Arab J Sci Eng (2017) 42:805–816
DOI 10.1007/s13369-016-2338-5

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

An Adaptive Non-symmetric Fuzzy Activation Function-Based Extreme Learning Machines for Face Recognition

Tripti Goel[1] · **Vijay Nehra**[1] · **Virendra P. Vishwakarma**[2]

Received: 14 February 2016 / Accepted: 13 October 2016 / Published online: 5 November 2016

Abstract The present investigation describes an adaptive non-symmetric fuzzy activation function-based extreme learning machines (ANF-ELM) for face recognition. ELMs are biologically inspired single hidden layer feed forward networks which present significant advantages over traditional back-propagation algorithm. Advantages of ELMs are low computational cost, fast learning speed, ease to implement and good performance results. In ELM, the hidden layer parameters are randomly selected and then the output weights are determined by calculating the inverse of the outputs of hidden layer. ANF-ELM is a simple single hidden layer feed forward network which gives nonlinear mapping from the input space to feature space by an adaptive non-symmetric fuzzy activation function (FAF), s. The s FAF is non-symmetric in nature with shifted origin which gives better performance compared to the symmetric activation function. The effectiveness of the ANF-ELM classifier is tested on four datasets: AT&T, Yale faces, CMU PIE and UMIST. The results show that an ANF-ELM provides good performance results and faster training speed when compared to other state of the art techniques.

Keywords Extreme learning machine · Face recognition · Fuzzy activation function · Single hidden layer feed forward network

1 Introduction

Automatic human face recognition, either from an image or from the video has a wide variety of applications in commercial and law enforcement. Face recognition problem has attracted many researchers since past twenty years [1,2]. However, the performance of face recognition system mainly depends on the classifier used. Many algorithms have been developed for face recognition which works better in outdoor conditions. But it is still an open and challenging task in real applications [3,4].

Traditionally, artificial neural networks (ANN) with back propagation (BP) have been used for face recognition [5]. In ANN, the networks parameters such as weights and biases are calculated by gradient descent-based algorithm which adjusts the parameters iteratively by back propagating error from the output layer to the input layer. Due to iterative process, BP algorithm suffers from the limitation of slow training speed, improper learning rate, over fitting and the presence of local minima. Various variants of ANN are presented in [6–8].

Huang et al. [9] presented an efficient single hidden layer feed forward neural network (SLFN) learning algorithm, ELM to overcome the drawbacks of BP. ELM is a single hidden layer feed forward neural network in which hidden layer's weights and biases are randomly generated. The output weights are calculated by performing the inverse operation on the output of hidden layer. In ELM, only the number of hidden neurons has to be chosen manually. Therefore, it overcomes the limitation of BP, in which a number of parameters have to be chosen such as number of epochs, performance

✉ Tripti Goel
triptigoel83@gmail.com

Vijay Nehra
nehra_vijay@yahoo.com

Virendra P. Vishwakarma
virendravishwa@rediffmail.com

1 Department of Electronics and Communication Engineering, Bhagat Phool Singh Mahila Vishwavidyalaya, Khanpur Kalan, Sonepat, Haryana, India

2 University School of Information and Communication Technology, Guru Gobind Singh Indraprastha University, Dwarka, Delhi, India

goal, batch size, number of layers, number of hidden neurons, etc. It is also shown in [9] that SLFN-ELM can classify a finite set of patterns with good performance results and low computational complexity.

The present approach, ANF-ELM classifier for face recognition uses an adaptive non-symmetric fuzzy activation function s, for ELM to train single layer hidden feed forward network. The s FAF [10] is an adaptive function having a shape, like sigmoid function. The motivation of s FAF is that it maps the outliers (the variation generated due to varying illumination, pose, expression, occlusion, etc.) to a concise range of membership value on either end, near to zero or one membership value. For average grayscale values, the mapping is approximately linear. In fuzzy activation function, the shape of the activation function has important role in recognition accuracy. Therefore, s fuzzy activation function is made adaptive whose steepness is controlled by parameter fuzzifier, f. The value of f is decided experimentally as depicted in Table 2.

Experimental analysis on four different databases, using the present approach has proved the considerable improvement in recognition results and learning speed for face recognition.

The rest of the paper is organized as follows: Review of existing methods for classification is discussed in Sect. 2. Section 3 describes the ANF-ELM learning algorithm for face recognition. Section 4 provides the brief overview of face databases used in the present study. Section 5 presents the experimental results and analysis of the present approach with other conventional algorithms for face recognition followed by conclusion.

2 Review of Existing Techniques

There are many methods for automatic face recognition which are divided into two groups: template-based and feature-based classifiers [11]. The template-based method aims to project the high-dimensional data into lower subspace dimension, where classification will become easy. In this approach, whole face region has been taken into account for the input into face detection system. The most well-known examples of the template-based algorithm are principal component analysis (PCA) [12–14], independent component analysis (ICA) [15] and linear discriminant analysis (LDA) [16]. PCA projects the set of training face images into a linear subspace of eigenvectors which represents the best distribution of face vectors. Test images are then projected into this subspace which selects the eigenvalues of most discriminant features. Another well-known approach LDA [16,17] aims to maximizes the Fisher's criteria. The Fisher's criteria maximize the interclass variances and minimize the intraclass variance. LDA normally needs more training time than PCA, because it generally uses PCA as a preprocessing step to avoid the small sample size problem [18]. Another algorithm, ICA searches for a linear transformation to express a set of random variables as linear combinations of statistically independent source variables [19,20]. However, template-based methods suffer from disadvantages that they are sensitive to variations like illumination and pose, and they consider many assumptions to be fulfilled for satisfied classification.

Feature-based algorithms find the set of local facial features from the face image like nose tip, eyes and mouth corner, etc., and then the system makes a unique ID for new face describing the extracted features and their relationship. Examples of feature-based algorithms are local feature analysis (LFA) [21] and elastic bunch graph method (EBGM) [22]. LFA reduces the dimensionality of the image by choosing a subset of output points that are decorrelated as much possible. EBGM represents the face image by labeled graph which is extracted by an elastic graph matching process. The main problem of the feature-based method is that it may cause some loss of useful information for face recognition and also, there is no any optimal method to extract a set of optimal features.

ANN has emerged as a powerful tool for automatic face recognition. ANN can achieve better performance by integrating both structural and statistical information of the face image. ANN adapts BP for training neural networks. BP learns the network parameters based on gradient descent learning which leads to slow learning speed [23,24]. Lawrence et al. [25] presented a method based on self-organizing map (SOM) and convolutional neural network (CNN) to classify the face images. SOM is used to cluster the set of training images and then construct a low-dimensional subspace. After using SOM, CNN with BP is applied for face recognition. Lin et al. [8] give probabilistic decision-based neural network which uses the advantages of both statistical techniques and neural network. Vapnik [26] introduced support vector machine (SVM) which constructs the optimal hyperplane by mapping input vectors into a high-dimensional space via certain nonlinear transformation. Fleming and Cottrell [27] proposed a nonlinear projection using auto associative neural network to extract the features and then multilayer perceptron, to classify these features. Wright et al. [28] devised a general classifier based on the sparse representation which is computed using l^1 minimization. Mi et al. [29] presented nearest subspace classifier using linear regression technique for face recognition.

Huang et al. [30] devised ELM, which is a single hidden layer feed forward neural network, gave better generalization performance. In ELM, hidden layer parameters are selected randomly and output weights are calculated by pseudo-inverse operation on the output of the hidden layer. Therefore, ELM provides better generalization performance with fast learning speed. Many works have been done a lot of work on

ELM for SLFN in [31–33]. Li and Er in [34] presented ELM with tunable active function for calculating the output of hidden neurons. Liang et al. [35] introduce a simple and efficient online sequential ELM (OS-ELM) that can train data one by one and also chunk by chunk. In OS-ELM, the observations are trained sequentially and presented to learning algorithm. In [36], standard ELM was extended to optimized ELM for classification. Huang et al. [37] gave kernel-based ELM for classification to improve the performance of classification. Huang et al. [38] presented an efficient learning algorithm specifically for sparse ELM. In [39], ELM is presented for unsupervised multilayer perceptron.

Ghosh et al. [40] integrate BP network with fuzzy-based feature extraction to extract the feature wise membership value of belonging of samples of all the classes to take advantage of both neural network and fuzzy logic. Kwak and Pedrycz in [41] devised the Fisher-face approach by integrating the refined information of the class membership of binary labeled faces. Yang et al. [42] also give the concept of integration of discriminative information present in the null space of the fuzzy. Vishwakarma et al. [43] gave a new approach for face recognition based on fuzzification to extract feature wise belonging of the face image. Castano et al. [44] presented PAC-ELM approach in which information retrieved from PCA is used as input to the hidden neurons of ELM. Xu and Zhu [45] gave a simple sparse method for classification. Vishwakarma in [46] presented the classification for face recognition using non-iterative SLFN_BVOI for single hidden layer feed forward network. In SLFN_BVOI classifier, the input weights and biases are allocated from approximate basis vectors of input training space. The present work nonlinearly maps the face images from the input space to the feature space by an adaptive non-symmetric s FAF.

After a brief review of existing techniques related to the present investigation, the forthcoming section presents the proposed approach of an adaptive non-symmetric fuzzy activation function-based extreme learning machines classifier.

3 An Adaptive Non-symmetric Fuzzy-Based Extreme Learning Machine

ELM is an SLFN that not only provides smaller error rate, but also learns at an extremely fast learning speed compared to ANN. A typical architecture of SLFN-ELM is depicted in Fig. 1.

The main elements of ELM are: r units in input layer, N hidden neurons, activation function $g(.)$ and C units in outputs layer. Here r is the dimension of input vector, N is the number of hidden neurons, and C is the number of classes for each database.

In the given SLFN ANF-ELM architecture, there are N training samples (x_i, t_i), where $i = 1, 2, 3 \ldots.. N$. Here,

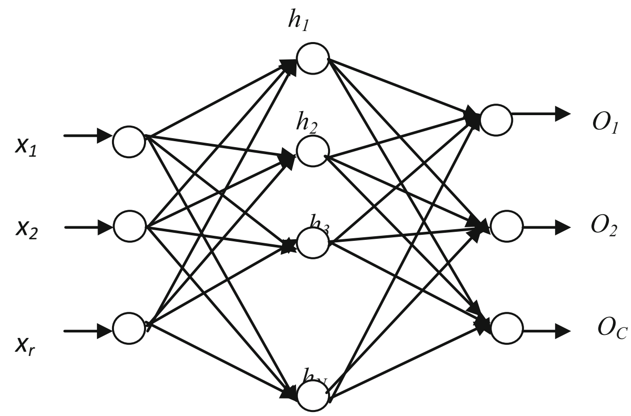

Fig. 1 Architecture of SLFN-ELM network [9]

$x_i = [x_1, x_2, \ldots\ldots.. x_n]$ and $t_i = [t_1, t_2, \ldots\ldots.. t_n]$ are input vector and target vector, respectively. The output of hidden layer corresponding to input pattern $x \in R^n$ is given using Eq. (1).

$$h_i = g(w_{ir}^T \cdot x_i + b_i) \tag{1}$$

where, $w_{ir} = [w_{i1}, w_{i2}, \ldots\ldots.. w_{in}]$is the weight vector which connects the rth input unit to ith hidden unit.

And $b_i = [b_1, b_2, \ldots\ldots.. b_n]$ represents the biases for hidden neurons. $g(.)$ denotes the adaptive non-symmetric FAF s, at hidden layer, which maps nonlinearly the data from the input space to feature space. The attraction of this FAF is that it maps the outliers (the variation generated due to varying illumination, pose, expression, occlusion, etc.) to a concise range of membership value on either end, near to zero and one membership value. For average grayscale values, the mapping is around linear which is controlled by the fuzzifier f. Also, s FAF is non-symmetric in nature which offers better performance compared to symmetric activation function [5] which has been experimentally proved in Sect. 5.

In the present investigation, an adaptive non-symmetric s FAF is used for mapping the input space into the feature space. Figure 2 shows the shape of this type of membership function [47,48]. The s FAF is defined using Eq. (2).

$$g(x, \alpha, \beta) = \begin{Bmatrix} 0 & x \le \alpha \\ 2^{f-1}\left(\frac{x-\alpha}{\beta-\alpha}\right)^f & \alpha \le x \le m \\ 1 - 2^{f-1}\left(\frac{x-\beta}{\beta-\alpha}\right)^f & m \le x \le \beta \\ 1 & x \ge \beta \end{Bmatrix} \tag{2}$$

The gradient of the s FAF is controlled by the parameter f, called fuzzifier. The value of fuzzifier f, which provides the optimum results, is obtained experimentally in Sect. 5. The FAF has two crossover points, α and β at 0 (minimum) and 1 (maximum) value, respectively, and center m at 0.5

(mean). α is calculated by $\min(x)$ β is calculated by $\max(x)$ and center m is calculated by $mean(x)$. When the feature values are away from the center in left side, its value gradually decreases and attains 0 at the crossover point α and when the feature values are away from center in right side, its value gradually increases and assign 1 at the crossover point β. For average grayscale values, the assignment of membership values is approximately linear.

After getting the hidden layer output, h_i the jth output corresponding to the i input unit is given using Eq. (3).

$$O_{ij} = h_i \beta_i \tag{3}$$

where $\beta_i = [\beta_1^T, \beta_2^T \ldots\ldots\ldots \beta_c^T]$ is the output weight vector which connects the ith hidden unit to the jth output unit.

Equation (3) can be compactly written as,

$$O = H\beta \tag{4}$$

where $H = [h_1, h_2, \ldots\ldots\ldots h_n]^T$ is the output vector of hidden layer which maps the data from the input space to fuzzy feature space.

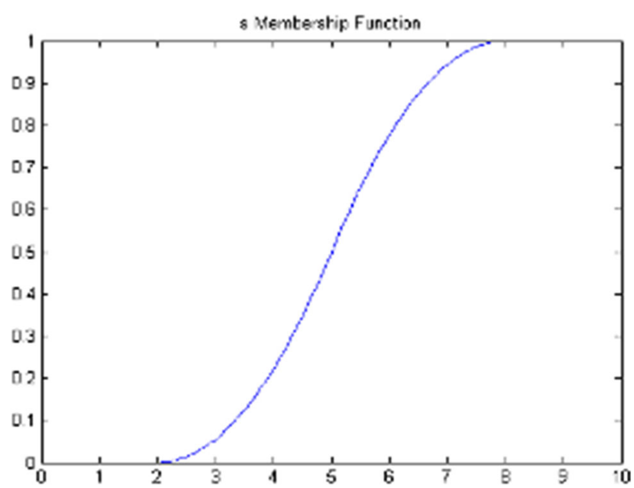

Fig. 2 s-Fuzzy activation function [10]

For improving the performance of the face recognition system, the training error should be minimized. To minimize the training error, Eq. (5) should be satisfied.

$$\|H\beta - T\| = 0 \tag{5}$$

which can be written in the generalized form using Eq. (6)

$$H\beta = T \tag{6}$$

Where $T = [t_1, t_2, \ldots\ldots\ldots t_n]^T$ represents the target vector corresponds to C classes. The smallest norm least -square solution of Eq. (6) is given using Eq. (7)

$$\hat{\beta} = H^{-1}T \tag{7}$$

where H^{-1} represents pseudo-inverse or Moore–Penrose inverse of matrix H. Also, other techniques can be used to calculate the inverse of matrix H which are: singular value decomposition (SVD), iterative method, orthogonal normalization method, orthogonal projection method, etc.

4 Brief Overview of Face Databases

In this section, a brief overview of AT&T [49], Yale faces [50], CMU PIE [51] and UMIST [52] face database used in the present study is described.

AT&T Database AT&T database contains 400 grayscale images of 40 individuals in *.pgm* format [49]. Each individual has 10 images. The images in this database include varying facial expressions open and closed eyes, smiling and not smiling faces and facial details with and without glasses. The resolution of each image is 112 × 92, and 256 gray levels. The pixel values of the images are normalized between −1 and +1. Further, no any normalization is done on these images. Images of this database contains scale variation up to 10%. Figure 3 shows the 10 sample face images of a same

Fig. 3 Sample face images of the same person of AT&T database [49]

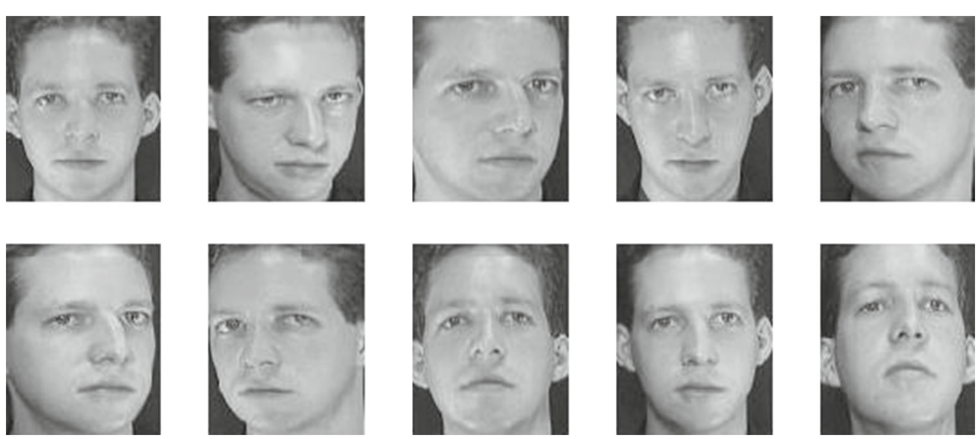

Fig. 4 Sample face images of the same person from Yale face database [50]

person. In the present study, the number of units in input layer is set to 10,304, number of neurons in hidden layer is randomly selected to 1500, and 40 units in output layer are set in AT&T database.

Yale Faces Database The Yale face database consists of total 165 grayscale images of 15 individuals, 11 images of each individual, in *.giff* format [50]. This database contains illumination variations (*eg*. left/right/center light) and varying facial expressions (smile, sad expression, open/closed mouth). The size of the original image is 243×320. In the present investigation, each image is resized to 122×160 pixel and normalized between -1 to $+1$. Figure 4 shows the 10 images of a particular individual of this database. In this database, the values chosen for r, N and C in this work are 19,520, 1500 and 15, respectively.

CMU_PIE Database CMU PIE database consists of 68 subjects with the total of 41,368 images containing, variations in pose, illumination and expressions (PIE) [51]. In the present study, only images with illumination variations are considered. There are 21 images per subject under illumination variations. The images of this database are cropped to include only the face with little hair and background. The original size of images after cropping is 239×197 and further resized to have the size of 119×98. The database images are colored images in *.jpeg* format. These images are converted to grayscale images and normalized between -1 and +1 values for experimental studies. Figure 5 shows the example of 21 images of a particular subject of this database. In the present study of classification, the parameters r, N and C for CMU PIE database are set equal to 11,662, 2000 and 68, respectively.

UMIST database UMIST database contains total 564 face images of 20 subjects [52]. The number of images per subject is not fixed, and it varies from 19 to 36. The size of each image is 92×112 and 256 shades of gray. Each subject consists in a wide range of poses variations from profile to frontal views. It also contains the variations of race, sex and appearance. Figure 6 shows the 10 images of a particular subject for this database. In the present work, first 19 face images per person are selected to form a new sub-database of 380 images. The parameters for proposed architecture for UMIST database are set equal to 10,304, 1500 and 20 for r, N and C, respectively.

Four databases are used in our experimentation. The next section details the promising results and discussion.

5 Results and Discussions

This section presents the significance of the present work, ANF-ELM, for the face recognition in terms of percentage error rate and learning speed on different databases. Studies are conducted on four different databases, AT&T [49], Yale faces [50], CMU PIE [51] and UMIST [52]. The optimum value of the parameter fuzzifier f is calculated for each database. The results of the present investigation on these databases are compared with conventional ELM with four other activation functions. ANF-ELM is also compared with other state of the art techniques on these databases to show the significance of the present work. To establish the improvements in the learning speed of the present investigation, the training time is also compared with conventional back-propagation network.

The studies have been conducted with different size of training set. The number of training images per person is varied, and fixed number of images are used for testing. For evaluating the efficacy of the present investigation, percentage error rate is used as the performance metric. Error rate signifies the percentage of images that are not correctly recognized by the classification scheme for the respective classifier.

Error rate (ER) for a face recognition system is calculated using Eq. (8)

$$ER = \frac{e}{t} \tag{8}$$

where t is total number of test images of all the subjects in a database, and e is the number of images out of these which are not correctly recognized. Percentage error rate is calculated by multiplying error rate by 100.

Fig. 5 Sample face images of the same person from CMU PIE database [51]

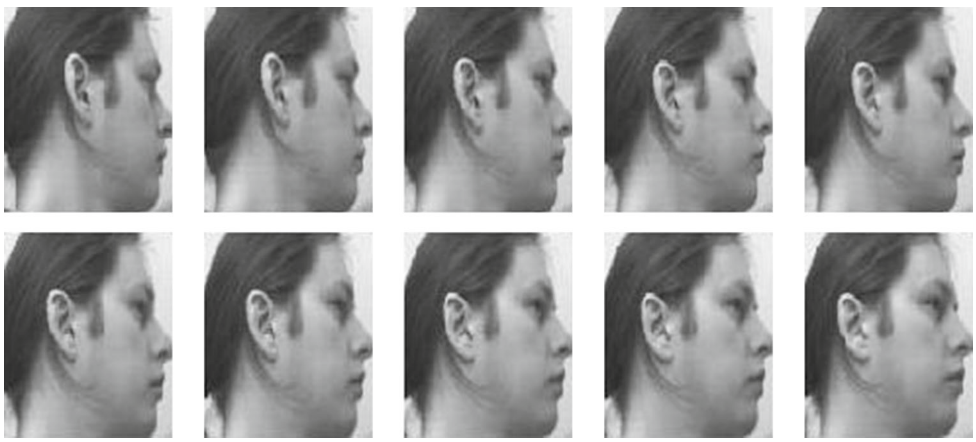

Fig. 6 Sample face images of the same person from UMIST database [52]

Acceptance rate (AR) for a face recognition system is computed from the error rate using Eq. (9)

$$\text{Acceptance Rate} = 100 - \text{Error Rate} \tag{9}$$

Therefore, acceptance rate has not been shown separately in the results and discussions part. For a particular method of classification, low error rate signifies high acceptance rate.

In the present study, *s* FAF is used which is non-symmetric in nature with shifted origin, as it gives better performance compared to a symmetric activation function. If *s* FAF have origin in the center, the error rate on Yale face database with number of training image equal to 1 is 45.6%, while for shifted origin *s* FAF, the error rate for the same number of training image is equal to 40%. Table 1 shows the comparison of *s* FAF with other non-symmetric FAF on Yale face database for number of training images per subject equal to 5. On Yale face database, error rate using *s* FAF is 8.66%, whereas error rate is 12.9, 12.3, and 21.78% for *pi*, *tri*, and *gauss* FAF, respectively. It is obvious from the same that non-symmetric FAF achieves lower error rate as compared to symmetric FAF.

Table 1 Error rate comparison of symmetric and non-symmetric FAF on Yale face database

S. no.	Fuzzy activation function used	Error rate (%)
1.	***s*-FAF**	**8.66 ± 0.99**
2.	*pi*-FAF	12.9 ± 1.32
3.	*tri*-FAF	12.3 ± 1.45
4.	*gauss*-FAF	21.78 ± 3.0

Bold indicates the results of the present algorithm, Fuzzy based extreme learning machine for face recognition on different databases. These results indicates the best results for recognition compared to conventional algorithms for face recognition

The performance of ANF-ELM also depends on the steepness of *s* FAF which is controlled by the parameter fuzzifier f. Studies are performed for different values of f to find its optimum value. Table 2 shows the error rate on Yale face database for different values of f for number of training images per subject equal to 5. The rest of the images are taken for the testing purpose, and there is no overlapping between training images and test images. It is obvious from Table 2 that the minimum error rate when fuzzifier f is equal to 2 is 8.66%

Table 2 Error rate on Yale face database for different values of fuzzifier f

S. no.	Value of f	Error rate (%)
1.	2	8.66 ± 0.99
2.	3	9.33 ± 0.99
3.	4	9.55 ± 0.60
4.	5	10.22 ± 0.49
5.	6	10.66 ± 0.60

with standard deviation 0.99%. Therefore, in the rest work, parameter f is chosen as equal to 2.

In the present work, experiments are carried out in three different approaches. In first approach, the comparison of ANF-ELM with ELM algorithm using different activation function at hidden layer nodes is presented. Comparisons are done with the following activation function in the hidden layer with ANF-ELM: sigmoid (*sig*), hyperbolic tangent (*tanh*), log sigmoid (*logsig*), and tan sigmoid (*tansig*). In second approach, the comparison of present work is done with other conventional algorithms for face recognition, such as BP with gradient descent algorithm [6], principal component analysis (PCA) [12], Fuzzy PCA [43], conventional ELM [9], Kernel ELM [37], PCA-ELM [44] Sparse representation method [45], SLFN_BVOI [46]. In last approach, comparison of learning speed of given ANF-ELM with BP algorithm is presented.

In this investigation, the number of image of training for each database is varied and the number of testing images is kept fixed. For each database, the number of training images is varied from 1 to 4 number of training images per subject and number of the images are kept fixed for testing purpose. In AT&T database, the number of testing images per subject are fixed to 6 per subject, 7 for Yale faces database, 17 for CMU PIE database and 15 for UMIST face database. As the number of training images is increased from 1 to 4, the error rate will decrease as the chance to recognize the person will increase.

Comparison with ELM Algorithm Using Different Activation Function

Table 3 depicts the percentage error rate comparison for all the four databases using ANF-ELM with different activation function for ELM algorithm. The comparisons are shown for 'sig', 'tanh', 'tansig' and 'logsig' activation function of ELM with the adaptive non-symmetric fuzzy activation function of ELM. Experimental results clearly justify the improvement in error rate by using an adaptive non-symmetric fuzzy activation function. As the number of training images are varied from 1 to 4, percentage error rate for each database will be decreasing. For AT&T database, error rate improvement is up to 20% using the proposed approach. For Yale face database, improvement in percentage error rate is up

Table 3 Error rates comparison for different ELM activation function with fuzzy activation function

Databases	AT&T				Yale				CMU PIE				UMIST			
% Error rate	Training images used				Training images used				Training images used				Training images used			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
ELM-sig	31.5	17.67	11.5	10.75	46.85	28.95	11.69	10.3	41.93	41.17	26.42	18.62	51.0	47.33	39.17	35.17
ELM-tanh	32.17	19.0	11.83	10.91	45.71	31.04	12.57	10.28	40.10	36.00	19.25	15.64	51.33	46.00	39.67	35.17
ELM-tansig	33.33	18.33	12.08	11.08	46.09	27.62	12.38	10.47	44.78	32.49	25.22	15.46	48.33	44.55	39.33	34.17
ELM-logsig	29.83	19.0	11.46	10.0	46.09	29.90	13.14	10.85	39.84	36.0	23.84	17.42	52.5	45.5	40	38.50
ANF-ELM	**29.58**	**16.67**	**9.67**	**9.5**	**40.19**	**26.67**	**10.47**	**10.28**	**3.35**	**0.92**	**0.80**	**0.49**	**36.83**	**31.67**	**28.67**	**28.67**

Bold indicates the results of the present algorithm, Fuzzy based extreme learning machine for face recognition on different databases. These results indicates the best results for recognition compared to conventional algorithms for face recognition to 20.32%. For CMU PIE database, the percentage error rate improvement is up to 99.73% for number of training images equal to 4. This result shows the best performance results for CMU PIE database. For UMIST database, the percentage error rate improvement is up to 33.08% which shows the utility of the present approach for classification. The mean value for percentage reduction in error rates for AT&T database is 10, 13, 14.11, and 8.5%, when the present approach, ANF-ELM is compared with ‘sig’, ‘tanh’, ‘tansig’ and ‘logsig’ activation function of ELM, respectively. The mean value for percentage reduction in error rates for Yale face database are 6, 11, 8.5, and 12.4%, when ANF-ELM is compared with ‘sig’, ‘tanh’, ‘tansig’ and ‘logsig’ activation function of ELM, respectively. The mean value for percentage reduction in error rates for CMU PIE face database are 96, 95.75, 97, and 96%, when the proposed approach, ANF-ELM is compared with ‘sig’, ‘tanh’, ‘tansig’ and ‘logsig’ activation function of ELM, respectively. The mean value for percentage reduction in error rates for UMIST face database is 27, 27, 24 and 28.5%, when ANF-ELM is compared with ‘sig’, ‘tanh’, ‘tansig’ and ‘logsig’ activation function of ELM, respectively.

Comparison with Other Algorithms for Face Recognition

Table 4 depicts the percentage error rate variation for different number of training images for all databases using conventional classifiers and the ANF-ELM classifier for face recognition. The corresponding performance graphs are shown in Fig. 7. The comparison has been done with BP with gradient descent algorithm [6], conventional ELM [9], PCA [12], Kernel ELM [37], Fuzzy PCA [43], PCA-ELM [44] Sparse representation [45] and SLFN_BVOI [46]. For experimental studies, back-propagation algorithm [6] uses *traingdx* training function [53] which updates the biases and weights of the network using gradient descent momentum training and adaptive learning rate. In ELM [9] learning algorithm, sigmoid activation function is used for all the databases. In PCA [12], N number of largest principal components have been chosen for classification. In fuzzy PCA [43], the π membership function is used for fuzzification. In kernel ELM [37], radial basis function (RBF) kernel function is used for hidden neurons of ELM. In PCA-ELM [44], first N largest principal components are extracted which are used for classification by ELM using sigmoid activation function.

For AT&T database [49], in BP [6], the numbers of the hidden neurons are selected equal to 70 with *traingdx* training function [53], which updates the bias and weights of the network using gradient descent momentum with adaptive learning rate. In ELM [9], sigmoid activation function is used for the hidden layer with 1500 neurons nodes at the hidden layer. In PCA [12], 100 largest principal components have been chosen for classification. In fuzzy PCA [43], the π membership function is used for fuzzification. In kernel

Table 4 Error rates comparison for different ELM activation function with fuzzy activation function

Databases	AT&T				Yale				CMU PIE				UMIST			
% Error rate	Training images used				Training images used				Training images used				Training images used			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
BP using gradient descent [6]	38.5	17.5	16.00	13.5	57.62	32.38	20.00	12.38	48.75	35.5	18.90	11.94	59.00	54.00	50.66	38.33
PCA [12]	33.33	25.42	18.75	15.42	52.38	28.57	12.38	10.52	39.84	37.88	30.65	25.57	52	45.67	41.00	42.33
Fuzzy PCA [43]	30.08	17.5	14.58	11.67	56.19	26.67	13.33	12.38	35.83	33.87	27.63	28.52	56.33	54.33	49.33	41.00
ELM [9]	31.5	17.67	11.5	10.75	46.85	28.95	11.69	9.3	41.93	41.17	26.42	18.62	51.00	47.33	39.17	35.17
Kernel ELM [37]	32.08	20	14.17	10.80	51.42	32.38	11.42	11.42	40.10	38.68	33.78	26.38	51.00	46.33	42.33	39.00
PCA-ELM algorithm [44]	30.21	20.58	13.72	15.33	47.23	30.28	13.71	12.38	35.29	27.45	13.81	11.41	58.00	53.00	48.00	46.00
Sparse representation method [45]	32.50	18.52	11.97	8.43	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
SLFN_BVOI [46]	N.A.*	N.A.	N.A.	N.A.	46.0	24.81	12.50	12.38	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
ANF-ELM	**29.58**	**16.67**	**9.67**	**9.5**	**40.19**	**26.67**	**10.47**	**10.28**	**3.35**	**0.92**	**0.80**	**0.49**	**36.83**	**31.67**	**28.67**	**28.67**

Bold indicates the results of the present algorithm, Fuzzy based extreme learning machine for face recognition on different databases. These results indicates the best results for recognition compared to conventional algorithms for face recognition

N.A. stands for not available. The results on other databases are not available

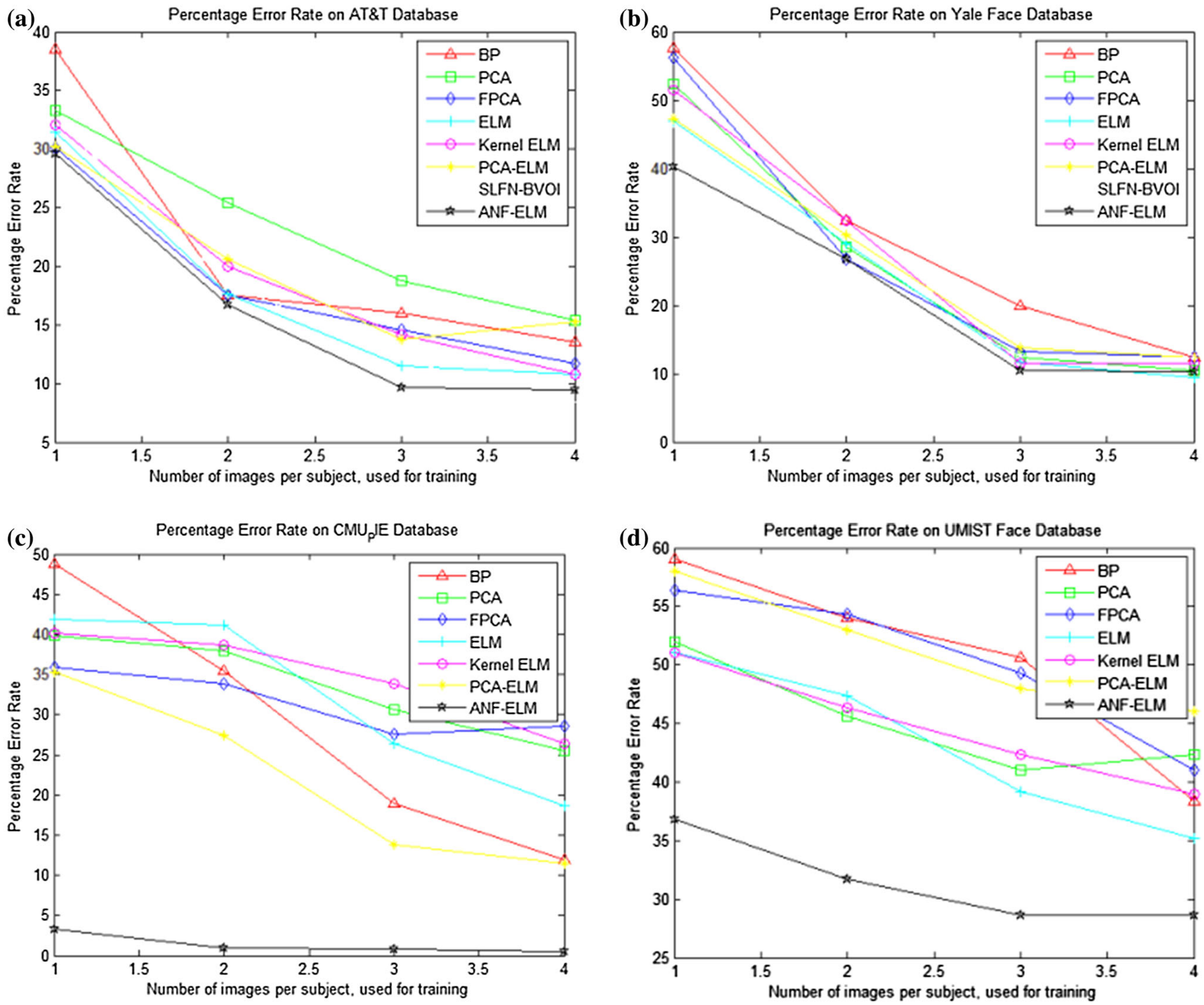

Fig. 7 **a** Percentage error rate variations on AT&T database. **b** Percentage error rate variations on Yale face database. **c** Percentage error rate variations on CMU_PIE database. **d** Percentage error rate variations on UMIST face database

ELM [37], RBF kernel parameter is set to 300 for hidden neurons of ELM. In PCA-ELM [44],100 largest principal components are extracted and given to ELM classifier which uses the sigmoid activation function. For the present algorithm ANE-ELM classifier, the number of units (r) in input layer is set to 10,304, number of hidden layer neurons (N) is set to 1500, and output layer units (C) are set to 40. Percentage error rate reduction using ANF-ELM is up to 48.42%, which shows the significance of the present approach. The mean value for percentage reduction in error rates is 25, 33, 16, 10, 17.25 and 27%, when the present approach, ANF-ELM is compared with BP, PCA, Fuzzy PCA, ELM, KELM and PCA-ELM, respectively.

In Yale face database [50], 7 numbers of images per subject is fixed for testing purpose and error rate is calculated for varying the number of training images from 1 to 4 per subject. For BP [6], the number of hidden neurons are selected equal to 70. Sigmoid activation function is used for ELM [9] algorithm with hidden layer with 1500 neurons at hidden layer. One hundred and fifty largest principal components have been chosen in PCA [12] algorithm for classification. π membership function is used for fuzzification in fuzzy PCA [43]. In kernel ELM [37], RBF kernel parameter is set to 300 for hidden neurons of ELM. In PCA-ELM [44], 150 largest principal components are extracted and given to ELM classifier which uses the sigmoid activation function. For the present algorithm ANE-ELM classifier, the values chosen for r, N and C for the proposed architecture are 19,520, 1500 and 15, respectively. Percentage error rate reduction with these face recognition techniques shows the significance of the present approach. Reduction in percentage error rate using ANF-ELM learning fluctuates up to 28.47% . The average value of reduction in error rate using ANF-ELM learning algorithm is 30, 12, 16, 6, 14.5 and 12.75% compared with that of BP, PCA, Fuzzy PCA, ELM, KELM and PCA-ELM, respectively.

In CMU PIE database [51], percentage error rate is calculated by varying the number of training images from 1 to 4 per subject and keeping the number of test images fixed to 17 per subject. For BP [6], the number of hidden neurons are selected equal to 500, as the size of database is large. ELM algorithm [9] with sigmoid activation function uses 3000 neurons at hidden layer. In PCA algorithm [12], 150 largest principal components have been chosen for classification. In fuzzy PCA [43], π membership function is used for fuzzification. In kernel ELM [37], RBF kernel parameter is set to 500 for hidden neurons of ELM. In PCA-ELM [44], 150 largest principal components are extracted and given to ELM classifier which uses the sigmoid activation function. For the proposed classifier, ANF-ELM, the parameters r, N and C for CMU PIE database are set equal to 11,662, 3000, and 68, respectively. Reduction in percentage error rate using present investigation is up to 98.08%. The average value of reduction in error rate using ANF-ELM learning algorithm is 95.25, 96, 96, 96.25 and 94% compared with that of BP, PCA, Fuzzy PCA, ELM, KELM and PCA-ELM, respectively.

In UMIST database [52], comparison of percentage error rate variations with varying number of training images from 1 to 4 for each subject, keeping the number of test images fixed to 15 per subject is shown. For BP [6], the number of hidden neurons are selected equal to 70. ELM [9] algorithm with sigmoid activation function uses 1500 neurons at hidden layer. In PCA algorithm [12], 100 largest principal components have been chosen for classification. In fuzzy PCA [43], π membership function is used for fuzzification. In kernel ELM [37], RBF kernel parameter is set to 200 for hidden neurons of ELM. In PCA-ELM [44],100 largest principal components are extracted and given to ELM classifier which uses the sigmoid activation function. The parameters for proposed architecture are set equal to 10,304, 1500 and 20 for r, N and C, respectively. Reduction in percentage error rate using present investigation is up to 43.41%. The average value of reduction in error rate using ANF-ELM learning algorithm is 37, 31, 37, 26.6, 30 and 39% compared with that of BP, PCA, Fuzzy PCA, ELM, KELM and PCA-ELM, respectively.

The reasons for best classification accuracy of ANF-ELM classifier are: (1) The number of neurons in hidden layer is very much higher than a number of hidden neurons in neural network trained using back propagation; (2) The weights and biases from the input to the hidden layer neurons are not trained like in back-propagation algorithm. Rather they are randomly initialized, if the activation functions in the hidden layer are infinitely differentiable and are fixed thereafter; (3) Use of adaptive non-symmetric s FAF maps the outliers (the

Table 5 Comparison of training time (in seconds) using BP and ANF-ELM training algorithms

Data bases	AT&T				Yale				CMU PIE				UMIST			
Training time (in sec)	Training images used				Training images used				Training images used				Training images used			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Training time using BP	256.9	264.7	324.4	393.6	426.2	441.4	348.8	465.3	257.2	235.3	198.8	214.7	103.8	130.3	205.0	117.7
Training time using ANF-ELM	**1.33**	**1.63**	**2.07**	**2.25**	**2.55**	**2.91**	**3.4**	**3.75**	**5.97**	**7.87**	**10.29**	**12.39**	**0.97**	**1.05**	**1.26**	**1.42**

Bold indicates the results of the present algorithm, Fuzzy based extreme learning machine for face recognition on different databases. These results indicates the best results for recognition compared to conventional algorithms for face recognition variation generated due to varying illumination, pose, expression, occlusion, etc.) to a concise range of membership value on either end, near to zero or one membership value which further improves the classification accuracy for face recognition.

Comparison of Learning Speed

ANF-ELM also gives fast learning in comparison with BP [6] because it does not require any iterative learning of network parameters for learning. Table 5 shows the comparison of training time using ANF-ELM and BP algorithm on all four database. For AT&T database, training time ratio of BP algorithm to that of ANF-ELM approach varies from 156.7 to 193.2 s with mean value of 172 sec. Training time ratio of BP algorithm to that of ANF-ELM approach for Yale face database varies from 102.6 to 167.2 s with the mean value 136.25 sec. For CMU PIE database, training time ratio for BP and ANF-ELM approach varies from 17.4 to 43.08 s with the mean value 28 sec. For UMIST database, training time ratio of BP algorithm to that of ANF-ELM approach varies from 82.9 to 162.7 s with mean value of 120 s. These results confirm the fast learning speed of ANF-ELM classifier compared to conventional algorithm BP, used for face recognition.

6 Conclusions

In the present investigation, a non-iterative approach, ANF-ELM for training single hidden layer neural network for face recognition is presented. ANF-ELM approach takes advantage of both, an adaptive, non-symmetric, FAF for nonlinear mapping of input space into fuzzy feature space and ELM for fast learning speed and better generalization results. In ELM learning algorithm, there is no requirement of iterative tuning of network parameters, like in traditional learning algorithm, BP. Therefore, the present approach, ANF-ELM for learning offers faster learning and better recognition results. The present work is investigated on four databases, AT&T, Yale face, CMU PIE and UMIST databases. Experimental outcomes show the significant performance improvement using ANF-ELM neural network over existing state of the art learning algorithms. The present algorithm is computationally efficient and can be also used in real-time automatic face recognition system.

The present work can be further extended to use nonlinear feature extraction before classification of the face images. Nonlinear feature extraction will extract the higher order statistics for efficiently extracting the features of the face image.

Acknowledgements The authors would like to express their sincere thanks to AT&T laboratories, Cambridge, Yale University, CMU and Image Engineering Laboratories for the use of AT&T, Yale, CMU PIE and UMIST face databases.

References

1. Chellappa, R.; Wilson, C.L.; Sirohey, S.: Human and machine recognition of faces: a survey. Proc. IEEE **83**(5), 705–740 (1995)
2. Zhao, W.; Chellappa, R.; Phillips, P.J.; Rosenfeld, A.: Face recognition: a literature survey. ACM Comput. Surv. **35**(4), 399–458 (2003)
3. Li, S.Z.; Jain, A.K.: Handbook of Face Recognition. Springer, Berlin (2005)
4. Zou, X.; Kittler, J.; Messer, K.: Illumination invariant face recognition: a survey. In: Proceedings of 1st IEEE International Conference on Biometric: Theory, Application and Systems, pp. 1–8 (2007)
5. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall, Englewood Cliffs (1999)
6. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, New York (1995)
7. Intrator, N.; Reisfeld, D.; Yeshurun, Y.: Face recognition using a hybrid supervised/unsupervised neural network. Pattern Recognit. Lett. **17**(1), 67–76 (1996)
8. Lin, S.H.; Kung, S.Y.; Lin, L.J.: Face recognition/detection by probabilistic decision-based neural networks. IEEE Trans. Neural Netw. **8**(1), 114–132 (1997)
9. Huang, G.B.; Zhu, Q.Y.; Siew, C.K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of International Joint Conference on Neural Networks (Budapest, Hungary), **2**(25–29), 985–990 (2004)
10. Haddadnia, J.; Ahmadi, M.: N-feature neural network human face recognition. Image Vis. Comput. **22**(12), 1071–1082 (2004)
11. Brunelli, R.; Poggio, T.: Face recognition: features versus templates. IEEE Trans. Pattern Anal. Mach. Intell. **15**(10), 1042–1053 (1993)
12. Turk, M.; Pentland, A.: Eigenfaces for recognition. J. Cogn. Neurosci. **3**(1), 71–96 (1991)
13. Wang, Z.; Li, X.: Face recognition based on improved PCA reconstruction. Intell. Control Autom. **8**, 6272–6276 (2010)
14. Huang, S.M.; Yang, J.F.: Improved principal component regression for face recognition under illumination variations. IEEE Trans. Signal Process. Lett. **19**(4), 179–182 (2012)
15. Bartlett, M.S.; Movellan, J.R.; Sejnowski, T.J.: Face recognition by Independent component analysis. IEEE Trans. Neural Netw. **13**(6), 1450–1464 (2002)
16. Etemad, K.; Chellapa, R.: Discriminant analysis for recognition of human face images. J. Opt. Soc. Am. **14**(8), 1724–1733 (1997)
17. Shu, X.; Gao, Y.; Lu, H.: Efficient linear discriminant analysis with locality preserving for face recognition. Pattern Recognit. **45**(5), 1892–1898 (2012)
18. Belhumeur, P.N.; Hespanha, J.P.; Kriegman, D.J.: Eigenfaces versus fisherfaces: recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 711–720 (1997)
19. Comon, P.: Independent component analysis, a new concept. Signal Process. **36**(3), 287–314 (1994)
20. Delac, K.; Grgic, M.; Grgic, S.: Independent comparative study of PCA, ICA, and LDA on the FERET data set. Int. J. Imaging Syst. Technol. **15**(5), 252–260 (2005)
21. Penev, P.S.; Atick, J.J.: Local feature analysis: a general statistical theory for object representation. Comput. Neural Syst. **7**, 477–500 (1996)
22. Wiskott, L.; Fellous, J.M.; Kruger, N.; Malsburg, C.V.: Face recognition by elastic graph matching. IEEE Trans. Pattern Recognit. Mach. Intell. **1**(7), 775–779 (1997)
23. Bishop, C.M.: Neural Networks for Pattern Recognition‘. Oxford University Press, New York (1995)
24. LeCun, Y.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.; Hubbard, W.; Jackel, L.: Back propagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989)

25. Lawrence, S.; Giles, C.L.; Tsoi, A.C.; Back, A.D.: Face recognition: a convolutional neural network approach. IEEE Trans. Neural Netw. **8**(1), 98–113 (1997)
26. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Berlin (1995)
27. Fleming, M.; Cottrell, G.: Categorization of faces using unsupervised feature extraction. In: Proceedings of International Joint Conference on Neural Network, **90**(2), 65–70 (1990)
28. Wright, J.; Yang, A.; Ganesh, A.; Sastry, S.; Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 1–18 (2009)
29. Mi, J.X.; Huang, D.S.; Wang, B.: The nearest-farthest subspace classification for face recognition. Neurocomputing **113**, 241–250 (2013)
30. Huang, G.B.; Zhu, Q.Y.; Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing **70**(1–3), 489–501 (2006)
31. Feng, G.; Huang, G.B.; Lin, Q.; Gay, R.: Error minimized extreme learning machine with growth of hidden nodes and incremental learning. IEEE Trans. Neural Netw. **20**(8), 1352–1357 (2009)
32. Huang, G.; Huang, G.B.; Song, S.; You, K.: Trends in extreme learning machines: a review. Neural Netw. **61**(1), 32–48 (2015)
33. Zhu, Q.Y.; Huang, G.B.; Siew, C.K.: A fast constructive learning algorithm for single hidden layer neural networks. In: Proceedings of IEEE International Conference on Control, Automation, Robotics and Vision, pp. 1907–1911 (2004)
34. Li, M.B.; Er, M.J.: Nonlinear system identification using extreme learning machine. In: IEEE International Conference on Control, Automation, Robotics and Vision, **6**, 1–4 (2006)
35. Liang, N.Y.; Huang, G.B.; Saratchandran, P.; Sundararajan, N.: A fast and accurate on-line sequential learning algorithm for feed-forward networks. IEEE Trans. Neural Netw. **17**(6), 1411–1423 (2006)
36. Huang, G.B.; Ding, X.; Zhou, H.: Optimization method based extreme learning machine for classification. Neurocomputing **74**, 155–163 (2010)
37. Huang, G.B.: An insight into extreme learning machines: random neurons, random features and kernels. Cogn. Comput. **6**(3), 376–390 (2014)
38. Bai, Z.; Huang, G.B.; Wang, D.; Wang, H.; Westover, M.B.: Sparse extreme learning machine for classification. IEEE Trans. Cybern. **44**(10), 1858–1870 (2014)
39. Tang, J.; Deng, C.; Huang, G.B.: Extreme learning machine for multilayer perceptron. IEEE Trans. Neural Netw. Learn. Syst. **27**(4), 809–821 (2016)
40. Ghosh, A.; Shankar, B.U.; Meher, S.K.: A novel approach to neuro-fuzzy classification. Neural Netw. **22**(1), 100–109 (2009)
41. Kwak, K.C.; Pedrycz, W.: Face recognition using a fuzzy fisher face classifier. Pattern Recognit. **38**, 1717–1732 (2005)
42. Yang, W.; Yan, H.; Wang, J.; Yang, J.: Face recognition using complete fuzzy LDA. In: Proceedings of 19th International Conference on Pattern Recognition, **19**, 1–4 (2008)
43. Vishwakarma, V.P.; Pandey, S.; Gupta, M.N.: Fuzzy based pixel wise information extraction for face recognition. Int. J. Eng. Technol. **2**(1), 117–123 (2009)
44. Castano, A.; Navarro, F.F.; Martinez, C.H.: PCA-ELM: a robust and pruned extreme learning machine approach based on principal component analysis. Neural Process. Lett. **37**(3), 377–392 (2013)
45. Xu, Y.; Zhu, Q.: A simple and fast representation-based face recognition method. Neural Comput. Appl. **22**(7), 1543–1549 (2013)
46. Vishwakarma, V.P.: A non-iterative learning based artificial neural network classifier for face recognition under varying illuminations. Commun. Comput. Inf. Sci. Contemp. Comput. **306**(5), 383–394 (2012)
47. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
48. Klir, G.J.; Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Application. Prentice Hall, New Jersey (1995)
49. AT&T face database, [Online]. Available: http://www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att_faces.tar.Z
50. Yale face database, [Online]. Available: http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html
51. Sim, T.; Baker, S.; Bsat, M.: The CMU pose, illumination, and expression database. IEEE Trans. Pattern Anal. Mach. Intell. **25**(12), 1615–1618 (2003)
52. UMIST Face Database, [Online]. Available: http://www.sheffield.ac.uk/eee/research/iel/research/face
53. MATLAB and Image Processing Toolbox Release (2010a), The MathWorks, Inc., Natick, Massachusetts, United States. retrieved on 05-05-2016