



Deterministic multikernel extreme learning machine with fuzzy feature extraction for pattern classification

Bhawna Ahuja¹ · Virendra P. Vishwakarma¹

Received: 20 August 2020 / Revised: 3 January 2021 / Accepted: 21 May 2021 /
Published online: 30 July 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

In this paper a novel multikernel deterministic extreme learning machine (ELM) and its variants are developed for classification of non-linear problems. Over a decade ELM is proved to be efficacious learning algorithms, but due to the non-deterministic and single kernel dependent feature mapping proprietary, it cannot be efficiently applied to real time classification problems that require invariant output solution. We address this problem by analytically calculation of input and hidden layer parameters for achieving the deterministic solution and exploiting the data fusion proficiency of multiple kernel learning. This investigation originates a novel deterministic ELM with single layer architecture in which kernel function is aggregation of linear combination of disparate base kernels. The weight of kernels depends upon perspicacity of problem and is empirically calculated. To further enhance the performance we utilize the capabilities of fuzzy set to find the pixel-wise coalition of face images with different classes. This handles the uncertainty involved in face recognition under varying environment condition. The pixel-wise membership value extracts the unseen information from images up to significant extent. The validity of the proposed approach is tested extensively on diverse set of face databases: databases with and without illumination variations and discrete types of kernels. The proposed algorithms achieve 100% recognition rate for Yale database, when seven and eight images per identity are considered for training. Also, the superior recognition rate is achieved for AT & T, Georgia Tech and AR databases, when compared with contemporary methods that prove the efficacy of proposed approaches in uncontrolled conditions significantly.

Keywords Pattern classification · ELM · KELM · MKL · Deterministic learning · Fuzzy feature extraction

✉ Virendra P. Vishwakarma
virendravishwa@rediffmail.com; vpv@ipu.ac.in

Bhawna Ahuja
Bhawna17.ahuja@gmail.com

¹ University School of Information, Communication & Technology, Guru Gobind Singh Indraprastha University, Sector-16C, Dwarka, New Delhi 110078, India

1 Introduction

Face recognition is nonintrusive and natural technique among other biometric modalities like iris and fingerprint. It is a visual pattern classification problem. Due to automation of face recognition in numerous applications including surveillance, biometric authentication and human computer interaction, has motivated many researchers to design a proficient and accurate classifier. The face recognition system performance deteriorates when the database incorporates a huge number of face images collected under uncontrolled environment conditions [10]. Face recognition system imply one to many matching of query face image from the set of facial images in the associated database. The recognition rate of face recognition system relies upon various factors like pose, age span, facial expression, facial wear, and illumination conditions [61]. It is a very challenging task in non-cooperative user scenarios specifically in surveillance video. The key challenges for face recognition system are i) immense variation in face images of same subject due to viewing direction, ii) non-convexity property of face, that requires a non-linear algorithm that can map high dimensional feature space to low dimension, iii) a system is trained on small number of face images [39, 61]. To handle all these challenges the face recognition system consists of mainly two modules: feature extraction and classification. The feature extraction module selects the discriminative features from normalized facial image and the objective of classification module is to design a powerful and robust classifier as the non-linearity and non-convexity of face image cannot be completely resolved by image normalization and efficient feature extraction.

The performance of face recognition system degrades due to variations in illumination conditions, as the change in the appearance of a face image due to variation in illumination is larger than face identity changes. The illumination normalization is central problem in face recognition system and various well-known techniques were developed to tackle this problem. The standard techniques to deal with illumination normalization are gamma correction, logarithmic transform, Histogram Equalization (HE), histogram match etc. However, these algorithms are unable to give satisfactory results due to variants illumination conditions. Hafed et al. had used discrete cosine transform (DCT) to obtain the feature vector of face image. Chen employed DCT in logarithmic domain by discarding low-frequency DCT coefficients to overcome the affect of illumination variations [13]. Xie et al. had normalized both large (low-frequency components) and small scale features for improved facial image recognition and restoration [56]. Vishwakarma proposed a fuzzy filter based illumination normalization algorithm in which low-frequency DCT coefficients were modified using fuzzy polynomial membership function [53].

After performing preprocessing on facial image, features are extracted. A variety of feature descriptors are developed and analyzed in literature [2, 4, 5, 8, 12, 14, 47]. Deniz et al. had introduced the fusion of Histogram of oriented gradients (HOG) at different scales to acquire the important features for face recognition [17]. Zhou et al. had proposed an approach for face recognition in which features are extracted by the integration of principal components and Linear discriminant analysis (LDA) [62]. Fernando et al. had introduced the concept of multi-scale gray level co-occurrence matrix for feature descriptors [14]. Furthermore, fuzzy approximation theory can be applied on pixel wise feature extraction of image. There is perpetually uncertainty in real pattern classification problems. Fuzzy logic can be employed to analyze these problems as it deals with approximate reasoning rather than precise [23]. Zadeh had introduced the concept of fuzzy logic for imprecise mode of reasoning that helps to take a decision in uncertain conditions [59, 60]. kim et al. designed a automatic facial expression

recognition classifier using neurofuzzy approach [37]. Kyoung et al. had implemented the concept of fuzzified feature vectors for face recognition. Hai-jun et al. had designed an online-sequential ELM with fuzzy activation functions for classification problems and function approximation [49].

The extracted features are recognized using classifier [1, 19, 21, 36, 41, 42]. The traditional gradient based learning algorithms are not efficient for challenging real time recognition applications. The structural parameters of these algorithms are iteratively calculated that leads to slow learning rate. Beside this, traditional algorithm are affected by overfitting, underfitting, out of memory and easily convergence to local minima problems [30, 33]. To work out with the issues of classical learning approaches, the concept of non-iterative learning algorithms was introduced [20, 24, 25, 29]. Huang et al. had proposed the concept of ELM, a fast and non-iterative single hidden layer framework for classification (binary and multiclass) and regression problems [29, 31]. Based on Bartlett concept of strong generalization performance, the objective of ELM is to minimize the output weight norm along with training error [7]. Benefitting from the non-linear mapping capability, ELM has been materialized in numerous applications like object recognition, imbalance learning, signal processing, traffic sign recognition [16, 34, 35, 67].

Although ELM has achieved some accomplishments, but there is a scope for enhancement. In previous few years researchers had done a lot of research to optimize the structure of ELM. Zhu et al. used the concept of differential evolutionary algorithm for selection of input weights [63]. Fei et al. proposed improved ELM for better performance by encoding the priori information for function approximation [26]. Huang et al. [32] schemed incremental ELM, where hidden nodes were incremented randomly with complex activation function. One of its variants was introduced by Feng et al. named as error minimized ELM (EM-ELM) [20]. In this design, hidden nodes can be incremented one by one or chunk by chunk. Deng et al. introduced the concept of regularized ELM, that works on the theory of weighted least square approach for the databases containing outliers [15]. To automate the architecture of ELM, Yoan et al. proposed optimally pruned ELM, in which statistical techniques were utilized to measure the significance of hidden nodes [46]. Wang et al. introduced effective ELM, in which the proper selection of bias and input weight were done to achieve the high performance rate [54]. Zong et al. had successfully implemented ELM on face recognition application and compared it with support vector machine (SVM) [65]. Zong et al. introduced the kernel version of ELM and implemented on face recognition application [66]. Jose et al. introduced regularized ELM, in which different regularization techniques were used to prune the architecture of ELM [45]. Huang et al. proved that ELM is better than SVM and least square SVM in terms of optimization constraints and efficiency [33]. Zong et al. schemed weighted ELM to balance the input data with imbalance class distribution by assigning different weights according to user's needs [67]. Hong et al. introduced the hierarchical ELM based on hierarchical structure for sequential learning [27]. Tang et al. introduced the concept of multi-layer ELM by adding more than one hidden layer to the original ELM design [52]. Zhiyong et al. had utilized the ELM concept for traffic sign recognition using HOG features [34]. In order to avoid the manual tuning of network parameters in multi layer-ELM Wong et al. introduced the kernel version of ML-ELM, entitled as ML-KELM [55]. Chengbo et al. introduced the concept of improved weighted ELM for classification of imbalance data based on majority voting [44]. Deng et al. had used the concept of ELM autoencoder along with online sequential ELM for feature extraction and classification respectively in visual tracking system [16].

For computer vision applications the single kernel classifier is not an appropriate choice. This motivates researchers to use the multiple kernel learning with classification techniques. Multiple kernel learning (MKL) is a method to select and integrate data derived from different sources such as strings, graphs and trees for a specific recognition problem [18, 57]. It strengthens the ability of machine learning by allowing the kernels to be chosen based on data. Gonen et al. proposed and analyzed combination and computational time of different methodologies for MKL on real time data [22]. Zhuang et al. explored the integration of different kernels in multi layer architecture [64]. Yanfeng et al. utilized the concept of MKL for classification of hyperspectral images, in which kernel weights were calculated by statistical significance. Bucack et al. had implemented different approaches of MKL for object recognition and analyzed that MKL is more efficient than single kernel function [11]. Fabio et al. proposed a new approach of MKL in which the kernel combination parameters were calculated by min-max algorithm [6]. Xinwang et al. proposed the sparse, non-sparse and radius incorporated methods to find the optimized kernel combination coefficient [43]. Xiao dong et al. implemented the convex combination approach to integrate different base kernels and exercised it for classification problems [40].

The objective of present investigation is to design a novel classification engine that can extract invariant features for efficient classification of non-linear face images captured under uncontrolled constraints. The essence of this novel work is:

- The extension of original ELM is proposed with deterministic and multiple kernel learning approach. In original ELM (non-deterministic) the input parameters are randomly assigned and hidden layer neurons are experimentally calculated. In deterministic ELM the input and hidden layer parameters are analytically evaluated, which results in steady and invariant output.
- To make it powerful and robust for real world heterogeneous databases and to discriminate between intra and inter class variance, fuzzification is envisioned for invariant feature extraction. Further the input space is mapped to optimized kernel feature space, which is integration of multiple base kernels.
- To handle the real time challenges (lightning variations) of face recognition system some pre-processing techniques are needed to be employed. As the face image is composed of low and high frequency components and illumination variations affect the low-frequency components. Therefore to surpass the effect of lightning conditions, low frequency components needed to be modified. In the proposed method we have used the fuzzy filter based normalization algorithm that modifies the low-frequency components using polynomial membership function.

Extensive set of experiments have been executed on face databases to analyze the performance of proposed approaches with contemporary techniques including OMKELM, KELM and ELM. The experiential results acknowledge the supremacy of proposed approaches with aforesaid techniques.

This paper is framed as follows. The concept of fuzzy logic, Extreme learning machine and multiple kernel learning are briefed in Section 2. The proposed work followed by analysis is presented in Section 3. In Section 4 empirical evaluation is performed on comprehensive set of databases and section 5 summarize our conclusions.

2 Preliminaries

2.1 Fuzzy logic

In the crisp set an individual is either a member or non-member of a given set Q . The characteristic function of crisp set Q is given as:

$$\mu_Q = \begin{cases} 1, x \in Q \\ 0, x \notin Q \end{cases} \tag{1}$$

Fuzzy logic is based on the principle of imprecise and approximate mode of reasoning, that helps in taking decision in an environment of imprecision and uncertainty [59]. Each element in fuzzy set is assigned a value that denotes its membership grade in fuzzy set. This grade is analogous to the degree with which an individual is compatible or similar with the concept illustrated by fuzzy set [38]. An individual with larger membership grade belongs to the fuzzy set with greater degree. These grades are characterized by real number in $[0, 1]$. The individual with a 1 value of membership grade indicates full membership and 0 value is for full-non membership. Crisp set is considered to be a restricted case of fuzzy set with only two values of membership grade i.e. 0 and 1. The membership grade of an individual in fuzzy set is determined by the membership function (MF). The MF maps individuals of a given set E (universal set) into real values in $[0, 1]$. The Membership grade μ_A for fuzzy set A is represented by

$$\mu_A : E \rightarrow [0, 1] \tag{2}$$

2.2 Pre-processing technique: Illumination normalization

The objective of normalization algorithm should be to reduce or nullify the effect of illumination variation on images, without changing the details of image. In this investigation fuzzy filter [53] is used for normalization of images. The key points behind this approach are i) the illumination variations remarkably affect low frequency (LF) elements and ii) the facial features which can segregate two facial images are in ascending order of effectiveness in frequency domain. Consider a face image of size $U \times V$, where U determines number of rows and V represents the number of columns. Perform contrast stretching by employing Adaptive histogram equalization (AHE) followed by logarithmic transform (LOG) written as AHE + LOG. After contrast enhancement, image is converted from spatial to frequency domain by applying DCT. The DCT for an image of dimension $U \times V$ is given as follows:

$$D(m, n) = \Omega(m)\Omega(n) \sum_{xc=0}^{U-1} \sum_{yc=0}^{V-1} f(xc, yc) \times \cos\left[\frac{\pi(2xc + 1)m}{2U}\right] \times \cos\left[\frac{\pi(2yc + 1)n}{2V}\right] \tag{3}$$

For $m = 0, 1, 2, \dots, U - 1$ and $n = 0, 1, 2, \dots, V - 1$.

The inverse DCT transform is given by

$$f(xc, yc) = \sum_{m=0}^{U-1} \sum_{n=0}^{V-1} \Omega(m)\Omega(n)D(m, n) \times \cos\left[\frac{\pi(2xc + 1)m}{2U}\right] \times \cos\left[\frac{\pi(2yc + 1)n}{2V}\right] \tag{4}$$

For $xc = 0, 1, 2, \dots, U - 1$ and $yc = 0, 1, 2, \dots, V - 1$.

where

$$\Omega(m) = \left\{ \begin{array}{l} \frac{1}{\sqrt{U}}, m = 0 \\ \sqrt{\frac{2}{U}}, m = 1, 2, \dots, U-1 \end{array} \right\} \tag{5}$$

$$\Omega(n) = \left\{ \begin{array}{l} \frac{1}{\sqrt{V}}, n = 0 \\ \sqrt{\frac{2}{V}}, n = 1, 2, \dots, V-1 \end{array} \right\}$$

Obtain the AC coefficients of image (representing the change in intensity values of pixels) in the ascending order of frequency. Determine the DCT coefficients for complete face image using Eq. (3).

$$D(c)_{c=0\dots(UV-1)} = D(m, n)_{n=0\dots(V-1), m=0\dots(U-1)} \tag{6}$$

The initial coefficients are acknowledged as low-frequency DCT (LF-DCT) coefficients. Consider a universal set c depicting the indices of $LF-DCT$ coefficients. To depreciate the effect of illumination, $LF-DCT$ coefficients are modified by fuzzy polynomial MF with increasing order of degree of membership of $LF-DCT$ indices. The polynomial MF is written as follows:

$$\mu_G = \frac{c^\omega}{C^\omega} \tag{7}$$

where C is index of largest $LF-DCT$ and ω is constant which is experimentally obtained. Let us consider LFN number of $LF-DCT$ coefficients. Obtain the fuzzy membership grade for LFN coefficients using Eq. (7). These fuzzy membership grades can be expressed in the form of vector as

$$G = [G_0, G_1, \dots, G_q, \dots, G_{LFN}] \tag{8}$$

The modified DCT coefficients of image can be obtained using:

$$MD(c)_{c=0\dots(UV-1)} = D(c)_{c=0\dots LFN} \times G + D(c)_{c=LFN+1\dots(UV-1)} \tag{9}$$

The fuzzy filter algorithm for illumination normalization is outlined as follows:

Algorithm 1: Fuzzy Filter algorithm for illumination Normalization.

Input: An input face image of dimension $U \times V$.

Output: Normalized image of dimension $U \times V$.

1. Perform the contrast stretching on input face image by performing AHE + LOG.
 2. Obtain the DCT coefficients $D(c)$ of complete input image by (6).
 3. Modify the LFN number of $LF-DCT$ coefficients using polynomial MF.
 4. Obtain the modified DCT coefficients of complete input image by (9).
 5. Perform the inverse DCT using Equation (4) to obtain the normalized image.
-

2.3 Extreme learning machine

ELM is non-deterministic learning approach with feedforward architecture, initially designed for single hidden layer [31, 33]. ELM foundation is based on the concept of non-iteratively calculation of input and hidden layer parameters. In ELM for accomplishing the higher accuracy rate the hidden layer parameters are experimentally tuned and input parameters are randomly projected. The output function of ELM is given as:

$$f_o = \sum_{j=1}^{\eta} \theta_j b_j(x_i) = \sum_{j=1}^{\eta} \theta_j b(iw_j \cdot x_i + a_j) = b(x)\theta = Y, i = 1, \dots, N \tag{10}$$

Where iw_j is the input weight vector connecting the i^{th} input neuron to j^{th} hidden node, a_j is learning parameter of hidden layer known as bias, θ_j is the weight vector related to output layer, connecting j^{th} hidden neuron to output nodes and f_o is the output of ELM. η and N denotes the count of hidden nodes and training samples respectively. Equation (10) can be compactly written as:

$$B\theta = Y \tag{11}$$

B is the output matrix of hidden layer that can be defined in the matrix form as

$$B = \begin{pmatrix} b(x_1) \\ \vdots \\ b(x_N) \end{pmatrix} \tag{12}$$

$$= \begin{pmatrix} b(iw_{1,1}, a_{1,1}, x_1) \cdots b(iw_{\eta,1}, a_{\eta,1}, x_1) \\ \vdots \\ b(iw_{1,N}, a_{1,N}, x_N) \cdots b(iw_{\eta,N}, a_{\eta,N}, x_N) \end{pmatrix}_{N \times \eta} \tag{13}$$

The q^{th} column of B is the output vector of q^{th} hidden node and q^{th} row of B is output of hidden layer analogous to input x_q . $x = [x_1, \dots, x_N]$ is the input data vector and $Y = [y_1, \dots, y_N]^T$ is target output vector for N count of input instances. $\theta = [\theta_1, \theta_2, \dots, \theta_\eta]^T$ denotes the output vector for η count of hidden neurons. Each element of θ is a vector of dimension equal to output nodes (number of classes).

The input weight matrix is defined as:

$$IW = \begin{pmatrix} iw_{11} iw_{12} \cdots iw_{1d} \\ \vdots \\ iw_{\eta,1} iw_{\eta,2} \cdots iw_{\eta,d} \end{pmatrix}_{\eta \times d} \tag{14}$$

The size of randomly assigned input weight vector for d dimensional input data is $\eta \times d$. IW is dynamic in size, as the count of hidden nodes is not static (experimentally calculated).

Equation (11) is linear system and its least-square solution is

$$\theta = B^\dagger Y \tag{15}$$

Where B^\dagger is Moore-Penrose Generalized inverse [48] of output matrix B of hidden layer. The Orthogonal projection approach is exploited to determine Moore-Penrose Generalized inverse of matrix B : $B^\dagger = B^T(BB^T)^{-1}$.

For obtaining the strong generalization performance and to determine the invariant output solution, a positive number W is added to the diagonal of BB^T [28].

$$f_o = b\theta = b(x)B^T \left(\frac{I}{W} + BB^T \right)^{-1} Y \tag{16}$$

W is regularization coefficient and $b(x)$ is ELM feature mapping, generally known to user.

Different from the iterative learning algorithms, ELM objective is to minimize the output weight along with training error. The objective of ELM for classification problems is:

$$\text{Min} : \frac{1}{2} \|\theta\|^2 + \frac{W}{2} \sum_{k=1}^N \|\delta\|^2$$

$$\text{Such that} : b(x)\theta = y_k^T - \delta_k^T \text{ for } k = 1, \dots, N \tag{17}$$

Where $\delta_k = [\delta_{k,1}, \dots, \delta_{k,h}]^T$ is training error vector of h output neurons corresponding to input data x_k .

2.4 Multiple kernel learning

In kernel based classifiers the efficiency relies on finding an appropriate kernel for representing a data. Single kernel based classifiers does not give good empirical performance to classify heterogeneous data, where each instance is represented by multiple source of data [3, 11]. To address this problem multiple kernel learning is utilized to determine optimal kernel from a set of base kernels. The objective of MKL is to combine the different sets of features by determining the optimal combination of different kernels [22, 58]. MKL is an approach to concatenate features from multiple sources and fed to a single learning algorithm. The optimal combination coefficient of different kernels is data dependent.

The MKL formulation using linear combination of set of u pre-defined kernels is given by:

$$K(x_i, x_k) = \sum_{v=1}^u \gamma_v K_v(x_i, x_k) \tag{18}$$

With $\gamma_v \& > 0$ and $\sum_v \gamma_v = 1$. $K(x_i, x_k)$ is resultant optimal kernel. $K_v(x_i, x_k)$ represents the v^{th} sub-kernel and γ_v is the weight (combination coefficient) of v^{th} kernel. Each sub-kernel uses different set of feature vectors for every instance. Depending upon the significance of features the weights are assigned to the different kernels to achieve accurate classification.

The Eq. (18) can be equivalently defined as

$$K(\cdot, \cdot; \gamma) = \sum_{v=1}^u \gamma_v K_v(\cdot, \cdot) \tag{19}$$

The choice of kernel $K(\cdot, \cdot)$ and its combination coefficient depends on perceptivity of the classification problem. In terms of feature mapping the MKL can be formulated as

$$\theta(\cdot; \gamma) = [\gamma_1 \theta_1(\cdot), \gamma_2 \theta_2(\cdot), \dots, \gamma_v \theta_v(\cdot), \dots, \gamma_u \theta_u(\cdot)] \tag{20}$$

Where $\theta(\cdot; \gamma)$ and $\theta_v(\cdot)$ are feature mapping corresponding to $K(x_i, x_k)$ and $K_v(x_i, x_k)$ respectively.

3 Proposed work

In this section, we propose novel progressive ELM based algorithms subject to multi-class pattern classification applications. The objective of proposed algorithms is to overcome the randomness of original ELM along with enhancement in classification performance by utilizing multikernel and fuzzy logic concepts. The suggested algorithms with single hidden layer feedforward neural network structure are a) Deterministic ELM, in which feature vectors are obtained from pixel intensity values and feature vector mapping is performed by employing multikernel approach b) Deterministic ELM that utilizes fuzzy MF for determination of feature vectors and multikernel technique for feature mapping. The proposed approaches are evaluated on face recognition application. The block diagram of system architecture is shown in Fig. 1. As the face images captured under real time environment contains illumination variations, the training and test images are pre-processed using illumination normalization technique. For this, fuzzy filter has been applied in DCT domain. The normalized face images are classified using proposed Deterministic ELM based classifiers.

3.1 Deterministic multikernel extreme learning machine (DMK-ELM)

The real world pattern classification application demands a fast and deterministic learning algorithm to classify complex data. The random practicing of structural parameters (weights related to input layer and biases of hidden neurons) in ELM results in varying and non-deterministic output solution [4, 65]. Different classification results are obtained for the same database by changing the count of hidden nodes. For obtaining the immutable and deterministic output solution the statistics of input and hidden layer should be calculated with the help of invariant parameters. Also, to deal with the complex data that require the representation of each instance with multiple features the single kernel feature mapping is not an optimal choice. In this paper, we design a deterministic ELM integrated with MKL approach. The proposed approach is capable to classify simple and complex data, by performing appropriate selection of kernels and optimal combination coefficients. In DMK-ELM for obtaining the deterministic output, the input and hidden layer parameters are obtained with the help of input data samples. In the proposed approach, the achieved output solution is stable as the structural parameters are

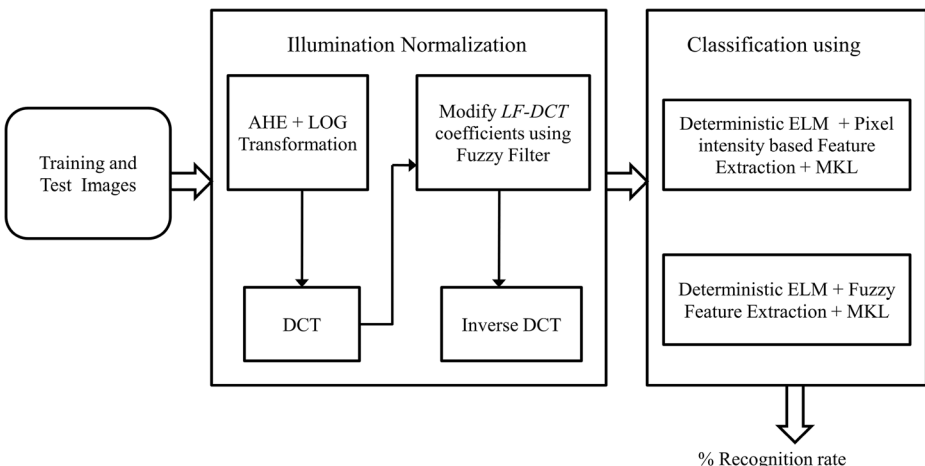


Fig. 1 Block diagram of proposed system

analytically calculated. In DMK-ELM the feature vector utilized for mapping input data to output classes is concatenation of feature vectors obtained with the help of pre defined sub-kernels.

The mathematical formulation of DMK-ELM is:

The input weight $IW = [iw_1, \dots, iw_\eta]$ can be determined with the help of input data x_i of dimension d as follows:

$$\frac{1}{m} \left[\frac{x_1}{Z_1}, \frac{x_2}{Z_2}, \dots, \frac{x_\eta}{Z_\eta} \right] \quad (21)$$

Where Z is analytically determined by evaluating the norm of input data as follows:

$$Z_i = \left[\sum_{j=1}^d (x_{ij})^2 \right]^{1/2} \quad (22)$$

And the value of constant m is determined by calculating norm of all Z_i :

$$m = \left[\sum_{i=1}^{\eta} (Z_i^2) \right]^{1/2} \quad (23)$$

As the counts of hidden neurons are equivalent to number of input samples, the Eq. (21) and (23) can be re-formulated as:

$$= \frac{1}{m} \left[\frac{x_1}{Z_1}, \frac{x_2}{Z_2}, \dots, \frac{x_N}{Z_N} \right] \quad (24)$$

$$m = \left[\sum_{i=1}^N (Z_i^2) \right]^{1/2} \quad (25)$$

For d dimensional input data, the bias of k^{th} hidden node is determined by mean of input weight vector IW of size $N \times d$.

$$a_k = \frac{1}{d} \sum_{i=1}^d iw_{ki} \quad (26)$$

The objective function of DMK-ELM for multi-class classification problem is

$$\text{Min} : \frac{1}{2} \|\theta^2\| + \frac{W}{2} \sum_{i=1}^N \|\delta_i\|^2$$

$$\text{Such that} : \theta^T b(x_i, iw_i : \gamma) = y_i - \delta_i, \forall i \sum_{v=1}^u \gamma_v = 1, \gamma_v \geq 0 \quad (27)$$

$B(:, \gamma) = [\gamma_1 B_1(\cdot), \dots, \gamma_u B_u(\cdot)]$ is the feature vector with respect to all the kernels from $1, 2, \dots, u$.

The mathematical re-formulation of Eq. (27) is

$$\text{Min} : \frac{1}{2} \|\theta^2\| + \frac{W}{2} \sum_{i=1}^N \|\delta_i\|^2$$

$$\text{such that : } \theta \sum_{v=1}^u \gamma_v b_v(x_i, iw_i) = y_i - \delta_i, \forall i \sum_{v=1}^u \gamma_v = 1, \gamma_v \geq 0 \tag{28}$$

Equation (28) is optimization problem (joint-convex) and its Lagrangian function is given by:

$$L(\theta, \delta, \gamma) = \frac{1}{2} \|\theta\|^2 + \frac{W}{2} \|\delta\|^2 - \sum_{\phi=1}^N \sum_{k=1}^h \alpha_{\phi k} \left(\theta_k \sum_{v=1}^u \gamma_v b_v(x_\phi, iw_\phi) - y_{\phi k} + \delta_{\phi k} \right) + \tau \left(\sum_{v=1}^u \gamma_v - 1 \right) \tag{29}$$

Where τ and $\alpha \in \mathcal{R}^{N \times h}$ are lagrangian multipliers. The optimal condition of Eq. (29) is mathematically defined using KKT theorem [9] as follows:

$$\frac{\partial L(\theta, \delta, \gamma)}{\partial \theta_k} = 0 \rightarrow \theta_k = \sum_{\phi=1}^N \sum_{v=1}^u \gamma_v \alpha_{\phi j} b_v(x_\phi, iw_\phi) \tag{30}$$

$$\frac{\partial L(\theta, \delta, \gamma)}{\partial \delta_{\phi k}} = 0 \rightarrow \delta_{\phi k} = \frac{\alpha_{\phi k}}{W}, \forall \phi, \forall k \tag{31}$$

$$\frac{\partial L(\theta, \delta, \gamma)}{\partial \alpha_{\phi k}} = 0 \rightarrow \theta_j \sum_{v=1}^u \gamma_v b_v(x_\phi, iw_\phi) - \tau_{\phi k} + \delta_{\phi k} = 0, \forall \phi, k \tag{32}$$

Equation (32) can be written as:

$$\left(B(iw_i, x_i; \gamma) B(iw_j, x_j; \gamma) + \frac{I}{W} \right) \alpha = Y^T \tag{33}$$

The output function of DMK-ELM is

$$f_o = B(iw_i, x_i; \gamma) B(iw_j, x_j; \gamma) \left(\frac{I}{W} + B(iw_i, x_i; \gamma) B(iw_j, x_j; \gamma) \right)^{-1} Y^T \tag{34}$$

The Deterministic multikernel ELM algorithm is outlined as follows:

Algorithm 2: Deterministic ELM with multiple kernel learning.

Input: A set of face image training samples $(x_k, y_k), k = 1, 2, \dots, N$ and specified u kernels $K_u(x_k, x_l)$

Output: Output function f_o , predicting class y .

1. Analytically obtain the input weight vector from Equation (21) by evaluating the norm and mean of input data x_i
 2. Determine the biases of hidden neurons by calculating the mean of input weight vector.
 3. Obtain the optimal kernel using $K(\cdot, \cdot; \gamma) = \sum_{v=1}^u \gamma_v K_v(\cdot, \cdot)$
 4. Evaluate the resultant deterministic output solution for DMK-ELM by practicing the feature vectors determined after step 3.
-

3.2 Deterministic extreme learning machine with fuzzy feature extraction (DMK-ELM-FFE)

In this section, DMK-ELM is further extended to DMK-ELM-FFE, by incorporating the concept of fuzzy logic for feature extraction. DMK-ELM is pixel wise algorithm that works on the pixel intensity of face image. In DMK-ELM-FFE the fuzzy theory is used to determine the pixel wise coalition of face image to distinct classes. It handles the uncertainty involved in face recognition due to varying environment conditions. The pixel-wise membership value extracts the unobserved information from face images likely to a considerable extent. It considers facial image as input and uses π MF to fuzzify the intensity of pixels for obtaining degree of membership of pixels to different classes. Consider a face image of dimension $U \times V$. This can be represented in the form of UV dimensional vector T as: $T = [t_1, t_2, \dots, t_d]$, which is the universal set in this context. The size of vector T is d , which denotes the number of pixels in face image i.e. UV . The MF considers each image as a vector and fuzzifies it. The π MF used for the fuzzification of face image is given by:

$$\pi(T : \alpha, \chi, \beta) = \left\{ \begin{array}{l} 0, T \leq \alpha \\ 2 \left(\frac{T-\alpha}{(x-\alpha)} \right)^2, \alpha < T \leq q_1 \\ 1 - 2 \left(\frac{(x-T)}{(x-\alpha)} \right)^2, q_1 < T \leq x \\ 2 \left(\frac{(T-x)}{(\beta-x)} \right)^2, \chi < T \leq q_2 \\ 1 - 2 \left(\frac{(\beta-T)}{(\beta-x)} \right)^2, q_2 < T \leq \beta \\ 0, T \geq \beta \end{array} \right. \tag{35}$$

Where α , β and χ represents min, max and mean of a specific pixel element in all training images. π MF provides highest membership grade equal to 1 at mean χ and 0.5 at two cross-over points (q_1 and q_2). The q_1 is between min and mean, whereas q_2 lies between mean and max. The value of membership grade is less than 0.5 beyond q_1 and q_2 . The pixels with membership grade value less than 0.5 are used to enhance the capability of face recognition system to assimilate the variation unwrapped by training data. The π MF is calculated based on the values of min, max and mean of a pixel in all training images using:

$$\begin{aligned} \alpha &= \min(TD) \\ \beta &= \max(TD) \\ \chi &= \text{mean}(TD) \end{aligned} \tag{36}$$

Where TD is a vector representing the intensity values of a particular pixel in all training images. For example if TX is pixel matrix of all the training images, α (1) will be the min in the first row of this matrix. Similarly β and χ are the max and mean values respectively of vector TD . After applying π MF the membership grade of face image is represented in the vector form as: $M = [m_1, \dots, m_2, \dots, m_d]$. M is a feature vector obtained after fuzzification of input image vector. Concatenate the pixel wise vector i.e. T and fuzzy feature vector M . The resultant

feature vector is

$$x_f = x_T \cup x_M \tag{37}$$

The objective function of DMK-ELM-FFE is:

$$\text{Min} : \frac{1}{2} \|\theta\|^2 + \frac{W}{2} \sum_{i=1}^N \|\delta\|^2$$

$$\text{such that} : \theta^T b(x_f, iw f_i : \gamma) = y_i - \delta_i, \forall i \sum_{v=1}^u \gamma_v = 1, \gamma_v \geq 0 \tag{38}$$

The objective function of DMK-ELM-FFE in Eq. (38) is formulated in the same way as achieved for DMK-ELM (Eq. (34)). The primary difference is that the DMK-ELM is pixel wise algorithm and DMK-ELM-FFE is multi-modal (pixel-wise and fuzzify). The output function of DMK-ELM-FFE is:

$$f_o = B((iw f_i, x_f i); \gamma) B((iw f_j, x_f j); \gamma) \left(\frac{I}{W} + B((iw f_i, x_f i); \gamma) B((iw f_j, x_f j); \gamma) \right)^{-1} Y^T \tag{39}$$

The Deterministic multikernel ELM with Fuzzy feature extraction is outlined in algorithm 3.

Algorithm 3: DMK-ELM with Fuzzy Feature Extraction.

Input: A set of face image training samples $(x_k, y_k), k = 1, 2, \dots, N$ and specified u kernels $K_u(x_k, x_i)$

Output: output function f_o , predicting class y .

1. Obtain the fuzzy feature vector x_M by Equations (35) and (36).
 2. Determine the resultant feature vector x_f from Equation (37).
 3. Analytically obtain the input weight vector from Equation (21) by evaluating the norm and mean of input data x_f .
 4. Determine bias by calculating the mean of input weight vector.
 5. Obtain the optimal kernel using $K(\cdot, \cdot; \gamma) = \sum_{v=1}^u \gamma_v K_v(\cdot, \cdot)$
 6. Evaluate the resultant deterministic output solution for DMK-ELM-FFE by practicing the feature vectors determine after step 4.
-

3.3 Analysis of proposed approaches

We conclude this section by comparing our proposed algorithms with ELM, KELM (with RBF and polynomial kernel functions) and OMKELM. The objective of DMK-ELM is to achieve deterministic and accurate solution for real time classification problems. The empirical results achieve with the help of ELM are variant. The parameters affecting the ELM performance are input weights, hidden layer biases and count of hidden neurons. In ELM, the input weights and biases are arbitrarily projected and hidden layer nodes are adjusted with respect to classification accuracy. In DMK-ELM these parameters are analytically calculated with the help of input data.

The DMK-ELM is computational efficient as the input weights and biases are obtained by norm and mean operations. DMK-ELM is more accurate and potential than KELM that exercises single kernel for mapping the feature vectors, which is not efficient to discriminate inter class variance.

To make the DMK-ELM more accurate, we use the concept of approximation theory of fuzzy logic to design DMK-ELM-FFE. In this the feature vectors are obtained with the help of fuzzification operation (π MF). It is computationally faster as the parameters of π MF are analytically calculated with the help of min, max and mean operation on input data. It performs better than OMKELM, as the feature vectors are powerful to extract the hidden information from training data that reveals the inter class variance.

4 Experimental evaluation

This section demonstrates the utility of proposed approaches of classification. For this the experiments are performed on prominent face databases AT& T [50], Yale [8], Georgia Tech [12] and AR [68]. A brief description of face databases is stated in sub section. The accomplished executed results of introduced techniques are compared with OMKELM, KELM and ELM.

4.1 Databases

- 1) AT& T database: consist of 400 images (gray scale) depicting 40 subjects (male and female). The images are captured with distinct expressions and decoration e.g. surprise, happiness, neutral, open/closed eyes. The dimension of each image is 112×92 . The face images of database are exemplified in Fig. 2.
- 2) Yale database: comprises of facial images representing 15 persons. The facial image of each person spans a variation in facial details like surprised, winking, center-light, left-light and illumination conditions. The variations are depicted in 11 facial images. The size of each image is 320×243 . The images of database are exemplified in Fig. 3.
- 3) Georgia Tech database: Comprises of 750 color facial images, depicting 50 individuals. The images of this database vary in terms of lightning conditions, appearance and facial



Fig. 2 Sample face images with different expressions from AT & T database



Fig. 3 Face images of Yale database with lightning variations

expressions. Each image is of dimension 120×90 . The images of this database are exemplified in Fig. 4.

- 4) AR database: consist of 4000 images representing 70 male and 56 female subjects. An individual with varying occlusions, face configuration and lightning conditions are depicted in 26 facial images. We have employed a subset of this database depicting 100 individuals. For each subject 14 images of dimension 165×120 with changing expressions and environmental conditions are considered. The images of database are exemplified in Fig. 5.

4.2 Experimental setup

In our simulations we have implemented the variants of ELM on AT& T, Yale, Georgia Tech (GT) and AR databases. For calculation of performance results, the cardinality of training set is increased with one image up to 8 images per subject. For exemplary, consider AT& T database, the minimum cardinality of training set is 40 and maximum is 320 images. Corresponding to this, the testing set has maximum cardinality of 360 images and minimum of 80 images. For ELM techniques utilizing the kernel formulation, we have exploited the RBF kernel:

$$K(a, a_i) = \exp\left(-\sigma|a-a_i|^2\right) \tag{40}$$

and polynomial kernel:

$$K(a, a_i) = (a \cdot a_i + \zeta)^\varphi \tag{41}$$

Here σ , ζ , and φ are kernel parameters, that are experimentally adjusted with respect to classification accuracy.

In case of ELM, employing random input weight iw and bias b , we have exploited sigmoidal activation function:



Fig. 4 Face images depicting persons of Georgia Tech database



Fig. 5 Face images of AR database with illumination variations

$$\psi(x, iw, b) = \frac{1}{(1 + \exp(-(iw \cdot x + b)))} \quad (42)$$

x is the image vector and its dimension for AT& T database is 10,304. We figure out the performance of ELM, utilizing random weights for a count of hidden nodes $\eta = (100, 200, \dots, 1000)$ and coefficient of regularization $W = (1, 5, 10, \dots, 100, 120)$. After trying a range of values for W and kernel parameters, the best classification results are obtained for $W = 10$, $\sigma = 100$, $\psi = 2$, $\zeta = 1$ and $\eta = 1000$. In OMKELM the kernel function exploited for the performance calculation is convex combination of RBF and polynomial kernel given as:

$$K(a, a_i) = \gamma_1 K_1(a, a_i) + \gamma_2 K_2(a, a_i), \sum_{i=1}^2 \gamma_i = 1 \quad (43)$$

The combination coefficient of different kernels is database dependent and has different optimal values for distinct databases.

4.3 Performance measures for classification

In our experiments, we utilize several metrics to characterize the performance of proposed algorithms on face databases. The main metrics employed in our research are:

- Testing accuracy: refers to the ratio of correctly classified instances from the total number of testing instances (testing dataset).

$$accuracy = \left(1 - \frac{m_class}{Total}\right) \times 100$$

where m_class represents the instances which are misclassified as another class and $Total$ represents the cardinality of testing dataset.

- Precision: refers to the proportion of predicted positive examples that are actually true positives. It is used to measure the correctly predicted labels. Precision is calculated as follows [51]:

$$precision = \frac{\sum_{i=1}^h TP_i}{\sum_{i=1}^h (TP_i + FP_i)}$$

where h is number of classes, TP is True Positives, FP is False

Positives

- Recall: refers to proportion of true positive examples that are predicted to be positive. It is used to measure the number of correct labels predicted by a classifier. Recall is calculated as follows [51]:

$$recall = \frac{\sum_{i=1}^h TP_i}{\sum_{i=1}^h (TP_i + FN_i)}$$

where FN is False Negative

- F-measure: is an accuracy measurement, which determines a weighted average between precision and recall. It is relationship between positively labeled data and actual prediction by a classifier based on per-class average [51].

$$F\text{-measure} = \frac{2 \times precision \times recall}{precision + recall}$$

4.4 Empirical results

Empirical results calculated by implementing competing approaches and proposed techniques on AT& T, Yale, GT and AR databases, utilizing testing accuracy as performance metric are revealed in Tables 1, 2, 3 and 4 respectively. The experimental results calculated for face databases, utilizing precision, recall and F-measure as performance metrics are revealed in Tables 5, 6, 7 and 8. The results are analyzed for different cardinality of training dataset.

Table 1 Performance results based on testing accuracy for different classifiers on AT & T database

Approach	Number of training images practiced per identity							
	1	2	3	4	5	6	7	8
ELM [65]	62.50	74.69	78.57	83.50	83.50	90.63	90.83	93.75
KELM [66](RBF)	67.22	77.81	80.36	87.08	89.50	95.00	95.00	96.25
KELM(Polynomial)	67.89	78.10	84.31	89.52	91.11	95.00	95.00	96.80
OMKELM [3]	68.33	79.37	85.00	91.25	92.00	95.00	95.83	97.50
DMK-ELM	73.89	86.56	85.71	91.25	92.50	95.37	95.83	97.50
DMK-ELM-FFE	75.28	89.32	93.58	95.83	98.50	96.81	98.22	98.75

The text written in bold style in the tables are showing the results of proposed techniques

Table 2 Performance results based on testing accuracy for different classifiers on Yale database

Approach	Number of training images practiced per identity							
	1	2	3	4	5	6	7	8
ELM [65]	54.10	82.41	85.35	89.52	88.80	94.00	95.00	96.67
KELM (RBF) [66]	55.33	80.00	89.17	89.52	88.89	96.00	95.00	97.78
KELM (Polynomial)	62.67	87.41	90.00	91.43	93.33	96.00	98.67	100
OMKELM [3]	64.67	88.15	92.50	93.33	94.44	96.00	98.67	100
DMK-ELM	65.33	89.63	95.00	96.19	95.56	98.67	100	100
DMK-ELM-FFE	65.33	91.89	95.56	96.19	97.50	98.33	100	100

The text written in bold style in the tables are showing the results of proposed techniques

4.4.1 Classification results on AT& T database

The AT& T database incorporates very less illumination variations, so no pre-processing is done on this database. The value of kernel combination coefficients γ_1 and γ_2 is 0.5. In AT& T database, for all the classifiers the accuracy rate increases with increase in size of training set. From the recognition results it can be stated that kernel formulation of ELM gives better results than ELM with random input parameters. For exemplary, when the cardinality of training set is 240 images, the recognition rate achieved with ELM, KELM (RBF) and KELM (polynomial) classifier is 93.75%, 96.25% and 96.80% respectively. Further, the results evaluated using multi kernels (OMKELM) are better than single kernel ELM. Although the results of KELM with polynomial kernel function are comparable with OMKELM. As compared with OMKELM, there is significant improvement in accuracy rate using DMK-ELM and DMK-ELM-FFE. For exemplary when two images per subject are used for training, the percentage increase in recognition rate is 9% with DMK-ELM and 12.5% for DMK-ELM-FFE. Figure 6(a) shows the performance comparison graph of different techniques on this database. To establish the efficacy of the proposed algorithms, the performance of all the classifiers are evaluated on other metrics for AT& T database and the values are listed in Table 5. It is clearly evoked by comparing the results using precision, recall and F-measure, that proposed classifiers are significantly superior to that of other existing ELM based classifiers.

Table 3 Performance results based on testing accuracy for different classifiers on GT database

Approach	Number of training images practiced per identity							
	1	2	3	4	5	6	7	8
ELM [65]	31.00	41.08	42.83	44.55	43.60	48.67	51.25	52.29
KELM (RBF) [66]	32.86	44.31	48.33	51.64	51.60	61.78	66.25	68.29
KELM (Polynomial)	34.43	47.08	51.78	55.20	58.19	65.87	69.00	72.14
OMKELM [3]	35.14	47.08	52.50	57.82	61.20	67.78	70.75	73.43
DMK-ELM	35.86	48.15	53.33	58.55	61.40	68.22	71.00	73.43
DMK-ELM-FFE	45.57	59.38	63.17	67.64	68.60	76.00	78.50	80.86

The text written in bold style in the tables are showing the results of proposed techniques

Table 4 Performance results based on testing accuracy for different classifiers on AR database

Approach	Number of training images practiced per identity							
	1	2	3	4	5	6	7	8
ELM [65]	67.54	66.42	68.64	70.50	74.22	71.50	60.71	59.50
KELM [66] (RBF)	60.31	62.00	64.91	74.00	80.33	81.13	82.86	86.50
KELM (Polynomial)	70.85	71.25	76.36	82.72	91.44	92.55	93.19	98.17
OMKELM [3]	70.85	74.75	76.36	83.80	91.44	93.25	93.89	98.17
DMK-ELM	77.69	82.33	84.55	87.70	93.44	94.63	96.29	97.69
DMK-ELM-FFE	78.08	82.25	84.27	91.00	95.44	96.75	96.86	98.50

The text written in bold style in the tables are showing the results of proposed techniques

4.4.2 Classification results on Yale database

The face images in Yale database are normalized using fuzzy filter, as the database contains large variation in terms of illumination and expressions. To reduce the effect of lightning variations, the numbers of LF-DCT coefficients considered are 21 with 0.5 value of fuzzy constant. The kernel combination coefficient for RBF and polynomial is 0.4 and 0.6 respectively. The recognition rate in this database is more than AT & T database in most of the cases. For example, when the training set size is eight images per subject we achieve 100 percentage recognition rate with KELM (polynomial), OMKELM, DMK-ELM and DMK-ELM-FFE. The accuracy results reveal that multi kernels technique is more accurate than single kernel. For example, on comparing the accuracy results of OMKELM and KELM (polynomial) the percentage increase in accuracy rate is 2%, when four images per subject are used for training.

Table 5 Performance results based on precision, recall and F-measure for different classifiers on AT & T database

Approach	Metric	Number of training images practiced per identity							
		1	2	3	4	5	6	7	8
ELM	<i>precision</i>	0.6685	0.7510	0.7714	0.8331	0.8675	0.8976	0.8617	0.9167
	<i>recall</i>	0.6472	0.7437	0.7564	0.8217	0.8550	0.8863	0.8750	0.9125
	<i>F-measure</i>	0.6376	0.7399	0.7529	0.8154	0.8492	0.8847	0.8586	0.9050
KELM (RBF)	<i>precision</i>	0.7584	0.8306	0.8425	0.8731	0.8999	0.9433	0.9487	0.9625
	<i>recall</i>	0.6806	0.7756	0.7929	0.8458	0.8850	0.9375	0.9417	0.9375
	<i>F-measure</i>	0.6801	0.7750	0.7970	0.8442	0.8863	0.9376	0.9409	0.9350
KELM (polynomial)	<i>precision</i>	0.7679	0.8495	0.8632	0.9175	0.9230	0.9442	0.9514	0.9650
	<i>recall</i>	0.7033	0.7810	0.8507	0.8942	0.9110	0.9437	0.9500	0.9625
	<i>F-measure</i>	0.7015	0.7800	0.8501	0.8921	0.9105	0.9436	0.9488	0.9600
OMKELM	<i>precision</i>	0.7679	0.8515	0.8732	0.9195	0.9330	0.9542	0.9604	0.9750
	<i>recall</i>	0.7033	0.7900	0.8517	0.9122	0.9200	0.9437	0.9576	0.9625
	<i>F-measure</i>	0.7015	0.7888	0.8502	0.9120	0.9185	0.9436	0.9488	0.9600
DMK-ELM	<i>precision</i>	0.7798	0.8845	0.8981	0.9195	0.9334	0.9592	0.9604	0.9750
	<i>recall</i>	0.7389	0.8637	0.8571	0.9122	0.9250	0.9557	0.9576	0.9625
	<i>F-measure</i>	0.7320	0.8518	0.8514	0.9120	0.9190	0.9473	0.9488	0.9600
DMK-ELM- FFE	<i>precision</i>	0.7964	0.8936	0.9365	0.9590	0.9856	0.9691	0.9886	0.9875
	<i>recall</i>	0.7506	0.8875	0.9329	0.9500	0.9800	0.9615	0.9783	0.9850
	<i>F-measure</i>	0.7419	0.8835	0.9262	0.9489	0.9786	0.9610	0.9725	0.9840

The text written in bold style in the tables are showing the results of proposed techniques

Table 6 Performance results based on precision, recall and F-measure for different classifiers on Yale database

Approach	Metric	Number of training images practiced per identity							
		1	2	3	4	5	6	7	8
ELM	<i>precision</i>	0.5583	0.7875	0.8515	0.8957	0.8903	0.9433	0.9344	0.9389
	<i>recall</i>	0.5410	0.7556	0.8535	0.8943	0.8756	0.9400	0.9167	0.9333
	<i>F-measure</i>	0.5337	0.7429	0.8494	0.8923	0.8741	0.9363	0.9139	0.9327
KELM (RBF)	<i>precision</i>	0.6963	0.8006	0.9033	0.9063	0.8992	0.9687	0.9533	0.9833
	<i>recall</i>	0.5533	0.8000	0.8917	0.8952	0.8833	0.9600	0.9500	0.9778
	<i>F-measure</i>	0.5527	0.7947	0.8914	0.8942	0.8809	0.9600	0.9430	0.9771
KELM (polynomial)	<i>precision</i>	0.7013	0.8859	0.9077	0.9167	0.9410	0.9678	0.9889	1
	<i>recall</i>	0.6267	0.8732	0.9000	0.9143	0.9333	0.9600	0.9867	1
	<i>F-measure</i>	0.6221	0.8704	0.9000	0.9109	0.9322	0.9592	0.9830	1
OMKELM	<i>precision</i>	0.7013	0.8959	0.9277	0.9367	0.9510	0.9678	0.9889	1
	<i>recall</i>	0.6467	0.8930	0.9250	0.9333	0.9444	0.9600	0.9867	1
	<i>F-measure</i>	0.6421	0.8904	0.9241	0.9309	0.9404	0.9592	0.9830	1
DMK-ELM	<i>precision</i>	0.7493	0.9089	0.9476	0.9645	0.9643	0.9896	1	1
	<i>recall</i>	0.6533	0.8963	0.9433	0.9619	0.9550	0.9867	1	1
	<i>F-measure</i>	0.6511	0.8917	0.9428	0.9605	0.9531	0.9864	1	1
DMK-ELM -FFE	<i>precision</i>	0.7493	0.9242	0.9599	0.9645	0.9810	0.9878	1	1
	<i>recall</i>	0.6533	0.9189	0.9550	0.9619	0.9750	0.9833	1	1
	<i>F-measure</i>	0.6511	0.9117	0.9545	0.9605	0.9716	0.9831	1	1

The text written in bold style in the tables are showing the results of proposed techniques

The results illustrate that proposed algorithms give better results than the OMKELM. For example when the training images are seven for each face identity, the proposed algorithms give 100% accuracy, while 98.67% accuracy is achieved with OMKELM. On average the increase in accuracy rate is more than two percentage by implementing the proposed approaches. Figure 6(b) shows the graph of comparison, based on percentage recognition rate of

Table 7 Performance results based on precision, recall and F-measure for different classifiers on GT database

Approach	Metric	Number of training images practiced per identity							
		1	2	3	4	5	6	7	8
ELM	<i>precision</i>	0.3168	0.4081	0.4460	0.4518	0.4380	0.4900	0.5182	0.5225
	<i>recall</i>	0.2943	0.4000	0.4167	0.4455	0.4360	0.4856	0.5125	0.5057
	<i>F-measure</i>	0.2840	0.3799	0.4052	0.4454	0.4294	0.4787	0.5051	0.4864
KELM (RBF)	<i>precision</i>	0.4022	0.4966	0.5591	0.5550	0.5403	0.6420	0.6789	0.7002
	<i>recall</i>	0.3286	0.4400	0.4833	0.5036	0.5140	0.6178	0.6625	0.6823
	<i>F-measure</i>	0.3237	0.4400	0.4800	0.4895	0.4912	0.6098	0.6623	0.6413
KELM (Polynomial)	<i>precision</i>	0.4164	0.5084	0.5851	0.6078	0.6170	0.6752	0.7369	0.7437
	<i>recall</i>	0.3443	0.4708	0.5178	0.5520	0.5819	0.6578	0.6890	0.7193
	<i>F-measure</i>	0.3410	0.4608	0.5148	0.5509	0.5790	0.6564	0.6889	0.7150
OMKELM	<i>precision</i>	0.4260	0.5084	0.5951	0.6278	0.6370	0.7352	0.7469	0.7537
	<i>recall</i>	0.3514	0.4708	0.5250	0.5782	0.6120	0.6778	0.7075	0.7343
	<i>F-measure</i>	0.3470	0.4608	0.5248	0.5709	0.6090	0.6764	0.7022	0.7340
DMK-ELM	<i>precision</i>	0.4440	0.5305	0.5518	0.6415	0.6650	0.7573	0.7505	0.7537
	<i>recall</i>	0.3586	0.4815	0.5333	0.5855	0.6140	0.6822	0.7100	0.7343
	<i>F-measure</i>	0.3531	0.4811	0.5300	0.5847	0.6123	0.6783	0.7091	0.7340
DMK-ELM -FFE	<i>precision</i>	0.4883	0.6034	0.6499	0.6512	0.6877	0.7748	0.7861	0.8041
	<i>recall</i>	0.4557	0.5938	0.6317	0.6764	0.6860	0.7600	0.7850	0.7986
	<i>F-measure</i>	0.4553	0.5934	0.6267	0.6731	0.6843	0.7528	0.7819	0.7954

The text written in bold style in the tables are showing the results of proposed techniques

Table 8 Performance results based on precision, recall and F-measure for different classifiers on AR database

	Metric	Number of training images practiced per identity							
		1	2	3	4	5	6	7	8
ELM	<i>precision</i>	0.7557	0.7466	0.7635	0.7487	0.7750	0.7282	0.6204	0.5948
	<i>recall</i>	0.6754	0.6642	0.6864	0.7050	0.7422	0.6963	0.6229	0.5550
	<i>F-measure</i>	0.6857	0.6740	0.6972	0.7002	0.7356	0.6854	0.5955	0.5438
KELM (RBF)	<i>precision</i>	0.8116	0.8111	0.8246	0.8407	0.8781	0.8871	0.8636	0.9059
	<i>recall</i>	0.6031	0.6200	0.6491	0.7400	0.8033	0.8113	0.8286	0.8650
	<i>F-measure</i>	0.6423	0.6534	0.6825	0.7502	0.8070	0.8142	0.8207	0.8654
KELM (Polynomial)	<i>precision</i>	0.8317	0.8350	0.8725	0.8872	0.9332	0.9408	0.9399	0.9844
	<i>recall</i>	0.7085	0.7175	0.7636	0.8272	0.9144	0.9255	0.9319	0.9817
	<i>F-measure</i>	0.7042	0.7140	0.7833	0.8223	0.9105	0.9251	0.9216	0.9811
OMKELM	<i>precision</i>	0.8317	0.8411	0.8725	0.8970	0.9332	0.9548	0.9490	0.9844
	<i>recall</i>	0.7085	0.7475	0.7636	0.8380	0.9144	0.9325	0.9386	0.9817
	<i>F-measure</i>	0.7342	0.7640	0.7833	0.8317	0.9105	0.9285	0.9326	0.9811
DMK-ELM	<i>precision</i>	0.8625	0.8554	0.8869	0.9038	0.9410	0.9615	0.9749	0.9827
	<i>recall</i>	0.7768	0.8233	0.8455	0.8770	0.9344	0.9463	0.9629	0.9768
	<i>F-measure</i>	0.7762	0.8225	0.8437	0.8755	0.9279	0.9415	0.9620	0.9761
DMK-ELM -FFE	<i>precision</i>	0.8634	0.8744	0.8913	0.9253	0.9598	0.9745	0.9812	0.9881
	<i>recall</i>	0.7808	0.8225	0.8426	0.9080	0.9544	0.9675	0.9686	0.9850
	<i>F-measure</i>	0.7801	0.8204	0.8419	0.9071	0.9519	0.9642	0.9674	0.9813

The text written in bold style in the tables are showing the results of proposed techniques

different approaches for Yale database. The performance results of proposed approaches based on precision, recall and F-measure are more promising than the comparative ELMs for all the size of training and test datasets. In Table 6, when seven and eight images per identity are considered for training the values of precision, recall and F-measure is equal to 1, which means all the test images are classified to their corresponding classes.

4.4.3 Classification results on GT database

The GT database contains color images, so before performance evaluation these images are converted to gray scale images without performing any pre-processing. The recognition rate of this database is very less irrespective of training size, when compared with other databases. The maximum accuracy is achieved when the cardinality of training dataset is 400, i.e. eight images per subject as there are 50 subjects with 15 images per subject. As it can be verified from the results that OMKELM gives better performance than ELM and KELM (with single kernel). The recognition results of OMKELM are comparable with DMK-ELM, but there is 10% improvement in accuracy rate with DMK-ELM-FFE irrespective of size of training dataset. On comparing the proposed approaches, the DMK-ELM-FFE results are more accurate than DMK-ELM irrespective of cardinality of training set. For example, when two images per subject are considered for training, the accuracy rate increases by 22% with DMK-ELM-FFE classifier. Figure 6(c) shows the graphical comparison, based on accuracy rate of proposed approaches with existing techniques in literature for GT database. The precision, recall and F-measure for GT database is less when compared with other databases. Although the values of these metrics are higher in case of proposed algorithms than existing ELM variants. On analyzing the performance results based on different metrics in Tables 3 and 7 for GT database, it can be concluded that among the proposed algorithms DMK-ELM-FFE is superior than DMK-ELM.

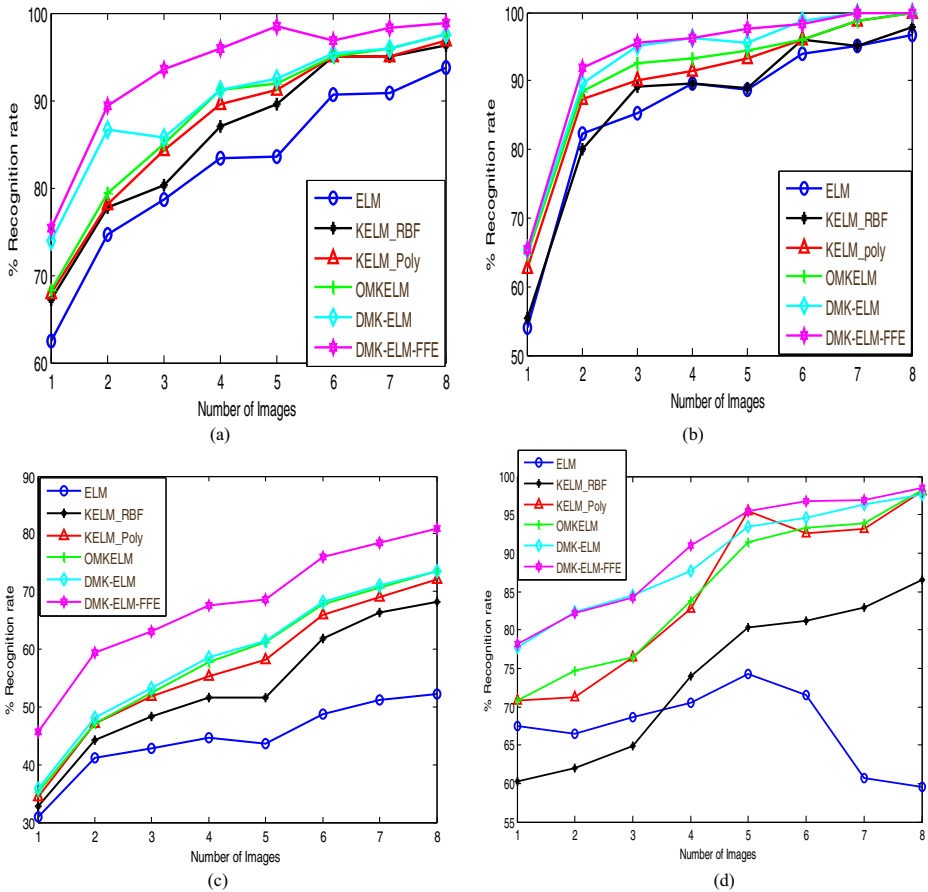


Fig. 6 Comparison of classification accuracy of DMK-ELM, DMK-ELM-FFE, OMKELM, KELM and ELM based on percentage recognition rate

4.4.4 Classification results on AR database

As the AR database contains colored images with non-uniform lightning variations, so the images in this database are normalized and converted to gray scale. The LF-DCT coefficients modified by employing fuzzy filter are 91 and the value of fuzzy constant is 0.5. The values of kernel coefficients γ_1 and γ_2 are 0.4 and 0.6 respectively. On analyzing the accuracy results it can be stated that OMKELM performs better than other existing techniques of classification. The percentage increase in accuracy rate is 11%, 20% and 4% with ELM, KELM (RBF) and KELM (polynomial) respectively, when the cardinality of training set is 200 images. The achieved recognition rates with proposed schemes are more than OMKELM. For exemplary, when four images per identity are considered, the increase in accuracy rate is more than 4.5 percentage. On analyzing the performance results of proposed approaches, DMK-ELM-FFE performs better than DMK-ELM. For example when the size of training dataset is 500 images, the increase in accuracy rate is 2%. Figure 6 (d) displays the graphical comparison of different classifiers for AR database, formed on recognition rate. In Table 8, the performance results of different classifiers based on precision, recall and F-measure for AR database are shown. The

empirical results based on all performance measures proves that proposed non-iterative DMK classifiers are more efficient than other state-of-the-art classifiers.

5 Conclusions

This paper prospects the issue of face recognition efficiently and deterministically by developing multi kernel learning based deterministic ELM along with fuzzy based feature extraction. The foundation of proposed approach is ELM classifier, whose structural parameters are analyzed in detail. Although the ELM is efficient classifier, but it cannot be employed in classification applications that demands deterministic and immutable output solution. To make it deterministic, the structural parameters of ELM are analytically devised. Furthermore, to make it more efficient, we employed data related multikernel approach, in which kernel and their combination coefficients are experimentally determined by investigating the perceptivity of the problem. To make it more accurate, we use fuzzy π MF that works on the intensity of pixels to find their association with different classes. From analyzing the results, it is evident that Kernel model of ELM is more accurate than ELM in most of the cases. Further, when single kernel approach is adopted the polynomial kernel gives more promising result than RBF kernel. For all the databases, OMKELM results are more accurate than single kernel approach, although the improvement in result is database dependent. On analyzing the empirical results, it can be concluded that the presented approaches outperform ELM, KELM and OMKELM. The experiential results acknowledge that the DMK-ELM-FFE performs superior than DMK-ELM for all the databases. The differences in performance results of proposed approaches are larger, when the databases are not pre-processed.

The objective of the proposed novel methods is to overcome the randomness of ELM along with to make it more efficient and accurate with optimized kernel and fuzzification of pixels. In contempt of proposed methods outshines other competitive algorithms, there is enough research work virtue to be explored in future such as to develop the variants of proposed algorithms for multilayer and to develop a potential algorithm that deals with extensive large massive training set emanated from augmentation of large-scale kernel functions.

References

1. Ahlawat S, Choudhary A, Nayyar A, Singh S, Yoon B (2020) Improved handwritten digit recognition using convolutional neural networks (CNN). *Sensors* 20:3344
2. Ahonen T, Hadid A, Pietikäinen M (2004) Face recognition with local binary patterns. In: *Eur Conf Comput Vis*. pp. 469–481
3. Ahuja B, Vishwakarma VP (2018) Optimised multikernels based extreme learning machine for face recognition. *Int J Appl Pattern Recognit* 5:330–340
4. Ahuja B, Vishwakarma VP (2019) Local feature extraction based KELM for face recognition. In: *2019 twelfth Int Conf Contemp Comput*. pp. 1–5
5. Ahuja B, Vishwakarma VP (2020) Local binary pattern based feature extraction with KELM for face identification. In: *2020 6th Int. Conf. Signal Process. Commun*. pp. 91–95
6. Aiolli F, Donini M (2015) EasyMKL: a scalable multiple kernel learning algorithm. *Neurocomputing* 169: 215–224
7. Bartlett PL (1998) The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Trans Inf Theory* 44:525–536

8. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans Pattern Anal Mach Intell* 19:711–720
9. Boyd S, Vandenberghe L (2004) *Convex optimization*. Cambridge university press
10. Brunelli R, Poggio T (1993) Face recognition: features versus templates. *IEEE Trans Pattern Anal Mach Intell* 15:1042–1052
11. Bucak SS, Jin R, Jain AK (2013) Multiple kernel learning for visual object recognition: a review. *IEEE Trans Pattern Anal Mach Intell* 36:1354–1369
12. Chen L, Man H, Nefian AV (2005) Face recognition based on multi-class mapping of fisher scores. *Pattern Recogn* 38:799–811
13. Chen W, Er MJ, Wu S (2006) Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain. *IEEE Trans Syst Man, Cybern Part B* 36:458–466
14. De Siqueira FR, Schwartz WR, Pedrini H (2013) Multi-scale gray level co-occurrence matrices for texture description. *Neurocomputing* 120:336–345
15. Deng W, Zheng Q, Chen L (2009) Regularized extreme learning machine. In: 2009 IEEE Symp Comput Intell data Min pp. 389–395
16. Deng C, Han Y, Zhao B (2019) High-performance visual tracking with extreme learning machine framework. *IEEE Trans Cybern*.
17. Déniz O, Bueno G, Salido J, la Torre F (2011) Face recognition using histograms of oriented gradients. *Pattern Recogn Lett* 32:1598–1603
18. Evgeniou T, Micchelli CA, Pontil M (2005) Learning multiple tasks with kernel methods. *J Mach Learn Res* 6:615–637
19. Fan X, Xiang C, Chen C, et al (2020) BuildSenSys: Reusing building sensing data for traffic prediction with cross-domain learning. *IEEE Trans Mob Comput*
20. Feng G, Huang G-B, Lin Q, Gay R (2009) Error minimized extreme learning machine with growth of hidden nodes and incremental learning. *IEEE Trans Neural Netw* 20:1352–1357
21. Gadekallu TR, Rajput DS, Reddy MPK, et al (2020) A novel PCA–whale optimization-based deep neural network model for classification of tomato plant diseases using GPU. *J Real-Time Image Process* 1–14.
22. Gönen M, Alpaydin E (2011) Multiple kernel learning algorithms. *J Mach Learn Res* 12:2211–2268
23. Gonzalez RC, Woods RE, others (2002) *Digital image processing [M]*. Publ house Electron Ind 141
24. Guo P (2018) A vest of the pseudoinverse learning algorithm. *arXiv Prepr. arXiv1805.07828*
25. Guo P, Lyu MR, Mastorakis NE (2001) Pseudoinverse learning algorithm for feedforward neural networks. *Adv Neural Networks Appl*.
26. Han F, Huang D-S (2006) Improved extreme learning machine for function approximation by encoding a priori information. *Neurocomputing* 69:2369–2373
27. Han H-G, Wang L-D, Qiao J-F (2014) Hierarchical extreme learning machine for feedforward neural network. *Neurocomputing* 128:128–135
28. Hoerl AE, Kennard RW (1970) Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* 12:55–67
29. Huang G-B, Zhu Q-Y, Siew C-K (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: *Neural Networks, 2004. Proceedings. 2004 IEEE Int. Jt Conf* pp 985–990
30. Huang G-B, Zhu Q-Y, Mao KZ et al (2006) Can threshold networks be trained directly? *IEEE Trans Circuits Syst II Express Briefs* 53:187–191
31. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. *Neurocomputing* 70:489–501
32. Huang G-B, Li M-B, Chen L, Siew C-K (2008) Incremental extreme learning machine with fully complex hidden nodes. *Neurocomputing* 71:576–583
33. Huang G-B, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. *IEEE Trans Syst Man, Cybern Part B* 42:513–529
34. Huang Z, Yu Y, Gu J, Liu H (2017) An efficient method for traffic sign recognition based on extreme learning machine. *IEEE Trans Cybern* 47:920–933
35. Jian Y, Huang D, Yan J, Lu K, Huang Y, Wen T, Zeng T, Zhong S, Xie Q (2017) A novel extreme learning machine classification model for e-nose application based on the multiple kernel approach. *Sensors* 17:1434
36. Khare N, Devan P, Chowdhary CL, Bhattacharya S, Singh G, Singh S, Yoon B (2020) SMO-DNN: spider monkey optimization and deep neural network hybrid classifier model for intrusion detection. *Electronics* 9:692
37. Kim D-J, Bien Z (2008) Design of “personalized” classifier using soft computing techniques for “personalized” facial expression recognition. *IEEE Trans Fuzzy Syst* 16:874–885
38. Klir G, Yuan B (1995) *Fuzzy sets and fuzzy logic*. Prentice hall New Jersey
39. Li SZ, Anil K (2005) Jain. *Handbook of Face Recognition*.
40. Li X, Mao W, Jiang W (2016) Multiple-kernel-learning-based extreme learning machine for classification design. *Neural Comput Appl* 27:175–184

41. Li Y, Hu H, Zhu Z, Zhou G (2020) SCANet: sensor-based continuous authentication with two-stream convolutional neural networks. *ACM Trans Sens Networks* 16:1–27
42. Li Y, Zou B, Deng S, Zhou G (2020) Using feature fusion strategies in continuous authentication on smartphones. *IEEE Internet Comput* 24:49–56
43. Liu X, Wang L, Huang G-B, Zhang J, Yin J (2015) Multiple kernel extreme learning machine. *Neurocomputing* 149:253–264
44. Lu C, Ke H, Zhang G, Mei Y, Xu H (2019) An improved weighted extreme learning machine for imbalanced data classification. *Memetic Comput* 11:27–34
45. Martínez JM, Escandell-Montero P, Soria-Olivas E et al (2011) Regularized extreme learning machine for regression problems. *Neurocomputing* 74:3716–3721
46. Miche Y, Sorjamaa A, Bas P et al (2009) OP-ELM: optimally pruned extreme learning machine. *IEEE Trans Neural Netw* 21:158–162
47. Ojala T, Pietikäinen M, Mäenpää T (2000) Gray scale and rotation invariant texture classification with local binary patterns. *Eur Conf Comput Vis*, In, pp 404–420
48. Rao CR, Mitra SK (1971) Generalized inverse of matrices and its applications.
49. Rong H-J, Huang G-B, Sundararajan N, Saratchandran P (2009) Online sequential fuzzy extreme learning machine for function approximation and classification problems. *IEEE Trans Syst Man, Cybern Part B* 39: 1067–1072
50. Samaria FS, Harter AC (1994) Parameterisation of a stochastic model for human face identification. In: *Appl. Comput. Vision, 1994.. Proc. Second IEEE Work.* pp 138–142
51. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. *Inf Process Manag* 45:427–437
52. Tang J, Deng C, Huang G-B (2016) Extreme learning machine for multilayer perceptron. *IEEE Trans neural networks Learn Syst* 27:809–821
53. Vishwakarma VP (2015) Illumination normalization using fuzzy filter in DCT domain for face recognition. *Int J Mach Learn Cybern* 6:17–34
54. Wang Y, Cao F, Yuan Y (2011) A study on effectiveness of extreme learning machine. *Neurocomputing* 74:2483–2490
55. Wong CM, Vong CM, Wong PK, Cao J (2016) Kernel-based multilayer extreme learning machines for representation learning. *IEEE Trans neural networks Learn Syst* 29:757–762
56. Xie X, Zheng W-S, Lai J et al (2010) Normalization of face illumination based on large-and small-scale features. *IEEE Trans Image Process* 20:1807–1821
57. Xu Z, Jin R, Yang H, et al (2010) Simple and efficient multiple kernel learning by group lasso. In: *Proc. 27th Int. Conf. Mach. Learn.* pp 1175–1182
58. Yang H, Xu Z, Ye J et al (2011) Efficient sparse generalized multiple kernel learning. *IEEE Trans Neural Netw* 22:433–446
59. Zadeh LA (1988) Fuzzy logic. *Computer (Long Beach Calif)* 21:83–93
60. Zadeh LA (1999) Fuzzy logic= computing with words. In: *Comput. with words information/intelligent Syst.* 1. Springer, pp 3–23
61. Zhao W, Chellappa R, Phillips PJ, Rosenfeld A (2003) Face recognition: a literature survey. *ACM Comput Surv* 35:399–458
62. Zhou C, Wang L, Zhang Q, Wei X (2013) Face recognition based on PCA image reconstruction and LDA. *Optik (Stuttg)* 124:5599–5603
63. Zhu Q-Y, Qin AK, Suganthan PN, Huang G-B (2005) Evolutionary extreme learning machine. *Pattern Recogn* 38:1759–1763
64. Zhuang J, Tsang IW, Hoi SCH (2011) Two-layer multiple kernel learning. *Proc Fourteenth Int Conf Artif Intell Stat*, In, pp 909–917
65. Zong W, Huang G-B (2011) Face recognition based on extreme learning machine. *Neurocomputing* 74: 2541–2551
66. Zong W, Zhou H, Huang G-B, Lin Z (2011) Face recognition based on kemelized extreme learning machine. In: *Int Conf Auton Intell Syst.* pp. 263–272
67. Zong W, Huang G-B, Chen Y (2013) Weighted extreme learning machine for imbalance learning. *Neurocomputing* 101:229–242
68. Zou C, Kou KI, Wang Y (2016) Quaternion collaborative and sparse representation with application to color face recognition. *IEEE Trans Image Process* 25:3287–3302