PERGAMON

Pattern Recognition 36 (2003) 1187–1202

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

A fuzzy hybrid learning algorithm for radial basis function neural network with application in human face recognition

Javad Haddadnia[a], Karim Faez[a], Majid Ahmadi[b,*]

[a]Electrical Engineering Department, Amirkabir University of Technology, Hafez Avenue, Tehran, 15914, Iran
[b]Electrical and Computer Engineering Department, University of Windsor, Windsor, Ont., Canada N9B 3P4

Received 9 May 2002; accepted 20 July 2002

Abstract

This paper presents a fuzzy hybrid learning algorithm (FHLA) for the radial basis function neural network (RBFNN). The method determines the number of hidden neurons in the RBFNN structure by using cluster validity indices with majority rule while the characteristics of the hidden neurons are initialized based on advanced fuzzy clustering. The FHLA combines the gradient method and the linear least-squared method for adjusting the RBF parameters and the neural network connection weights. The RBFNN with the proposed FHLA is used as a classifier in a face recognition system. The inputs to the RBFNN are the feature vectors obtained by combining shape information and principal component analysis. The designed RBFNN with the proposed FHLA, while providing a faster convergence in the training phase, requires a hidden layer with fewer neurons and less sensitivity to the training and testing patterns. The efficiency of the proposed method is demonstrated on the ORL and Yale face databases, and comparison with other algorithms indicates that the FHLA yields excellent recognition rate in human face recognition.

Keywords: RBF neural network (RBFNN); Face recognition; Fuzzy system; Learning algorithm

1. Introduction

Neural networks have been used in a number of applications such as remote sensing [1], dynamic modeling and medicine [2] and pattern classification [3–5]. Neural networks have been employed and compared to conventional classifiers. The results have shown that the performance of the neural network approaches is equal to, or slightly better than, other methods [6,7]. These properties make neural networks an attractive tool for many pattern classification problems. A central issue in neural networks is the problem of learning algorithm. The choice of learning algorithm, network topology, weight initialization and input signal presentations are important factors in the learning performance. In particular the choice of learning algorithm determines the rate of convergence and the suitability of the solution [8]. Recently radial basic function neural networks (RBFNNs) have been found to be very attractive for many engineering problems. An important property of the RBFNNs is that they form a unifying link among many different research fields such as function approximation, regularization, noisy interpolation and pattern recognition [11]. The increasing popularity of the RBFNNs is partly due to their simple topological structure, their locally tuned neurons and their ability to have a fast learning algorithm in comparison with other multilayer feed forward neural networks [7,9,10]. Jang demonstrated that under simple conditions an RBFNN could function like fuzzy inference system (FIS) [12,13]. The functional equivalence provides a shortcut to a better design for both RBFNNs and FISs [14,15]. The analysis and learning algorithms for RBFNNs are also applicable to FIS and the fuzzy modeling

[*] Corresponding author. Tel.: +1-519-253-3000, ext. 2576; fax: +1-519-971-3695.
E-mail addresses: javad@uwindsor.ca (J. Haddadnia), kfaez@aut.ac.ir (K. Faez), ahmadi@uwindsor.ca (M. Ahmadi).

0031-3203/03/$30.00 © 2002 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.
PII: S0031-3203(02)00231-5

procedure could be a good means of initializing an RBFNN before training. The use of fuzzy clustering for designing an RBFNN has been proposed in the literature [13,16,17].

There are several learning algorithms for RBFNN. A common learning algorithm is based on first choosing randomly a set of data points as RBF parameters and then using singular value decomposition to solve for the weights of the network. The orthogonal least-square (OLS) algorithm [18] can be employed as a forward regression procedure to select the RBF parameters. The forward selection algorithm (FSA) [19] assigns one RBF unit for each class under training provided that feature vectors of the patterns belonging to the same class are close to each other. In situations where the feature vector of the pattern is not satisfying the above condition an additional neuron is added to the hidden layer. In this training algorithm, one RBF unit is generated at a time through an iterative process until the sum of squared errors reaches the minimum. The Gaussian mixer model (GMM) [19] regards the basis function as the components of a mixture density model whose RBF parameters are to be optimized by maximum likelihood method. In GMM learning algorithm the number of hidden neurons is used as an input parameter to the model so that the unsupervised procedure optimizes the Gaussian parameters based on the training set. The RBF parameters are determined by tuning the mixture model with circular covariance using the expectation–maximization algorithm and the maximum inter-center square distance. The minimal resource allocating network (MRAN) [9] is a sequential minimal learning algorithm that adds one hidden neuron to the network based on the novelty of the input data under training. This training algorithm starts with no hidden neuron and grows by allocating a new hidden neuron based on the novelty of the observation in a sequential manner. If an input data has no novelty then the existing parameters of the network are adjusted. An optimal design of RBFNN for fuzzy-rule extraction in high-dimensional data has been proposed in Ref. [17]. Fuzzy clustering method was used to find the number of hidden neurons and RBF parameters are set by Teoplitz covariance matrix estimator and finally back propagation algorithm was used to determine the connection weights of the output layer. A supervised fuzzy clustering for the RBFNN training has been proposed in Ref. [20]. The learning methodology follows the Pedrycz's conditional fuzzy clustering where data agglomeration is based on both input and output pattern values. One drawback that is commonly encountered in this method is that it is regarded as supervised, therefore it suffers from the need for human interaction to determine classes and training regions.

The main objective of this paper is to present a novel and efficient learning algorithm for the RBFNN with applications in face recognition. The proposed learning algorithm takes advantage of the recent advances in optimal fuzzy clustering known as the fuzzy-C-mean (FCM) algorithm [21,22] to initialize the RBF parameters. The use of appropriate cluster validity indices with majority rule to define the number of hidden neurons is discussed. The adjustment of the parameters and the determination of the neural network connection weights are done via a hybrid method based on the linear least square and the gradient decent [23]. The analysis of the sensitivity of the proposed learning algorithm to select training and testing sets is studied. To show the usefulness of the FHLA, the proposed learning algorithm has been compared with other algorithms such as MRAN, GMM and FSA. In order to conduct a comparative study, the designed RBFNN with the above learning algorithms has been applied to the face recognition problem using 400 face images on the ORL database and 165 face images on the Yale database [4]. The feature extraction technique used in this study combines structural and statistical approaches, including the shape information presented in Ref. [24] and the principle component analysis (PCA) [25]. This study intends to demonstrate the effect of the FHLA on the number of required hidden neurons, the number of epochs required for training, the sensitivity to select training and testing sets and the overall recognition rate. Also the effect of the irrelevant data in face image on the recognition rate with different learning algorithms is studied. The paper is organized as follows: Section 2 gives a brief description of the RBFNN classifier. In Section 3, the proposed FHLA is presented. The face recognition method and the design of experiment are described in Section 4. Experimental results are presented in Section 5 and the conclusions are attained in Section 6.

2. Radial basis function neural network

The construction of the RBFNN involves an input layer, a hidden layer and an output layer with feed forward architecture. The input layer of this network is a set of n units, which accept the elements of an n-dimensional input feature vector. The input units are fully connected to the hidden layer with r hidden units. Connections between the input and the hidden layer have unit weights and, as a result, do not have to be trained. In this structure the hidden units are referred to as the RBF units. The goal of the RBF units is to cluster the data and reduce its dimensionality with a nonlinear transformation and to map the input data to a new space. The RBF units are also fully connected to the output layer. The output layer, which contains s units, implements a linear combination on this new space.

Consider the finite set of n-dimensional input feature vectors $X = \{x_1, x_2, \ldots, x_M\} \subset \Re^n$, and the prototype vectors $C = \{c_1, c_2, \ldots, c_r\} \subset \Re^n$. The overall input–output response of the RBFNN with r RBF units and s output units, is a mapping $\Re^n \rightarrow \Re^s$ described in Refs. [12,13,19,20,23,35] as follows:

$$y_j(x) = \sum_{i=1}^{r} w_{ij} R(\|x - c_i\|, \sigma_i), \tag{1}$$

where $y_j(x)$ is the response of the jth output unit to input x, w_{ij} is the connection weight of the ith RBF unit to the jth output unit, $R_i(.)$ is a real-valued activation function of the ith RBF unit and $\|.\|$ denotes the Euclidean norm. In this model, each RBF unit is described by two parameters, the center (c_i) and the width (σ_i). The activation function of the hidden units, $R(.)$, is radially symmetric in the input space and the output of each RBF unit is determined by the distance between the input vector x and the center parameter c_i of the RBF unit. There are several forms of activation functions. An axiomatic approach indicates that any activation function $R(.)$ must satisfy the following requirements [26]:

Axiom 1: The RBF activation function is a mapping $\mathfrak{R}^n \to \mathfrak{R}^+$, i.e.

$$\forall x, c, \sigma : R(\|x-c\|, \sigma) > 0. \tag{2}$$

Axiom 2: $R(.)\infty$ for all x and c.

Axiom 3: $R(.)$ is a monotonically decreasing function. This requires that

$$\forall x_i, x_k : \|x_i - c_j\|^2 < \|x_k - c_j\|^2$$
$$\Rightarrow R_j(\|x_i - c_j\|, \sigma_j) > R_j(\|x_k - c_j\|, \sigma_j). \tag{3}$$

Axiom 4: If $\partial R/\partial x_k$ denotes the gradient of $R(.)$ with respect to the corresponding input x_k, then:

$$\forall x_i, x_k : \|x_i - c_j\|^2 < \|x_k - c_j\|^2$$
$$\Rightarrow \frac{\|\partial R_j/\partial x_i\|^2}{\|x_i - c_j\|^2} > \frac{\|\partial R_j/\partial x_k\|^2}{\|x_k - c_j\|^2}. \tag{4}$$

Axiom 5: The gradient of $R(.)$ must satisfy the following condition:

$$\forall x_i, x_k : \|x_i - c_j\|^2 < \|x_k - c_j\|^2$$
$$\Rightarrow \|\partial R_j/\partial x_i\|^2 > \|\partial R_j/\partial x_k\|^2. \tag{5}$$

One of the most common activation functions for the RBF units is the Gaussian function, which satisfies all of the above axioms [19,20,23,26]. The Gaussian function is defined by

$$R_i(x) = \exp\left(-\frac{\|x - c_i\|^2}{\sigma_i^2}\right), \qquad i = 1, 2, \ldots, r, \tag{6}$$

where σ_i is the width of the RBF unit. Note that σ_i^2 represents the diagonal entries of the covariance matrix of the Gaussian function.

3. Fuzzy hybrid learning algorithm

The proposed FHLA method for designing and training the RBFNN classifier involves two steps: (1) RBFNN structure determination and initialization and (2) RBFNN parameters adjustment. In the first step, the RBFNN structure is determined and initialized by selecting of the number of neurons in the input, hidden and output layers. The center (c_i) and the width (σ_i) of the RBF units in the hidden layer are initialized with a good initial value. In the second step, the centers, widths and the output connection weights (w_{ij}) of the designed RBFNN are adjusted based on the gradient and the linear least-squared methods as an optimization procedure. The proposed two-step FHLA method for designing and training the RBFNN is presented in the following sections.

3.1. RBFNN structure determination and initialization

The RBFNN structure determination involves defining the number of neurons in the input, hidden and output layers. The number of feature vector elements sets the number of input neurons in the input layer while the number of classes determines the number of output neurons. The number of RBF units as well as their characteristic initializations is carried out using a fuzzy clustering technique. It is easy to verify that the performance of the RBFNN is strongly affected by the choice of c_i, σ_i and r (number of RBF units). In all learning algorithms the c_i, σ_i and r, could be obtained by an optimization procedure. However, most such procedures unfortunately suffer from local minimum. It is therefore recommended that the procedure be repeated for several initial points and the best local optimum be chosen.

The construction of the hidden layer in the proposed FHLA is based on three steps. (1) Determination of the number of RBF units by using a set of cluster validity indices. (2) The use of fuzzy clustering for defining the initial point for c_i. (3) Overlapping criteria between clusters determines the initial value for σ_i. Fig. 1 shows the flow chart of the RBFNN structure determination and initialization while the corresponding solutions for each step are presented in the following.

3.1.1. Fuzzy clustering and initialization RBF centers

In the RBFNN structure, a clustering technique associates a cluster to each RBF units. Bezdek introduced several clustering algorithms based on fuzzy set theory and an extension of the least-squares error criterion [21]. Most analytical fuzzy clustering approaches are derived from Bezdeck's FCM [21,22]. FCM is a data clustering algorithm that each data point is associated with a cluster through a membership degree. This technique partitions a collection of N_T data points into r fuzzy groups and finds a cluster center in each group, such that a cost function of a dissimilarity measure is minimized. The algorithm employs fuzzy partitioning such that a given data point can belong to several groups with a degree specified by membership grades between 0 and 1. A fuzzy r-partition of input feature vector $X = \{x_1, x_2, \ldots, x_{N_\mathrm{T}}\} \subset \mathfrak{R}^n$ is represented by a matrix $\mathbf{U} = [\mu_{ik}]$, where the entries satisfy the following constraints:

$$\mu_{ik} \in [0, 1], \qquad 1 \leqslant i \leqslant r, \qquad 1 \leqslant k \leqslant N_\mathrm{T}, \tag{7}$$

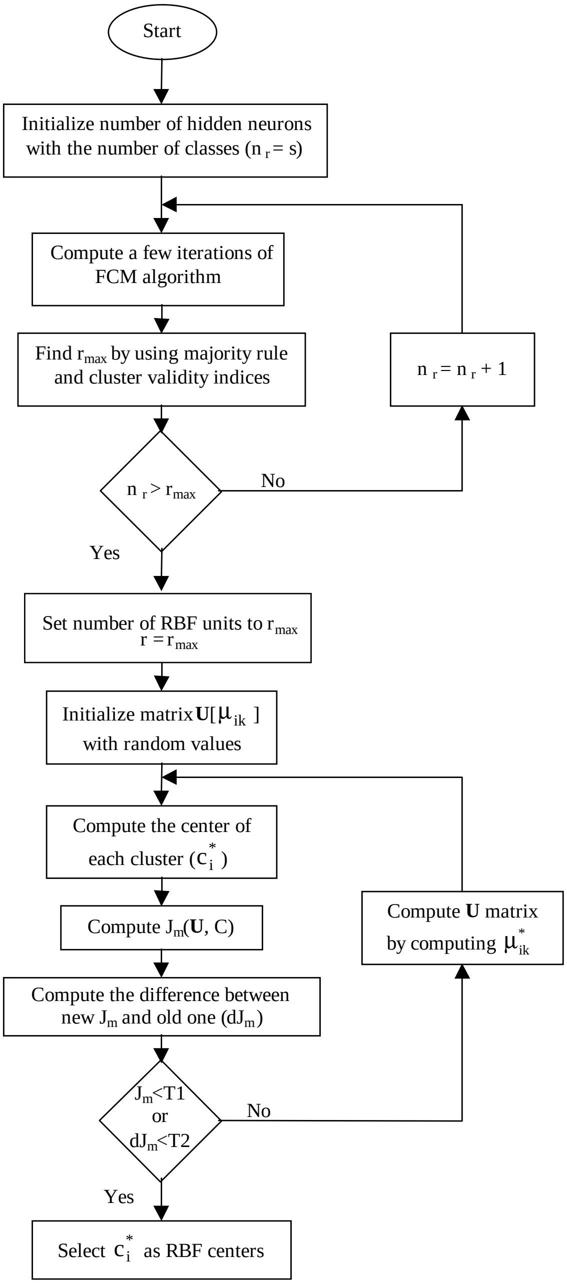
Start
Initialize number of hidden neurons with the number of classes (n r = s)
Compute a few iterations of FCM algorithm
Find r max by using majority rule and cluster validity indices
n r = n r + 1
n r > r max
No
Yes
Set number of RBF units to r max
r = r max
Initialize matrix U[μ ik]
with random values
Compute the center of each cluster (c* i)
Compute J m(U, C)
Compute U matrix by computing μ* ik
Compute the difference between new J m and old one (dJ m)
J m<T1
or
dJ m<T2
No
Yes
Select c* i as RBF centers

Fig. 1. The RBFNN structure determination and initialization with the FHLA method.

$$\sum_{i=1}^{r} \mu_{ik} = 1, \qquad 1 \leqslant k \leqslant N_{\mathrm{T}} \tag{8}$$

$$0 < \sum_{k=1}^{N_{\mathrm{T}}} \mu_{ik} < N_{\mathrm{T}}, \qquad 1 \leqslant i \leqslant r. \tag{9}$$

$\mathbf{U}$ can be used to describe the cluster structure of X by interpreting μ_{ik} as the degree of membership of μ_{ik} to cluster i. A proper partition $\mathbf{U}$ of X may be defined by the minimization of the following objective function [21]:

$$J_m(U,C) = \sum_{k=1}^{N_{\mathrm{T}}} \sum_{i=1}^{r} (\mu_{ik})^m \, d_{ik}^2, \tag{10}$$

where $m \in [1,+\infty)$ is a weighting exponent called fuzzifier, $C = \{c_1, c_2, \ldots, c_r\}$ is the vector of the cluster centers, and d_{ik} is the distance between x_k and the ith cluster. Bezdek [21] proved that if $m \geqslant 1, d_{ik}^2 > 0, 1 \leqslant, i \leqslant r$, then $\mathbf{U}$ and C minimize $J_m(U,C)$ only if their entries are computed as follows:

$$\mu_{ik}^* = \frac{1}{\sum_{j=1}^{r} (d_{ik}/d_{jk})^{2/(m-1)}}, \tag{11}$$

$$c_i^* = \frac{\sum_{k=1}^{N_{\mathrm{T}}} (\mu_{ik})^m x_k}{\sum_{k=1}^{N_{\mathrm{T}}} (\mu_{ik})^m}. \tag{12}$$

One of the major factors that influence the determination of appropriate clusters of points is the dissimilarity measure chosen for the problem. Indeed, the computation of the membership degrees μ_{ik}^* depends on the definition of the distance measure d_{ik}, which is the inner product norms (quadratic norms) on $\Re^n$. The squared quadratic norm (distance) between a pattern vector x_k and the center c_i of the ith cluster is defined as follows:

$$d_{ik}^2 = \|x_k - c_i\|_G = (x_k - c_i)^{\mathrm{T}} G (x_k - c_i), \tag{13}$$

where G is any positive definite $(n \times n)$ matrix. The identity matrix is the simplest and most popular choice of G.

The FCM algorithm consists of a series of iterations alternating between Eqs. (11) and (12). This algorithm converges to either a local minimum or a saddle point of $J_m(U,C)$ [27]. We use the FCM to determine the cluster centers c_i and the membership matrix $\mathbf{U}$ for a given r value as follows:

Step 1: Initially the membership matrix is constructed using random values between 0 and 1; such that constrains (7)–(9) are satisfied.

Step 2: For each cluster i $(i = 1, 2, \ldots, r)$, the fuzzy cluster centers c_i are calculated using Eq. (13).

Step 3: For each cluster i, the distance measures d_{ik} are computed using Eq. (13).

Step 4: The cost function in Eq. (10) is computed and if either it is found to be below a certain tolerance value, or its improvement over the previous iteration (dJ_m) is below a certain threshold, then it is stopped and the clustering procedure is terminated.

Step 5: A new $\mathbf{U}$ using Eq. (11) is computed and steps 2–5 are repeated.

By using the above fuzzy clustering procedure, we divide the training patterns into r clusters. The center of each cluster is considered as an initial value for the RBF centers.

3.1.2. *RBF width initialization*

The width of the RBF units is another parameter that should be initialized with a good initial point. The amount of overlap between classes is controlled by the widths of the RBF units. For each cluster k, the distance $\mathrm{dc}(k,j)$ between the center of this cluster and the center of any other cluster is computed by

$$\mathrm{dc}(k,j) = \|c_k - c_j\|, \quad j = 1,2,\ldots,r \text{ and } j \neq k. \tag{14}$$

The minimum among the distances $(d_{\min}(k,l))$ is computed as follows:

$$d_{\min}(k,l) = \min(\mathrm{dc}(k,j)), \quad j = 1,2,\ldots,r \text{ and } j \neq k. \tag{15}$$

We have used the following overlapping criterion to initialize the width of the RBF units. For each cluster k, having determined $d_{\min}(k,l)$ from Eq. (15), the choice of the value for σ_k is calculated using

$$\sigma_k = \gamma d_{\min}(k,l), \tag{16}$$

where γ is a parameter determining the degree of overlap between the classes.

3.1.3. *Determining the number of RBF units based on cluster validity indices*

The criteria for the definition of an optimal partitioning of the data into subgroups are generally based on three requirements [28]: (1) Clear separation between the resulting clusters. (2) Minimal volume of the clusters. (3) Maximal number of relevant data points concerned in the vicinity of the cluster centered. Initially we have considered the number of clusters (number of RBF units) to be known a priori. However, when this number cannot be determined according to some a priori knowledge, a cluster validity criterion is required in order to determine the optimal number of clusters [28,29]. There are a number of cluster validation indices available in the literature. Pal and Bezdek performed an analysis of several indices [29,30]. Tracking the optimal partition consists of varying the number of clusters between fixed minimum and maximum values, and for each given number, computing the cluster validity index. An optimal partition corresponds to a minimum or a maximum value, depending on the index used. There is no general cluster validity index which includes all possible applications with all possible combinations of the parameters of a clustering algorithm. In this paper, a method is presented to select a cluster validity index based on the application of the existing techniques and fusing the outcome through the majority rule. Table 1 presents six validity indices that have been used in this work. In this table F_i is a fuzzy covariance matrix for each cluster i and is defined by

$$F_i = \frac{\sum_{k=1}^{N_\mathrm{T}} \mu_{ik}(x_k - c_i)(x_k - c_i)^\mathrm{T}}{\sum_{k=1}^{N_\mathrm{T}} \mu_{ik}}, \quad 1 \leqslant i \leqslant r. \tag{17}$$

We determine the number of RBF units based on cluster validity indices, assuming that N_T and s, the number of training sample patterns and the number of classes, respectively, are known. Let n_r be the number of RBF units and $r_{\max}$ be the maximum value of n_r computed from the cluster validity indices using the majority rule fusion technique. The following procedure has been used to determine n_r.

Step 1: Initially n_r is set to s.

Step 2: Compute several iterations of the FCM clustering algorithm (as described in Section 3.1.1); then for each cluster i $(i = 1,2,\ldots,n_r)$, compute the fuzzy covariance matrix in Eq. (17).

Step 3: Compute $r_{\max}$ by using the majority rule and the cluster validity indices in Table 1. If $n_r > r_{\max}$ go to step 4; otherwise, increase n_r by 1 and go back to step 2.

Step 4: The number of RBF units is set equal to $r_{\max}$ $(r = r_{\max})$.

The above procedure determines the number of RBF unit (r) in the RBFNN structure.

3.2. *RBFNN parameters adjustment*

The second step in the proposed FHLA method is to adjust the RBFNN parameters. The training of the RBF neural network involves estimating output connection weights, centers and widths of the RBF units. Here we propose a hybrid scheme, which combines the gradient method and the linear least-squared method for adjusting the RBFNN parameters [23]. Let n and s be the number of input and output units, respectively, and we assume that a total of u RBF units are generated for all training patterns based on the procedure described in Section 3.1.3. The hybrid optimization process updates the c_i, σ_i and w_{ij} in two steps. In the first step, the neural network connection weights in the output of the RBF units (w_{ij}) are adjusted under the assumption that the centers and the widths of the RBF units are known a priori. In the second step, the centers and the widths (c_i, σ_i) of the RBF units are updated as described later. The corresponding solutions for each step of the hybrid optimization procedure are presented in the following sections.

3.2.1. *Output connection weights adjustment*

For any input feature vector $x_k \in \mathfrak{R}^n$, the output of the RBFNN in Eq. (1) can be determined in a more compact form as follows:

$$\mathbf{W} \times \mathbf{R} = \mathbf{Y}, \tag{18}$$

where $\mathbf{R} \in \mathfrak{R}^{u \times N_\mathrm{T}}$ is the matrix of the RBF units, $\mathbf{W} \in \mathfrak{R}^{s \times u}$ is the output connection weight matrix, $\mathbf{Y} \in \mathfrak{R}^{s \times N_\mathrm{T}}$ is the output matrix and N_T is the total number of training sample patterns. Since N_T is usually greater than s, this is an over determined problem and generally, there is no exact procedure to solve for $\mathbf{W}$ in Eq. (18). Instead, an iterative method based on the linear least-squared method is utilized to obtain an approximate solution $\mathbf{W}' \in \mathfrak{R}^{s \times u}$, which is close to $\mathbf{W}$ in a least-squared sense. To find $\mathbf{W}'$, the squared error function is determined by

$$\mathrm{SE} = \|\mathbf{T} - \mathbf{W} \times \mathbf{R}\|^2, \tag{19}$$

where $\mathbf{T} = (t_1, t_2, \ldots, t_s)^\mathrm{T} \in \mathfrak{R}^{s \times N_\mathrm{T}}$ is the target matrix consisting of 1's and 0's with each column having only one nonzero element that identifies the processing pattern to

Table 1
Cluster validity indices for using with majority rule for finding cluster number

Validity index	Formula	Cluster no.
Partition density	$V_{\mathrm{D}} = \sum_{i=1}^{r} \frac{S_i}{[\det(F_i)]^{1/2}}$ $S_i = \sum_{k=1}^{N} \mu_{ik}$ $\forall x_k \in \{x_k/(x_k - c_i)F_i^{-1}(x_k - c_i) < 1\}$	Maximum
Average partition density (using determinant of F_i)	$V_{\mathrm{AD}} = \frac{1}{r} \sum_{i=1}^{r} \frac{S_i}{[\det(F_i)]^{1/2}}$	Maximum
Average partition density (using the trace of F_i)	$D_{AD} = \frac{1}{r} \sum_{i=1}^{r} \frac{S_i}{\mathrm{tr}(F_i)}$	Maximum
Trace of between-cluster scatter matrix	$V_{\mathrm{B}} = \mathrm{tr}(S_{\mathrm{B}})$ $S_{\mathrm{B}} = \sum_{i=1}^{r} S_i(c_i - c)(c_i - c)^{\mathrm{T}}$ $S_i = \sum_{k=1}^{N} \mu_{ik}$ $c = \frac{1}{c_i} \sum_{i=1}^{r} c_i$	Minimum
Trace of within-cluster scatter matrix	$V_{\mathrm{w}} = \mathrm{tr}(S_{\mathrm{w}})$ $S_{\mathrm{w}} = \sum_{i=1}^{r} \left(F_i \sum_{k=1}^{N} \mu_{ik} \right)$	Maximum
Hyper volume	$V_{\mathrm{HV}} = \sum_{i=1}^{r} [\det(F_i)]^{1/2}$	Minimum

which the given exemplar belongs. By using the linear least-squared method we can find $\mathbf{W}'$ such that

$$\mathbf{W}' \times \mathbf{R} = \mathbf{T}. \tag{20}$$

The optimal $\mathbf{W}'$ can be obtained by [13]

$$\mathbf{W}' = \mathbf{T}(\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}, \tag{21}$$

where $(\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}$ is the pseudoinverse of $\mathbf{R}$ and $\mathbf{R}^{\mathrm{T}}$ is the transpose of $\mathbf{R}$. From Eq. (21) we can now compute the output connection weights.

3.2.2. *Center and width adjustment*

Assuming that the given training data set has N_{T} sample patterns, we can define the error measure for the mth pattern of the training data as the sum of squared errors (SSE) [13]:

$$E^m = \frac{1}{2} \sum_{k=1}^{s} (t_k^m - y_k^m)^2, \tag{22}$$

where y_k^m and t_k^m represent the kth real output and target output of the mth training pattern, respectively. The center and width of the RBF units are adjusted by taking the negative gradient of the E^m over the RBF parameters. To calculate the negative gradient, first we have to calculate the error rate for the mth training pattern and for the kth output node using

$$\frac{\partial E^m}{\partial y_k^m} = (y_k^m - t_k^m), \qquad k = 1, 2, \ldots, s. \tag{23}$$

For the jth RBF unit, the error rate can be derived by the chain rule

$$\frac{\partial E^m}{\partial R_j^m} = \sum_{k=1}^{s} \frac{\partial E^m}{\partial y_k^m} \frac{\partial y_k^m}{\partial R_j^m} = \sum_{k=1}^{s} (y_k^m - t_k^m) \frac{\partial y_k^m}{\partial R_j^m}. \tag{24}$$

The error rate for the RBF units based on the variation of the center c and the width σ of the RBF units can be derived from Eq. (24) by the chain rule and using Eqs. (1), (6) and

(23), as follows:

$$\frac{\partial E^m}{\partial c_j^m(i)} = 2\sum_{k=1}^{s} y_k^m w_{ik}^m R_j^m (P^m(i) - c_j^m(i))/(\sigma_j^m)^2, \qquad (25)$$

$$\frac{\partial E^m}{\partial \sigma_j^m} = 2\sum_{k=1}^{s} y_k^m w_{ik}^m R_j^m (P^m(i) - c_j^m(i))/(\sigma_j^m)^3, \qquad (26)$$

where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, u, P^m(i)$ is the ith input variable of the mth training pattern, $c_j^m(i)$ is the ith input variable of the center of the jth RBF unit for the mth training pattern, and σ_j^m is the width of the jth RBF unit for the mth training pattern. Accordingly, the update formulas for the center c and the width σ of the RBF units are:

$$\Delta c_j^m(i) = -\xi \frac{\partial E^m}{\partial c_j^m(i)}$$

$$= 2\xi \sum_{k=1}^{s} y_k^m w_2^m(i,k) R_j^m (P^m(i) - c_j^m(i))/(\sigma_j^m)^2, \qquad (27)$$

$$\Delta \sigma_j^m = -\xi \frac{\partial E^m}{\partial \sigma_j^m}$$

$$= 2\xi \sum_{k=1}^{s} y_k^m w_2^m(i,k) R_j^m (P^m(i) - c_j^m(i))/(\sigma_j^m)^3, \qquad (28)$$

where ξ is the learning rate, $\Delta c_j^m(i)$ is the updating value for the ith input variable of the center of the jth RBF unit based on the mth training pattern, and $\Delta \sigma_j^m$ is the updating value for the width of the jth RBF unit for the mth training pattern. The initial conditions for c and σ is considered based on Section 3.1.1 for c and Section 3.1.2 for σ.

3.2.3. Adjusting procedure

Each epoch of the adjusting procedure is composed of a forward pass and a backward pass. In the forward pass, we supply input feature elements to the RBFNN until the matrix $\mathbf{W}'$ is obtained based on Eq. (21). After adjusting the elements of $\mathbf{W}'$ for the input training pattern, the error measure from Eq. (22) is calculated. In the backward pass, the error rates propagate from the output toward the input, and the center and the width (c and σ) of the RBF units are updated by the gradient method, using Eqs. (27) and (28) based on the input training pattern.

Another parameter, which affects the adjustment procedure, is the learning rate (ξ). To ensure the learning algorithm is not trapped in a local minimum and to avoid oscillating around the optimum value, we have selected the learning rate according to the following equation:

$$\xi = \max(\eta_{\max}\lambda^i, \eta_{\min}), \qquad (29)$$

where $\eta_{\max}$ and $\eta_{\min}$ are the maximum and minimum values of the learning rate, $0 < \lambda < 1$ is a descent coefficient, and i is the number of epochs.

3.2.4. The FHLA sensitivity analysis

Let $N_{\rm T}$ and s be the total number of sample patterns and the number of classes, respectively, with each class k having N^k sample patterns. During the learning phase of the neural network a subset of the sample patterns in each class ($N_{\rm t}^k$) is usually selected to train the classifier, where $N_{\rm t}^k < N^k$, therefore the number of available training sets ($n_{\rm t}^k$) for each class k can be computed by

$$n_{\rm t}^k = \frac{N^k!}{N_{\rm t}^k!(N^k - N_{\rm t}^K)!}, \qquad k = 1, 2, \ldots, s. \qquad (30)$$

The total number of possible training sets ($N_{\rm t}^S$) for all s classes is determined as

$$N_{\rm t}^S = \prod_{k=1}^{s} n_{\rm t}^k. \qquad (31)$$

It should be noted that $N_{\rm t}^S$ is usually a very large number. In this work one of the databases that has been used is the ORL database, which has 40 classes ($s = 40$) with 10 sample patterns in each class ($N^k = 10$). We have selected five sample patterns for the training purpose in each class ($N_{\rm t}^k = 5$), therefore the total number of available training sets is $N_{\rm t}^s = 252^{40}$. On the other hand, selecting a different training set in the RBFNN would result in changes in the center of the RBF unit, which subsequently affects the output of the RBF unit as can be seen from Eqs. (6) and (7). Finding the best training set among the $N_{\rm t}^S$ to best train the RBFNN is a difficult task. Randomly selecting the training set is possible if the sensitivity of the learning algorithm to the training set selection is found to be low. To determine the sensitivity of the proposed learning algorithm to the choice of the training set, we observe that the sensitivity of the $R_i(.)$ of the ith RBF unit in Eq. (6) with the respect to the corresponding center c_i can be obtained using [26]

$$S_i = \frac{\|\partial R_i/\partial c_i\|^2}{\|x_k - c_i\|^2}, \qquad k = 1, 2, \ldots, N_{\rm t}^k, \qquad (32)$$

where x_k is the kth input feature vector. It can easily be shown that Eqs. (6) and (32) give

$$S_i = \left(\frac{2}{\sigma_i^2}\right)^2 \exp\left(-2\frac{\|x_k - c_i\|^2}{\sigma_i^2}\right). \qquad (33)$$

Replacing σ_i from Eq. (16) in Eq. (33) gives

$$S_i = \frac{4}{\gamma^4 {\rm d}_{\min}^2(i,l)} \exp\left(-2\frac{\|x_k - c_i\|^2}{\gamma^2 {\rm d}_{\min}^2(i,l)}\right). \qquad (34)$$

The above equation indicates the sensitivity of the FHLA to the selection of the training and testing sets. The clustering procedure described in Section 3.1.1 requires

$$\|x_k - c_i\|^2 < d_{\min}^2(i,l), \qquad (35)$$

where $d_{\min}(i,l)$ has been defined in Eq. (15). Considering inequality (35), it is evident that by properly selecting γ, one can obtain

$$\frac{\|x_k - c_i\|^2}{\gamma^2 d_{\min}^2(i,l)} < 1. \tag{36}$$

This condition forces the exponent part of Eq. (34) to be less than unity, resulting in S_i to be less sensitive to variations in c_i and x_k.

It is also noted from Eq. (25) that changing the center of the RBF unit influences the overall classifier error. Therefore, we can measure the sensitivity of the learning algorithm to the training and testing patterns using the standard deviation of the overall classifier error. The standard deviation is computed based on randomly selecting different training and testing sets.

4. Design of experiment

The human face recognition system has been used as a benchmark in this study. A complete conventional human face recognition system should include two stages. The first stage requires extraction of pertinent features from the facial images and the creation of the feature vectors. The second stage involves classification of facial images based on the derived feature vector obtained in the first stage. The designed RBFNN with the proposed learning algorithm has been used as a classifier in the second stage of the human face recognition system.

The aim of the feature extraction in the first stage of the human face recognition system is to produce a feature vector containing all pertinent information of the face to be recognized. Two main approaches for feature extraction have been extensively used by other researchers [4,31]. The first method is based on extracting structural and geometrical facial features that constitute local structure of facial images, for example, the shapes of the eyes, nose and mouth. The structural-based approaches deal with local data instead of global data. It has been shown that, due to explicit modeling of facial features, the structural-based approaches suffer from the unpredictability of facial appearances and environmental conditions [4]. The second method is a statistical-based approach that extracts features from the whole image and, therefore, uses global data instead of local data. Since the global data of an image are now used to determine the feature elements, the data that are irrelevant to the facial portion of the image, such as hair, shoulders and background, may contribute to the creation of erroneous feature vectors that can affect the recognition results [32]. In this paper, we have used PCA [25] as a feature domain that uses global data to create the feature vector elements. However, to keep only the important data about the face images, and to eliminate the irrelevant data the feature extraction is done in two steps. In the first step, by using the shape information based on [5,24], we have created a subimage to enclose only the important information needed for the recognition algorithm, while in the second step, the feature vector has been obtained through calculation of the PCA of the derived subimage.

4.1. Subimage formation

The subimage encloses all the pertinent information around the face in an ellipse while the pixel value outside the ellipse is set to zero. A technique is presented in Refs. [5,24], which finds the best-fit ellipse to enclose the facial region of the human face in a frontal view of the facial image. Unfortunately through creation of the subimage with the best-fit ellipse many unwanted regions of the face image may still appear in this subimage. These include hair portion, neck and part of the background as an example. This is shown in Fig. 2. Instead of using the best-fit ellipse for creating a subimage we have defined another ellipse. The new ellipse has the same orientation and center as the best-fit ellipse but the length of its major and minor axes are calculated from the length of the major and minor axes of the best-fit ellipse as follows:

$$A = \rho\alpha, \tag{37}$$

$$B = \rho\beta, \tag{38}$$

where A and B are, respectively, the length of the major and minor axes of the new ellipse, α and β are the length of the major and minor axes of the best-fit ellipse [24,33], and ρ is a coefficient that varies between 0 and 1. Fig. 3 shows the effect of changing ρ on facial images, while Fig. 4 shows the corresponding subimages.

Our experimental results with 400 images show that the best value for ρ is around 0.87. By using the ρ parameter for subimage creation, data that are irrelevant to facial portion, such as hair, neck, shoulders and background, are discarded.

4.2. Principle component analysis (PCA)

PCA is a well-known statistical technique for feature extraction. Each $M \times N$ image in the training set was row concatenated to form $MN \times 1$ vectors x_k. Given a set of N_{T} training images $\{x_k\}_{k=0,1,\dots N_{\mathrm{T}}}$ the mean vector of the training set was obtained as [25]

$$\bar{x} = \frac{1}{N_{\mathrm{T}}} \sum_{k=1}^{N_{\mathrm{T}}} x_k. \tag{39}$$

A $N_{\mathrm{T}} \times MN$ training set matrix $\mathbf{X} = [x_k - \bar{x}]$ can now be built. The basis vectors are obtained by solving the eigenvalue problem:

$$\Lambda = V^{\mathrm{T}} \mathbf{\Sigma_X} V, \tag{40}$$

where $\mathbf{\Sigma_X} = \mathbf{X}\mathbf{X}^{\mathrm{T}}$ is the covariance matrix, V is the eigenvector matrix of $\mathbf{\Sigma_X}$ and Λ is the corresponding diagonal matrix of eigenvalues. As the PCA has the property of packing the greatest energy into the least number of principal

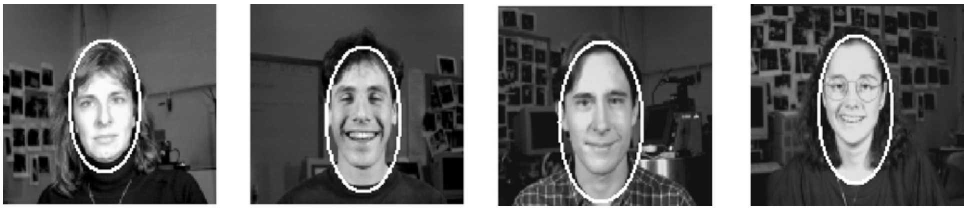

Fig. 2. Face localization based on best-fit ellipse.

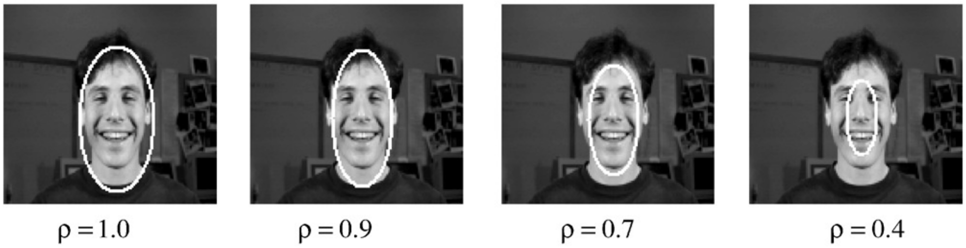

Fig. 3. Different ellipses with pertinent values of ρ.

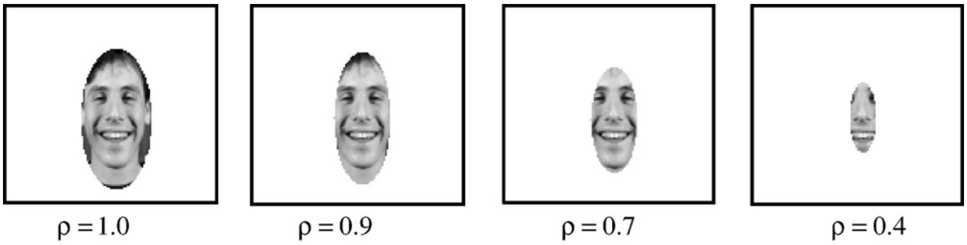

Fig. 4. Subimage formation based on different values of ρ.

components, eigenvectors corresponding to the m largest eigenvalues in the PCA are selected to form a lower-dimensional subspace. It is proven that the residual reconstruction error generated by discarding the $N_T - m$ components is low even for small m [25].

4.3. Experimental design

To check the utility of the proposed algorithm experimental studies are carried out on the ORL database images of Cambridge University and the Yale face database of Yale University. The ORL database contains 400 face images from 40 individuals in different states (available at http://www.uk.research.att.com/facedatabase.html). The total number of images for each person is 10. None of the 10 samples is identical to any other sample. They vary in position, rotation, scale and expression. The changes in orientation have been accomplished by rotating the person a maximum of 20° in the same plane; also each person has changed his/her facial expression in each of 10 samples (open/close eye, smiling/not smiling). The changes in scale have been achieved by changing the distance between the person and the video camera. For some individuals, the images were taken at different times, varying facial details (glasses/no glasses). Each image was digitized and presented by a 112 × 92 pixel array whose gray levels ranged between 0 and 255. Samples of the ORL database are shown in Fig. 5.

The Yale face database contains 165 face images of 15 individuals. There are 11 images per subject, one for each facial expression or configuration: center-light, glasses/no glasses, happy, normal, left-light, right-light, sad, sleepy, surprised and wink. Samples of Yale face database are shown in Fig. 6. (This database is available at http://cvc.yale.edu/projects/yalefaces/yalefaces.html.)

Fig. 7 illustrates the experiments carried out in this work. Each experiment consists of four steps: subimage formation, generation of the feature vector, training the classifier and testing the classifier. In the first step, the subimage is created using the method described in Section 4.1. The training and testing set is selected, by randomly choosing five images for

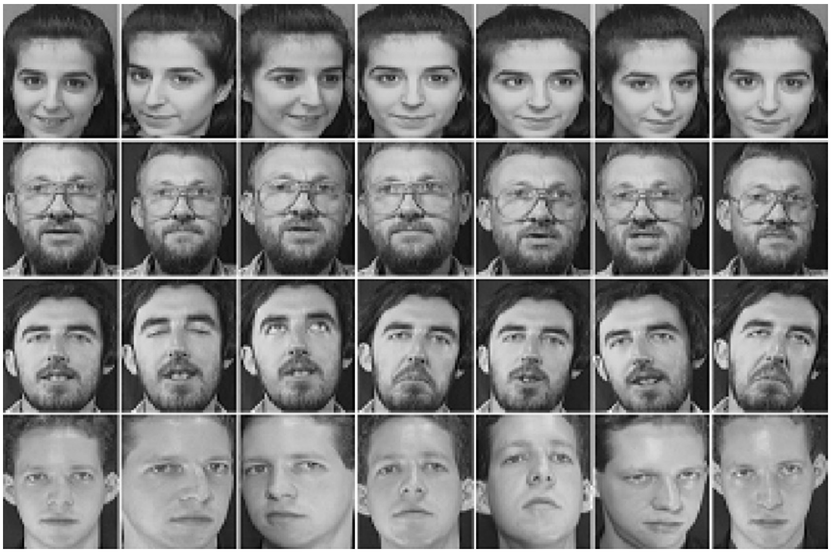

Fig. 5. Samples of face images in the ORL database.

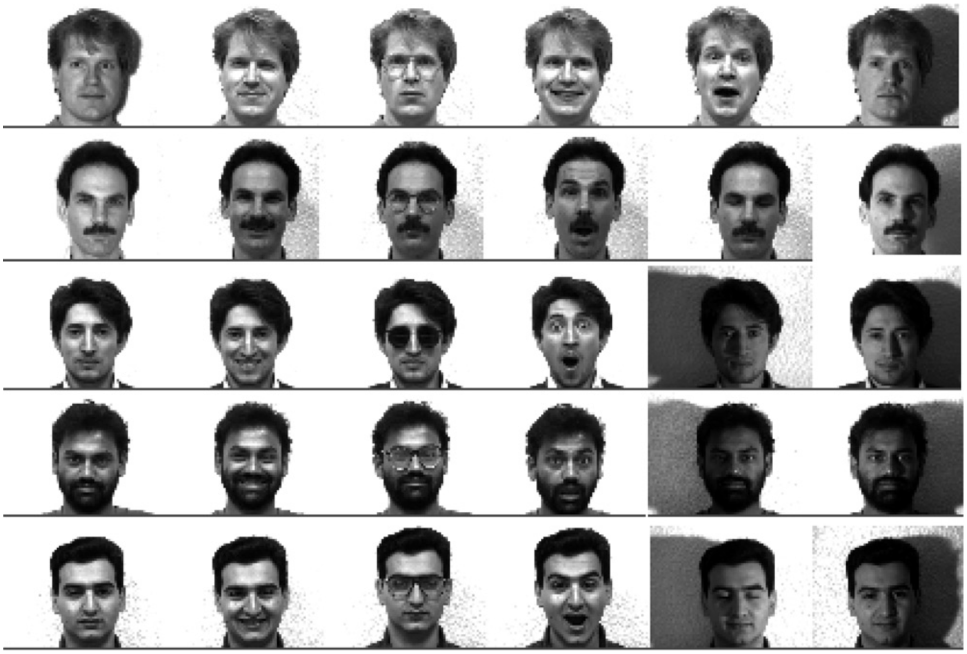

Fig. 6. Samples of face images in the Yale face database.

each subject from the ORL database and six images from the Yale database. Therefore, in the ORL database a total of 200 images are used as the training set and another 200 are used as the testing set while in the Yale database a total of 90 images are used for training and the rest are used for testing. In the second step PCAs are generated inside the subimages. In the third step, the classifier is designed and trained. Finally in the fourth step, performance of the classification is evaluated. This procedure has been repeated for each learning algorithm by randomly choosing different training and testing sets.

5. Experimental results and discussion

The experimental study conducted in this paper evaluates the effect of different learning algorithms on the number of required hidden neurons (complexity of the RBFNN),

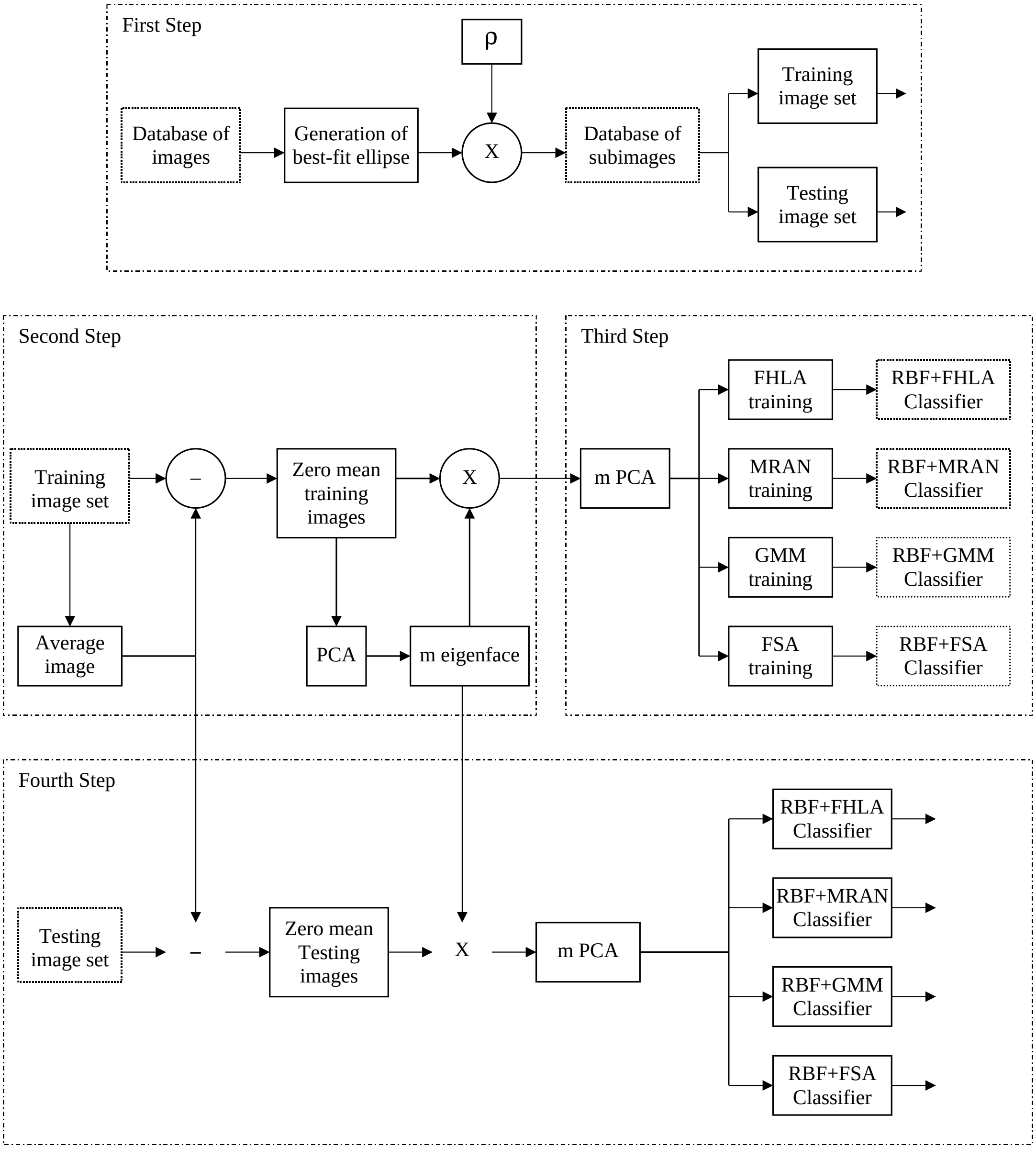

Fig. 7. Experimental design carried out in this work.

training speed, sensitivity of the learning algorithms to the training and testing set and the overall recognition rate. Also, the effect of the irrelevant data on the overall recognition rate is studied. Finally the proposed human face recognition system with the FHLA learning algorithm is compared with other human face recognition systems.

5.1. *Effect of the learning algorithms*

For the purpose of evaluating the different learning algorithms, we have designed and trained the RBFNN with FHLA, MRAN, GMM and FSA. The experiment has been repeated 40 times by randomly choosing different training and testing sets on each database. Also the number of principle components to represent the feature vectors was set to eleven different values: 10, 15, 20, 25, 30, 40, 50, 55, 60, 70, 80. A total of 440 runs were executed for each learning algorithm on each database. Table 2 shows the parameters for the FHLA during training phase for the ORL and the Yale databases. Fig. 8 shows the average classifier error rate as a function of the number of principle components for the FHLA and MRAN among 40 runs for the ORL database.

Table 2
FHLA and its parameters in training phase

No. of PCA	Training phase parameters							
	λ	γ	$\eta_{\max}$	$\eta_{\min}$	ORL database		Yale database	
					Epochs	SSE	Epochs	SSE
10	0.99	0.9–1	8×10^{-2}	8×10^{-3}	15–35	0.16–0.11	10–25	0.12–0.09
20	0.99	1–1.1	2.5×10^{-3}	5×10^{-4}	25–45	0.12–0.08	15–30	0.9–0.07
30	0.99	1.1–1.2	9×10^{-3}	3×10^{-4}	40–55	0.09–0.06	25–40	0.07–0.05
40	0.99	1.1–1.2	6.5×10^{-4}	4.5×10^{-5}	55–65	0.07–0.05	35–55	0.05–0.03
50	0.98	1.2–1.5	8.5×10^{-4}	2×10^{-5}	60–75	0.05–0.02	50–65	0.05–0.02
55	0.99	1.5–1.7	2×10^{-5}	1.2×10^{-6}	70–90	0.03–0.01	55–70	0.03–0.01
60	0.97	1.7–1.9	5×10^{-5}	3.5×10^{-6}	85–100	0.06–0.03	65–80	0.04–0.02
70	0.99	1.8–2	9×10^{-5}	6×10^{-6}	90–110	0.08–0.05	70–90	0.05–0.03
80	0.99	1.8–2	5×10^{-6}	3×10^{-7}	105–125	0.11–0.08	85–105	0.09–0.05

Note: $\eta_{\max}$—maximum learning rate, $\eta_{\min}$—minimum learning rate, λ—descent coefficient for learning rate, γ—overlapping parameter, SSE—sum of squared errors.

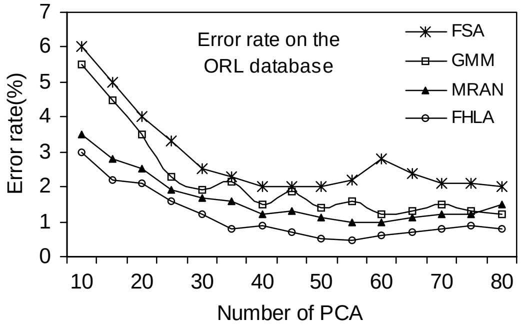

Fig. 8. Average error rate with respect to PCA number for different learning algorithms for the ORL database.

Table 3
Minimum average error rate for different learning algorithms

Learning algorithm	Number of PCA	Min. average error	
		ORL database (%)	Yale database (%)
FSA	50	2.0	1.6
GMM	60	1.5	1.3
MRAN	60	1.1	0.82
FHLA	55	0.45	0.25

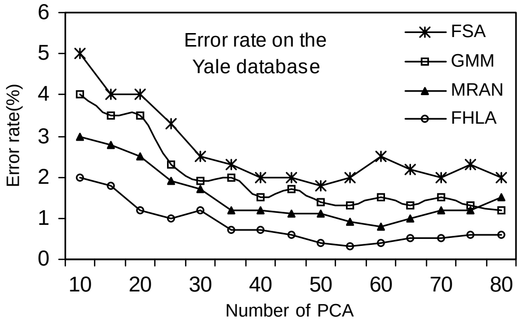

Fig. 9. Average error rate as a function of PCA number for different learning algorithms for the Yale database.

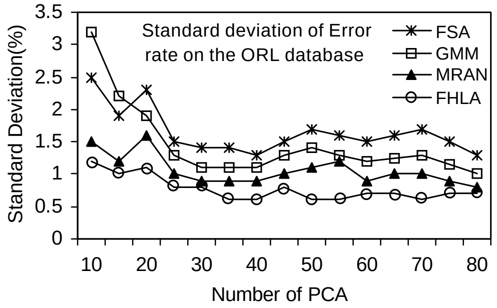

Fig. 10. Standard deviation of classifier error for different learning algorithms on the ORL database.

In this figure, however, the average classifier error rate for GMM and FSA was computed for 25 runs as reported in Ref. [19]. Also Fig. 9 shows the average error rate for different learning algorithms for the Yale database. The minimum average classifier error rate for different learning algorithms for the two databases has been shown in Table 3. The average classifier error rate curves and Table 3 show the FHLA has a small error rate than the other training algorithms.

Fig. 10 presents the standard deviation of the error rate computed for 40 runs for the FHLA, MRAN, GMM and FSA as a function of the number of the PCA for the ORL database, while Fig. 11 shows the corresponding values for

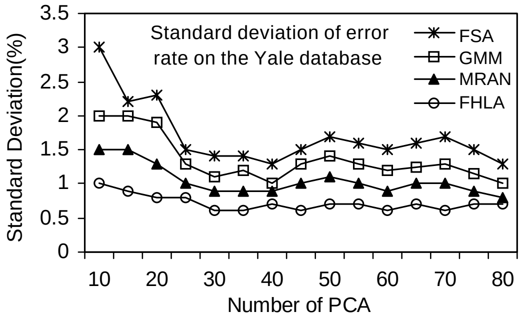

Fig. 11. Standard deviation of classifier error for different learning algorithms on the Yale database.

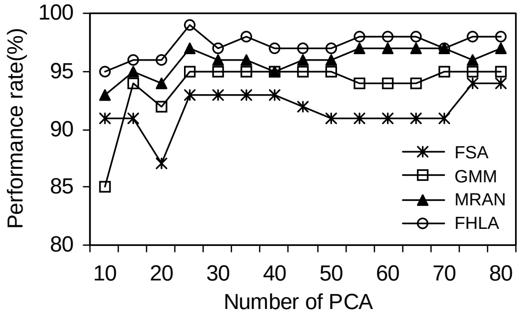

Fig. 12. Minimum recognition rate among 40 runs for different learning algorithms for the ORL database.

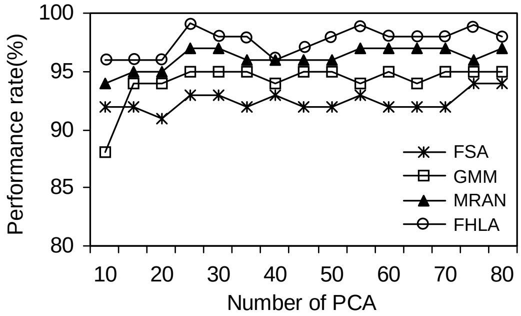

Fig. 13. Minimum recognition rate among 40 runs for different learning algorithms for the Yale database.

the Yale database. These graphs indicate the sensitivity of the results to the choice of the training and testing sets. The FHLA method again presents the lowest standard deviation for all the PCA numbers and, therefore, is less sensitive to the selection of training and testing patterns.

Figs. 12 and 13 present the minimum recognition rate among 40 runs for the learning algorithms for the ORL and the Yale databases, respectively. It is seen that, among the various learning algorithms, the FHLA yields the highest recognition rate.

The FHLA method has been compared with the MRAN, GMM and FSA in terms of the training parameters during the learning phase. These include the number of required hidden neurons (RBF units) and the average number of epochs. The average number of epochs for 40 runs for the ORL and Yale databases has been shown in Table 4. As Table 4 indicates, the FHLA method has converged faster than the other learning methods for all PCA numbers. Another important parameter is the number of RBF units required in each learning algorithm. The maximum and minimum number of required RBF units among 40 runs has been indicated in Table 5 where the FHLA and MRAN outperform the other two techniques. However, for low numbers of PCA, the MRAN requires a smaller number of neurons in the hidden layer than does the FHLA, while for high numbers of the PCA, the FHLA requires fewer neurons in the hidden layer than does the MRAN. In our study we have found the best recognition rate is obtained with the first 55 largest PCA numbers, as can be seen from Figs. 8, 9 and Table 3. The FHLA yields a better recognition rate for high PCA numbers while Table 5 indicates the FHLA also requires fewer neurons in the hidden layer than the other learning algorithms. It is interesting to note that for face recognition, the FHLA results in a neural network with the lowest number of hidden neurons while producing the best recognition rates.

5.2. *Effect of the irrelevant data on recognition rate*

For the purpose of evaluating how the irrelevant data of a face image such as hair, neck, shoulder and background will influence the recognition results we have first set the input feature vector with the 55 largest PCA numbers for the FHLA, the 60 largest PCA numbers for MRAN and GMM and finally 50 for FSA, which have yielded the best results (Table 3). We have varied the ρ value in Eqs. (37) and (38) as can be seen in Fig. 4 and evaluated the recognition rate for different learning algorithms. Fig. 14 shows the effect of changing ρ on the system error rate for the ORL database. As Fig. 14 shows, the best recognition rate can be achieved at $\rho = 0.87$ for all learning algorithms. We have used the best value of $\rho = 0.87$ while was empirically derived, in our recognition system.

5.3. *Comparison with the other human face recognition systems*

To evaluate the effectiveness of our proposed method in comparison with other human face recognition systems, we have chosen the 55 largest PCA numbers for feature extraction. We have also selected $\rho = 0.87$ and an RBF neural network with the FHLA learning algorithm as the classifier. In our study, the ORL database was used in the experiments and the methods reported in Refs. [34–37] were used for

Table 4
Average number of epochs among 40 runs for different learning algorithms

No. of PCA	ORL database				Yale database			
	FHLA	MRAN	GMM	FSA	FHLA	MRAN	GMM	FSA
10	23	29	53	62	18	21	45	52
20	34	38	67	73	23	30	57	64
30	46	62	94	110	31	53	85	91
40	59	76	108	121	43	66	93	102
50	67	89	118	132	54	79	102	112
55	81	101	136	144	67	88	121	134
60	92	125	159	172	74	101	142	155
70	101	138	171	185	79	127	153	164
80	112	146	193	202	98	135	172	188

Table 5
Minimum and maximum number of required RBF unit among 40 runs for different learning algorithms

No. of PCA	ORL database				Yale database			
	FHLA	MRAN	GMM	FSA	FHLA	MRAN	GMM	FSA
10	43–46	42–45	46–50	47–50	18–19	17–19	19–22	19–23
20	45–48	44–47	47–52	48–53	19–21	18–20	19–22	20–23
30	47–50	47–50	48–53	49–53	19–21	18–20	20–23	22–24
40	47–51	48–52	50–54	52–55	20–22	20–23	22–25	23–25
50	48–52	50–54	53–56	54–57	20–22	22–25	24–27	25–28
55	48–51	51–55	54–57	55–57	20–23	24–26	25–28	26–29
60	49–52	53–56	55–57	56–58	21–24	24–26	27–30	28–30
70	51–54	55–57	57–60	58–61	21–25	26–29	28–32	29–32
80	53–56	56–60	59–63	60–63	22–26	27–29	30–33	31–34

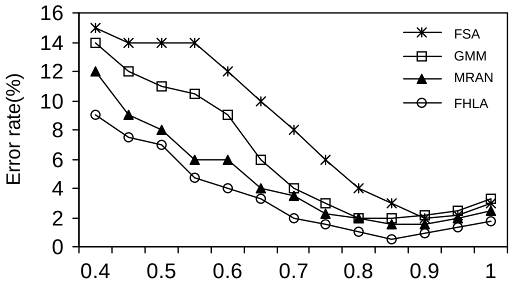

Fig. 14. Classifier error rate variation with respect to changing ρ for different learning algorithms.

Table 6
Error rates for different human face recognition systems on the ORL database

Methods	No. of experimental (m)	E_{ave}%
CNN [35]	3	3.83
NFL [36]	4	3.125
FT [37]	1	1.75
SINN [34]	4	1.323
Proposed method	4	0.45

comparison purpose. To make this comparison meaningful, an average overall error rate is defined as [34–37]

$$E_{\mathrm{ave}} = \frac{\sum_{i=1}^{m} N_m^i}{MN_{\mathrm{t}}}, \tag{41}$$

where m is the number of experimental runs which are performed on the random partitions of the database, N_m^i is the number of misclassification for the ith run, and N_{t} is the number of total testing images for each run. Table 6 shows the outcome of this comparative study, where SINN denotes the shape information with the neural network that was reported in Ref. [34], CNN is the convolution neural network method in Ref. [35], NFL is the nearest feature line method in Ref. [36], and FT denotes the fractal transformation technique in Ref. [37]. In this table, the proposed method yielded an error rate of 0.45%, which is the best obtained in our experiment for the ORL database.

6. Conclusion

This paper presents an alternative learning algorithm for the RBFNN with applications in the recognition of human facial images. The proposed algorithm offers faster convergence, less sensitivity to training and testing sets and less classification error in comparison with other algorithms. The proposed fuzzy hybrid learning algorithm substantially decreases the dimensions of the search space in the gradient method, which is crucial in the optimization of high-dimensional problem such as human face recognition. It also results in an RBFNN, which has a small number of hidden neurons. A comparative study demonstrates the superiority of the proposed learning algorithm in human face recognition in comparison with three popular learning techniques. A recognition rate of 99.55% for the ORL database and 99.75% for the Yale database was obtained using the proposed learning algorithm.

Acknowledgements

The authors wish to thank Dr. Reza Lashkari for his constructive criticism of the manuscript and Natural Science and Engineering Research Council of Canada (NSERC) and Micronet for supporting this research.

References

[1] L. Bruzzone, D.F. Prieto, S.B. Serpico, A neural-statistical approach to multitemporal and multisource remote-sensing image classification, IEEE Trans. Geosci. Remote Sensing (30) (1999) 1350–1359.

[2] J.-N. Hwang, S.-Y. Kung, M. Niranjan, J.C. Principe, The past, present, and future of neural networks for signal processing, IEEE Signal Process. Mag. (14) (1997) 28–48.

[3] M.A. Grudin, On internal representation in face recognition systems, Pattern Recognition 33 (6) (2000) 1161–1177.

[4] M.H. Yung, D.J. Kreigman, N. Ahuja, Detecting face in images: a survey, IEEE Trans. Pattern Anal. Mach. Int. 34 (1) (2002) 34–58.

[5] J. Haddadnia, K. faez, Human face recognition based on shape information and pseudo Zernike moment, Proceedings of 5th International Fall Workshop on Vision, Modeling and Visualization, Saarbrucken, Germany, November 22–24, 2000, pp. 113–118.

[6] C. Lee, D.A. Landgrebe, Decision boundary feature extraction for neural networks, IEEE Trans. Neural Networks 8 (1997) 75–83.

[7] W. Zhou, Verification of the nonparametric characteristics of backpropagation neural networks for image classification, IEEE Trans. Geo. Remote Sensing 37 (2) (1999) 771–779.

[8] Y. Hara, R.G. Atkins, S.H. Yueh, R.T. Shin, J.A. Kong, Application on neural networks to radar image classification, IEEE Trans. Geo. Remote Sensing 32 (1) (1994) 100–109.

[9] L. Yingwei, N. Sundarajan, P. Saratchandran, Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm, IEEE Trans. Neural Networks 9 (2) (1998) 308–318.

[10] J. Haddadnia, K. Faez, P. Moallem, Neural network based face recognition with moments invariant, IEEE International Conference On Image Processing, Vol. I, Thessaloniki, Greece, 7–10, October 2001, pp. 1018–1021.

[11] C.M. Bishop, Neural Network for Pattern Recognition, Oxford University Press, New York, 1995.

[12] J.S.R. Jang, C.T. Sun, Functional equivalence between radial basis function network and fuzzy inference system, IEEE Trans. Neural Networks 4 (1) (1993) 156–158.

[13] J-S.R. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Trans. Systems Man Cybern. 23 (3) (1993) 665–684.

[14] Z.Q. Liu, F. Yan, Fuzzy neural network in case-based diagnostic system, IEEE Trans. Fuzzy Systems 5 (2) (1997) 209–222.

[15] C.F. Juang, C.T. Lin, An on-line self-construction neural fuzzy inference network and its application, IEEE Trans. Fuzzy Systems 6 (1) (1998) 12–32.

[16] M. Delgado, A.F. Gomez-Skarmeta, F. Martin, A fuzzy clustering-based rapid prototype for fuzzy rule-based modeling, IEEE Trans. Fuzzy Systems 5 (2) (1997) 223–233.

[17] F. Behloul, B.P.F. Lelieveldt, A. Boudraa, J.H.C. Reiber, Optimal design of radial basis function neural networks for fuzzy-rule extraction in high dimensional data, Pattern Recognition J. 35 (3) (2002) 659–675.

[18] S. Chen, C.F.N. Cowan, P.M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Trans. Neural Networks 2 (2) (1991) 302–309.

[19] C.E. Thomaz, R.Q. Feitosa, A. Veiga, Design of radial basis function network as classifier in face recognition using eigenfaces, IEEE Proceedings of Vth Brazilian Symposium on Neural Network, 1998, pp. 118–123.

[20] R. Tagliaferri, D. Scala, A supervised fuzzy clustering for radial basis function neural networks training, Joint 9th IFSA World Conferences and 20th NAFIPS International Conferences, Vol. 3, 2001, pp. 1804–1809.

[21] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum, New York, 1981.

[22] S. Abe, R. Thawonmas, M. Kayama, A fuzzy cluster with ellipsoidal regions for diagnosis problem, IEEE Trans. Systems Man Cybern. Part C: Appl. Rev. 29 (1) (1999) 140–149.

[23] J. Haddadnia, M. Ahmadi, K. Faez, A hybrid learning RBF neural network for human face recognition with pseudo Zernike moment invariant, IEEE International Joint Conference on Neural Network, Honolulu, HI, May 12–17, 2002, accepted for publication.

[24] J. Haddadnia, M. Ahmadi, K. Faez, An efficient method for recognition of human face recognition using higher order pseudo Zernike moment invariant, The 5th IEEE International Conference on Automatic Face and Gesture Recognition, Washington, DC, USA, May 20–21, 2002, accepted for publication.

[25] M. Truk, A. Pentland, Eigenface for Recognition, J. Cog. Neurosci. 3 (1) (1991) 71–86.

[26] N.B. Karayiannis, Gradient descent learning of radial basis neural networks, IEEE International Conference on Neural Networks, Houston, TX, June 9–12, 1997, Vol. 3, pp. 1815–1820.

[27] M.S. Yang, Convergence properties of the generalized fuzzy-C-means clustering algorithms, Comput. Math. Appl. 25 (11) (1993) 3–11.

[28] M.R. Rezaee, B.P.F. Lelieveldt, J.H.C. Reiber, A new cluster validity index for the fuzzy-C-maen, Pattern Recognition Lett. 19 (1998) 237–246.

[29] N.R. Pal, J.C. Bezdek, On cluster validity for fuzzy-C-mean model, IEEE Trans. Fuzzy Syst. 3 (3) (1995) 370–379.

[30] N.R. Pal, J.C. Bezdek, Correction to on cluster validity for the fuzzy-C-mean model, IEEE Trans. Fuzzy Systems 5 (1) (1997) 152–153.

[31] J. Daugman, Face detection: a survey, Comput. Vision Image Understanding 83 (3) (2001) 236–274.

[32] L.F. Chen, H.M. Liao, J. Lin, C. Han, Why recognition in a statistic-based face recognition system should be based on the pure face portion: a probabilistic decision-based proof, Pattern Recognition 34 (6) (2001) 1393–1403.

[33] K. Sobotta, I. Pitas, Face localization and facial feature extraction based on shape and color information, IEEE International Conference on Image Processing, Vol. 3, Switzerland, 16–19 September 1996, pp. 483–486.

[34] J. Haddadnia, K. Faez, Human face recognition with moments invariant, Proceeding of The IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing, Maryland, USA, June 3–6, 2001.

[35] S. Lawrence, C.L. Giles, A.C. Tsoi, A.D. Back, Face recognition: a convolutional neural networks approach, IEEE Trans. Neural Networks (Special Issue on Neural Networks and Pattern Recognition) 8 (1) (1997) 98–113.

[36] S.Z. Li, J. Lu, Face recognition using the nearest feature line method, IEEE Trans. Neural Networks 10 (1999) 439–443.

[37] T. Tan, H. Yan, Face recognition by fractal transformations, IEEE Int. Conf. Acoustics, Speech Signal Process. 6 (1999) 3537–3540.

About the Author—JAVAD HADDADNIA received his B.Sc. degree in Electrical and Electronic Engineering with the first rank in 1993 and M.Sc. in Electronic Engineering in 1995, from Amirkabir University of Technology, Tehran, Iran, respectively. He is currently a Ph.D. candidate in the Electrical Engineering Department of Amirkabir University of Technology, Tehran, Iran. His research interests include neural network, digital image processing, computer vision, face detection and recognition. He has published several papers in the above area. He has served as a visiting research scholar at the University of Windsor, Canada during 2001–2002. He is a member of SPIE, CIPPR and IEICE.

About the Author–KARIM FAEZ was born in Semnan, Iran. He received his B.S. degree in Electrical Engineering from Tehran Polytechnic University as the first rank in June 1973, and his M.S. and Ph.D. degrees in Computer Science from University of California at Los Angeles (UCLA) in 1977 and 1980 respectively. Prof. Faez was with Iran Telecommunication Research Center (1981–1983) before joining Amirkabir University of Technology (Tehran Polytechnic) in Iran, where he is now a professor of Electrical Engineering. He was the founder of the Computer Engineering Department of Amirkabir University in 1989 and he has served as the first chairman during April 1989–September 1992. Professor Faez was the chairman of planning committee for Computer Engineering and Computer Science of Ministry of Science, research and Technology (during 1988–1996). His research interests are in Pattern Recognition, Image Processing, Neural Networks, Signal Processing, Farsi Handwritten Processing, Earthquake Signal Processing, Fault Tolerant System Design, Computer Networks, and Hardware Design. He is a member of IEEE, IEICE, and ACM.

About the Author—MAJID AHMADI received B.Sc.(EE) degree from Arya Mehr University in Tehran, Iran, and Ph.D. degree from Imperial College of London University, London, UK, in 1970 and 1977, respectively. He has been with the Department of Electrical and Computer Engineering, University of Windsor, Canada, since 1980 and holds the rank of Professor. His research interests include pattern recognition, machine vision, neural network and design, stability and realization of 2-D filters. Dr. Ahmadi co-authored the book on *Digital Filtering in* 1-D *and* 2-D *Design and Applications*, published by Plenum, 1989. He has published over 300 articles in the above area. Dr. Ahmadi is a fellow of the Institution of Electrical Engineers in England (FIEE) and is a Fellow member of IEEE.