Face Recognition using Principal Component Analysis and RBF Neural Networks

S. Thakur[1], J. K. Sing[*2], D. K. Basu[2], M. Nasipuri[2], M. Kundu[2]

[1]Department of Information Technology, Netaji Subhas Engineering College, Kolkata, India
[2]Department of Computer Science & Engineering, Jadavpur University, Kolkata, India
[*]Corresponding author: Email: jksing@ieee.org.

***Abstract*— In this paper, an efficient method for face recognition using principal component analysis (PCA) and radial basis function (RBF) neural networks is presented. Recently, the PCA has been extensively employed for face recognition algorithms. It is one of the most popular representation methods for a face image. It not only reduces the dimensionality of the image, but also retains some of the variations in the image data. After performing the PCA, the hidden layer neurons of the RBF neural networks have been modelled by considering intra-class discriminating characteristics of the training images. This helps the RBF neural networks to acquire wide variations in the lower-dimensional input space and improves its generalization capabilities. The proposed method has been evaluated using the AT&T (formerly ORL) and UMIST face databases. Experimental results show that the proposed method has encouraging recognition performance.**

***Keywords*- Face recognition, RBF neural networks, PCA, AT&T (ORL) and UMIST face databases.**

I. INTRODUCTION

Face recognition has been an active research area in the pattern recognition and computer vision domains [Turk and Pentland, 1991; Samal and Iyengar, 1992; Er, Wu, Lu, Toh, 2002; Sing, Basu, Nasipuri and Kundu, 2007; Yang and Paindovoine, 2003]. It has many potential applications, such as, surveillance, credit cards, passport, security, etc. A number of methods have been proposed in the last decades [Chellapa, Wilson and Sirohey, 1995]. In the field of face recognition, the dimension of the facial images is very high and require considerable amount of computing time for classification. The classification and subsequent recognition time can be reduced by reducing dimension of the image data. Principal component analysis (PCA) [Turk and Pentland, 1991] is one of the popular methods used for feature extraction and data representation. It not only reduces the dimensionality of the image, but also retains some of the variations in the image data and provides a compact representation of a face image. The key idea of the PCA method is to transform the face images into a small set of characteristics feature images, called *eigenfaces*, which are the principal components of the initial training set of the face images. PCA yields projection directions that maximize the total scatter across all classes, i.e., across all face images.

In recognition process a test image is projected into the lower-dimension face space spanned by the eigenfaces and then classified either by using statistical theory or a classifier.

The PCA method was developed in 1991 [Turk and Pentland, 1991]. In [Belhumeur, Hespanha and Kriegman, 1997], the PCA method is used for dimension reduction for linear discriminate analysis (LDA), generating a new paradigm, called *fisherface*. The fisherface approach is more insensitive to variations of lighting, illumination and facial expressions. However, this approach is more computationally expensive than the PCA approach.

In this paper, we propose a new method for face recognition using PCA and RBF neural networks. The RBF neural networks have been used due to its simple structure and faster learning ability [Moody and Darken, 1989; Girosi and Poggio, 1990]. The face features are extracted by the PCA method, reducing the dimensionality of input space. It has been seen that variations between the images of the same subject due to variation in pose, orientation, etc. are quite high. Therefore, to achieve high recognition rate, structural information of face images of the same subject is considered for classification process. This has been realized by identifying sub-clusters corresponding to a subject separately using a clustering algorithm. Then the prototypes of these sub-clusters are used to model the hidden layer neurons of the RBF neural networks. This process also improves its generalization capabilities.

The remaining part of the paper is organized as follows. Section II provides the procedure of extracting face features using the PCA method. Section III presents the design methodologies and training procedure of the RBF neural network. The experimental results on the AT&T and UMIST face databases are presented in Section IV. Section V presents the details of implementation of the proposed method and timing. Finally, Section VI draws the conclusion remarks.

II. FEATURE EXTRACTION USING PCA

The facial features are extracted using the PCA method. Let there are R face images in the training set and each image $\boldsymbol{X}_i$ is a 2-dimensional array of size $m \times n$ of intensity values. An image $\boldsymbol{X}_i$ can be converted into a vector of D ($D = m \times n$) pixels, where, $\boldsymbol{X}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. The rows of pixels of the image are placed one after another to form the vector. Define the training set of R images by $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_R) \subset \Re^{D \times R}$. The covariance matrix is defined as follows:

$$\Gamma = \frac{1}{R}\sum_{i=1}^{R}\left(X_i - \overline{X}\right)\left(X_i - \overline{X}\right)^T = \boldsymbol{\Phi}\boldsymbol{\Phi}^T \qquad (1)$$

where $\boldsymbol{\Phi} = (\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2, \ldots, \boldsymbol{\Phi}_R) \subset \Re^{D \times R}$ and $\overline{X} = \frac{1}{R}\sum_{i=1}^{R} X_i$, which is the mean image of the training set. The dimension of the covariance matrix $\boldsymbol{\Gamma}$ is $D \times D$. Then, the eigenvalues and eigenvectors are calculated from the covariance matrix $\boldsymbol{\Gamma}$. Let $\mathbf{Q} = (\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_r) \subset \Re^{D \times R}$ ($r<R$) be the r eigenvectors corresponding to r largest non-zero eigenvalues. Each of the r eigenvectors is called an *eigenface*. Now, each of the face images of the training set $\mathbf{X}_i$ is projected into the eigenface space to obtain its corresponding eigenface-based feature $\mathbf{Z}_i \subset \Re^{r \times R}$, which is defined as follows:

$$\mathbf{Z}_i = \mathbf{Q}^T\mathbf{Y}_i, \; i = 1, 2, \ldots, R \qquad (2)$$

where $\mathbf{Y}_i$ is the mean-subtracted image of $\mathbf{X}_i$.

In order to recognize the test images, each of the test images is transformed into the eigenface space using the equation (2) and then fed to the RBF neural networks as inputs for classification.

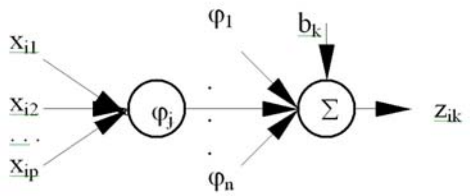

Fig. 1. Structure of a hidden and output layer neuron of an RBF-NN

III. DESIGN METHODOLOGIES AND TRAINING PROCEDURE OF THE RBF NEURAL NETWORKS

We have used RBF neural networks for classifying the images due to its simple structure and faster learning abilities [Moody and Darken, 1989; Girosi and Poggio, 1990]. Image features are extracted using the PCA method as described in the previous section. The performance of the RBF neural networks heavily depends on its training procedure. The structural information and training procedure of the proposed RBF neural networks are discussed in the following sub-sections.

A. The Structure and Function of the RBF Neural Networks

An RBF neural network (RBF-NN) with three layers is employed for classification in the proposed method. The architecture of a hidden and output layer neuron of the RBF-NN is shown in Fig. 1. The function of an RBF-NN can be viewed as a process, which maps a p-dimensional input pattern $\mathbf{x}_i$ from input space to a decision space of m-dimension. In doing so, a non-linear function and a linear function are used in the hidden layer and output layer, respectively. Conventionally, a Gaussian function is used as a non-linear function, which is defined as follows:

$$\varphi_j(x_i) = \exp\left(-\frac{\| x_i - c_j \|^2}{2\sigma_j^2}\right), \; j = 1,2,\ldots,n; i = 1,2,\ldots,N \qquad (3)$$

where $\mathbf{c_j}$ and σ_j are the centre and the width of the receptive field of the j^{th} hidden layer neuron, respectively, n and N are the total number of hidden layer neurons and total number of input patterns, respectively. The output of the k^{th} output layer neuron is generated by the following linear function:

$$z_{ik} = \sum_{j=1}^{n}\varphi_j(x_i)w_{kj} + b_k w_k; \; k = 1, 2, \ldots, m \qquad (4)$$

where w_{kj} is the weight of the link between the j^{th} hidden layer neuron and the k^{th} output layer neuron, b_k and w_k are unit positive bias and weight of the link to the k^{th} output layer neuron from the bias neuron, respectively and m is the number of actual class of the problem at hand. In the present study, the value of m is equals to the total number of individuals in the database.

B. Training Procedure of the RBF Neural Networks

The RBF-NN has been trained in two stages. In the first stage, the centres of the Gaussian functions associated with each of the hidden layer neurons and their corresponding widths are selected. Then, in the second stage, weights of the links between the hidden layer and output layer are estimated.

The choices of hidden layer neurons are very important to the RBF-NN to achieve good generalization. Because, an RBF-NN acquire knowledge from these locally tuned neurons, which represent the subspaces of the input space. In the face recognition problem, since the variations between the face images of a person, taken by varying pose, orientation, etc., are quite high, it is reasonable to find the sub-clusters from the image space associated with each person. This process helps to acquire structural information more appropriately from the image space spanned by the training set. The prototypes of these sub-clusters are modelled as the hidden layer neurons of the

RBF-NN. To realize the above process we have clustered the training images associated with each person separately. The clustering process is defined as follows:

1) Let there are *N* images of a person in the training set. We have to find *K (K< N)* sub-clusters from the image space spanned by the *N* training images.
2) Initially, all the training images are assigned as *N* distinct clusters. Set $k = N$.
3) Compute inter-cluster distance d(i, j) by the following equation:

$$\underset{i \neq j}{d(i, j)} = \left\| C_i - C_j \right\|; i, j = 1,2,..., k \quad (5)$$

where $\mathbf{C}_i$ and $\mathbf{C}_j$ are the ^{i}th and ^{j}th clusters. $\|\cdot\|$ is the Euclidean norm.

4) Find the two closest clusters $\mathbf{C}_i$ and $\mathbf{C}_j$ by the following equation:

$$d_{min}(i, j) = \underset{i,j}{\arg\min}\{d(i, j)\}; i, j = 1,2,..., N, i \neq j \quad (6)$$

5) Find a new cluster by averaging the above two closest clusters and set $k = k$-1.
6) Repeat the steps 3 - 5 until $k = K$.
7) Repeat steps 1 - 6 for all the subjects in the training set separately.

Once the hidden layer neurons are selected, the next step is to estimate the widths of the Gaussian functions (basis functions) associated with each of them. In RBF-NN, the hidden layer neurons acquire knowledge from the overlapping sub-spaces in the feature space. The amount of overlapping is controlled by the widths of the corresponding Gaussian functions. If the overlapping is too small, the outputs of the RBF neurons will be small for the inputs for which they have been designed. Therefore, RBF-NN will not generalize well. On the other hand, if the overlapping is too large, interaction with the other classes will be high and RBF neurons will produce high output for inputs belonging to the other classes. This also leads to poor generalization of the RBF-NN. Therefore, the widths are to be estimated in such a way, which minimizes the overlapping between different classes and maximizes the generalization ability of the RBF-NN. To realize the above criteria we have estimated the widths in the following ways:

1) Compute the distance between mean cluster and members of the cluster corresponding to a class *k*, i. e. intra cluster distances.
2) Find maximum distance d_{max} from the mean and define the width σ_A^k as follows:

$$\sigma_A^k = \frac{d_{max}}{\sqrt{\ln \beta}} \quad (7)$$

where β is a confidence factor.

3) Compute distance between mean clusters; i. e. inter cluster distances and find the nearest class distance d_{min}.
4) Determine another form of widths σ_B^k by considering an overlapping factor η for class *k* by

$$\sigma_B^k = \eta \times d_{min} \quad (8)$$

where η controls the overlapping between different classes.

5) Finally, the widths of the Gaussian functions associated with class *k* is defined as follows:

$$\sigma^k = \max(\sigma_A^k, \sigma_B^k) \quad (9)$$

The values for β and η are selected for which RBF-NN provides best performance. In our experiments we have found β=0.75 and η=1.25 after many experimental runs. The weights of the links between hidden layer and output layer are estimated using least-mean-square (LMS) algorithm [Haykin, 1999].

IV. EXPERIMENTAL RESULTS

The performance of the proposed method has been evaluated on the AT&T Laboratories Cambridge database (formerly ORL database) [ORL face database] and the UMIST face database [Graham and Allinson, 1998]. The AT&T database is used to test performance of the proposed method under the condition of minor variations of rotation and scaling, the UMIST database is used to examine the performance of the method when the angle of rotation of the facial images is quite large.

Many experiments have been carried out, by considering different configurations of the proposed method, to test its performance. The recognition rate is defined as the ratio of the total number of correct recognition by the method to the total number of images in the test set for a single experimental run. Therefore, the average recognition rate, R_{avg}, of the method is defined as follows:

$$R_{avg} = \frac{\sum_{i=1}^{q} n_{cls}^i}{q \times n_{tot}} \quad (10)$$

where q is the number of experimental runs. The n_{cls}^i is the number of correct recognition in the ith run and n_{tot} is the total number of faces under test in each run. Consequently, the average error rate, E_{avg}, may be defined as 100-R_{avg}.

The performances of the method have also been evaluated using rejection criteria. Here, each of the output layer units of the RBF-NN should recognize all the faces of its own class and reject the faces belonging to the other classes (intruders). To calculate the success rate of the method two parameters namely, the sensitivity and specificity of the method have been evaluated. Sensitivity is defined as the probability of correctly recognizing a face, whereas specificity refers to the probability of correct rejection of an intruder. They can be computed as follows:

$$Sensitivity = \frac{T_P}{T_P + F_N} \tag{11}$$

$$Specificity = \frac{T_N}{T_N + F_P} \tag{12}$$

where T_P is the total number of faces correctly recognized (true positive) and F_N is the total number of faces falsely recognized as intruders (false negative) in each run. T_N is the total number faces of the other classes truly rejected as intruders (true negative) and F_P is the total number of faces of other classes falsely recognized as its own (false positive) in each run. If a face is falsely rejected as an intruder then, the same face is also falsely recognized by another class as its own. Therefore, the values of F_N and F_P will be same. Thus, the percentage of the sensitivity is also called as the recognition rate. In our experiments, first we estimated the parameters of the method by evaluating the recognition or sensitivity rate. Then the specificity of the method is evaluated.

A. Experiments on the AT&T Face Database

The AT&T database contains 400 gray-scale images of 40 persons. Each person has 10 images, each having a resolution of 112×92, and 256 gray levels. Images of the individuals have been taken varying light intensity, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background, with tilt and rotation up to 20^o and scale variation up to 10%. Sample face images of a person are shown in Fig. 2.

The experiments were carried out in three different strategies; mainly randomly partitioning the database, n-fold cross validation test and leave-one out strategy.

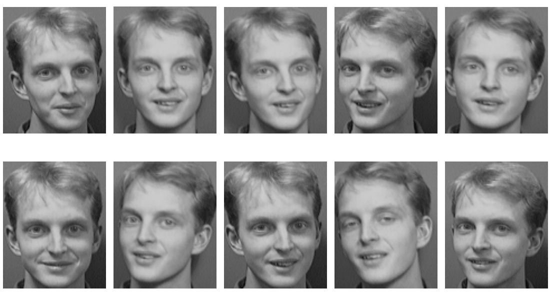

Fig. 2. Sample images of a subject from the AT&T database

1) Randomly partitioning the database: In this experimental strategy, five images (s=5) from each person of the database are selected randomly for training set and the rest of the face images are included in the test set. Therefore, a total of 200 faces are used to train and another 200 faces are used to test the RBF-NN. It should be noted that there is no overlap between the training and test images. In this way ten (q=10) different training and test sets have been generated. The eigenfaces from one of the 10 training sets corresponding to 10 largest eigenvalues are shown in Fig. 3.

Fig. 3. Eigenfaces from one of the 10 training sets corresponding to 10 largest eigenvalues

In these experiments, three sub-clusters are selected from the five images of a person using the clustering algorithm defined in sub-section III.B. Therefore 120 (3×40) hidden layer neurons are used in the RBF-NN. The average recognition rates by varying number of principal components (PCs) in 10 experimental runs are shown in Fig. 4. The best average recognition rate (93.05%) is achieved when 60, 70, 90 and 100 PCs are used. The maximum and minimum recognition rates among the 10 experimental runs (for PCs = 10-100) are found to be 95.50% and 85%, respectively.

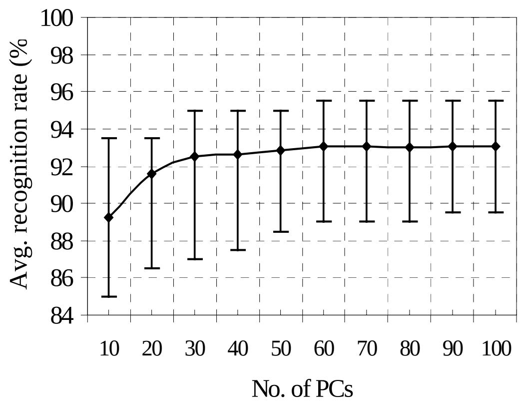

Fig. 4. Recognition rate on the ORL database by varying PCs. The upper and lower extrema of the error bars represents the maximum and minimum values, respectively

We have also evaluated the performance of the proposed method using rejection criteria. In these experiments, each of the output layer units of the RBF-NN should recognize all the faces of its own class and reject the other faces belonging to the other classes. For example, with s = 5, each of the output layer unit (associated to a class) should recognize five faces of its own class and reject the 195 (5×39) faces of the remaining other 39 classes. Table I shows the average specificity rate of the proposed method using 60 PCs in 10 experimental runs. The average specificity rate is found to be 99.82% for s=5.

TABLE I. Average specificity rate for s=5 with 60 PCs and 120 hidden layer neurons in 10 experimental runs on the ORL database

Number of samples/individual	s=5
Number of total samples/training set	200
Number of Test samples	8000
Number of faces to be recognized	200
Number of faces to be rejected	7800
Average number of faces recognized (T_P)	186
Average number of faces rejected (T_N)	7786
Average false positive (F_P)	14
Average false negative (F_N)	14
Average Specificity rate	99.82%

2) *N-fold cross validation test:* In this study, the AT&T database is randomly divided into ten-folds, taking one image of a person into a fold. Therefore, each fold consists of 40 images, each one corresponding to a different person. For ten-folds cross validation test, in each experimental run, nine folds are used to train the RBF-NN and remaining one fold is used for testing. Therefore, training and test sets consist of 360 and 40 images, respectively in a particular experimental run. Also in this experiment, 120 (3×40) hidden layer neurons are selected in the same way that has been discussed in the previous sub-section. The average recognition rates by varying number of principal components (PCs) in ten-folds cross validation tests are shown in Fig. 5. The best average recognition rate (97.00%) is achieved when 50 and 60 PCs are used in the system. The maximum and minimum recognition rates among the ten-folds experimental runs (for PCs = 20-60) are found to be 100% and 85%, respectively.

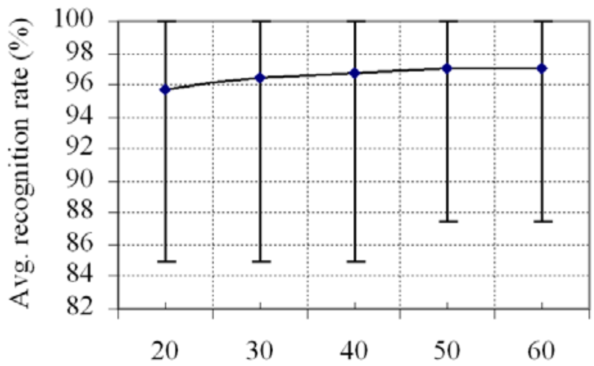

Fig. 5. Recognition rate on the ORL database by varying PCs in ten-folds cross validation test. The upper and lower extrema of the error bars represents the maximum and minimum values, respectively

Table II shows the average specificity rate of the proposed method using 60 PCs in 10-folds cross validation test. The average specificity rate is found to be 99.94%.

3) *Leave-one-out method:* In this section, experiments were carried out using the "*leave-one-out*" strategy. To classify an image of a subject, the image is removed from the database of M images and placed into a test set. Remaining M-1 images are used in the corresponding training set. In this way, experiments are performed M times, removing one image from the database at a time. For the ORL database, we have performed 400 experimental runs for the database of 400 images. Table III shows the average sensitivity and specificity rates using 60 PCs and 120 hidden layer neurons. We have achieved 97.75% and 99.94% average sensitivity and specificity rates, respectively.

TABLE II. Average specificity rate using 60 PCs and 120 hidden layer neurons in 10-folds cross validation test on the ORL database

Number of samples/individual	s=9
Number of total samples/training set	360
Number of Test samples	1600
Number of faces to be recognized	40
Number of faces to be rejected	1560
Average number of faces recognized (T_P)	39
Average number of faces rejected (T_N)	1559
Average false positive (F_P)	1
Average false negative (F_N)	1
Average Specificity rate	99.94%

TABLE III. Experimental results using leave-one-out strategy on the ORL database

Method	# of hidden layer neurons	# of PCs	Success rate (%)
Sensitivity	120	60	97.75
Specificity	120	60	99.94

TABLE IV. Comparison of the performances between the proposed method and the other methods on the ORL database

Experiment	Method	R_{avg} (%)
Randomly partition, 5 images/subject	**Present method**	93.05
	PCA+FLD approach, as reported by [Xiong, Swamy and Ahmad, 2005] in comparing with their method	90.98
Leave-one-out	**Present method**	97.75
	Eigenface approach, as reported by [Yang, Ahuja, Kriegman, 2000] in comparing with their method	97.25
	Eigenface approach, as reported by [Liu, Tang, Lu and Ma, 2006] in comparing with their method	97.25

4) *Comparison with other methods:* Recently, a number of methods have been reported based on PCA method and evaluated on the ORL database. For fair comparison, we have compared the performance of our proposed method with those, which have used the same experimental methodology. Table IV shows the comparison of the performances between the proposed method and the methods, as reported by [Xiong, Swamy and Ahmad, 2005; Yang, Ahuja, Kriegman, 2000; Liu, Tang, Lu and Ma, 2006] in comparing with their methods. It may be noted that the present method outperforms the other methods, in both the experiments, which are carried out by partitioning the database randomly (5 images per subject in the training and test sets) and leave-one-out strategy.

B. Experiments on the UMIST Face Database

The UMIST face database is a multi-view database, consisting of 575 gray-scale images of 20 people (subject), each covering a wide range of poses from profile to frontal views. Each subject also covers a range of race, sex and appearance. Each image has a resolution of 112×92, and 256 gray levels. Unlike the ORL database, the number of images per people is not fixed; it varies from 19 to 48. Fig. 6 shows some of the sample images of a subject from the database.

Fig. 6. Some sample images of a subject from the UMIST database.

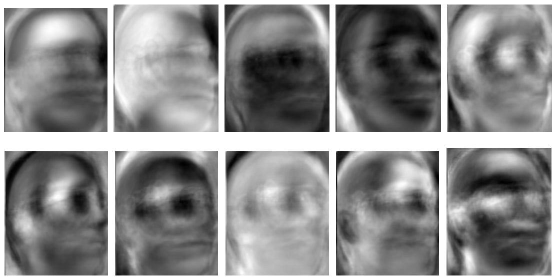

Fig. 7. Eigenfaces from one of the 10 training sets corresponding to 10 largest eigenvalues

1) *Randomly partitioning the database:* Eight images per subject have been selected randomly to form a training set of 160 images. Remaining 415 images have been used in the corresponding test set. In this way 10 different training and test sets have been formed to evaluate the performance of the proposed method. The recognition rate of the proposed method has been evaluated by averaging the recognition rates obtained over these 10 experimental runs. It should be again noted that there is no overlap between the training and test sets in a particular experimental run. On the ORL database, we have achieved best average recognition rate using 60 PCs. Therefore, in this experiment, we have used 60 PCs and performance of the method is evaluated by varying the number of hidden layer neurons of the RBF-NN. Fig. 7 shows the eigenfaces from one of the 10 training sets corresponding to 10 largest eigenvalues.

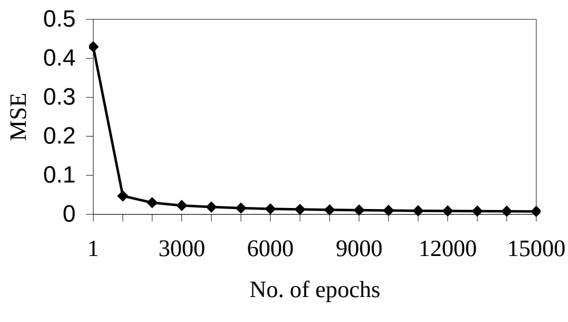

Fig. 8. Convergence of MSE on the UMIST database

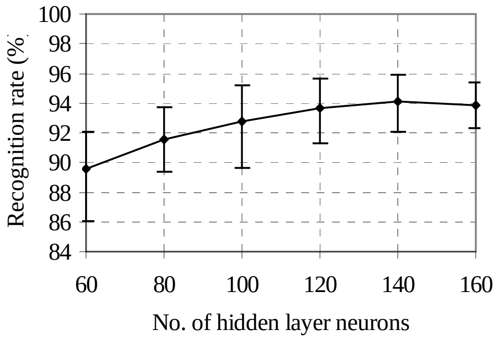

Fig. 9. Recognition rate on the UMIST database by varying hidden layer neurons. The upper and lower extrema of the error bars represents the maximum and minimum values, respectively

Fig. 8 shows the convergence of mean square error (MSE) at the output layer of the RBF-NN with the increasing numbers of training epochs. The graph is plotted with the average values in 10 experimental runs.

Fig. 9 shows the average recognition rate of the proposed method over the 10 experimental runs. We have achieved best average recognition rate of 94.10% using 140 hidden layer neurons and 60 PCs. The maximum and minimum recognition rates over 10 experimental runs are found to be 95.90% and 92.05%, respectively.

Table V shows the average specificity rate of the proposed method using 60 PCs in 10 experimental runs. The average specificity rate is found to be 99.70%.

TABLE V. Average specificity rate for s=8 with 60 PCs and 140 hidden layer neurons in 10 experimental runs on the UMIST database

Number of samples/individual	s=8
Number of total samples/training set	160
Number of Test samples	8300
Number of faces to be recognized	415
Number of faces to be rejected	7885
Average number of faces recognized (T_P)	391
Average number of faces rejected (T_N)	7861
Average false positive (F_P)	24
Average false negative (F_N)	24
Average Specificity rate	99.70%

2) N-fold cross validation test: Since the number of images per subject varies from 19 to 48, we have randomly divided the database into 19 folds, taking one image of a subject into a fold. Therefore, in each fold there are 20 images, each one corresponding to a different subject. For 19-folds cross validation test, in each experimental run, 18 folds are used to train the RBF-NN and remaining one fold is used for testing. Therefore, training and test sets consist of 360 and 20 images, respectively in a particular experimental run. In this experiment, 300 (15×20) hidden layer neurons are generated in the same way, which has been discussed in sub-section III.B. The success rates in terms of sensitivity and specificity are presented in Table VI. The average sensitivity and specificity are found to be 98.42% and 99.92%, respectively.

TABLE VI. Experimental results using 19-folds cross validation test on the UMIST database

Method	# of hidden layer neurons	# of PCs	Success rate (%)
Sensitivity	300	60	98.42
Specificity	300	60	99.92

3) Leave-one-out method: In this section, experiments were carried out using the “leave-one-out” strategy as discussed in sub-section IV.A.3. In this strategy, we have performed 575 experimental runs for the database of 575 images. Table VII shows the average sensitivity and specificity rates using 60 PCs and 300 hidden layer neurons. We have achieved 99.48% and 99.97% average sensitivity and specificity rates, respectively.

TABLE VII. Experimental results using leave-one-out strategy on the UMIST database

Method	# of hidden layer neurons	# of PCs	Success rate (%)
Sensitivity	300	60	99.48
Specificity	300	60	99.97

4) Comparison with other methods: We have also compared the performances of the proposed method with other method, which have used the UMIST database and adopted same experimental methodology. Table VIII shows the comparison of the performances between proposed method and the method, as reported by [Xiong, Swamy and Ahmad, 2005] in comparing with their method. It may be noted that the present method outperforms the other method.

TABLE VIII. Comparison of the performances between the proposed method and the other methods on the UMIST database

Experiment	Method	R_{avg} (%)
Randomly partition, 8 images/subject	**Present method**	94.10
	PCA approach, as reported by [Xiong, Swamy and Ahmad, 2005] in comparing with their method	93.60

V. Implementation and Timing

The proposed face recognition method has been implemented in C programming language on an IBM Intel Pentium 4 Hyper-Threading technology, 3.0 GHz computer with 2 GB DDR-II RAM running on Fedora 9 Linux Operating Systems. In case of UMIST database, the proposed method (with 60 PCA features, 140 hidden layer neurons, 20 output layer neurons, β=0.75 and η=1.25) takes approximately 1 minute 50 seconds to complete a run of 10000 epochs, each one having 160 training images for the determination of the required parameters. Once the parameters are determined, it takes approximately 1.5 milliseconds to recognize a face. Thus, the present method will be able to recognize faces in interframe periods of video and also in other real-time applications.

VI. Conclusion

This paper presents a face recognition method based on PCA and RBF neural networks. Facial features are extracted by the PCA method, which reduces the dimensionality of the original face images while preserving some discriminating features within the training images. After performing the PCA, structural information is acquired corresponding to each person from lower-dimensional training images. This structural information is used to model the hidden layer neurons of the RBF-NN. The proposed method is evaluated on the AT&T and the UMIST face databases using three different testing strategies; i) randomly partitioning the database, ii) n-folds cross validation test and iii) leave-one-out method. The experimental results obtained on both the face databases are found to be quite promising and better than some of the PCA-based methods reported earlier.

Acknowledgment

This work was partially supported by the BOYSCAST Fellowship 2006, Department of Science & Engineering, Government of India, (Ref. No. SR/BY/E-23/05), CMATER and the SRUVM projects of the Department of Computer Science & Engineering, Jadavpur University, Kolkata, India. The author, S. Thakur would like to thank Netaji Subhas Engineering College, Kolkata for providing computing facilities and allowing time for conducting research works. The author, J. K. Sing would like to thank Dr. Punam Kumar Saha, Associate Professor, Department of Electrical & Computer Engineering, University of Iowa, Iowa City, Iowa, USA, for his constructive suggestions and providing necessary computing facilities in his laboratory. The author, D. K. Basu would also like to thank the AICTE for providing him the Emeritus Fellowship (F. No.: 1-51/RID/EF(13)/2007-08, dated 28-02-2008).

References

[1] Belhumeur P. N., Hespanha J. P., and Kriegman D. J. 1997, "Eigenfaces versus fisherfaces: recognition using class specific linear projection", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 7, pp. 711-720.

[2] Chellapa R., Wilson C., Sirohey S. 1995, "Human and machine recognition of faces: a survey", *Proc. of the IEEE*, vol. 83, no. 5, pp. 705-741.

[3] Er M. J., Wu S., Lu J. and Toh H. L. 2002, "Face recognition with radial basis function (RBF) neural networks", *IEEE Trans. Neural Networks*, vol. 13, no. 3, pp. 697-710.

[4] Girosi F. and Poggio T. 1990, "Networks and the best approximation property", *Biol. Cybern.*, vol. 63, no. 3, pp. 169-176.

[5] Graham D. B and Allinson N. M. 1998, "Characterizing Virtual Eigensignatures for General Purpose Face Recognition", *(in) Face Recognition: From Theory to Applications*, NATO ASI Series F, Computer and Systems Sciences, vol. 163. H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman-Soulie and T. S. Huang (eds), pp. 446-456.

[6] Haykin S. 1999, Neural Networks a Comprehensive Foundation. Prentice-Hall Inc., 2nd Ed.

[7] Liu O., Tang X., Lu H., and Ma S. 2006, "Face Recognition using kernel scatter-difference-based discriminant analysis", *IEEE Trans Neural Networks*, vol. 17, no. 4, pp. 1081-1085.

[8] Moody J. and Darken C. J. 1989, "Fast learning in networks of locally tuned processing units", *Neural Computation*, vol. 1, no. 2, pp. 281-294.

[9] ORL face database. *AT&T Laboratories*, Cambridge, U. K. [Online] Available: http://www.uk.research.att.com/facedatabase.html.

[10] Samal A. and Iyengar P. 1992, "Automatic recognition and analysis of human faces and facial expressions: A survey", *Pattern Recognition*, vol. 25, no. 1, pp. 65-77.

[11] Sing J. K. Basu D. K., Nasipuri M. and Kundu M. 2007, "Face recognition using point symmetry distance-based RBF network", *Applied Soft Computing*, vol. 7, no. 1, pp. 58-70.

[12] Turk M. and Pentland A. 1991, "Eigenface for recognition", *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86.

[13] Xiong H., Swamy M. N. S. and Ahmad M. O. 2005, "Two-dimensional FLD for face recognition", *Pattern Recognition*, vol. 38, no. 7, pp. 1121-1124.

[14] Yang F., Paindovoine M. 2003, "Implementation of an RBF neural network on embedded systems: real-time face tracking and identity verification", *IEEE Trans. Neural Networks*, vol. 14, no. 5, pp. 1162-1175.

[15] Yang M.-H., Ahuja N. and Kraegman D., 2000, "Face recognition using kernel eigenfaces", *Proc. IEEE Int'l. Conf. Image Proc.*, pp. 37-40.

AUTHOR'S BIOGRAPHIES

Sweta Thakur received her B.E. (Computer Technology) degree from Nagpur University and M.E. (Computer Science & Engineering) degree from Jadavpur University in 2001 and 2004, respectively. Ms. Thakur has been a faculty member of the Department of Information Technology, Netaji Subhas Engineering College, Kolkata since August 2004. She is currently pursuing her doctorate in engineering at the Jadavpur University. Her research interests include face recognition, pattern recognition, image processing.

Jamuna Kanta Sing received his B.E. (Computer Science & Engineering) degree from Jadavpur University in 1992, M.Tech. (Computer & Information Technology) degree from Indian Institute of Technology (IIT), Kharagpur in 1993 and Ph.D. (Engineering) degree from Jadavpur University in 2006. Dr. Sing has been a faculty member of the Department of Computer Science & Engineering, Jadavpur University since March 1997. He has published more than 15 papers in International Journals and Conference Proceedings. He is a member of the IEEE, USA. His research interest includes face recognition/detection, medical image processing, and pattern recognition.

Dipak Kumar Basu received his B.E.Tel.E., M.E.Tel., and Ph.D. (Engg.) degrees from Jadavpur University, in 1964, 1966 and 1969 respectively. Prof. Basu was a faculty member of the Department of Computer Science & Engineering, Jadavpur University from 1968 to January 2008. He is presently an A.I.C.T.E. Emeritus Fellow at the Department of Computer Science & Engineering, Jadavpur University. His current fields of research interest include pattern recognition, image processing, and multimedia systems. He is a senior member of the IEEE, USA, Fellow of IE (India) and WBAST, Kolkata, India and a former Fellow of Alexander Von Humboldt Foundation, Germany.

Mita Nasipuri received her B.E.E., M.E.Tel.E. and Ph.D. (Engg.) degrees from Jadavpur University, in 1979, 1981 and 1990, respectively. Prof. Nasipuri has been a faculty member of the Department of Computer Science & Engineering, Jadavpur University since 1987. Her current research interest includes image processing, pattern recognition, and multimedia systems. She is a senior member of the IEEE, USA, Fellow of IE (India) and WBAST, Kolkata, India.

Mahantapas Kundu received his B.E.E., M.E.Tel.E. and Ph.D. (Engg.) degrees from Jadavpur University, in 1983, 1985 and 1995, respectively. Prof. Kundu has been a faculty member of the Department of Computer Science & Engineering, Jadavpur University since 1988. His current research interest includes image processing, pattern recognition, multimedia database, and artificial intelligence.