l}{\partial W^l} = \delta^l \circ x^{l-1}, \\ \frac{\partial E_0}{\partial b^l} &= \frac{\partial E_0}{\partial a^l} \circ \frac{\partial a^l}{\partial b^l} = \delta^l. \end{aligned} \quad (4)$$

The change values of W^l and b^l can be calculated by

$$\begin{aligned} \Delta W^l &= -\eta \frac{\partial E_0}{\partial W^l}, \\ \Delta b^l &= -\eta \frac{\partial E_0}{\partial b^l}, \end{aligned} \quad (5)$$

where η represents the learning rate.

2.3. The Output Layer. If the number of neurons in the output layer is N , the CNN eventually divides the input images into N categories. In the forward propagation process, the input of the output layer is $z_k^L = W_k^L x^{L-1} + b_k^L$, $k \in [1, 2, \dots, N]$, since the output of the softmax activation function provides the probability of each class to which a sample belongs. Thus, unlike the middle layer of the CNN, we use the softmax activation function instead of the Relu function in the output layer, which is the key in our proposed method. The output is normalized by the softmax function, which can be expressed as

$$a_k = \frac{e^{\theta_k z_k^L}}{\sum_{j=1}^N e^{\theta_j z_j^L}}, \quad (6)$$

where a_k is the output of the k th neuron in the output layer and θ_k is the parameter of the softmax function. It is obvious that $\sum_{k=1}^N a_k = 1$, and if one item increases, all the other items will decrease accordingly.

3. The Proposed Method

First, we define the symbols to be used in this section. The training dataset X is composed of two parts: $X = [L, U] \in R^{d \times N}$, where $L = [x_1, x_2, \dots, x_l] \in R^{d \times l}$ represents the set of the labelled samples and $U = [x_{l+1}, x_{l+2}, \dots, x_{l+u}] \in R^{d \times u}$ represents the set of the unlabelled samples. $l + u = N$ is the total number of the training samples. The training process of the proposed methods is composed of two stages. As shown in Figure 1, first, the most informative and reliable samples selected by the active learning method are labelled based on the information entropy. Then the class probability information extracted from the remaining samples is designed as a regularization term, which will be added to the loss function of the CNN for retraining. When the training process has finished, the unlabelled samples go to the CNN and obtain the labels which can be calculated from the softmax layer of the CNN.

3.1. Active Learning. To improve the robustness of the classification model when there are insufficient labelled samples, it is necessary to extend the initial training set. Especially for the CNN model with a large number of parameters, the model will be overfitting and weak in generalization if the labelled samples are inadequate. Thus, an active learning method is utilized to increase the number of training samples.

The CNN model employed in this paper is shown in Figure 2, where Conv, Max pool, and Flatten represent the convolution layer, the subsampling layer, and the fully connected layer, respectively. First of all, CNN is trained using the initially labelled samples. After that, the unlabelled samples go into the CNN, and the information entropy of each unlabelled sample is calculated using the output of the softmax layer. The information entropy measures the uncertainty of the samples. The greater the value of the information entropy is, the greater the uncertainty of a sample is. In other words, if the sample is closer to the classification plane, the probability of the sample belonging to a specific category is more uniform.

The probability of each class to which a sample belongs is represented by $[p_1^i, p_2^i, \dots, p_N^i]$, which can be obtained by (6). Then, the information entropy of the unlabelled sample x_i can be calculated by

$$H^i = -\sum_{k=1}^N p_k^i \log(p_k^i). \quad (7)$$

To extract the most informative samples, the top M unlabelled samples with the maximum information entropy are selected and manually labelled. Then they are added to the initial training set. Since the certainty of the samples with small information entropy is high, the labels obtained by the CNN model are considered reliable. Therefore, P samples

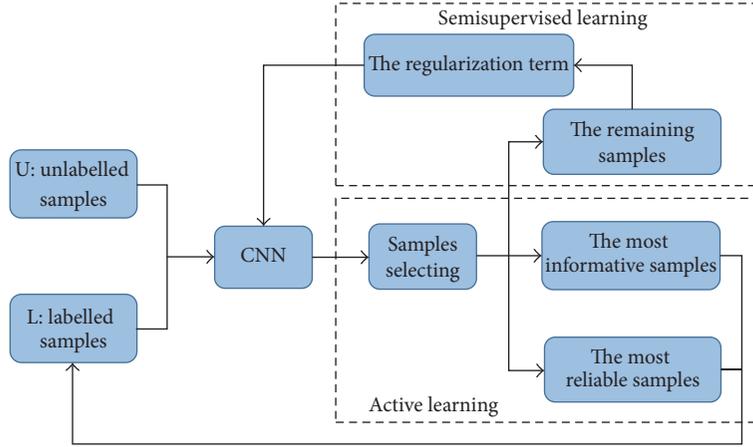


FIGURE 1: The flowchart of the training process of the proposed algorithm.

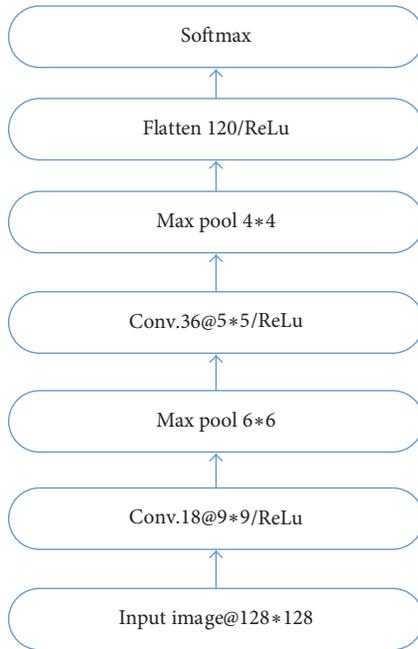


FIGURE 2: The CNN model employed in this paper.

with small entropy are selected and labelled with the CNN. Then these samples are added to the initial training set as well. At this point, the training set is expanded.

Since the initial set of the labelled samples is small, the ability of a CNN model to measure the uncertainty of the samples is weak, which may cause a large deviation. Thus, instead of selecting M samples, it is more reasonable to select in an iterative manner. At the beginning of the training process, the epochs are small because the labelled samples are insufficient. With the increase of the training set, the epochs of the CNN can be increased gradually. In this paper, we select the epochs according to the following rule:

$$N = 50 + 4i, \quad (8)$$

where i denotes the iterations of the active learning algorithm and N denotes the epochs of the CNN during each iteration.

When a certain number of the unlabelled samples have been selected manually, the remaining samples are supposed to be reliable. If we continue selecting the unlabelled samples based on the active learning method, the improvement of generalization ability and the classification accuracy of the CNN is not ensured. Thus, after a certain number of the unlabelled samples are manually labelled, the iteration should be terminated. The main drawback of the CNN is that the training process costs too much time. In order to achieve a trade-off between the number of the unlabelled samples and the complexity of the algorithm, the iterations should be chosen appropriately.

3.2. Semisupervised Learning. Although the uncertainty of the remaining unlabelled samples is not significant, the class probability information contained in the samples can be used to improve the generalization ability of the CNN. We design a regularization term based on the class probability information of the unlabelled samples, which is added to the loss function of the CNN. The commonly used cost functions for the CNN include quadratic cost function, cross-entropy cost function, and log likelihood cost function. When the softmax layer is used as the output layer, the log likelihood cost function is simple and effective. The expression of the log likelihood cost function is as follows:

$$E_0 = - \sum_{x_i \in L} \sum_k y_k^i \log(a_k^i) = - \sum_{x_i \in L} y^i \cdot \log(a^i), \quad (9)$$

where a_k^i represents the output of the k th neuron in the output layer corresponding to the input sample x_i and y_k^i represents the expected value of the k th neuron. y^i and a^i denote the actual vector and the expected vector output from the softmax layer, respectively.

The maximum class probability output from the softmax layer is used to design the regularization term. The maximum

output of the softmax layer relevant to the unlabelled input sample x_i is expressed as follows:

$$q_{\max}^i = \max([a_1^i, a_2^i, \dots, a_N^i]). \quad (10)$$

The regularization term of the unlabelled samples is expressed by

$$E_1 = - \sum_{x_i \in \mathcal{U}} \sum_k q_{\max}^i y_k^i \log(a_k^i), \quad (11)$$

where y_k^i denotes the label of the unlabelled samples assigned by the CNN. q_{\max}^i plays the role of a constraint. A higher value of q_{\max}^i indicates that the sample plays a more important role in the training process of the CNN model. After introducing the regularization term of the unlabelled samples, we have the cost function as follows:

$$E = E_0 + b * E_1, \quad (12)$$

where b is the weighting parameter.

For E_0 , the error vector of the output layer is

$$\delta_0^L = \frac{\partial E_0}{\partial z^L} = \sum_{x_i \in \mathcal{L}} (a^i - y^i). \quad (13)$$

For E_1 , the error vector of the output layer is

$$\delta_1^L = \frac{\partial E_1}{\partial z^L} = b \sum_{x_i \in \mathcal{U}} q_{\max}^i (a^i - y^i). \quad (14)$$

According to (1), (13), and (14), the sensitivity of the modified cost function can be obtained as

$$\begin{aligned} \delta' &= \delta_0^L + b * \delta_1^L \\ &= \sum_{x_i \in \mathcal{L}} (a^i - y^i) + b \sum_{x_i \in \mathcal{U}} q_{\max}^i (a^i - y^i). \end{aligned} \quad (15)$$

As the error vector of the output layer is obtained, the sensitivity of each layer can be calculated using (3) iteratively, and the parameters of each layer can be updated according to (4) and (5).

4. Experiment

We perform experiments on the Moving and Stationary Target Acquisition and Recognition (MSTAR) database, which is cofunded by National Defense Research Planning Bureau (DARPA) and the US Air Force Research Laboratory (AFRL). Ten types of vehicle targets in the MSTAR database are chosen in our experiment, that is, 2S1, ZSU234, BMP2, BRDM2, BTR60, BTR70, D7, ZIL131, T62, and T72. The SAR and corresponding optical images of each type are shown in Figure 3. Table 1 lists the detailed information of target chips involved in this experiment.

4.1. The Effectiveness of the Active Learning Method. We select 5% of the samples as the initial labelled training set which is

used to train the CNN. Then the active learning method based on information entropy is utilized to expand the labelled training set in an iterative manner. During each iteration, the active learning method manually labels 8 samples with the largest information entropy, and 10 samples with the minimum information entropy are labelled by the CNN. To demonstrate the effectiveness of the active learning method, we compare it with the random selection method in which 8 unlabelled samples are randomly selected and manually labelled during each iteration. The classification accuracy of the above two methods varies with the iterations, as shown in Figure 4. Obviously, the classification accuracy of the active learning method is higher than that of the random selection method. Thus, it validates that the active learning method can select the unlabelled samples more effectively. The selected samples are then added to the labelled training set, which is helpful for improving the classification accuracy.

4.2. The Effectiveness of Semisupervised Learning Method.

After the expansion of the initial labelled training set, the semisupervised learning method is utilized by obtaining the regularization term based on the remaining samples and we then add it to the cost function of the CNN. To demonstrate the effectiveness of the regularization method, we compare it with the method without any regularization. The comparison result is shown in Figure 5. As can be seen, the classification accuracy of the regularization method is higher than those without any regularization. The convergence speed of the regularization method is also faster because the method without any regularization needs 100 epochs for convergence, but the regularization method only needs 80 epochs. Thus, it validates that the regularization method is capable of utilizing the information contained in the unlabelled samples, which is helpful to improve the classification accuracy and the convergence speed of the proposed method.

4.3. Comparison with Other Methods.

In this section, we compare the performance of our method with that of the CNN [13], label propagation (LP) [19], and progressive semisupervised SVM with diversity (PS3VM-D) [18]. The CNN is a fully supervised algorithm which only utilizes the labelled samples to train the classification model. The LP and PS3VM-D are both semisupervised methods. The LP establishes a similar matrix and propagates the label of the labelled samples to the unlabelled samples according to the degree of similarity. The PS3VM-D selects the reliable unlabelled samples to expand the initial labelled training set. The comparison results are shown in Figure 6.

We find that our method outperforms the CNN method significantly when the labelled samples are insufficient, and this is due to our active learning and regularization strategy. The generalization ability of the CNN method is weak due to the small number of the labelled samples. With the increase of the labelled samples, the classification accuracy of the CNN method gradually increases, and ultimately it is the same as our method. Furthermore, the classification accuracy of our method is better than that of the other two semisupervised methods. The LP and PS3VM-D assign pseudolabels to the unlabelled samples; however, if the pseudolabels are wrong,

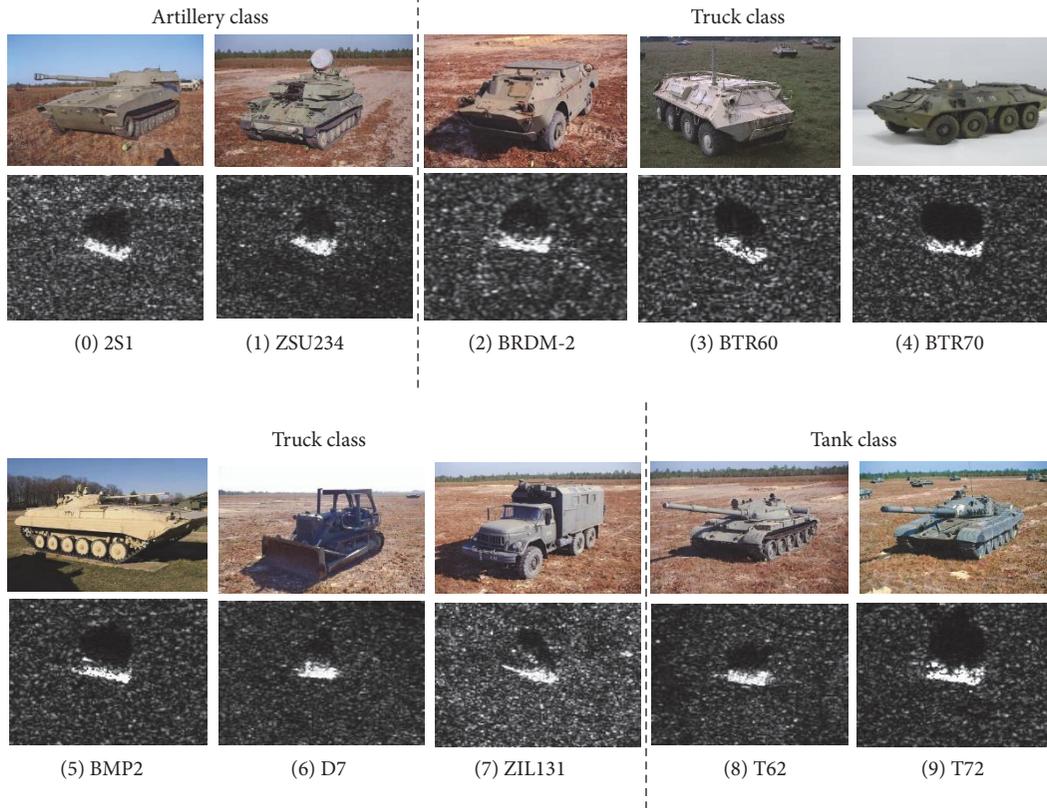


FIGURE 3: SAR images and corresponding optical images of ten types of targets in the MSTAR database.

TABLE 1: The training and testing set of our experiment.

Type	Tops	Model	Training set		Testing set	
			Depression	Number	Depression	Number
2S1	Artillery	B_01	17°	299	15°	274
ZSU234		D_08	17°	299	15°	274
BRDM2	Truck	E_71	17°	298	15°	274
BTR60		K10YT_7532	17°	256	15°	195
BMP2		SN_9563	17°	233	15°	195
BTR70		C_71	17°	233	15°	196
D7		92V_13015	17°	299	15°	274
ZIL131		E_12	17°	299	15°	274
T62		Tank	A_51	17°	299	15°
T72	#A64		17°	299	15°	274
			Sum: 2814		Sum: 2503	

the labels will have a negative influence on the subsequent classifier training process. In contrast, our method utilizes the unlabelled samples by designing a regularization term based on the class probability information; then the regularization term is added to the loss function of the CNN for the retraining purpose. In summary, our method is effective and has a strong generalization ability especially when the labelled samples are insufficient.

5. Conclusion

The CNN has achieved a great success in the field of image recognition. To improve the classification accuracy of the SAR image when the labelled samples are insufficient, a new active semisupervised CNN method has been proposed in this paper. First, we query the most informative and reliable samples by utilizing the active learning method, and then

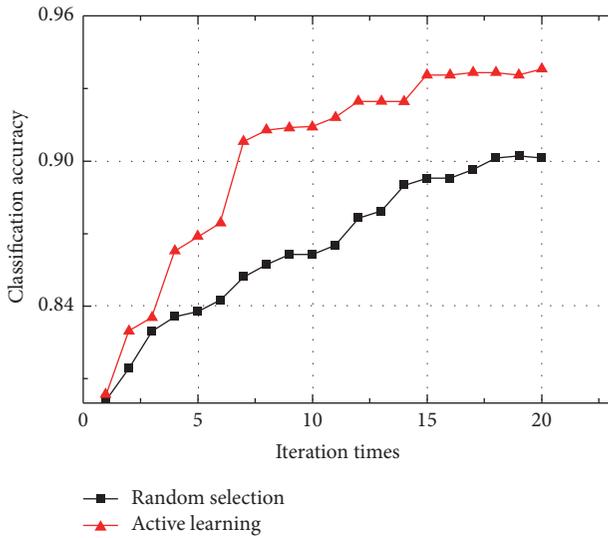


FIGURE 4: Classification accuracy of active learning method and random selection method.

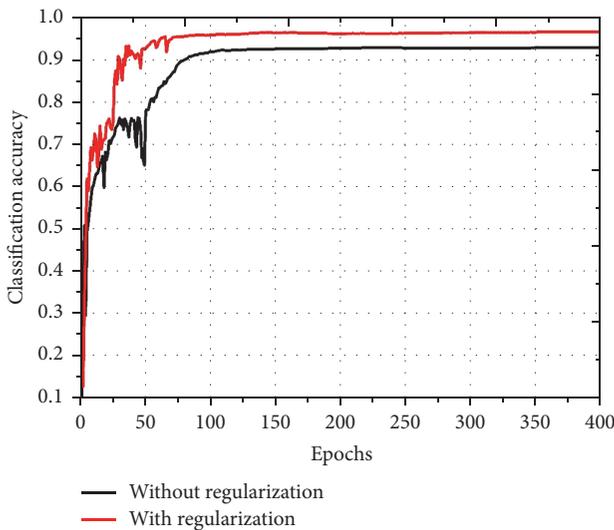


FIGURE 5: Classification accuracy of the methods with regularization and without regularization.

the semisupervised regularization is designed based on the remaining unlabelled samples. The main contributions of this paper are summarized as follows:

- (1) We used the active learning method to select the most informative and reliable samples, which are labelled manually and by the CNN, respectively. Thus, the overfitting issue is handled by adding the selected samples to the initial training set.
- (2) We designed a regularization term based on the class probability information of the unlabelled samples, and then the regularization term is added to the loss function of the CNN for retraining. Hence, the classification accuracy and generalization ability of the CNN were improved effectively.

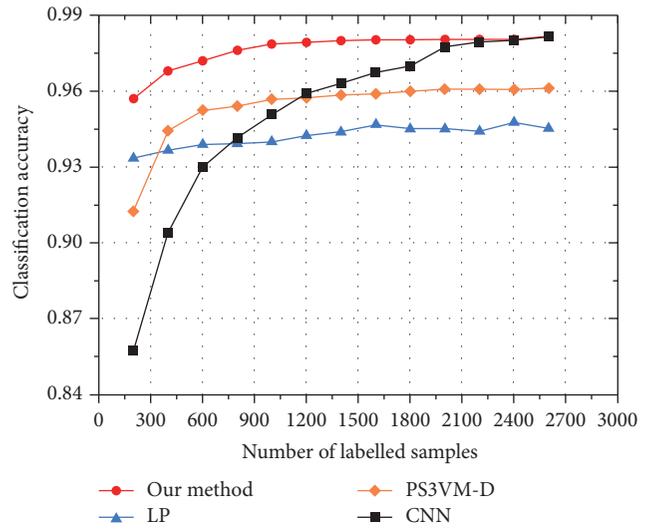


FIGURE 6: Classification accuracy of different classification methods.

From the experiment results, we observe that our method is effective and has a strong generalization ability especially when the labelled samples are insufficient; for example, the classification accuracy of our method is 95.7% when the number of the labelled samples is 236, which is apparently higher than the other methods using the same number of the labelled samples.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61771027; 61071139; 61471019; 61171122; 61501011; 61671035), the Aeronautical Science Foundation of China (20142051022), and the Preresearch Project (9140A07040515HK01009). Dr. E. Yang is supported in part by the RSE-NNSFC Joint Project (2017–2019) (6161101383) with China University of Petroleum (Huadong). Dr. H. Zhou is supported by UK EPSRC under Grants EP/N508664/1, EP/R007187/1, and EP/N011074/1 and Royal Society-Newton Advanced Fellowship under Grant NA160342.

References

- [1] Q. Zhao and J. C. Principe, “Support vector machines for SAR automatic target recognition,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 2, pp. 643–654, 2001.
- [2] W. Middelman, A. Ebert, and U. Thoennessen, “Automatic target recognition in SAR images based on a svm classification scheme,” in *Proceedings of the International Conference on Adaptive and Natural Computing Algorithms*, pp. 492–499, Springer, 2007.
- [3] Ö. Aytekin, M. Koc, and I. Ulusoy, “Local primitive pattern for the classification of SAR images,” *IEEE Transactions on*

- Geoscience and Remote Sensing*, vol. 51, no. 4, pp. 2431–2441, 2013.
- [4] S. Fukuda and H. Hirose, “A wavelet-based texture feature set applied to classification of multifrequency polarimetric SAR images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 5, pp. 2282–2286, 1999.
 - [5] X. Zhan, R. Zhang, D. Yin, and C. Huo, “SAR image compression using multiscale dictionary learning and sparse representation,” *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 5, pp. 1090–1094, 2013.
 - [6] S. Yang, Y. Zhang, and Y. Han, “Speckle reduction of SAR image through dictionary learning and point target enhancing approaches,” in *Proceedings of the 6th International Conference on Radar, RADAR 2011*, pp. 1926–1929, October 2011.
 - [7] G. Dong, G. Kuang, N. Wang, and W. Wang, “Classification via sparse representation of steerable wavelet frames on Grassmann manifold: application to target recognition in SAR image,” *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2892–2904, 2017.
 - [8] C. O. Dumitru and M. Datcu, “Information content of very high resolution SAR images: Study of feature extraction and imaging parameters,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 8, pp. 4591–4610, 2013.
 - [9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.
 - [10] P. Vasuki, S. Mohamed, and M. Roomi, “Automatic target classification in SAR images by multilayer back propagation neural network,” *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, no. 24, pp. 5510–5514, 2012.
 - [11] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *Computer Science*, 2014.
 - [12] J. Zhao, W. Guo, S. Cui, Z. Zhang, and W. Yu, “Convolutional Neural Network for SAR image classification at patch level,” in *Proceedings of the 36th IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2016*, pp. 945–948, July 2016.
 - [13] S. Chen and H. Wang, “SAR target recognition based on deep learning,” in *Proceedings of the 2014 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2014*, pp. 541–547, November 2014.
 - [14] Y. Zhou, H. Wang, F. Xu, and Y. Q. Jin, “Polarimetric SAR image classification using deep convolutional neural networks,” *IEEE Geoscience & Remote Sensing Letters*, no. 99, pp. 1–5, 2016.
 - [15] T. Wang, Y. Li, and H. Xiong, “A novel locally active learning method for SAR image classification,” in *Proceedings of the Joint 2014 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2014 and the 35th Canadian Symposium on Remote Sensing, CSRS 2014*, pp. 4596–4599, July 2014.
 - [16] M. Babaei, S. Tsoukalas, G. Rigoll, and M. Datcu, “Visualization-Based Active Learning for the Annotation of SAR Images,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 10, pp. 4687–4698, 2015.
 - [17] A. Samat, P. Gamba, P. Du, and J. Luo, “Active extreme learning machines for quad-polarimetric SAR imagery classification,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 35, pp. 305–319, 2015.
 - [18] C. Persello and L. Bruzzone, “Active and semisupervised learning for the classification of remote sensing images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 6937–6956, 2014.
 - [19] F. Wang and C. Zhang, “Label propagation through linear neighborhoods,” *IEEE Transactions on Knowledge & Data Engineering*, vol. 20, no. 1, pp. 55–67, 2007.
 - [20] R. Duan, W. Jiang, and H. Man, “Semi-supervised image classification in likelihood space,” in *Proceedings of the 2006 IEEE International Conference on Image Processing, ICIP 2006*, pp. 957–960, October 2006.
 - [21] F. Li, D. A. Clausi, L. Wang, and L. Xu, “A semi-supervised approach for ice-water classification using dual-polarization SAR satellite imagery,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2015*, pp. 28–35, June 2015.
 - [22] R. Hänsch and O. Hellwich, “Semi-supervised learning for classification of polarimetric SAR-data,” in *Proceedings of the 2009 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2009*, pp. III987–III990, July 2009.
 - [23] Liu P, Zhang H, Eom K B. Active Deep Learning for Classification of Hyperspectral Images. *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*, 2016, PP(99):1-13.
 - [24] K. Wang, D. Zhang, Y. Li et al., “Cost-effective active learning for deep image classification,” *IEEE Transactions on Circuits & Systems for Video Technology*, no. 1, article 99, 2017.
 - [25] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, “A semi-supervised convolutional neural network for hyperspectral image classification,” *Remote Sensing Letters*, vol. 8, no. 9, pp. 839–848, 2017.

Research Article

Application of the Intuitionistic Fuzzy InterCriteria Analysis Method with Triples to a Neural Network Preprocessing Procedure

Sotir Sotirov,¹ Vassia Atanassova,² Evdokia Sotirova,¹ Lyubka Doukovska,³ Veselina Bureva,¹ Deyan Mavrov,¹ and Jivko Tomov¹

¹Laboratory of Intelligent Systems, University “Prof. Dr. Assen Zlatarov”, 1 “Prof. Yakimov” Blvd., 8010 Burgas, Bulgaria

²Department of Bioinformatics and Mathematical Modelling, Institute of Biophysics and Biomedical Engineering, 105 “Acad. G. Bonchev” Str., 1113 Sofia, Bulgaria

³Department of Intelligent Systems, Institute of Information and Communication Technologies, 2 “Acad. G. Bonchev” Str., 1113 Sofia, Bulgaria

Correspondence should be addressed to Sotir Sotirov; ssotirov@btu.bg

Received 9 February 2017; Accepted 14 June 2017; Published 10 August 2017

Academic Editor: George A. Papakostas

Copyright © 2017 Sotir Sotirov et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The approach of InterCriteria Analysis (ICA) was applied for the aim of reducing the set of variables on the input of a neural network, taking into account the fact that their large number increases the number of neurons in the network, thus making them unusable for hardware implementation. Here, for the first time, with the help of the ICA method, correlations between triples of the input parameters for training of the neural networks were obtained. In this case, we use the approach of ICA for data preprocessing, which may yield reduction of the total time for training the neural networks, hence, the time for the network’s processing of data and images.

1. Introduction

Working with neural networks presents many difficulties; for example, the number of neurons in the perception of the individual values can be too large, and since a proportionally larger amount of memory and computing power is necessary to train the networks, this would lead to a longer periods for training. Therefore, researchers are forced to look for better methods for training neural networks. Backpropagation is the most applied such method—in it neural networks are trained with uplink (applied on a Multilayer Perceptron). There are, however, many other methods that accelerate the training of neural networks [1–3], by reducing memory usage, which in turn lowers the needed amount of computing power.

In the stage of preprocessing, the data at the input of the neural network can be used as a constant threshold value to distinguish static from dynamic activities, as it was done in [4]. This way, the amount of incidental values due to unforeseen circumstances is reduced.

Another approach is to use a wavelet-based neural network classifier to reduce the power interference in the training of the neural network or randomly stumbled measurements [5]. Here the discrete wavelet transform (DWT) technique is integrated with the neural network to build a classifier.

Particle Swarm Optimization (PSO) is an established method for parameter optimization. It represents a population-based adaptive optimization technique that is influenced by several “strategy parameters.” Choosing reasonable parameter values for PSO is crucial for its convergence behavior and depends on the optimization task. In [6] a method is presented for parameter metaoptimization based on PSO and it is applied to neural network training. The idea of Optimized Particle Swarm Optimization (OPSO) is to optimize the free parameters of PSO by having swarms within a swarm.

When working with neural networks it is essential to reduce the amount of neurons in the hidden layer, which

also reduces the number of weight coefficients of the neural network as a whole. This leads to a smaller dimension of the weight matrices, and hence the used amount of memory. An additional consequence from this is the decreased usage of computing power and the shortened training time [7].

Multilayer Perceptrons are often used to model complex relationships between sets of data. The removal of nonessential components of the data can lead to smaller sizes of the neural networks, and, respectively, to lower requirements for the input data. In [8] it is described that this can be achieved by analyzing the common interference of the network outputs, which is caused by distortions in the data that is passed to the neural network's inputs. The attempt to find superfluous data is based on the concept of sensitivity of linear neural networks. In [9] a neural network is developed, in which the outputs of the neurons of part of the layers are not connected to the next layer. The structure thus created is called a "Network in a Network." In this way part of the inputs of the neural network are reduced, which removes part of the information, and along with it part of the error accumulated during training and data transfer. The improved local connection method given in [9] produces a global collation by fundamental cards in the classification layer. This layer is easier to interpret and less prone to overloading than the traditional fully connected layers.

In this paper, we apply the intuitionistic fuzzy sets-based method of InterCriteria Analysis to reduce the number of input parameters of a Multilayer Perceptron. This will allow the reduction of the weight matrices, as well as the implementation of the neural network in limited hardware, and will save time and resources in training.

The neural network is tested after reducing the data (effectively the number of inputs), so as to obtain an acceptable relation between the input and output values, as well as the average deviation (or match) of the result.

2. Presentation of the InterCriteria Analysis

The InterCriteria Analysis (ICA) method is introduced in [10] by Atanassov et al. It can be applied to multiobject multicriteria problems, where measurements according to some of the criteria are slower or more expensive, which results in delaying or raising the cost of the overall process of decision-making. When solving such problems it may be necessary to adopt an approach for reasonable elimination of these criteria, in order to achieve economy and efficiency.

The ICA method is based on two fundamental concepts: intuitionistic fuzzy sets and index matrices. Intuitionistic fuzzy sets were first defined by Atanassov [11–13] as an extension of the concept of fuzzy sets defined by Zadeh [14]. The second concept on which the proposed method relies is the concept of index matrix, a matrix which features two index sets. The theory behind the index matrices is described in [15].

According to the ICA method, a set of objects is evaluated or measured against a set of criteria, and the table with these evaluations is the input for the method. The number of criteria can be reduced by calculating the correlations (differentiated in ICA to: positive consonance, negative

consonance, and dissonance) in each pair of criteria in the form of intuitionistic fuzzy pairs of values, that is, a pair of numbers in the interval $[0, 1]$, whose sum is also a number in this interval. If some (slow, expensive, etc.) criteria exhibit positive consonance with some of the rest of the criteria (that are faster, cheaper, etc.), and this degree of consonance is considered high enough with respect to some predefined thresholds, with this degree of precision the decision-maker may decide to omit them in the further decision-making process. The higher the number of objects involved in the measurement, the more precise the evaluation of the intercriteria consonances (correlations). This makes the approach completely data-driven and ongoing approbations over various application problems and datasets are helping us better perceive its reliability and practical applicability.

Let us consider a number of C_q criteria, $q = 1, \dots, n$, and a number of O_p objects, $p = 1, \dots, m$; that is, we use the following sets: a set of criteria $C_q = \{C_1, \dots, C_n\}$ and a set of objects $O_p = \{O_1, \dots, O_m\}$.

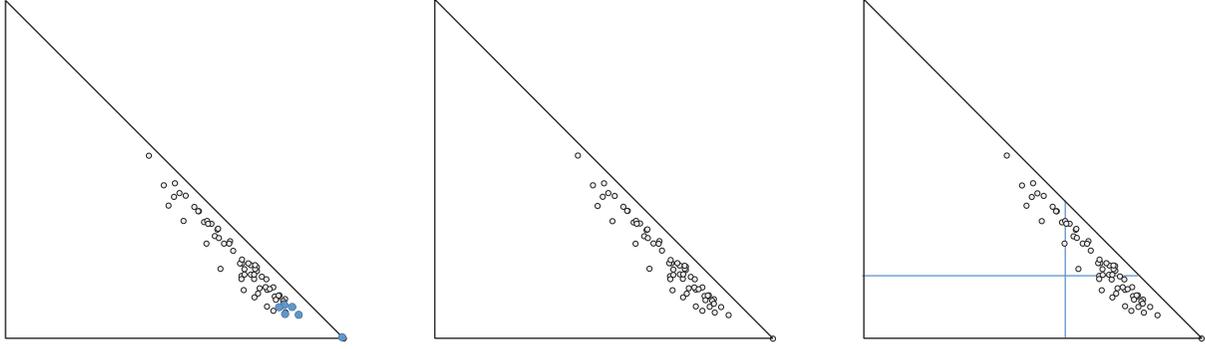
We obtain an index matrix M that contains two sets of indices, one for rows and another for columns. For every p , q ($1 \leq p \leq m$, $1 \leq q \leq n$), O_p in an evaluated object, C_q is an evaluation criterion, and a_{O_p, C_q} is the evaluation of the p th object against the q th criterion, defined as a real number or another object that is comparable according to a relation R with all the other elements of the index matrix M .

$$M = \begin{array}{c|cccccc} & C_1 & \cdots & C_k & \cdots & C_l & \cdots & C_n \\ \hline O_1 & a_{O_1, C_1} & \cdots & a_{O_1, C_k} & \cdots & a_{O_1, C_l} & \cdots & a_{O_1, C_n} \\ \cdots & \cdots \\ O_i & a_{O_i, C_1} & \cdots & a_{O_i, C_k} & \cdots & a_{O_i, C_l} & \cdots & a_{O_i, C_n} \\ \cdots & \cdots \\ O_j & a_{O_j, C_1} & \cdots & a_{O_j, C_k} & \cdots & a_{O_j, C_l} & \cdots & a_{O_j, C_n} \\ \cdots & \cdots \\ O_m & a_{O_m, C_1} & & a_{O_m, C_k} & & a_{O_m, C_l} & \cdots & a_{O_m, C_n} \end{array} \quad (1)$$

The next step is to apply the InterCriteria Analysis for calculating the evaluations. The result is a new index matrix M^* with intuitionistic fuzzy pairs $\langle \mu_{C_k, C_l}, \nu_{C_k, C_l} \rangle$ that represents an intuitionistic fuzzy evaluation of the relations between every pair of criteria C_k and C_l . In this way the index matrix M that relates the evaluated objects with the evaluating criteria can be transformed to another index matrix M^* that gives the relations among the criteria:

$$M^* = \begin{array}{c|ccc} & C_1 & \cdots & C_n \\ \hline C_1 & \langle \mu_{C_1, C_1}, \nu_{C_1, C_1} \rangle & \cdots & \langle \mu_{C_1, C_n}, \nu_{C_1, C_n} \rangle \\ \cdots & \cdots & \cdots & \cdots \\ C_n & \langle \mu_{C_n, C_1}, \nu_{C_n, C_1} \rangle & \cdots & \langle \mu_{C_n, C_n}, \nu_{C_n, C_n} \rangle \end{array} \quad (2)$$

The last step of the algorithm is to determine the degrees of correlation between groups of indicators depending of the chosen thresholds for μ and ν from the user. The correlations between the criteria are called "positive consonance," "negative consonance," or "dissonance." Here we use one of the


 FIGURE 1: Three alternatives for constructing the subset Σ [17].

Type of correlations between the criteria
 strong positive consonance [0,95; 1]
 positive consonance [0,85; 0,95]
 weak positive consonance [0,75; 0,85)
 weak dissonance [0,67; 0,75)
 dissonance [0,57; 0,67)
 strong dissonance [0,43; 0,57)
 dissonance [0,33; 0,43)
 weak dissonance [0,25; 0,33)
 weak negative consonance [0,15; 0,25)
 negative consonance [0,15; 0,05)
 strong negative consonance [0,05; 0]

Box 1: Type of correlations.

possible approaches to defining these thresholds, namely, the scale shown in Box 1 [16].

3. InterCriteria Analysis with Triples

The algorithm for identifying intercriteria triples is introduced in [17] by Atanassova et al.

Step 1. Starting from the input dataset of m objects measured against n criteria, we calculate the total number of $n(n-1)/2$ intuitionistic fuzzy pairs standing for the intercriteria consonances and plot these pairs as points onto the intuitionistic fuzzy triangle. Instead of maintaining a pair of two numbers for each pair of criteria C_i-C_j , namely, $\langle \mu_{ij}, \nu_{ij} \rangle$ we calculate (see [18]) for each pair the number d_{ij} :

$$d_{ij} = \sqrt{(1 - \mu_{ij})^2 + \nu_{ij}^2} \quad (3)$$

giving its distance from the (1;0) point, that is, the image of the complete Truth onto the intuitionistic fuzzy triangle. Our aim is to identify top-down all the $n(n-1)/2$ calculated values that are closest to the (1;0) and, at the same time, closest to each other; hence we sort them in ascending order by their distance to (1;0); see the example in Table 2.

Step 2. Let us denote with Σ the subset of the closest to (1;0) triples of criteria. The way we construct the subset Σ may

slightly differ per user preference or external requirement, with at least three possible alternatives, as listed below (see Figure 1):

- (2.1) Select top p or top $q\%$ of the $n(n-1)/2$ ICA pairs (predefined number of elements of the subset Σ).
- (2.2) Select all ICA pairs whose corresponding points are within a given radius r from the (1;0) point.
- (2.3) Select all ICA pairs whose corresponding points fall within the trapezoid formed between the abscissa, the hypotenuse, and the two lines corresponding to $y = \alpha$ and $x = \beta$ for two predefined numbers $\alpha, \beta \in [0; 1]$.

Step 3. Check if there are triples of criteria, each pair of which corresponds to a point, belonging to the subset Σ . If no, then no triples of criteria conform with the stipulated requirements. However, if triples are to be found, then we extend the subset Σ accordingly, by either taking a larger number p or q (Substep (2.1)), or a larger radius r (Substep (2.2)), or smaller α and/or larger β (Substep (2.3)). If now the subset Σ contains triples of criteria that simultaneously fulfil the requirements, then go to Step 4.

Step 4. We start top-down with the first pair of criteria, let it be C_i-C_j , that is, the pair with the smallest d_{ij} , thus ensuring maximal proximity of the corresponding point, say, P_{ij} , to (1;0) point. We may pick the third criterion in the triple either as C_k which is the next highest correlating criterion with C_i , that is, P_{ik} with $d_{ik} (>d_{ij})$, or as C_l which is the next highest correlating criterion with C_j , that is, P_{jl} with $d_{jl} (>d_{ij})$, noting that it is possible to have $d_{ik} = d_{jl}$. Then, we check the distances to (1;0) of the respective third points P_{jk} and P_{il} , taking that triple of criteria $C_i-C_j-C_k$ or $C_i-C_j-C_l$ that has the

$$\min(d_{ij} + d_{ik} + d_{jk}, d_{ij} + d_{il} + d_{jl}). \quad (4)$$

Then for each triple of criteria $C_i-C_j-C_x$ (where $x \in \{k, l\}$), we calculate the median point of the so formed triangle, which is a point plotted in the intuitionistic fuzzy triangle with coordinates:

$$\langle \tilde{\mu}, \tilde{\nu} \rangle = \left\langle \frac{\mu_{ij} + \mu_{jx} + \mu_{xi}}{3}, \frac{\nu_{ij} + \nu_{jx} + \nu_{xi}}{3} \right\rangle. \quad (5)$$

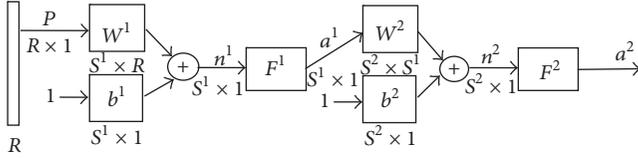


FIGURE 2: Abbreviated notation of a classical Multilayer Perceptron.

This pair gives us the level of $(\bar{\mu}, \bar{\nu})$ -consonance of the whole triple. Repeat Step 4 until the number of the triples in the subset Σ is exhausted.

4. Artificial Neural Networks

The artificial neural networks [4, 19] are one of the tools that can be used for object recognition and identification. In the first step, it has to be learned and after that we can use for the recognitions and for predictions of the properties of the materials. Figure 2 shows in abbreviated notation of a classic two-layered neural network.

In the two-layered neural networks, one layer's exits become entries for the next one. The equations describing this operation are

$$a^2 = f^2(w^2 f^1(w^1 p + b^1) + b^2), \quad (6)$$

where

- (i) a^m is the exit of the m th layer of the neural network for $m = 1, 2$;
- (ii) w^m is a matrix of the weight coefficients of the each of the entries of the m th layer;
- (iii) b is the neuron's entry bias;
- (iv) f^1 is the transfer function of the 1st layer;
- (v) f^2 is the transfer function of the 2nd layer.

The neuron in the first layer receives p outside entries. The neurons' exits from the last layer determine the neural network's exits as a .

The "backpropagation" algorithm [20] is used for learning the neural networks. When the multilayer neural network is trained, usually the available data has to be divided into three subsets. The first subset, named "Training set," is used for computing the gradient and updating the network weights and biases. The second subset is named "Validation set." The error of the validation set is monitored during the training process. The validation error normally decreases during the initial phase of training, as does the training set error. Sometimes, when the network begins to overfit the data, the error of the validation set typically begins to rise. When the validation error increases for a specified number of iterations, the training stops and the weights and biases at the minimum of the validation error are returned [4]. The last subset is named "test set." The sum of these three sets has to be 100% of the learning couples.

For this investigation we use MATLAB and neural network structure 8:45:1 (8 inputs, 45 neurons in hidden layer,

and one output) (Figure 2). The numbers of the weight coefficients are $9 \times 45 = 405$.

The proposed method is focused on removing part of the number of neurons (and weight coefficients) and thus does not reduce the average deviation of the samples, used for the learning testing and validating the neural network.

5. Testing

We consider a number of C_q criteria, $q = 1, \dots, n$, and a number of O_p measurements of cetane number of crude oil, $p = 1, \dots, m$; that is, we use the following sets: a set of group of criteria $C_q = \{C_1, \dots, C_n\}$ and a set of measurements of cetane number $O_p = \{O_1, \dots, O_m\}$.

The ICA method was applied to the 140 crude oil probes, measured against 8 criteria as listed below:

- (I) density at 15°C g/cm³;
- (II) 10% (v/v) ASTM D86 distillation, °C;
- (III) 50% (v/v) ASTM D86 distillation, °C;
- (IV) 90% (v/v) ASTM D86 distillation, °C;
- (V) refractive index at 20°C;
- (VI) H₂ content, % (m/m);
- (VII) aniline point, °C;
- (VIII) molecular weight g/mol.

So we work with a 140×8 table, and a software application that implements the ICA algorithm returns the results in the form of two index matrices (see Tables 1 and 2), containing, respectively, the membership and the nonmembership parts of the intuitionistic fuzzy correlations detected between each pair of criteria (28 pairs). The values in the matrix are colored in red-yellow-green color scale for the varying degrees of consonance and dissonance from green (highest values) to yellow. Naturally, each criterion best correlates with itself, which gives the respective intuitionistic fuzzy pairs $\langle 1; 0 \rangle$, or 1s and 0s, along the main diagonals of Tables 1 and 2.

In Table 3 the relations between the pairs of criteria obtained by applying the ICA method are shown.

The calculated distance d_{ij} for each pair of criteria C_i - C_j from the $(1; 0)$ point in the intuitionistic fuzzy triangle is shown in Table 4 (note that $d_{ij} \in [0, \sqrt{2}]$).

The next step is to choose the pair C_i - C_j with the smallest d_{ij} , thus ensuring maximal proximity of the corresponding point to $(1; 0)$ point. We pick the third criterion in the triple either as C_k that is the next highest correlating criterion with C_i , or as C_l that is the next highest correlating criterion with C_j , taking that triple of criteria C_i - C_j - C_k or C_i - C_j - C_l that has the $\min(d_{ij} + d_{ik} + d_{jk}, d_{ij} + d_{il} + d_{jl})$. In Table 5 the pairs of criteria C_i - C_j in "strong positive consonance," "positive consonance," and "weak positive consonance" are shown.

On the input of the neural network we put the experimental data for obtaining cetane number of crude oil. Testing is done as at the first step; all the measurements of the 140 crude oil probes against the 8 criteria are analyzed in order to make a comparison of the obtained results thereafter. For this comparison to be possible, the predefined weight coefficients

TABLE 1: Membership parts of the IF pairs, giving the InterCriteria correlations.

μ	(I)	(II)	(III)	(IV)	(V)	(VI)	(VII)	(VIII)
(I)	1	0.699	0.770	0.658	0.956	0.176	0.446	0.703
(II)	0.699	1	0.787	0.597	0.676	0.408	0.640	0.775
(III)	0.770	0.787	1	0.777	0.728	0.395	0.665	0.922
(IV)	0.658	0.597	0.777	1	0.627	0.468	0.674	0.771
(V)	0.956	0.676	0.728	0.627	1	0.134	0.404	0.661
(VI)	0.176	0.408	0.395	0.468	0.134	1	0.730	0.473
(VII)	0.446	0.640	0.665	0.674	0.404	0.730	1	0.743
(VIII)	0.703	0.775	0.922	0.771	0.661	0.473	0.743	1

TABLE 2: Nonmembership parts of the IF pairs, giving the InterCriteria relations.

ν	(I)	(II)	(III)	(IV)	(V)	(VI)	(VII)	(VIII)
(I)	0	0.288	0.217	0.326	0.042	0.822	0.552	0.295
(II)	0.288	0	0.204	0.391	0.312	0.580	0.348	0.213
(III)	0.217	0.204	0	0.212	0.261	0.595	0.325	0.068
(IV)	0.326	0.391	0.212	0	0.359	0.518	0.312	0.215
(V)	0.042	0.312	0.261	0.359	0	0.866	0.596	0.339
(VI)	0.822	0.580	0.595	0.518	0.866	0	0.270	0.527
(VII)	0.552	0.348	0.325	0.312	0.596	0.270	0	0.257
(VIII)	0.295	0.213	0.068	0.215	0.339	0.527	0.257	0

TABLE 3: Correlations between the pairs of criteria.

Type of InterCriteria Relation	Pairs of criteria
Strong positive consonance [0.95; 1]	(I-V)
Positive consonance [0.85; 0.95]	(III-VIII)
Weak positive consonance [0.75; 0.85]	(II-III, III-IV, II-VIII, IV-VIII, I-III)
Weak dissonance [0.67; 0.75]	(VII-VIII, III-V, VI-VII, I-II, I-VIII, II-V, IV-VII)
Dissonance [0.57; 0.67]	(III-VII, I-IV, V-VIII, II-VII, IV-V, II-IV)
Strong dissonance [0.43; 0.57]	(IV-VI, VI-VIII, I-VII)
Dissonance [0.33; 0.43]	(II-VI, V-VII, III-VI)
Weak dissonance [0.25; 0.33]	0
Weak negative consonance [0.15; 0.25]	(I-VI)
Negative consonance [0.15; 0.05]	(V-VI)
Strong negative consonance [0.05; 0]	0

and offsets that are normally random values between -1 and 1 are now established and are the same in all studies with coefficients 1.

For the learning process, we set the following parameters: performance (MSE) = 0.00001; validation check = 25. The input vector is divided into three different parts: training (70/100); validation (15/100); and testing (15/100). For target we use the cetane number ASTM D613.

At the first step of the testing process, we use all the 8 criteria listed above, in order to train the neural network.

After the training process all input values are simulated by the neural network.

The average deviation of the all 140 samples is 1,8134. The coefficient R (regression R values measure the correlation between outputs and targets) obtained from the MATLAB program is 0.97434 (see Table 6).

At the next step of the testing process, we make a fork and try independently to remove one of the columns and experiment with data from the remaining seven columns. We compare the results in the next section, "Discussion." First, we make a reduction of column 1 (based on Table 5) and put the data on the input of the neural network.

After the training process all input values are simulated. The average deviation of all the 140 samples is 1.63 and the coefficient R is 0.9772.

At the next step, we alternatively perform reduction of column 3 (according to Table 5), and put the data on the input of the neural network.

After the training process all input values are simulated. The average deviation of the all 140 samples is 1.8525 and the coefficient R is 0.97256. After that we can proceed with columns 5, 2, 8, and 4.

Now, at the next step, we proceed with feeding the neural network with 6 inputs, with the reduction of both columns, 3 and 5, according to the data from Table 5. The average deviation of all the 140 samples is 1.7644 and the coefficient R is 0.97089. In the same way we can reduce the inputs: 1 and 5, 1 and 3, 2 and 3, 3 and 8, 3 and 4, and 4 and 8, simultaneously.

At the next step, we reduce the number of inputs with one more, that is, we put on the input of the neural network experimental data from 5 inputs, with removed columns 1, 3, and 5. The average deviation of all the 140 samples is 1.857 and

TABLE 4: Distance d_{ij} for each pair of criteria C_i-C_j .

d	(I)	(II)	(III)	(IV)	(V)	(VI)	(VII)	(VIII)
(I)	0	0.416	0.316	0.473	0.061	1.165	0.783	0.419
(II)	0.416	0	0.295	0.561	0.450	0.829	0.501	0.310
(III)	0.316	0.295	0	0.307	0.377	0.849	0.467	0.104
(IV)	0.473	0.561	0.307	0	0.518	0.742	0.452	0.314
(V)	0.061	0.450	0.377	0.518	0	1.225	0.843	0.480
(VI)	1.165	0.829	0.849	0.742	1.225	0	0.382	0.745
(VII)	0.783	0.501	0.467	0.452	0.843	0.382	0	0.363
(VIII)	0.419	0.310	0.104	0.314	0.480	0.745	0.363	0

TABLE 5: Distance d_{ij} for pair of criteria C_i-C_j in positive consonance.

C_i	C_j	m_{ij}	d_{ij}	C_k	m_{ik}	d_{ik}	d_{jk}	C_l	m_{jl}	d_{il}	d_{jl}	$\min(d_{ij}+d_{ik}+d_{jk}, d_{ij}+d_{il}+d_{jl})$	Chosen triple of criteria	$\langle \bar{\mu}, \bar{\nu} \rangle$
(I)	(V)	0.956	0.061	(III)	0.770	0.319	0.377	(III)	0.728	0.319	0.377	0.756	$C_{(I)}-C_{(V)}-C_{(III)}$	$\langle 0.818; 0.173 \rangle$
(III)	(VIII)	0.922	0.104	(II)	0.787	0.295	0.310	(II)	0.775	0.295	0.310	0.709	$C_{(III)}-C_{(VIII)}-C_{(II)}$	$\langle 0.828; 0.162 \rangle$
(II)	(III)	0.787	0.295	(VIII)	0.775	0.310	0.104	(IV)	0.777	0.561	0.307	0.709	$C_{(II)}-C_{(III)}-C_{(VIII)}$	$\langle 0.828; 0.162 \rangle$
(III)	(IV)	0.777	0.307	(I)	0.770	0.319	0.473	(VIII)	0.771	0.104	0.314	0.725	$C_{(III)}-C_{(IV)}-C_{(VIII)}$	$\langle 0.823; 0.165 \rangle$
(II)	(VIII)	0.775	0.310	(I)	0.699	0.416	0.418	(IV)	0.771	0.561	0.314	1.144	$C_{(II)}-C_{(VIII)}-C_{(I)}$	$\langle 0.726; 0.265 \rangle$
(IV)	(VIII)	0.771	0.314	(VII)	0.674	0.452	0.363	(VII)	0.743	0.452	0.363	1.129	$C_{(IV)}-C_{(VIII)}-C_{(VII)}$	$\langle 0.729; 0.261 \rangle$
(I)	(III)	0.770	0.316	(VIII)	0.703	0.418	0.104	(V)	0.728	0.061	0.377	0.753	$C_{(I)}-C_{(III)}-C_{(V)}$	$\langle 0.818; 0.173 \rangle$

TABLE 6: Correlation coefficients for pair of criteria C_i-C_j according to Pearson.

C_i	C_j	Correlation coefficient C_i-C_j	C_k	Correlation coefficient C_i-C_k	C_l	Correlation coefficient C_j-C_l	$\max(\text{correlation coefficient } C_i-C_j + \text{correlation coefficient } C_i-C_k; \text{correlation coefficient } C_i-C_j + \text{correlation coefficient } C_j-C_l)$	Chosen triple of criteria
(I)	(V)	0,989	(III)	0,616	(III)	0,495	1,605	(I-V-III)
(III)	(VIII)	0,971	(IV)	0,819	(II)	0,797	1,789	(III-VIII-IV)
(VI)	(VII)	0,831	(VIII)	0,024	(VIII)	0,576	1,406	(VI-VII-VIII)
(III)	(IV)	0,819	(VIII)	0,971	(VIII)	0,796	1,789	(III-IV-VIII)

the coefficient R is 0.97208 (see Table 6). In the same way are removed the parameters 2, 3, and 8 and 3, 4, and 8.

Finally, we experiment with the reduction of the fourth column, feeding the neural network with only 4 inputs. After the reduced columns 1, 2, and 4, the fourth reduced column is column 5. After the simulation the average deviation of the all 140 samples is 2.19 and the coefficient R obtained from the MATLAB program is 0.95927.

6. Discussion

In support of the method, Tables 6, 7, and 8 present the correlation coefficients between the different criteria. The tables also present the maximal values of the coefficient sums per criteria. In the last column, the triples of selected criteria are given, as sorted in the descending way by the *correlation coefficient* C_i-C_j .

In Table 9 compilations between ICA approach and correlation analysis according to Pearson, Kendall, and Spearman are shown.

The selected pairs, based on the four methods, are identical in the first row. In the second row three of the methods yield identical results (ICA, Kendall, and Spearman), and the only difference is in the selected criteria as calculated by the Pearson method. In the third row, the situation is the same. Here the triples are the same with precision of ordering. Only the triple of correlation criteria calculated by the *Pearson* method is different. In the fourth row, the triples are quite similar. The triples calculated by *ICA* and *Pearson* are identical. The triple determined by *Kendall* correlation coincides with the first row of the table. The last triple, defined by the *Spearman* correlation, coincides with the second and third row of the triples defined by the correlation analyses of *ICA*, *Pearson*, and *Spearman*.

So far, such a detailed comparison between the four methods has been conducted over medical [21, 22] and petrochemical [23] data. It was observed that considerable divergence of the ICA results from the results obtained by the rest of the methods is only found when the input data contain mistakes, as a result of misplacing the decimal point with at

TABLE 7: Correlation coefficients for pair of criteria C_i-C_j according to Kendall.

C_i	C_j	Correlation coefficient C_i-C_j	C_k	Correlation coefficient C_i-C_k	C_l	Correlation coefficient C_j-C_l	max(correlation coefficient C_i-C_j + correlation coefficient C_i-C_k ; correlation coefficient C_i-C_j + correlation coefficient C_j-C_l)	Chosen triple of criteria
(I)	(V)	0,915	(III)	0,557	(III)	0,470	1,472	(I-V-III)
(III)	(VIII)	0,858	(II)	0,582	(II)	0,566	1,440	(III-VIII-II)
(II)	(III)	0,582	(VIII)	0,566	(VIII)	0,566	1,147	(II-III-VIII)
(I)	(III)	0,557	(V)	0,915	(VIII)	0,858	1,472	(I-III-V)

TABLE 8: Correlation coefficients for pair of criteria C_i-C_j according to Spearman.

C_i	C_j	Correlation coefficient C_i-C_j	C_k	Correlation coefficient C_i-C_k	C_l	Correlation coefficient C_j-C_l	max(correlation coefficient C_i-C_j + correlation coefficient C_i-C_k ; correlation coefficient C_i-C_j + correlation coefficient C_j-C_l)	Chosen triple of criteria
(I)	(V)	0,988	(III)	0,728	(III)	0,641	1,716	(I-V-III)
(III)	(VIII)	0,962	(II)	0,762	(II)	0,753	1,724	(III-VIII-II)
(II)	(III)	0,762	(VIII)	0,753	(VIII)	0,962	1,724	(II-III-VIII)
(II)	(VIII)	0,753	(III)	0,762	(III)	0,962	1,715	(II-VIII-III)

TABLE 9

	ICA	Pearson	Kendall	Spearman
(1)	(I-V-III)	(I-V-III)	(I-V-III)	(I-V-III)
(2)	(III-VIII-II)	(III-VIII-IV)	(III-VIII-II)	(III-VIII-II)
(3)	(II-III-VIII)	(VI-VII-VIII)	(II-III-VIII)	(II-III-VIII)
(4)	(III-IV-VIII)	(III-IV-VIII)	(I-III-V)	(II-VIII-III)

least one position to the left or to the right. We anticipate in the future a theoretical research for checking the validity of this practical observation. If it proves to be true, then ICA, together with the rest three types of analysis, will turn into a criterion for data correctness.

As we stated above, reducing the number of input parameters of a classical neural network leads to reduction of the weight matrices, resulting in implementation of the neural network in limited hardware and saving time and resources in training. For this aim, we use the intuitionistic fuzzy sets-based approach of InterCriteria Analysis (ICA), which gives dependencies between the criteria and thus helps us reduce the number of highly correlating input parameters, yet keeping high enough the level of precision.

Table 10 summarizes the most significant parameters of the process of testing the neural network with different numbers of inputs, gradually reducing the number in order to discover optimal results. These process parameters are the NN-specific parameters “average deviation,” “regression coefficient R ,” and “number of the weight coefficients.”

The average deviation when we use 8 input vectors is 1.8134 with number of weight coefficients 405. By reducing the number of the inputs the number of weight coefficients is also decreased which theoretically is supposed to reduce the matching coefficient. In this case the removal of column 1 (and therefore one input is removed) causes further

decreasing the average deviation of 1.6327. The additional information (without column 5) used for training the neural network is very little, and the total Mean Square Error is less. The result is better compared to the formerly used attempt by training the neural network with 8 data columns.

When we use 7 columns (and 7 inputs of neural networks) excluding some of the columns gives better result than the previous one. This shows that, while maintaining the number of weight coefficients and reducing the maximal membership in the intercriteria IF pairs, the neural network receives an additional small amount of information which it uses for further learning.

Best results (average deviation = 1.5716) are obtained by removing the two columns (6 inputs without inputs 1 and 3) with the greatest membership components of the respective d .

In this case, the effect of reducing the number of weight coefficients from 360 to 315 and the corresponding MSE is greater than the effect of the two columns.

The use of 5 columns (without columns 1, 3, and 5) leads to a result which is less than the previous, that is, 1.857. This shows that with reducing the number of weight coefficients (and the total MSE) and the information at the input of the neural network a small amount of information is lost with which the network is trained. As a result, the overall accuracy of the neural network is decreased.

TABLE 10: Table of comparison.

Number of inputs	Average deviation	Regression coefficient R	Number of the weight coefficients
8 inputs	1.8134	0.97434	405
7 inputs without input 1	1.6327	0.9772	360
7 inputs without input 3	1.8525	0.97256	360
7 inputs without input 5	1.6903	0.9734	360
7 inputs without input 2	2.1142	0.96511	360
7 inputs without input 8	1.7735	0.97511	360
7 inputs without input 4	1.9913	0.96932	360
6 inputs without inputs 3, 5	1.7644	0.97089	315
6 inputs without inputs 1, 5	1.8759	0.97289	315
6 inputs without inputs 1, 3	1.5716	0.97881	315
6 inputs without inputs 2, 3	2.0716	0.96581	315
6 inputs without inputs 3, 8	1.9767	0.97213	315
6 inputs without inputs 3, 4	1.9792	0.97163	315
6 inputs without inputs 4, 8	2.0174	0.96959	315
5 inputs without inputs 1, 3, 5	1.857	0.97209	270
5 inputs without inputs 2,3, 8	2.0399	0.96713	270
5 inputs without inputs 3, 4, 8	2.0283	0.96695	270
4 inputs without inputs 1, 2, 4, 5	2.217	0.95858	225
4 inputs without inputs 2, 3, 4, 8	2.1989	0.95927	225

The worst results (average deviation = 2.217) are obtained in the lowest number of columns—4. In this case, columns 1, 2, 4, and 5 are removed. Although the number of weight coefficients here is the smallest, the information that is used for training the neural network is less informative.

7. Conclusion

In the paper we apply the newest leg of theoretical research on InterCriteria Analysis to a dataset with the measurements of 140 probes of crude oil against 8 physicochemical criteria. On the first step we put all data from these measurements in the input of a classical neural network. After performing ICA analysis of the pairwise intercriteria correlations, we apply the recently developed method for identification of intercriteria triples in attempt to reduce the inputs of the neural network, without significant loss of precision. This leads to a reduction of the weight matrices, thus allowing implementation of the neural network on limited hardware and saving time and resources in training.

Very important aspect of the testing of the neural network after reducing some of the data (resp., the number of inputs) is to obtain an acceptable correlation between the input and output values, as well as the average deviation (or match) of the result.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are thankful for the support provided by the Bulgarian National Science Fund under Grant Ref. no. DFNI-I-02-5 “*InterCriteria Analysis: A New Approach to Decision Making.*”

References

- [1] S. Bellis, K. M. Razeeb, C. Saha et al., “FPGA implementation of spiking neural networks - An initial step towards building tangible collaborative autonomous agents,” in *Proceedings of the 2004 IEEE International Conference on Field-Programmable Technology, FPT '04*, pp. 449–452, December 2004.
- [2] S. Himavathi, D. Anitha, and A. Muthuramalingam, “Feedforward neural network implementation in FPGA using layer multiplexing for effective resource utilization,” *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 880–888, 2007.
- [3] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, “Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 156–167, 2006.
- [4] S. Haykin, *Neural Networks: A Comprehensive Foundation*, NY: Macmillan, 1994.
- [5] Z.-L. Gaing, “Wavelet-based neural network for power disturbance recognition and classification,” *IEEE Transactions on Power Delivery*, vol. 19, no. 4, pp. 1560–1568, 2004.
- [6] M. Meissner, M. Schmuker, and G. Schneider, “Optimized Particle Swarm Optimization (OPSO) and its application to

- artificial neural network training,” *BMC Bioinformatics*, vol. 7, article 125, 2006.
- [7] S. Sotirov, V. Atanassova, E. Sotirova, V. Bureva, and D. Mavrov, “Application of the intuitionistic fuzzy InterCriteria analysis method to a neural network preprocessing procedure,” in *Proceedings of the 16th World Congress of the International Fuzzy Systems Association (IFSA) 9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT)*, pp. 1559–1564.
- [8] J. M. Zurada, A. Malinowski, and I. Cloete, “Sensitivity analysis for minimization of input data dimension for feedforward neural network,” in *Proceedings of the 1994 IEEE International Symposium on Circuits and Systems. Part 3 (of 6)*, pp. 447–450, June 1994.
- [9] M. Lin, Q. Chen, and S. Yan, “Network in network,” arXiv preprint arXiv:1312.4400, 2013.
- [10] K. Atanassov, D. Mavrov, and V. Atanassova, “InterCriteria decision making. A new approach for multicriteria decision making, based on index matrices and intuitionistic fuzzy sets,” in *Issues in IFS and GN*, p. 11, 11, 1–7, 2014.
- [11] K. T. Atanassov, “Intuitionistic fuzzy sets,” *Fuzzy Sets and Systems*, vol. 20, no. 1, pp. 87–96, 1986.
- [12] K. Atanassov, *Intuitionistic Fuzzy Sets: Theory and Applications*, Physica-Verlag, Heidelberg, Germany, 1999.
- [13] K. T. Atanassov, “Intuitionistic Fuzzy Relations (IFRs),” in *On Intuitionistic Fuzzy Sets Theory*, vol. 283 of *Studies in Fuzziness and Soft Computing*, pp. 147–193, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [14] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [15] K. T. Atanassov, *Index matrices: towards an augmented matrix calculus*, vol. 573 of *Studies in Computational Intelligence*, Springer, Cham, 2014.
- [16] K. Atanassov, V. Atanassova, and G. Gluhchev, “InterCriteria analysis: ideas and problems,” in *Notes on Intuitionistic Fuzzy Sets*, vol. 21, pp. 81–88, 1 edition, 2015.
- [17] V. Atanassova, L. Doukowska, A. Michalíková, and I. Radeva, “InterCriteria analysis: from pairs to triples,” *Notes on Intuitionistic Fuzzy Sets*, vol. 22, no. 5, pp. 98–110, 2016.
- [18] V. Atanassova, D. Mavrov, L. Doukowska, and K. Atanassov, “Discussion on the threshold values in the InterCriteria decision making approach,” *Notes on Intuitionistic Fuzzy Sets*, vol. 20, no. 2, pp. 94–99, 2014.
- [19] M. Hagan, H. Demuth, and M. Beale, *Neural Network Design*, PWS Publishing, Boston, MA, USA, 1996.
- [20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [21] S. Krumova, S. Todinova, D. Mavrov et al., “InterCriteria analysis of calorimetric data of blood serum proteome,” *Biochimica et Biophysica Acta (BBA) - General Subjects*, vol. 1861, no. 2, pp. 409–417, 2017.
- [22] S. Todinova, D. Mavrov, S. Krumova et al., “Blood plasma thermograms dataset analysis by means of interCriteria and correlation analyses for the case of colorectal cancer,” *International Journal Bioautomation*, vol. 20, no. 1, pp. 115–124, 2016.
- [23] D. S. Stratiev, S. Sotirov, I. Shishkova et al., “Investigation of relationships between bulk properties and fraction properties of crude oils by application of the interCriteria analysis,” *Petroleum Science and Technology*, vol. 34, no. 13, pp. 1113–1120, 2016.

Research Article

Patch-Based Principal Component Analysis for Face Recognition

Tai-Xiang Jiang, Ting-Zhu Huang, Xi-Le Zhao, and Tian-Hui Ma

School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 610054, China

Correspondence should be addressed to Ting-Zhu Huang; tingzhuhuang@126.com

Received 1 March 2017; Revised 4 June 2017; Accepted 7 June 2017; Published 11 July 2017

Academic Editor: George A. Papakostas

Copyright © 2017 Tai-Xiang Jiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We have proposed a patch-based principal component analysis (PCA) method to deal with face recognition. Many PCA-based methods for face recognition utilize the correlation between pixels, columns, or rows. But the local spatial information is not utilized or not fully utilized in these methods. We believe that patches are more meaningful basic units for face recognition than pixels, columns, or rows, since faces are discerned by patches containing eyes and noses. To calculate the correlation between patches, face images are divided into patches and then these patches are converted to column vectors which would be combined into a new “image matrix.” By replacing the images with the new “image matrix” in the two-dimensional PCA framework, we directly calculate the correlation of the divided patches by computing the total scatter. By optimizing the total scatter of the projected samples, we obtain the projection matrix for feature extraction. Finally, we use the nearest neighbor classifier. Extensive experiments on the ORL and FERET face database are reported to illustrate the performance of the patch-based PCA. Our method promotes the accuracy compared to one-dimensional PCA, two-dimensional PCA, and two-directional two-dimensional PCA.

1. Introduction

The principal component analysis, one of the most popular *multivariate statistical techniques* [1], has been widely used in the areas of pattern recognition and signal processing [2]. It is a statistical method under the broad title of *factor analysis* [3]. The modern instantiation PCA was formalized by Hotelling [1, 4] who also coined the term *principal component*, but in fact we can trace its origin back to [5] or even Cauchy [6]. PCA analyzes the observed data which is usually described by several dependent and intercorrelated variables. Its goal is to extract the important information from the data and to express this information as a set of new orthogonal variables called principal components.

There are numerous PCA-based methods for face recognition, from one-dimensional PCA [7] to two-directional two-dimensional PCA known as $(2D)^2$ PCA [8]. All these methods rely on two points. Firstly, the pattern of similarity of the observations and the variables can be represented as points in maps by PCA [2, 9, 10]. Secondly, the similarity of face images can be in some sense “calculated” by evaluating the distance of these points.

The main idea of one-dimensional PCA method for face recognition is *eigenspace projection*. A projection matrix is

obtained by maximum the image covariance, which shows the correlation between pixels in each training data (or say labeled face image). The next step is projecting the 1D vectors (previously constructed from 2D images) into the feature space [11]. In addition, the eigenvectors corresponding to large eigenvalues (or say the principle components), which would resemble a human face after transforming into matrix of the same size of the original face image, are called *eigenface*. Then the nearest neighbor (NN) classifier is adopted by computing the distance in the eigenspace to verify the identity of unlabeled face images. For instance, we would be sure that the face belongs to the 1st individual, if an unlabeled face image is nearest to one of the 1st individual’s labeled face images in the eigenspace. However transforming 2D images into 1D vectors always leads to a very high-dimensional space, in which the calculating of the covariance matrix, which shows the correlation of pixels, is difficult. The size of the covariance matrix achieves 10000×10000 , if the size of face images is 100×100 . Hence, it would consume a lot of time to evaluate the eigenvectors of a such large size covariance matrix.

Two-dimensional principal component analysis (2DPCA) [12], as opposed to eigenface, projects face images into a subfeature space directly without image-to-vector

conversion. This direct projection not only enables the preservation of partial image spatial information but also reduces computational burden [13]. The so-called image covariance matrix of 2DPCA, which is constructed directly using the original face image matrixes, is much smaller than the covariance matrix of eigenface method. In 2DPCA, the image covariance (scatter) matrix, which is somehow the same as the covariance matrix in the eigenface, shows the correlation of each column of each image. Motivated by 2DPCA, $(2D)^2$ PCA [8] calculate the correlation from two directions of both of the columns and rows. 2DPCA and $(2D)^2$ PCA have achieved good results in face recognition. However these methods fail to fully explore the local spatial information.

In order to further explore the local spatial information, let us take a look at the track of existing methods. Eigenface method only calculates the correlation of pixels, while the 2DPCA only calculates the correlation of columns. And $(2D)^2$ PCA calculates the correlation of both columns and rows in the same time. Accuracy is promoted from one-dimensional PCA to $(2D)^2$ PCA, when the basic unit is changing from pixels to both columns and rows. Then what is the best basic unit if this evolution continues? We believe that patch is the most meaningful basic unit for these linear classification methods (e.g., people is discerned by eyes and nose). The local spatial information of eyes and nose is contained in the patches. So it is more intuitive to consider the correlation of different patches. From another aspect, patch is successfully used in the field of image processing recently, not only face recognition [14–16] but also image denoising [17–19], image superresolution [20, 21], and image decomposition (cartoon-texture [22, 23] or illumination-reflectance [24] and further retinex image enhancement [25]). Patch is becoming a basic tool in these above-mentioned literatures. Motivated by our idea that the patch is the most meaningful basic unit for these linear classification methods and the widely successful application of patch, we intend to calculate the correlation of the patches in the computation of our PCA.

For the purpose of calculating the correlation of the patches, we simply add patch preprocessing before the frame work of 2DPCA. That is, we first divide the face images into patches and then we convert these patches into columns. The columns, in 2DPCA frame work, are substituted by our patch-unfold-columns, so the correlation between columns in the 2DPCA becomes the correlation between patches after calculating the image covariance (scatter) matrix. Then the orthonormal eigenvectors of the image covariance (scatter) matrix can be the optimal projection axes which are used for feature extraction. The optimal projection axes are used to form a matrix, which is called the feature matrix or feature image of the training images [12]. The test images are projected on this projection matrix and then classified by finding out the nearest neighbor of the projections of the test images. We call this method *patch-based principal component analysis* (PPCA). As a result, the main contribution of the proposed method is that the most meaningful basic unit patch is incorporated in the frame work of 2DPCA, so that the correlation between the most meaningful basic units is

utilized to promote the accuracy rate. This is confirmed our experiments. Besides, the proposed method can be easily implemented.

In fact, we can choose the support vector machine (SVM) as classifier and this may improve the accuracy rate. But SVM is not necessary for the comparison among our method and eigenface method, 2DPCA, and $(2D)^2$ PCA. In another aspect, we know that PCA is one of global techniques [26], so that it is difficult to utilize both the local spatial correlation between pixels in each patch and the nonlocal spatial correlation between patches as [17]. But we consider that the global computation could somehow compensate the utilization of the nonlocal spatial correlation between patches.

It is noteworthy that there has been great progress of face recognition nowadays. It is very hard for an improved version of an old method to challenge the recent deep learning [27, 28] based methods. Please refer to [29] for a more extensive overview on face recognition. However the improvement of an old method is still meaningful, since that many old methods are being widely employed, e.g., the alternating direction method of multipliers (ADMM) [30–35] and block coordinate decent (BCD) algorithm [36]. Meanwhile, what we focus on is the improvement of the PCA-based classification method. Moreover, the experimental results in Section 3 have indeed validated that our method outperforms other PCA-based methods.

The outline of this paper is given as follows. In Section 2, we present our PPCA method for face recognition. In Section 3, experimental results are reported to demonstrate the performance of the proposed method. Finally, some conclusions are drawn in Section 4.

2. Patch-Based Principal Component Analysis

In 2DPCA, an image matrix $\mathbf{A}_{\text{image}}$ of size $n_1 \times n_2$ is directly projected on n_2 -dimensional unitary column vectors: $\mathbf{Y} = \mathbf{A}_{\text{image}}\mathbf{X}$. By maximizing the total scatter $J(\mathbf{X}) = \text{tr}\{E[(\mathbf{Y} - E\mathbf{Y})(\mathbf{Y} - E\mathbf{Y})^T]\} = \text{tr}\{\mathbf{X}^T E[(\mathbf{A} - E\mathbf{A})^T(\mathbf{A} - E\mathbf{A})]\mathbf{X}\}$, we obtain the projection matrix. Then the following steps are feature extraction and classification. Our PPCA just adds a patch preprocessing prior to this frame work above. Then, same as 2DPCA, we calculate the image covariance matrix and optimal projection axes for feature extraction and classification.

2.1. Patch Preprocessor. Suppose that we have m training facial images. For the j -th training sample, we divide the image of size $n_1 \times n_2$ into N patches of size $p \times q$ ($1 \leq p \leq n_1$, $1 \leq q \leq n_2$). If n_1 (or n_2) is not divisible by p (or q), we would add overlap θ_1 (or θ_2), so that $(n_1 - \theta_1)/(p - \theta_1)$ (or $(n_2 - \theta_2)/(q - \theta_2)$) would always be integer no matter the choice of p (or q). Generally, for the sake of reducing computational burden, we choose the smallest one of the overlaps for each selected p (or q). Then we can get the number of patches of every face image:

$$N(p, q) = \frac{n_1 \times n_2}{p \times q}, \quad (1)$$

or with the overlap (θ_1, θ_2)

$$N(p, q, \theta_1, \theta_2) = \frac{(n_1 - \theta_1) \times (n_2 - \theta_2)}{(p - \theta_1) \times (q - \theta_2)}. \quad (2)$$

Then we convert each patch into a column vector of size $M (= p \times q)$:

$$\mathbf{a}_i = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_M \end{pmatrix}, \quad (i = 1, 2, \dots, N). \quad (3)$$

More details about the patch-to-vector conversion are given in the Section 3. Then let A_j represent all of the reshaped vectors of the j -th training facial image

$$\mathbf{A}_j = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N], \quad (4)$$

where the size of \mathbf{A}_j is $M \times N$, and $j = 1, 2, \dots, m$.

It should be noted that we adopt the 2D-PCA framework rather than $(2D)^2$ PCA. As mentioned before, $(2D)^2$ PCA takes both the correlations of columns and rows into consideration, while the 2DPCA method concentrates on the correlations between column vectors. Meanwhile our patch preprocessing convert patches into vectors. Therein, it is reasonable to adopt the 2D-PCA framework rather than $(2D)^2$ PCA.

2.2. Total Scatter. Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ be a matrix with orthonormal columns, $N > d$. Then we project matrix \mathbf{A} of size $M \times N$ onto \mathbf{X} by the following linear transformation [37, 38]:

$$\mathbf{Y} = \mathbf{A}\mathbf{X}. \quad (5)$$

\mathbf{Y} is an M -dimensional projected vector (i.e., the projected feature vector [12]) of matrix \mathbf{A} . Same as 2DPCA, we use the total scatter of the projected samples to measure the discriminatory power of the projection matrix \mathbf{X} :

$$\begin{aligned} J(\mathbf{X}) &= \text{tr} \left\{ E \left[(\mathbf{Y} - E\mathbf{Y})(\mathbf{Y} - E\mathbf{Y})^T \right] \right\} \\ &= \text{tr} \left\{ E \left[(\mathbf{A}\mathbf{X} - E(\mathbf{A}\mathbf{X}))(\mathbf{A}\mathbf{X} - E(\mathbf{A}\mathbf{X}))^T \right] \right\} \quad (6) \\ &= \text{tr} \left\{ \mathbf{X}^T E \left[(\mathbf{A} - E\mathbf{A})^T (\mathbf{A} - E\mathbf{A}) \right] \mathbf{X} \right\}. \end{aligned}$$

Let us define

$$\mathbf{G} := E \left[(\mathbf{A} - E\mathbf{A})^T (\mathbf{A} - E\mathbf{A}) \right], \quad (7)$$

which is called the image covariance (scatter matrix). The average matrix of all the L preprocessed images is

$$\bar{\mathbf{A}} = \frac{1}{L} \sum_{j=1}^L \mathbf{A}_j. \quad (8)$$

Then \mathbf{G} can be evaluated by

$$\mathbf{G} = \frac{1}{L} \sum_{j=1}^L (\mathbf{A}_j - \bar{\mathbf{A}})^T (\mathbf{A}_j - \bar{\mathbf{A}}). \quad (9)$$

It is easy to verify that \mathbf{G} is a semipositive matrix. We can evaluate \mathbf{G} directly using the L training samples. The total scatter of the projected samples can be expressed by

$$J(\mathbf{X}) = \mathbf{X}^T \mathbf{G} \mathbf{X}, \quad (10)$$

where \mathbf{X} is a unitary column vector. This is called *generalized total scatter criterion* [12]. The unitary vector \mathbf{X} is called the optimal projection axis that maximizes the criterion.

2.3. Optimization. It has been proved that the optimal projection axis \mathbf{X}_{opt} , which maximizes the total scatter of the projected samples, is the eigenvectors of \mathbf{G} corresponding to the largest eigenvalues [38]. In general, we choose the orthonormal eigenvectors $\mathbf{X}_1, \dots, \mathbf{X}_d$ of \mathbf{G} corresponding to the first d largest eigenvalues. They are equivalent to

$$\begin{aligned} \mathbf{X}_1, \dots, \mathbf{X}_d &= \arg \max_{\mathbf{X}} J(\mathbf{X}), \\ \mathbf{X}_i^T \mathbf{X}_j &= 0, \\ i &\neq j, \quad i, j = 1, \dots, d. \end{aligned} \quad (11)$$

The first eigenvector is required to have the largest possible variance (i.e., this component will “explain” or “extract” the largest part of the pattern information of the preprocessed face images [1]). We can simply control the value of d by a threshold θ as follows [8]:

$$\frac{\sum_{i=1}^d \lambda_i}{\sum_{i=1}^N \lambda_i} \geq \theta, \quad (12)$$

where λ_i ($i = 1, 2, \dots, N$) are the first N largest eigenvalues. We can determine d by presetting θ or even referring to the results from different face database.

2.4. Feature Extraction and Classification. For each *patch-preprocessed* facial image in training set \mathbf{A}_j , let

$$\mathbf{Y}_j = \mathbf{A}_j \mathbf{X}, \quad (13)$$

where $\mathbf{X} = [\mathbf{X}_1, \dots, \mathbf{X}_d]$ of size $N \times d$ is the projection matrix. We call $\mathbf{Y}_j = [\mathbf{Y}_j^{(1)}, \dots, \mathbf{Y}_j^{(d)}]$ of size $M \times d$ the *patch-based feature matrix* and $\mathbf{Y}_j^{(i)}$ ($i = 1, \dots, d$) the *patch-based principal components (vectors)* of the j -th sample image.

After patch preprocessing and 2DPCA projection, facial images in the training set have been transformed into the *patch-based feature matrixes*. We use the nearest neighbor (NN) classifier [39] for classification. We define the distance between two arbitrary patch-based feature matrixes by

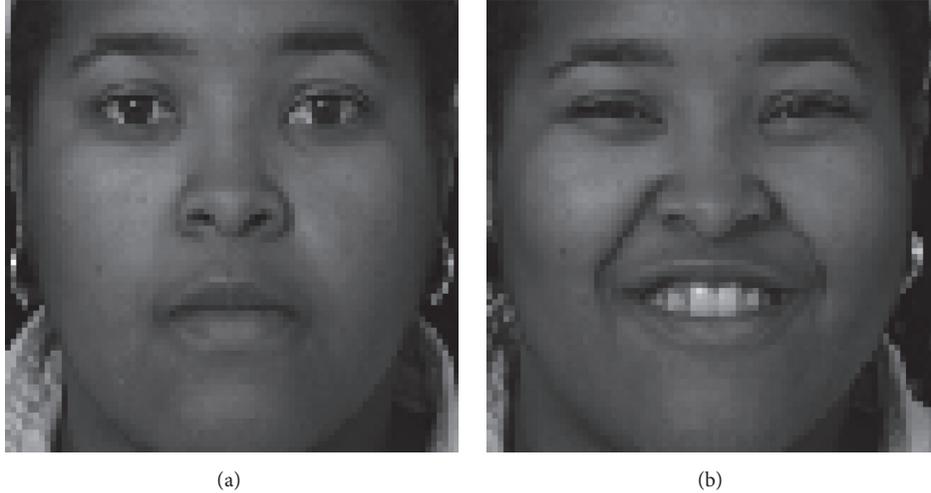
$$d(\mathbf{Y}_i, \mathbf{Y}_j) = \sum_{k=1}^d \left\| \mathbf{Y}_i^{(k)} - \mathbf{Y}_j^{(k)} \right\|_2, \quad (14)$$

where $\|\cdot\|_2$ denotes the Euclidean distance.

We have L training facial images, each of which is assigned a given identity. Given a test facial image, we first

TABLE 1: Comparison of the four methods on recognition accuracy on the partial FERET database.

Method	Accuracy (%)	Patch size (pixel)	θ_1	θ_2	Time (s)	Threshold θ
1DPCA	79.0	—	—	—	1.454	0.99
2DPCA	84.0	—	—	—	1.734	0.80
(2D) ² PCA	83.0	—	—	—	2.104	0.90
PPCA	86	20 × 19	0	18	26.658	0.92

FIGURE 1: Two different images of one individual in the partial FERET database. (a) belongs to the subset **fa**, while (b) belongs to the subset **fb**.

do a patch preprocessing and obtain a preprocessed matrix \mathbf{A}_{test} . Then we project \mathbf{A}_{test} onto \mathbf{X} and obtain \mathbf{Y}_{test} . If

$$d(\mathbf{Y}_{\text{test}}, \mathbf{Y}_l) = \min_j d(\mathbf{Y}_{\text{test}}, \mathbf{Y}_j) \leq \omega, \quad (15)$$

where ω is a preset thresholding, the test image results to the same kind of \mathbf{Y}_l , that is, the test facial image and the l -th training image, belongs to the same person. Otherwise, if $\min_j d(\mathbf{Y}_{\text{test}}, \mathbf{Y}_j) \geq \omega$, the test sample does not belong to any identity in this training data.

3. Experimental Results

In this section, the performance among our proposed PPCA and the eigenface method (or say the 1DPCA method), the 2DPCA method, and the (2D)²PCA method is evaluated on two well-known face image databases (ORL and FERET). To our point of view, experiments on constrained face databases are sufficient to validate the superiority of the proposed method among these methods. Thus, unconstrained face databases, for example, LFW database, are not taken into consideration.

First, the recognition accuracies of these four methods are compared with the experimental strategy that use half of the images in the database for training. After that, more experimental results show the influence from reordering and the size of patches. All experiments are performed using Matlab (R2014a) on a desktop with 3.40 GHz Intel core i7-2600 CPU and 12 GB RAM equipped with Windows 7 OS. If not specified, the preset threshold θ , which controls the

number of projection vectors, is set to 0.90 in the latter experiments. That is, we extract 90 percentage energy of the whole training images.

3.1. Recognition Accuracy Results on the FERET Database. The FERET database [40, 41] is a standard dataset used for facial recognition system evaluation. The Face Recognition Technology (FERET) program is managed by the Defense Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology (NIST). Until 2003, there are 2,413 facial images representing 856 individuals in the FERET database. The performance of the above 4 methods are tested on the partial FERET face database, which contains 400 images (with the cropped size 80×80) from 200 individuals, each providing 2 different images. The so-called **fa** subset, which contains 100 images, is used as training data, while the so-called **fb** subset, containing remaining 100 images, is used as testing data. Figure 1 shows 2 images of one individual in the ORL database.

From Table 1, we can see that the PPCA method achieves the highest accuracy on the FERET database. To get the highest accuracy, parameter θ is set referring to the results. The recognition accuracy is improved from 84.0 percentage of 2DPCA and 83.0 percentage of (2D)²PCA to 86 percentage. It means that the PPCA method recognized 2 more images than 2DPCA and 3 more images than (2D)²PCA on the FERET database. We remarked here that images of cropped size 60×60 were used in [8] and 83%, 84.5%, and 85% accuracy rates were got, respectively, by 1DPCA, 2DPCA, and (2D)²PCA.

TABLE 2: Comparisons of the four methods on recognition accuracy on the ORL database.

Method	Accuracy (%)	Patch size (pixel)	θ_1	θ_2	Time (s)
1DPCA	88.0	—	—	—	2.673
2DPCA	90.5	—	—	—	5.026
(2D) ² PCA	90.5	—	—	—	4.044
PPCA	91.0	2 × 24	0	7	7.186
		4 × 17	0	2	6.677
		31 × 23	4	0	8.617
		24 × 11	2	2	7.475



FIGURE 2: 10 different images of one individual in the ORL database. First 5 images are used as training data while the other 5 images are used as testing data.

3.2. Recognition Accuracy Results on the ORL Database.

The ORL database contains images from 40 individuals, each providing 10 different images with the size 112×92 (<http://rduin.nl/prhtml/prdatafiles/orl.html>). Figure 2 gives 10 images of 1 individual in the ORL database. As previously mentioned, the first five images of each individual are used as training data, and the remaining five images are used as testing data.

Table 2 gives the results on the comparisons of the four methods on recognition accuracy. Both 2DPCA and (2D)²PCA reach 90.5% accuracy, which is higher than eigenface method. Our method achieves the highest accuracy on this database. The recognition accuracy is improved from 90.5 percentage to 91.0 percentage with four different sizes of the patch. That is, the PPCA method could recognize 1 more face image than 2DPCA and (2D)²PCA on the ORL database. The CPU time of the PPCA method is not desirable but less serious in its consequences.

3.3. Influence of Reordering the Patch. The patch-to-vector conversion has a significant impact on the performance of our method. Our initial patch preprocessor converts a patch into a column vector by directly concatenating small columns

in the patch. This indeed increased the recognition accuracy that our method achieves 91.0% recognition accuracy with four different sizes of patch. However this improvement does not satisfy us. Employing the idea of clustering, we convert a patch into a column vector by reordering pixels by values for the sake of placing the approximative values together. The concatenating strategy is compared with reordering strategy by contrasting the results of recognition accuracy and CPU time on five different sizes of patch in Table 3.

Table 3 shows that the reordering strategy achieves better performances on recognition accuracy than concatenating strategy. Although reordering strategy implies an additional step of ranking the values in order, its CPU time is not always more than concatenating strategy. We further analyze the eigenvalues of the image covariance (scatter matrix) \mathbf{G} , which is defined in (9). The patch size 24×20 is selected and the comparison is conducted on the ORL database. Here, the size of image covariance matrix \mathbf{G} was 25×25 , so it was very easy to calculate its eigenvalues. In Figure 3, the magnitude (eigenvalues divided by the sum of eigenvalues) of the eigenvalues by these two strategy is plotted in decreasing order.

TABLE 3: Comparison of two strategies on recognition accuracy and CPU time on the ORL database.

Patch size	θ_1	θ_2	Concatenating		Reordering	
			Accuracy (%)	Time (s)	Accuracy (%)	Time (s)
24×11	2	2	91.0	8.093	92.0	8.523
24×20	2	2	90.5	8.356	93.0	7.924
24×23	2	0	90.0	7.702	93.5	9.060
38×6	1	4	90.0	13.184	93.0	10.873
38×7	1	2	90.0	7.053	93.0	8.335

TABLE 4: Comparison of different patch sizes on recognition accuracy and CPU time on the ORL database and the FERET database.

Patch size	ORL database				FERET database				
	θ_1	θ_2	Accuracy (%)	Time (s)	Patch size	θ_1	θ_2	Accuracy (%)	Time (s)
2×2	0	0	87.0	21.553	2×2	0	0	75.5	9.175
28×8	0	1	87.5	5.584	2×19	0	18	85.0	36.974
24×11	2	2	91.0	7.475	10×9	0	8	82.0	12.490
22×23	4	0	89.0	7.835	10×18	0	16	84.0	13.793
24×23	2	0	90.0	8.356	16×13	0	12	83.0	17.326
31×20	4	2	90.5	8.832	20×19	0	18	86.0	30.658
40×40	4	14	86.5	10.076	40×40	0	18	79.0	6.855

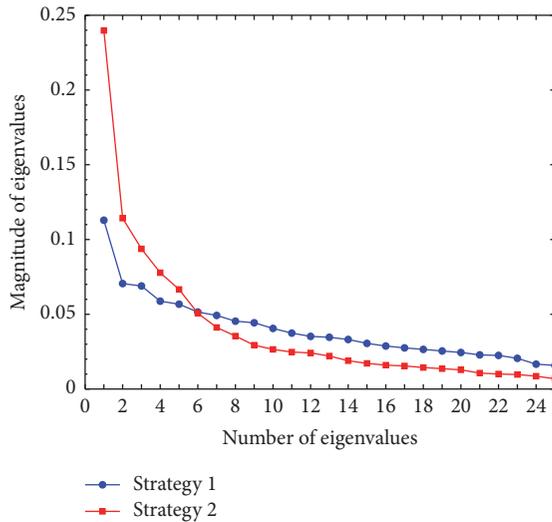


FIGURE 3: The plot of the magnitude of the eigenvalues of in decreasing order. Strategy 1 represents the “concatenating” strategy, while strategy 2 represents “reordering” strategy.

As depicted in Figure 3, the magnitude of the eigenvalues with strategy 2 decreases faster than that with strategy 1. That is, the first small number of eigenvalues by reordering strategy is larger than the same number of eigenvalues by concatenating strategy. This implies that the energy of a patch-preprocessed facial image is concentrated on its first small number of component vectors. Therefore, it is reasonable to use these component vectors for recognition purposes [12]. In addition, the more concentrated the energy on the first small number of eigenvalues is, the smaller the value of d in (11) would be. The smaller value of d brings less computation complexity and less CPU time,

which is exactly consistent with the results of CPU time in Table 3.

We remarked here that though reordering strategy achieves higher accuracy on the ORL database, we have to admit that it may not be stable. Reordering strategy does not achieve higher accuracy than concatenating strategy on the FERET database. This potential instability may be attributed to the patch-to-vector procedure, which might more or less lose the structural information. Hence, in our further work, we will attempt to find better ways to preserve more local spatial structural information rather than better strategies for patch-to-vector conversion.

3.4. Influence of the Patch Size. The PPCA method can somehow be considered as a generalization of 2DPCA method. 2DPCA method is a particular case of the PPCA method when the patch size is $n_1 \times 1$. When the size of patch is 1×1 or $n_1 \times n_2$, the PPCA method resembles one-dimensional PCA. Table 4 illustrates that the choice of patch size affects the performance of our method both on the recognition accuracy and on CPU time. A bad choice of patch size might generate a negative result. We would better not choose patches with too large or too small sizes. Therefore, it is important but not easy to choose a patch size with high recognition accuracy. Besides, the computation complexity is so large when the patches are highly overlapped (e.g., patch size 20×19 with overlaps 0 and 18). That is, our method would take much more time than 1DPCA, 2DPCA, and $(2D)^2$ PCA, if the patches are highly overlapped. Hence, it is better to choose patches of moderate sizes with small overlaps.

With the further analysis of the results, we find that there is a difference among the results who are identified of different patches. Our early experiment shows that 2DPCA and $(2D)^2$ PCA both get the same results of identified people. We can conclude that their capability of identification is

TABLE 5: Differences (with respect to 2DPCA) in the identifying results with different size of patches and “concatenating” strategy.

Patch size (pixel)	Accuracy (%)	θ_1	θ_2	Number of more identified images	Number of images failed to be identified
2×24	91.0	0	7	42, 133	195
4×17	91.0	0	2	200	NA
31×23	91.0	4	0	60, 133	198
24×11	91.0	2	2	42, 200	152

TABLE 6: Differences (with respect to 2DPCA) in the identifying results with different size of patches and “sorting” strategy.

Patch size (pixel)	Accuracy (%)	θ_1	θ_2	Number of more identified images	Number of images failed to be identified
24×11	92.0	2	2	25, 50, 53, 98, 176, 180, 196, 200	49, 83, 84, 114, 160
24×20	93.0	2	2	25, 50, 53, 98, 176, 180, 196, 200	83, 84, 160
24×23	93.5	2	0	25, 42, 50, 53, 98, 176, 180, 196, 200	83, 84, 160
38×6	93.0	1	4	25, 50, 53, 69, 98, 176, 180, 196, 200	49, 83, 84, 160
38×7	93.0	1	2	25, 42, 50, 53, 69, 98, 176, 180, 196, 200	49, 83, 84, 134, 160

same. Thus, the results of our method are compared with results of 2DPCA. The results, with respect to 2DPCA, on the ORL database from two strategies are shown, respectively, in Tables 5 and 6. The item named “number of more identified images” refers to the number of facial images in the testing set which our method correctly recognizes but 2DPCA fails to identify. The item named “number of images failed to be identified” refers to the number of facial images in the testing set which our method does not identify but 2DPCA recognizes.

From Table 4, we can find that the 195th face image in the test set is not recognized by our method of patch size 2×24 , while the 42nd and 133rd are recognized in contrast with results given by 2DPCA and $(2D)^2PCA$. And the 200th face image in the test set is recognized with the patch size of 4×17 . The 69th and 133rd face images are recognized and the 198th failed to be identified with the patch size of 31×23 . The 42nd and 200th face images are recognized and the 152nd is not identified with the patch size of 24×11 . Table 5 gives the comparison of our method, with different size of patches and “sorting” strategy. For the sake of simplicity, the details would no longer be listed.

We can see from Table 4 that different sizes of patches bring different identifying results though they achieve the same accuracy. As observed from Table 5, the PPCA method with similar size of patches performs approximately the same. For instance, our method with patch sizes 38×6 and 38×6 recognizes the 69th image in the ORL database, while patch sizes 24×11 , 24×11 , and 24×11 do not contribute to the recognition of the 69th image.

These differences in the results between our PPCA and 2DPCA (and $(2D)^2PCA$), differences among our method from different size of patches, and similarities in results from similar size of patches reveal that the capability of extracting different features would differ on account of the choice of patch size. This indeed validates our belief that “*patch is the meaningful basic unit for classification (e.g., people is discerned by eye and nose)*,” with being aware that eye or nose and so forth are of different sizes.

4. Conclusions

We have presented a patch-based PCA method to deal with face recognition. By simply doing a patch preprocessing, before the computation of projection matrix of 2DPCA, we can directly calculate the correlation of the patches instead of the rows or columns of face images. Comparisons of recognition accuracy are made with the 1DPCA [7], 2DPCA [12], and $(2D)^2PCA$ [8] methods on the ORL face database and the FERET database. Numerical experiments are represented to illustrate that the use of patch promotes the accuracy compared to former 1DPCA, 2DPCA, and $(2D)^2PCA$. Meanwhile, the results demonstrate our belief that patch is the most meaningful basic unit for classification.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by 973 Program (2013CB329404), the National Science Foundation of China (61370147, 61402082), and the Fundamental Research Funds for the Central Universities (ZYGX2016J132). Portions of the research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office. The authors would like to express their great gratitude to AT&T Laboratories Cambridge for using the face images in the ORL database.

References

- [1] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.
- [2] I. T. Jolliffe, *Principal Component Analysis*, Wiley Online Library, New York, NY, USA, 2002.

- [3] K. Kim, "Face recognition using principle component analysis," in *Proceedings of the In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–591, 1996.
- [4] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.
- [5] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.
- [6] I. Grattan-Guinness, "The search for mathematical roots, 1870–1940," in *Logics, Set Theories and The Foundations of Mathematics from Cantor through Russell to Gode*, Princeton University Press, Princeton, NJ, USA, 2011.
- [7] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 586–591, IEEE, Maui, HI, USA, 1991.
- [8] D. Zhang and Z. Zhou, "(2D)2 PCA: two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, no. 1–3, pp. 224–231, 2005.
- [9] J. Edward Jackson, *A Users Guide to Principal Components*, John Wiley & Sons, Hoboken, NJ, USA, 2005.
- [10] G. Saporta and N. Niang, "Principal component analysis: application to statistical process control," *Data Analysis*, pp. 1–23, 1996.
- [11] M. A. Turk and A. P. Pentland, "Recognition in face space," in *Proceedings of the Intelligent Robots and Computer Vision IX: Algorithm and Techniques*, pp. 43–54, SPIE, Boston, Mass, USA, 1990.
- [12] J. Yang, D. Zhang, A. F. Frangi et al., "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.
- [13] Y. Sun, X. Tao, Y. Li, and J. Lu, "Robust 2D principal component analysis: a structured sparsity regularized approach," *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2515–2526, 2015.
- [14] C. Hu, J. Harguess, and J. K. Aggarwal, "Patch-based face recognition from video," in *Proceedings of the International Conference on Image Processing, ICIP '09*, pp. 3321–3324, IEEE, Cairo, Egypt, November 2009.
- [15] C. Jung, L. Jiao, B. Liu, and M. Gong, "Position-patch based face hallucination using convex optimization," *IEEE Signal Processing Letters*, vol. 18, no. 6, pp. 367–370, 2011.
- [16] Y. Wong, S. Chen, S. Mau, C. Sanderson, and B. C. Lovell, "Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition," in *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2011*, IEEE, Colorado Springs, CO, USA, June 2011.
- [17] K. Dabov, A. Foi, V. Katkovich, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [18] Y. Chen, T.-Z. Huang, L.-J. Deng, X.-L. Zhao, and M. Wang, "Group sparsity based regularization model for remote sensing image stripe noise removal," *Neurocomputing*, 2017.
- [19] J.-J. Mei, Y. Dong, T.-Z. Huang, and W. Yin, "Cauchy noise removal by nonconvex admm with convergence guarantees," *Journal of Scientific Computing*, 2017.
- [20] L.-J. Deng, W. Guo, and T.-Z. Huang, "Single image super-resolution by approximated Heaviside functions," *Information Sciences*, vol. 348, pp. 107–123, 2016.
- [21] L.-J. Deng, W. Guo, and T.-Z. Huang, "Single image super-resolution via an iterative reproducing kernel Hilbert space method," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, 2015.
- [22] H. Schaeffer and S. Osher, "A low patch-rank interpretation of texture," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 226–262, 2013.
- [23] Y. Fan, T. Huang, T. Ma, and X. Zhao, "Cartoon-texture image decomposition via non-convex low-rank texture regularization," *Journal of the Franklin Institute*, vol. 354, no. 7, pp. 3170–3187, 2017.
- [24] W. Wang and M. K. Ng, "A nonlocal total variation model for image decomposition: illumination and reflectance," *Numerical Mathematics. Theory, Methods and Applications*, vol. 7, no. 3, pp. 334–355, 2014.
- [25] H. Chang, M. K. Ng, W. Wang, and T. Zeng, "Retinex image enhancement via a learned dictionary," *Optical Engineering*, vol. 54, no. 1, Article ID 140914, 2015.
- [26] L. J. P. van der Maaten, E. O. Postma, and J. H. van den Herik, "Dimensionality reduction: a comparative review," *Journal of Machine Learning Research*, vol. 10, no. 40, pp. 66–71, 2009.
- [27] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: closing the gap to human-level performance in face verification," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1701–1708, IEEE, Columbus, Ohio, USA, June 2014.
- [28] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: a unified embedding for face recognition and clustering," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 815–823, IEEE, Boston, Mass, USA, June 2015.
- [29] M. P. Beham and S. M. M. Roomi, "A review of face recognition methods," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 27, no. 4, Article ID 1356005, 2013.
- [30] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.
- [31] X.-L. Zhao, F. Wang, T.-Z. Huang, M. K. Ng, and R. J. Plemmons, "Deblurring and sparse unmixing for hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 7, pp. 4045–4058, 2013.
- [32] J. Liu, T.-Z. Huang, I. W. Selesnick, X.-G. Lv, and P.-Y. Chen, "Image restoration using total variation with overlapping group sparsity," *Information Sciences*, vol. 295, pp. 232–246, 2015.
- [33] T.-Y. Ji, T.-Z. Huang, X.-L. Zhao, T.-H. Ma, and L.-J. Deng, "A non-convex tensor rank approximation for tensor completion," *Applied Mathematical Modelling*, vol. 48, pp. 410–422, 2017.
- [34] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR '17)*, IEEE, Honolulu, Hawaii, USA, 2017.
- [35] T.-H. Ma, T.-Z. Huang, X.-L. Zhao, and Y. Lou, "Image deblurring with an inaccurate blur kernel using a group-based low-rank image prior," *Information Sciences*, vol. 408, pp. 213–233, 2017.
- [36] T.-Y. Ji, T.-Z. Huang, X.-L. Zhao, T.-H. Ma, and G. Liu, "Tensor completion using total variation and low-rank matrix factorization," *Information Sciences*, vol. 326, pp. 243–257, 2016.

- [37] K. Liu, Y.-Q. Cheng, and J.-Y. Yang, "Algebraic feature extraction for image recognition based on an optimal discriminant criterion," *Pattern Recognition*, vol. 26, no. 6, pp. 903–911, 1993.
- [38] J. Yang and J.-Y. Yang, "From image vector to matrix: a straightforward image projection technique-IMPCA vs. PCA," *Pattern Recognition*, vol. 35, no. 9, pp. 1997–1999, 2002.
- [39] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, New York, NY, USA, 2012.
- [40] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.
- [41] P. Jonathon Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.