An Efficient Feature Extraction Method with Pseudo-Zernike Moment in RBF Neural Network-Based Human Face Recognition System

Javad Haddadnia

Engineering Department, Tarbiat Moallem University of Sabzevar, Sabzevar, Khorasan 397, Iran
Email: haddadnia@sttu.ac.ir

Majid Ahmadi

Electrical and Computer Engineering Department, University of Windsor, Windsor, Ontario, Canada N9B 3P4
Email: ahmadi@uwindsor.ca

Karim Faez

Electrical Engineering Department, Amirkabir University of Technology, Tehran 15914, Iran
Email: kfaez@aut.ac.ir

This paper introduces a novel method for the recognition of human faces in digital images using a new feature extraction method that combines the global and local information in frontal view of facial images. Radial basis function (RBF) neural network with a hybrid learning algorithm (HLA) has been used as a classifier. The proposed feature extraction method includes human face localization derived from the shape information. An efficient distance measure as facial candidate threshold (FCT) is defined to distinguish between face and nonface images. Pseudo-Zernike moment invariant (PZMI) with an efficient method for selecting moment order has been used. A newly defined parameter named axis correction ratio (ACR) of images for disregarding irrelevant information of face images is introduced. In this paper, the effect of these parameters in disregarding irrelevant information in recognition rate improvement is studied. Also we evaluate the effect of orders of PZMI in recognition rate of the proposed technique as well as RBF neural network learning speed. Simulation results on the face database of Olivetti Research Laboratory (ORL) indicate that the proposed method for human face recognition yielded a recognition rate of 99.3%.

Keywords and phrases: human face recognition, face localization, moment invariant, pseudo-Zernike moment, RBF neural network, learning algorithm.

1. INTRODUCTION

Face recognition has been a very popular research topic in recent years because of wide variety of application domains in both academia and industry. This interest is motivated by applications such as access control systems, model-based video coding, image and film processing, criminal identification and authentication in secure systems like computers or bank teller machines, and so forth [1]. A complete face recognition system should include three stages. The first stage is detecting the location of the face, which is difficult because of unknown position, orientation, and scaling of the face in an arbitrary image [2, 3, 4]. The second stage involves extraction of pertinent features from the localized facial image obtained in the first stage. Finally, the third stage requires classification of facial images based on the derived feature vector obtained in the previous stage.

In order to design a high recognition rate system, the choice of feature extractor is very crucial and extraction of pertinent features from two-dimensional images of human face plays an important role in any face recognition system. There are various techniques reported in the literature that deal with this problem. A recent survey of the face recognition systems can be found in [1, 5]. Two main approaches to feature extraction have been extensively used by other researchers [5]. The first one is based on extracting structural and geometrical facial features that constitute the local structure of facial images, for example, the shapes of the

eyes, nose, and mouth [6, 7]. The structure-based approaches deal with local information instead of global information, and, therefore, they are not affected by irrelevant information in an image. However, because of the explicitness model of facial features, the structure-based approaches are sensitive to the unpredictability of face appearance and environmental conditions [5]. The second method is a statistics-based approach that extracts features from the whole image and, therefore uses global information instead of local information. Since the global data of an image are used to determine the feature elements, information that is irrelevant to facial portion, such as hair, shoulders, and background, may create erroneous feature vectors that can affect the recognition results [8].

In recent years, many researchers have noticed this problem and tried to exclude the irrelevant data while performing face recognition. This can be done by eliminating the irrelevant data of a face image with a dark background [9] and constructing the face database under constrained conditions, such as asking people to wear dark jackets and to sit in front of a dark background [10]. Turk and Pentland [11] multiplied the input image by a two-dimensional Gaussian window centered on the face to diminish the effects caused by the nonface portion. Sung and Poggio [12] tried to eliminate the near-boundary pixels of a normalized face image by using a fixed size mask. In [13], Liao et al. proposed a face-only database as the basis for face recognition.

In this paper, an efficient feature extraction technique is developed, based on the combination of local and global information of face images. At first, face localization based on shape information [2, 14] with a new definition for distance measure threshold called facial candidate threshold (FCT) for distinguishing between nonface image and facial image candidate is introduced. We present the effect of varying the FCT on the recognition rate of the proposed technique. A new parameter, called the axis correction ratio (ACR), is defined to eliminate irrelevant data from the face images and to create a subimage for further feature extraction. We have shown how ACR can improve the recognition rate. Once the face localization process is completed, pseudo-Zernike moment invariant (PZMI) with a new method to select moment orders is utilized to obtain the feature vector of the face under recognition. In this paper, PZMI was selected over other types of moments because of its utility in human face recognition approaches in [14, 15]. The last step in human face recognition requires classification of the facial image into one of the known classes based on the derived feature vector obtained in the previous stage. The radial basis function (RBF) neural network is also used as the classifier [15, 16]. The training of the RBF neural network is done, based on the hybrid learning algorithm (HLA) [17] and we have shown that the proposed feature extraction method with an RBF neural network classifier gives a faster training phase and yields a better recognition rate. The organization of this paper is as follows. Section 2 presents the face localization method. In Section 3, the face feature extraction is presented. Classifier techniques are described in Section 4 and finally, Sections 5 and 6 present the experimental results and conclusions.
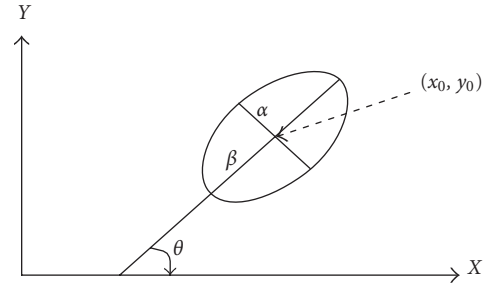

Figure 1: Face model based on ellipse model.

2. FACE LOCALIZATION METHOD

To ensure a robust and accurate feature extraction, the exact location of the face in an image is needed. The ultimate goal of the face localization is finding an object in an image as a face candidate whose shape resembles the shape of a face and, therefore, one of the key problems in building automated systems that perform face recognition task is face localization. Many algorithms have been proposed for face localization and detection, which are based on using shape [2, 4, 14], color information [3], motion [18], and so forth. A critical survey on face localization and detection can be found in [5]. In this paper, we have used a modified version of the shape information technique for the face localization presented in [2, 14].

Many researchers have concluded that an ellipse can generally approximate the face of a human. The localization algorithm utilizes the information about the edges of the facial image or the region over which the face is located [3, 14, 15]. The advantage of the region-based method is its robustness in the presence of noise and changes in illumination. In the region-based method, the connected components are determined by applying a region growing algorithm [3, 14], then, for each connected component with a given minimum size, the best-fit ellipse is computed using the properties of the geometric moments. To find a face region, an ellipse model with five parameters is used: X_0, Y_0 are the centers of the ellipse, θ is the orientation, and α and β are the minor and the major axes of the ellipse, respectively, as shown in Figure 1. To calculate these parameters, first we review the geometric moments. The geometric moments of order $p + q$ of a digital image are defined as

$$M_{pq} = \sum_x \sum_y f(x, y) x^p y^q, \tag{1}$$

where $p, q = 0, 1, 2, \ldots$ and $f(x, y)$ is the gray-scale value of the digital image at x and y location. The translation invariant central moments are obtained by placing origin at the center of the image

$$\mu_{pq} = \sum_x \sum_y f(x, y)(x - x_0)^p (y - y_0)^q, \tag{2}$$

where $x_0 = M_{10}/M_{00}$ and $y_0 = M_{01}/M_{00}$ are the centers of the connected components. Therefore, the center of the ellipse is

given by the center of gravity of the connected components. The orientation θ of the ellipse can be calculated by determining the least moment of inertia [2, 3, 14]

$$\theta = \frac{1}{2} \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right), \tag{3}$$

where μ_{pq} denotes the central moment of the connected components as described in (2). The length of the major and the minor axes of the best-fit ellipse can also be computed by evaluating the moment of inertia. With the least and the greatest moments of inertia of an ellipse defined as

$$\begin{aligned} I_{\text{Min}} &= \sum_x \sum_y \left[(x - x_0) \cos \theta - (y - y_0) \sin \theta \right]^2, \\ I_{\text{Max}} &= \sum_x \sum_y \left[(x - x_0) \sin \theta - (y - y_0) \cos \theta \right]^2, \end{aligned} \tag{4}$$

the length of the major and the minor axes are calculated from [3, 4, 14] as

$$\begin{aligned} \alpha &= \frac{1}{\pi \left[I_{\text{Max}}^3 / I_{\text{Min}} \right]^{1/8}}, \\ \beta &= \frac{1}{\pi \left[I_{\text{Min}}^3 / I_{\text{Max}} \right]^{1/8}}. \end{aligned} \tag{5}$$

To assess how well the best-fit ellipse approximates the connected components, we define a distance measure between the connected components and the best-fit ellipse as follows:

$$\phi_i = \frac{P_{\text{inside}}}{\mu_{00}}, \qquad \phi_o = \frac{P_{\text{outside}}}{\mu_{00}}, \tag{6}$$

where the P_{inside} is the number of background points inside the ellipse, P_{outside} is the number of points of the connected components that are outside the ellipse, and μ_{00} is the size of the connected components.

The connected components are closely approximated by their best-fit ellipses when ϕ_i and ϕ_o are as small as possible. We have named the threshold values for ϕ_i and ϕ_o as FCT. Our experimental study indicates that when FCT is less than 0.1, the connected component is very similar to ellipse; therefore it is a good candidate as a face region. If ϕ_i and ϕ_o are greater than 0.1, there is no face region in the input image, therefore, we reject it as a nonface image. An example of application of this method for locating face region candidates and rejecting nonface images has been presented in Figure 2.

3. FEATURE EXTRACTION TECHNIQUE

The aim of the feature extractor is to produce a feature vector containing all pertinent information about the face while having a low dimensionality. In order to design a good face recognition system, the choice of feature extractor is very crucial. To design a system with low to moderate complexity, the feature vectors created from feature extraction stage should contain the most pertinent information about the face to be recognized. In the statistics-based feature extraction approaches, global information is used to create a set of

feature vector elements to perform recognition. A mixture of irrelevant data, which are usually part of a facial image, may result in an incorrect set of feature vector elements. Therefore, data that are irrelevant to facial portion such as hair, shoulders, and background should be disregarded in the feature extraction phase.

Face recognition systems should be capable of recognizing face appearances in a changing environment. Therefore we use PZMI to generate the feature vector elements [14, 15]. Also the feature extractor should create a feature vector with low dimensionality. The low-dimensional feature vector reduces the computational burden of the recognition system; however, if the choice of the feature elements is not properly made, this in turn may affect the classification performance. Also, as the number of feature elements in the feature extraction step decreases, the neural network classifier becomes small with a simple structure. The proposed feature extractor in this paper yields a feature vector with low dimensionality, and, by disregarding irrelevant data from face portion of the image, it improves the recognition rate. The proposed feature extraction is done in two steps. In the first step, after face localization, we create a subimage which contains information needed for the recognition algorithm. In the second step, the feature vector is obtained by calculating the PZMI of the derived subimage.

3.1. Creating a subimage

To create a subimage for feature extraction phase, all pertinent information around the face region is enclosed in an ellipse while pixel values outside the ellipse are set to zero. Unfortunately, through creation of the subimage with the best-fit ellipse, as described in Section 2, many unwanted regions of the face image may still appear in this subimage, as shown in Figure 2. These include hair portion, neck, and part of the background as an example. To overcome this problem, instead of using the best-fit ellipse for creating a subimage, we have defined another ellipse. The proposed ellipse has the same orientation and center as the best-fit ellipse but the lengths of its major and minor axes are calculated from the lengths of the major and minor axes of the best-fit ellipse as follows:

$$A = \rho \cdot \alpha, \qquad B = \rho \cdot \beta, \tag{7}$$

where A and B are the lengths of the major and minor axes of the proposed ellipse, and α and β are the lengths of the major and minor axes of the best-fit ellipse that have been defined in (5). The coefficient ρ is called ACR and varies from 0 to 1. Figure 3 shows the effect of changing ACR while Figure 4 shows the corresponding subimages.

Our experimental results with 400 face images show that the best value for ACR is around 0.87. By using the above procedure, data that are irrelevant to facial portion are disregarded. The feature vector is then generated by computing the PZMI of the subimage obtained in the previous stage. It should be noted that the speed of computing the PZMI is considerably increased due to smaller pixel content of the subimages.
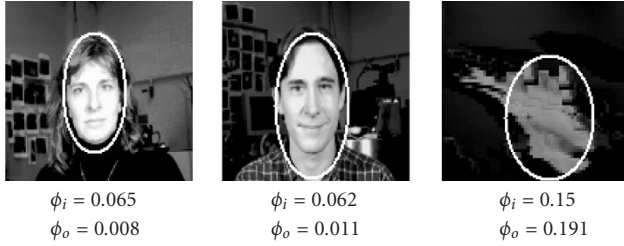
$\phi_i = 0.065$ $\phi_i = 0.062$ $\phi_i = 0.15$
$\phi_o = 0.008$ $\phi_o = 0.011$ $\phi_o = 0.191$

FIGURE 2: Distinguishing between face and nonface using best-fit ellipse and FCT threshold.

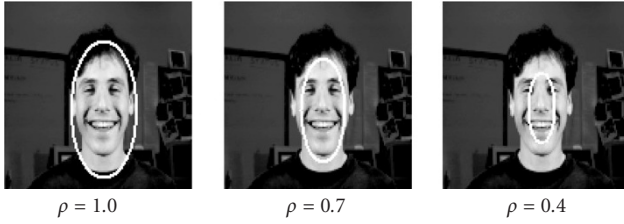$\rho = 1.0$ $\rho = 0.7$ $\rho = 0.4$

FIGURE 3: Different ellipses with respect to ACR.

FIGURE 4: Subimage formation based on different ellipses and ACR values.

3.2. Pseudo-Zernike moment invariant

Statistics-based approaches for feature extraction are very important in pattern recognition for their computational efficiency and their use of global information in an image for extracting features [15]. The advantages of considering orthogonal moments are that they are shift, rotation, and scale invariants and are very robust in the presence of noise. The invariant properties of moments are utilized as pattern sensitive features in classification and recognition applications [14, 19, 20, 21]. Pseudo-Zernike polynomials are well known and widely used in the analysis of optical systems. Pseudo-Zernike polynomials are orthogonal sets of complex-valued polynomials defined as (see [20, 21])

$$V_{nm}(x, y) = R_{nm}(x, y) \exp\left(jm\tan^{-1}\left(\frac{y}{x}\right)\right), \quad (8)$$

where $x^2 + y^2 \leq 1$, $n \geq 0$, $|m| \leq n$, and the radial polynomials $\{R_{n,m}\}$ are defined as

$$R_{n,m}(x, y) = \sum_{s=0}^{n-|m|} D_{n,|m|,s}(x^2 + y^2)^{(n-s)/2}, \quad (9)$$

where

$$D_{n,|m|,s} = (-1)^s \frac{(2n + 1 - s)!}{s!(n - |m| - s)!(n - |m| - s + 1)!}. \quad (10)$$

The PZMI of order n and repetition m can be computed using the scale invariant central moments $\text{CM}_{p,q}$ and the radial geometric moments $\text{RM}_{p,q}$ as follows [21, 22]:

$$
\begin{aligned}
\text{PZMI}_{nm} \\
= \frac{n+1}{\pi} \sum_{(n-m-s)\,\text{even},\,s=0}^{n-|m|} D_{n,|m|,s} \\
\times \sum_{a=0}^{k} \sum_{b=0}^{m} \binom{k}{a}\binom{m}{b}(-j)^b \text{CM}_{2k+m-2a-b,\,2a+b} \\
+ \frac{n+1}{\pi} \sum_{(n-m-s)\,\text{odd},\,s=0}^{n-|m|} D_{n,|m|,s} \\
\times \sum_{a=0}^{d} \sum_{b=0}^{m} \binom{d}{a}\binom{m}{b}(-j)^b \text{RM}_{2d+m-2a-b,\,2a+b},
\end{aligned} \quad (11)
$$

where $k = (n - s - m)/2$, $d = (n - s - m + 1)/2$, and also $\text{CM}_{p,q}$ and $\text{RM}_{p,q}$ are as follows:

$$
\begin{aligned}
\text{CM}_{p,q} &= \frac{\mu_{pq}}{M_{00}^{(p+q+2)/2}}, \\
\text{RM}_{p,q} &= \frac{\sum_x \sum_y f(x, y)(\widehat{x}^2 + \widehat{y}^2)^{1/2} \widehat{x}^p \widehat{y}^q}{M_{00}^{(p+q+2)/2}},
\end{aligned} \quad (12)
$$

where $\widehat{x} = x - x_0$, $\widehat{y} = y - y_0$, and x_0, y_0, μ_{pq}, and M_{pq} are defined in (1) and (2).

### 3.3. Selecting feature vector elements

After face localization and subimage creation, we calculate the PZMI inside each subimage as face features. For selecting the best order of the PZMI as face feature elements, we define a feature vector in face recognition application whose elements are based on the PZMI orders as follows:

$$\text{FV}_j = \{\text{PZMI}_{km}\}, \quad k = j, j+1, \ldots, N, \quad (13)$$

where j varies from 1 to $N - 1$, therefore, FV_j is a feature vector which contains all the PZMI from order j to N. Table 1 shows samples of feature vector elements for $j = 3, 5, 9$ when $N = 10$. Also, Figure 5 shows the number of feature vector elements relative to j value. As Figure 5 shows, when j increases, the number of elements in each feature vector (FV_j) decreases. These results are based on a value of $N = 10$. Our experimental study indicates that this method of selecting the pseudo-Zernike moment order as the feature elements allows the feature extractor to have a lower-dimensional vector while maintaining a good discrimination capability.

TABLE 1: Feature vector elements based on PZMI.

| j value | PZMI feature elements | | Number of feature elements |
	n value	M value	
3	3	0,1,2,3	60
	4	0,1,2,3,4	
	5	0,1,2,3,4,5	
	6	0,1,2,3,4,5,6	
	7	0,1,2,3,4,5,6,7	
	8	0,1,2,3,4,5,6,7,8	
	9	0,1,2,3,4,5,6,7,8,9	
	10	0,1,2,3,4,5,6,7,8,9,10	
6	6	0,1,2,3,4,5,6	45
	7	0,1,2,3,4,5,6,7	
	8	0,1,2,3,4,5,6,7,8	
	9	0,1,2,3,4,5,6,7,8,9	
	10	0,1,2,3,4,5,6,7,8,9,10	
9	9	0,1,2,3,4,5,6,7,8,9	21
	10	0,1,2,3,4,5,6,7,8,9,10	
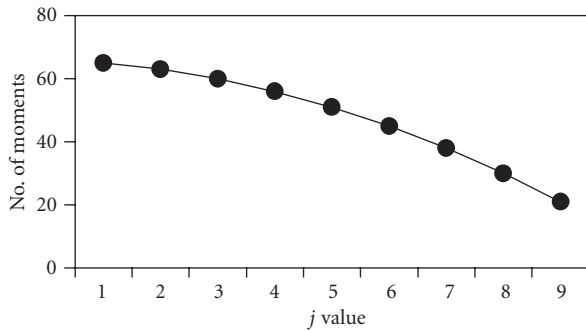
FIGURE 5: Number of feature elements (PZMI) with respect to j.

4. CLASSIFIER DESIGN

Neural network is widely used as a classifier in many face recognition systems. Neural networks have been employed and compared to conventional classifiers for a number of classification problems. The results have shown that the accuracy of the neural network approaches is equivalent to, or slightly better than, other methods. Also, due to the simplicity, generality, and good learning ability of the neural networks, these types of classifiers are found to be more efficient [23].

RBF neural networks have been found to be very attractive for many engineering problems because (1) they are universal approximators, (2) they have a very compact topology, and (3) their learning speed is very fast because of their locally tuned neurons [16, 17, 23, 24]. An important property of RBF neural networks is that they form a unifying link between many different research fields such as function approximation, regularization, noisy interpolation, and pattern recognition. Therefore, RBF neural networks serve as an
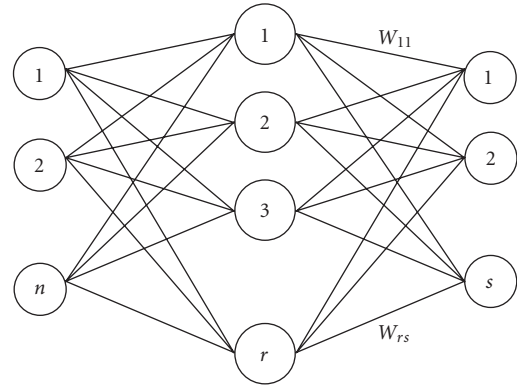

FIGURE 6: RBF neural network structure.

excellent candidate for pattern classification where attempts have been carried out to make the learning process in this type of classification faster than normally required for the multilayer feedforward neural networks [23, 25].

In this paper, an RBF neural network is used as a classifier in a face recognition system where the inputs to the neural network are feature vectors derived from the proposed feature extraction technique described in the previous section.

4.1. RBF neural network structure

An RBF neural network structure is shown in Figure 6. The construction of the RBF neural network involves three different layers with feedforward architecture. The input layer of the neural 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 hidden layers have unit weights and, as a result, do not have to be trained. The goal of the hidden layer is to cluster the data and reduce its dimensionality. In this structure, the hidden units are referred to as the RBF units. The RBF units are also fully connected to the output layer. The output layer supplies the response of the neural network to the activation pattern applied to the input layer. The transformation from the input space to the RBF-unit space is nonlinear (nonlinear activation function), whereas the transformation from the RBF-unit space to the output space is linear (linear activation function). The RBF neural network is a class of neural networks where the activation function of the hidden units is determined by the distance between the input vector and a prototype vector. The activation function of the RBF units is expressed as follows [24, 25]:

$$R_i(x) = R_i\left(\frac{\|x - c_i\|}{\sigma_i}\right), \quad i = 1, 2, \ldots, r, \qquad (14)$$

where x is an n-dimensional input feature vector, c_i is an n-dimensional vector called the center of the RBF unit, σ_i is the width of the RBF unit, and r is the number of the RBF units. Typically, the activation function of the RBF units is chosen as a Gaussian function with mean vector c_i and variance

vector σ_i as follows:

$$R_i(x) = \exp\left(-\frac{||x - c_i||^2}{\sigma_i^2}\right). \tag{15}$$

Note that σ_i^2 represents the diagonal entries of the covariance matrix of the Gaussian function. The output units are linear and the response of the jth output unit for input x is

$$y_j(x) = b(j) + \sum_{i=1}^{r} R_i(x)w_2(i, j), \tag{16}$$

where $w_2(i, j)$ is the connection weight of the ith RBF unit to the jth output node and $b(j)$ is the bias of the jth output. The bias is omitted in this network in order to reduce the neural network complexity [17, 24, 25]. Therefore,

$$y_j(x) = \sum_{i=1}^{r} R_i(x) \times w_2(i, j). \tag{17}$$

4.2. RBF neural network classifier design

To design a classifier based on RBF neural networks, we have set the number of input nodes in the input layer of the neural network equal to the number of feature vector elements. The number of nodes in the output layer is then set to the number of image classes. The number of RBF units as well as their characteristic initialization is carried out using the following steps [17].

Step 1. Initially, the RBF units are set equal to the number of outputs.

Step 2. For each class k $(k = 1, 2, \ldots, s)$, the center of the RBF unit is selected as the mean value of the sample features, belonging to the same class, that is,

$$C^k = \frac{\sum_{i=1}^{N^k} p^k(n, i)}{N^k}, \quad k = 1, 2, \ldots, s, \tag{18}$$

where $p^k(n, i)$ is the ith sample with n as the number of features belonging to class k and N^k is the number of images in the same class.

Step 3. For each class k, compute the distance d_k from the mean C^k to the farthest point p_f^k belonging to class k:

$$d_k = \left| p_f^k - C^k \right|, \quad k = 1, 2, \ldots, s. \tag{19}$$

Step 4. For each class k, compute the distance $dc(k, j)$ between the mean of the class and the mean of other classes as follows:

$$dc(k, j) = \left| C^k - C^j \right|, \quad j = 1, 2, \ldots, s, \ j \neq k. \tag{20}$$

Then, find $d_{\min}(k, l) = \min(dc(k, j))$ and check the relationship between $d_{\min}(k, l)$, d_k, and d_l. If $d_k + d_l \leq d_{\min}(k, l)$, then class k is not overlapping with other classes. Otherwise, class

k is overlapping with other classes and misclassifications may occur in this case.

Step 5. If two classes are overlapped strongly, we first split one of the classes into two to remove the overlap. If the overlap is not removed, the second class is also split. This requires addition of a new RBF unit to the hidden layer.

Step 6. Repeat Steps 2 to 5 until all the training sample patterns are classified correctly.

Step 7. The mean values of the classes are selected as the centers of RBF units.

4.3. Hybrid learning algorithm

The training of the RBF neural networks can be made faster than the methods used to train multilayer neural networks. This is based on the properties of the RBF units, and can lead to a two-stage training procedure. The first stage of the training involves determining output connection weights, which requires solution of a set of linear equations which can be done fast. In the second stage, the parameters governing the basis function (corresponding to the RBF units) are determined using an unsupervised learning method that requires solution of a set of nonlinear equations. The training of the RBF neural networks involves estimating output connection weights, centers, and widths of the RBF units. Dimensionality of the input vector, and the number of classes set the number of input and output units, respectively. In this paper, an HLA, which combines the gradient method and the linear least square (LLS) method, is used for training the neural network [17]. This is done in two steps. In the first step, the neural network connection weights in the output of the RBF units $(w_2(i, j))$ 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 widths (c and σ) of the RBF units are updated as described later.

4.3.1. Computing connection weights

Let r and s be the number of inputs and outputs, respectively, and assume that the number of u RBF outputs is generated for all training face patterns. For any input $P_i(p_{1i}, p_{2i}, \ldots, p_{ri})$, the jth output y_j of the RBF neural network in (14) can be calculated in a more compact form as follows:

$$W_2 \times R = Y, \tag{21}$$

where $R \in \mathfrak{R}^{u \times N}$ is the matrix of the RBF units, $W_2 \in \mathfrak{R}^{s \times u}$ is the output connection weight matrix, $Y \in \mathfrak{R}^{s \times N}$ is the output matrix, and N is the total number of sample face patterns. The relationship for error is defined by

$$E = ||T - Y||, \tag{22}$$

where $T = (t_1, t_2, \ldots, t_s)^T \in \mathfrak{R}^{s \times N}$ is the target matrix consisting of ones and zeros with each column having only

one nonzero element and identifies the processing pattern to which the given exemplar belongs.

Our objective is to find an optimal coefficient matrix $W_2' \in \mathfrak{R}^{s \times u}$ such that $E^T E$ is minimized. This is done by the well-known LLS method [16] as follows:

$$W_2' \times R = Y. \tag{23}$$

The optimal W_2' is given by

$$W_2' = T \times R^+, \tag{24}$$

where R^+ is the pseudoinverse of R and is given by

$$R^+ = (R^T R)^{-1} R^T. \tag{25}$$

We can compute the connection weights using (22) and (23) by knowing matrix R as follows:

$$W_2' = T (R^T R)^{-1} R^T. \tag{26}$$

4.3.2. Defining the center and width of the RBF units

Here, the center and width of the RBF units (R matrix) are adjusted by taking the negative gradient of the error function, E^n, for the nth sample pattern which is given by [25]

$$E^n = \frac{1}{2} \sum_{k=1}^{s} \left(t_k^n - y_k^n \right)^2, \quad n = 1, 2, \dots, N, \tag{27}$$

where y_k^n and t_k^n represent the kth real output and target output of the nth sample face pattern, respectively.

For the RBF units with the center C and the width σ, the update value for the center can be derived from (25) by the chain rule as follows:

$$
\begin{aligned}
\Delta C^n(i, j) &= -\xi \frac{\partial E^n}{\partial C^n(i, j)} \\
&= 2\xi \sum_{k=1}^{s} y_k^n \cdot w_2^n(k, j) \cdot R_j^n \cdot \frac{(P_i^n - C^n(i, j))}{(\sigma_j^n)^2}
\end{aligned} \tag{28}
$$

and the update value for the width is computed as follows:

$$
\begin{aligned}
\Delta \sigma_j^n &= -\xi \frac{\partial E^n}{\partial \sigma_j^n} \\
&= 2\xi \sum_{k=1}^{s} y_k^n \cdot w_2^n(k, j) \cdot R_j^n \cdot \frac{(P_i^n - C^n(i, j))}{(\sigma_j^n)^3},
\end{aligned} \tag{29}
$$

where $i = 1, 2, \dots, r$, $j = 1, 2, \dots, u$, P_i^n is the ith input variable of the nth sample face pattern, and ξ is the learning rate. $\Delta C^n(i, j)$ is the update value for the ith variable of the center of the jth RBF unit based on the nth training pattern. $\Delta \sigma_j^n$ is the update value for the width of the jth RBF unit with respect to the nth training pattern.

FIGURE 7: Samples of facial images in ORL database.

5. EXPERIMENTAL RESULTS

To check the utility of our proposed algorithm, experimental studies are carried out on the ORL database images of Cambridge University. This database contains 400 facial images from 40 individuals in different states, taken between April 1992 and April 1994. The total number of images for each person is 10. None of the 10 images is identical to any other. They vary in position, rotation, scale, and expression. The changes in orientation have been accomplished by each person rotating a maximum of 20 degrees in the same plane, as well as each person changing his/her facial expression in each of the 10 images (e.g., open/close eyes, 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 and varying facial details (glasses/no glasses). All the images were taken against a dark homogeneous background. Each image was digitized and presented by a 112×92 pixel array whose gray levels ranged between 0 and 255. Samples of database used are shown in Figure 7.

Experimental studies have been done by dividing database images into training and test sets. A total of 200 images are used to train and another 200 are used to test. Each training set consists of 5 randomly chosen images from the same class in the training stage. There is no overlap between the training and test sets. In the face localization step, the shape information algorithm with FCT has been applied to all images. Subsequently, calculating the PZMI of the subimage, which is created with ACR value, creates the feature vector. The RBF classifier is trained using the HLA method with the training sets, and finally the classifier error rate is computed with respect to the test images. In this study, the classifier error rate is considered as the number of misclassifications in the test phase over the total number of test images. The experimental study conducted in this paper evaluates the effect of the PZMI orders, ACR, FCT, and the presence of noise in images on the recognition rate. Also the utility of the learning algorithm on the recognition rate is studied.
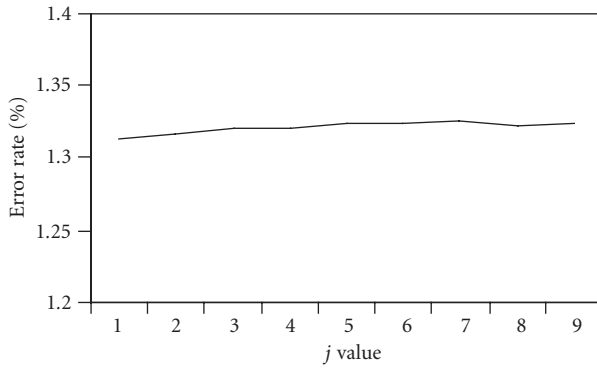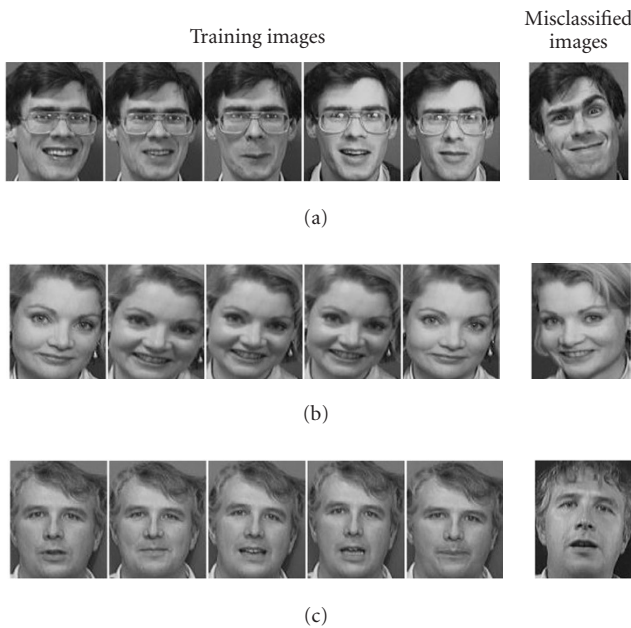
FIGURE 8: Error rate with respect to j.

FIGURE 10: Error rate variation with respect to ACR value.

FIGURE 9: Misclassified images with corresponding training images.

5.1. Effect of moment orders

In this phase of the experiment, simulation has been done, based on the j value defined in (13). The ACR has been set equal to one and the RBF neural network classifier has been trained for each j value based on the training images. Figure 8 shows the error rate of the system with respect to j. This figure shows that when j increases, the error rate almost remains unchanged. In contrast, as Figure 5 has shown, when j increases, the number of feature elements of the feature vector in the feature extraction step decreases. This observation is interesting because in spite of the decrease in the number of feature elements, the error rate has remained unchanged. Also, these results show that higher orders of the PZMI contain more and useful information for face recognition process. Figure 9 shows the misclassified images and their corresponding training sets for the value of $j = 9$. As indicated in Figure 9a, the misclassified image in this set is
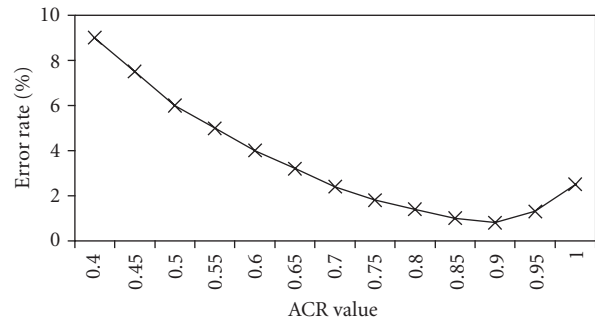
substantially different from the training set in terms of its facial expression, while the reason for misclassification of the images in Figures 9b and 9c can be explained with the effect of the irrelevant data in the test images with respect to their training sets.

5.2. Effect of ACR when disregarding irrelevant data

For the purpose of evaluating how the irrelevant data of a facial image such as hair, neck, shoulders, and background will influence the recognition results, we have chosen the PZMI of orders 9 and 10 (set $j = 9$) for feature extraction. We have also selected FCT = 0.1 for the face localization algorithm and the RBF neural network with the HLA as the classifier. We varied the ACR value and evaluated the recognition rate of the proposed algorithm. Figure 10 shows the effect of ACR values on the error rate.

As Figure 10 shows, the error rate varies as the ACR values change. At ACR = 1, a recognition rate of 98.7% is obtained (Section 5.1). Now, by changing ACR and calculating the correct recognition rate, it is observed that at ACR = 0.87, a recognition rate of 99.3% can be achieved. This clearly indicates the importance of the ACR in improving the recognition performance.

5.3. Effect of FCT when distinguishing between face and nonface regions

To evaluate the effect of FCT in the face localization step and distinguishing between face and nonface images, we prepared 20 nonface images and applied them to the system. Figure 2 shows a sample of such images with $\phi_i = 0.13$ and $\phi_o = 0.191$. We varied the FCT value and evaluated the number of nonface images that passed through the system. Experimental results showed that FCT = 0.1 is a good threshold for distinguishing between face and nonface images. Figure 11 shows this result.

5.4. Effect of the HLA method on the RBF classifier

To investigate the effect of the learning method HLA on the RBF neural network, the ACR has been set equal to one and we have created four categories of feature vectors based on the order (n) of the PZMI. In the first category with $n = 1, 2, \ldots 6$, all the moments of PZMI are considered
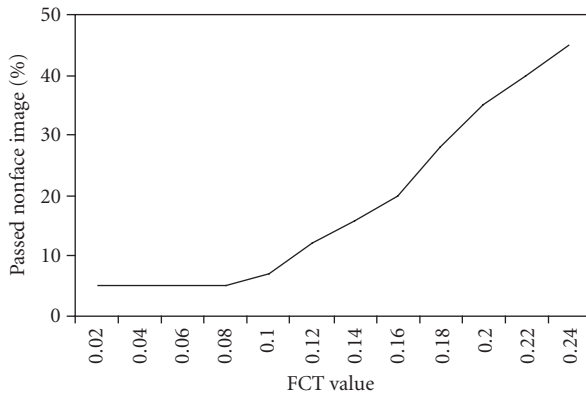
FIGURE 11: Effect of FCT.

FIGURE 12: Error rate with respect to noise amplitude.

as feature vector elements. The number of the feature vector elements in this category is 27. In the second category, $n = 4, 5, 6, 7$ is chosen. All the moments of each order included in this category are summed up to create a feature vector of size 26. In the third category, $n = 6, 7, 8$ is considered, and the feature vector has 24 elements. Finally, in the last category, $n = 9, 10$ is considered with 21 feature elements. The neural network classifier was trained in each category based on the training images, and subsequently, the system was tested using the test images. The experimental results are shown in Table 2. This table indicates that in the training phase of the RBF neural network classifier, the number of epochs decreases when the PZMI orders increase. On the other hand, the RBF neural network with the HLA learning method has converged faster in the training phase when higher orders of the PZMI are used in comparison with the lower orders of PZMI. Also, this table indicates that the HLA method in the training phase has a lower root mean squared error (RMSE) with a good discrimination capability.

To compare the HLA with other learning algorithms, we have developed the k-mean clustering algorithm for training the RBF neural networks [17, 26] and applied it to the database with the same feature extraction technique. Table 3 shows the comparison between the two learning methods. It is seen from Table 3 that the HLA method converges faster than the k-mean clustering which needs fewer epochs in the training phase. Also, the RMSE during the training phase for the HLA is smaller than that for the k-mean clustering learning algorithm.

5.5. Performance evaluation in the presence of noise

To evaluate the performance of the feature extraction method with ACR parameter, PZMI, and the RBF neural network for human face recognition in the presence of noise, a white Gaussian noise of zero mean and different amplitudes (in gray-level image) has been added to the clean images. The recognition process was then applied to the noisy images. Figure 12 shows the error rate of the recognition process with respect to different values of the noise amplitude. This figure indicates that the proposed technique for human face recog-
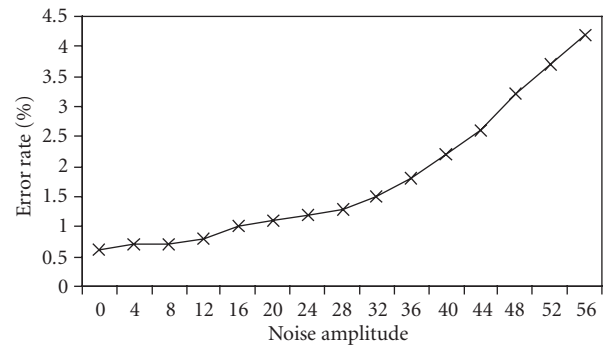
nition is very robust in the presence of noise. In Figure 13, samples of noisy images have been shown.

5.6. Comparison with other human face recognition systems

To compare the effectiveness of the proposed method with other algorithms, the PZMI of orders 9 and 10 with 21 feature elements, FCT = 0.1, ACR = 0.87, and the RBF neural network with the HLA learning algorithm have been used. This study compares the proposed technique with the methods that used the same ORL database. These include the shape information neural network (SINN) [15], convolution neural network (CNN) [27], nearest feature line (NFL) [28], and the fractal transformation (FT) [29]. In this comparison, the training set and the test set were derived in the same way as was suggested in [15, 27, 28, 29]: the 10 images from each class of the 40 persons were randomly partitioned into sets, resulting in 200 training images and 200 test images, with no overlap between the two. Also in this study, the error rate was defined, as was used in [15, 27, 28, 29], to be the number of misclassified images in the test phase over the total number of test images. To conduct the comparison, an average error rate which has been used in [15, 27, 28, 29] was utilized as defined below:

$$E_{\text{ave}} = \frac{\sum_{i=1}^{m} N_m^i}{m N_t}, \tag{30}$$

where m is the number of experimental runs, each being performed on a random partitioning of the database into sets, N_m^i is the number of misclassified images for the ith run, and N_t is the number of total test images for each run. Table 4 shows the comparison between the different techniques using the same ORL database in terms of E_{ave}. In this table, the CNN error rate was based on the average of three runs as given in [27], while for the NFL, the average error rate of four runs was reported in [28]. Also an average run of one for the FT [29] and four runs for the SINN [15] were carried out as suggested in the respective papers. The average error rate of the proposed method for the four runs is 0.682%, which yields the lowest error rate of these techniques on the ORL database.

TABLE 2: Effect of the HLA method in learning phase.

Category	Features vectors		Training phase		Testing phase	
	No. of feature flements		No. of epochs	RMSE	No. of misclassified	Error rate
$n = 1, 2, \ldots, 6$	27		$80 \sim 100$	$0.09 \sim 0.06$	15	7.5%
$n = 4, 5, 6, 7$	26		$60 \sim 80$	$0.06 \sim 0.04$	9	4.5%
$n = 6, 7, 8$	24		$45 \sim 60$	$0.04 \sim 0.02$	6	3%
$n = 9, 10$	21		$30 \sim 45$	$0.04 \sim 0.01$	3	1.3%

TABLE 3: Comparison between the two learning techniques.

Feature category	K-mean clustering		HLA method	
	No. of epochs	RMSE	No. of epochs	RMSE
$n = 1, 2, \ldots, 6$	$135 \sim 120$	$0.12 \sim 0.09$	$80 \sim 100$	$0.09 \sim 0.06$
$n = 4, 5, 6, 7$	$120 \sim 95$	$0.09 \sim 0.06$	$60 \sim 80$	$0.06 \sim 0.04$
$n = 6, 7, 8$	$95 \sim 70$	$0.06 \sim 0.04$	$45 \sim 60$	$0.04 \sim 0.02$
$n = 9, 10$	$70 \sim 50$	$0.04 \sim 0.02$	$30 \sim 45$	$0.04 \sim 0.01$

Noise amplitude = 10

Noise amplitude = 20

Noise amplitude = 40

Noise amplitude = 50
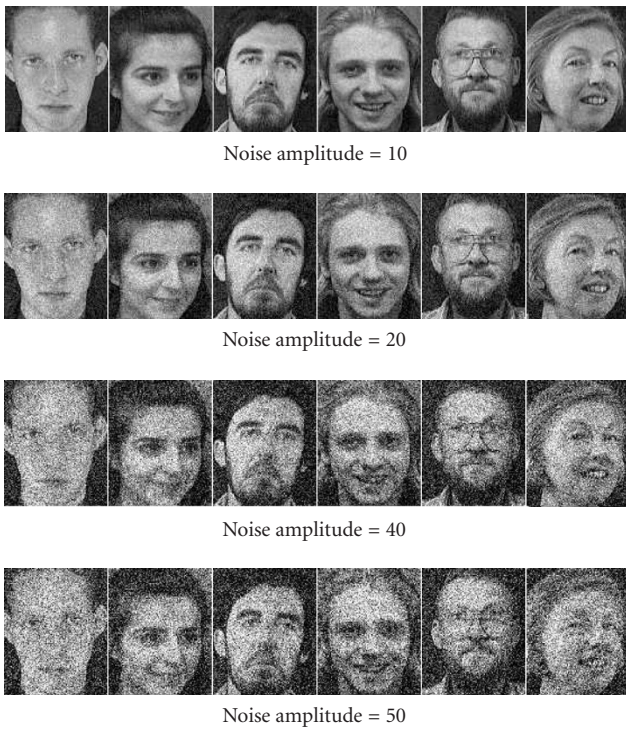
FIGURE 13: Samples of noisy images with different noise value.

TABLE 4: Error rates of different approaches.

Methods	No. of experimental (m)	E_{ave}%
CNN [27]	3	3.83
NFL [28]	4	3.125
FT [29]	1	1.75
SINN [15]	4	1.323
Proposed method	3	0.682

The paper introduces several parameters for efficient and robust feature extraction technique as well as the RBF neural network learning algorithm. These include FCT, ACR, and selection of the PZMI orders and the HLA method. Exhaustive experimentation was carried out to investigate the effect of varying these parameters on the recognition rate. We have shown that high order PZMI contains very useful information about the facial images, and that the HLA method affects the learning speed. We have also indicated the optimum values of the FCT and ACR corresponding to the best recognition results on the ORL database. The robustness of the proposed algorithm in the presence of noise is also investigated. The highest recognition rate of 99.3% with ORL database was obtained using the proposed algorithm. We have implemented and tested some of the existing face recognition techniques on the same ORL database. This comparative study indicates the usefulness and the utility of the proposed technique.

6. CONCLUSIONS

This paper presented an efficient method for the recognition of human faces in frontal view of facial images. The proposed technique utilizes a modified feature extraction technique, which is based on a flexible face localization algorithm followed by PZMI. An RBF neural network with the HLA method was used as a classifier in this recognition system.

REFERENCES

[1] M. A. Grudin, "On internal representation in face recognition systems," *Pattern Recognition*, vol. 33, no. 7, pp. 1161–1177, 2000.

[2] J. Haddadnia and K. Faez, "Human face recognition using shape information and pseudo Zernike moments," in *Proc. 5th International Fall Workshop Vision, Modeling and Visualization (VMV '00)*, pp. 113–118, Saarbrücken, Germany, November 2000.

[3] K. Sobottka and I. Pitas, "Face localization and facial feature extraction based on shape and color information," in *Proc. IEEE International Conference on Image Processing (ICIP '96)*, vol. 3, pp. 483–486, Lausanne, Switzerland, September 1996.

[4] J. Wang and T. Tan, "A new face detection method based on shape information," *Pattern Recognition Letter*, vol. 21, no. 6-7, pp. 463–471, 2000.

[5] E. Hjelmas and B. K. Low, "Face detection: a survey," *Computer Vision And Image Understanding*, vol. 83, no. 3, pp. 236–274, 2001.

[6] M. Bichsel and A. P. Pentland, "Human face recognition and the face image sets topology," *CVGIP: Image Understanding*, vol. 59, no. 2, pp. 254–261, 1994.

[7] V. Bruce, P. J. B. Hancock, and A. M. Burton, "Comparisons between human and computer recognition of faces," in *Proc. 3rd International Conference On Automatic Face and Gesture Recognition (FG '98)*, pp. 408–413, Nara, Japan, April 1998.

[8] L.-F. Chen, H.-Y. M. Liao, J.-C. Lin, and C.-C. Han, "Why recognition in a statistics-based face recognition system should be based on the pure face portion: a probabilistic decision-based proof," *Pattern Recognition*, vol. 34, no. 7, pp. 1393–1403, 2001.

[9] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[10] F. Goudail, E. Lange, T. Iwamoto, K. Kyuma, and N. Otsu, "Face recognition system using local autocorrelations and multiscale integration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 1024–1028, 1996.

[11] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[12] K. K. Sung and T. Poggio, *Example-Based Learning for View-Based Human Face Detection*, vol. 1521 of *A.I. Memo*, MIT Press, Cambridge, Mass, USA, 1994.

[13] H.-Y. M. Liao, C.-C. Han, and G.-J. Yu, "face + hair + shoulders + background = face," in *Proc. Workshop on 3D Computer Vision '97*, pp. 91–96, The Chinese University of Hong Kong, Hong Kong, China, May 1997.

[14] J. Haddadnia, M. Ahmadi, and K. Faez, "An efficient method for recognition of human faces using higher orders pseudo Zernike moment invariant," in *Proc. 5th International Conference On Automatic Face and Gesture Recognition (FG '02)*, pp. 330–335, Washington, DC, USA, May 2002.

[15] J. Haddadnia, K. Faez, and P. Moallem, "Neural network based face recognition with moments invariant," in *Proc. IEEE International Conference on Image Processing (ICIP '01)*, vol. 1, pp. 1018–1021, Thessaloniki, Greece, October 2001.

[16] J. Haddadnia and K. Faez, "Human face recognition using radial basis function neural network," in *Proc. 3rd International Conference On Human and Computer (HC '00)*, pp. 137–142, Aizu, Japan, September 2000.

[17] J. Haddadnia, M. Ahmadi, and K. Faez, "A hybrid learning RBF neural network for human face recognition with pseudo Zernike moment invariant," in *IEEE International Joint Conference On Neural Network (IJCNN '02)*, pp. 11–16, Honolulu, Hawaii, USA, May 2002.

[18] R. Herpers, G. Verghese, K. Derpanis, et al., "Detection and tracking of faces in real environments," in *IEEE International Workshop on Recognition, Analysis, and Tracking of Face and Gesture in Real-Time Systems*, pp. 96–104, Corfu, Greece, September 1999.

[19] S. X. Liao and M. Pawlak, "On the accuracy of Zernike moments for image analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1358–1364, 1998.

[20] C. H. Teh and R. T. Chin, "On image analysis by the methods of moments," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 496–513, 1988.

[21] S. O. Belkasim, M. Shridhar, and M. Ahmadi, "Pattern recognition with moment invariants: a comparative study and new results," *Pattern Recognition*, vol. 24, no. 12, pp. 1117–1138, 1991.

[22] R. R. Bailey and M. Srinath, "Orthogonal moment features for use with parametric and non-parametric classifiers," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 4, pp. 389–399, 1996.

[23] W. Zhou, "Verification of the nonparametric characteristics of backpropagation neural networks for image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 771–779, 1999.

[24] L. Yingwei, N. Sundararajan, and P. Saratchandran, "Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm," *IEEE Transactions on Neural Networks*, vol. 9, no. 2, pp. 308–318, 1998.

[25] J.-S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[26] S. Gutta, J. R. J. Huang, P. Jonathon, and H. Wechsler, "Mixture of experts for classification of gender, ethnic origin, and pose of human faces," *IEEE Transactions on Neural Networks*, vol. 11, no. 4, pp. 948–960, 2000.

[27] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: a convolutional neural network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[28] S. Z. Li and J. Lu, "Face recognition using the nearest feature line method," *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp. 439–443, 1999.

[29] T. Tan and H. Yan, "Face recognition by fractal transformations," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP '99)*, vol. 6, pp. 3537–3540, Phoenix, Ariz, USA, March 1999.

Javad Haddadnia received his B.S. and M.S. degrees in electrical and electronic engineering with the first rank from Amirkabir University of Technology, Tehran, Iran, in 1993 and 1995, respectively. He received his Ph.D. degree in electrical engineering from Amirkabir University of Technology, Tehran, Iran in 2002. He joined Tarbiat Moallem University of Sabzevar in Iran. His research interests include neural network, digital image processing, computer vision, and face detection and recognition. He has published several papers in these areas. 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.
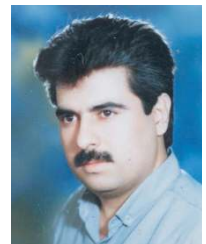
Majid Ahmadi received his B.S. in electrical engineering from Arya-Mehr University and Ph.D. degree in electrical engineering from Imperial College of London University in 1971 and 1977, respectively. Dr. Ahmadi has been a Professor in the Department of Electrical and Computer Engineering, University of Windsor since 1980. Dr. Ahmadi has conducted research in the areas of 2D signal processing, image processing and computer vision, pattern recognition, neural network architectures, applications and VLSI realization, computer arithmetic, and MEMS. He has published over 300 papers in these areas. He is a Fellow of the IEEE (USA) and a Fellow of the IEE (UK).

Karim Faez was born in Semnan, Iran. He received his B.S. degree in electrical engineering from Tehran Polytechnic University with 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. Professor 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 from April 1989 to September 1992. Professor Faez was the chairman of planning committee for Computer Engineering and Computer Science of Ministry of Science, Research, and Technology (from 1988 to 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.