6

PCA and LDA based Neural Networks for Human Face Recognition

Alaa Eleyan and Hasan Demirel
Eastern Mediterranean University
Northern Cyprus

1. Introduction

After 9/11 tragedy, governments in all over the world started to look more seriously to the levels of security they have at their airports and borders. Countries annual budgets were increased drastically to have the most recent technologies in identification, recognition and tracking of suspects. The demand growth on these applications helped researchers to be able to fund their research projects. One of most common biometric recognition techniques is face recognition. Although face recognition is not as accurate as the other recognition methods such as fingerprints, it still grabs huge attention of many researchers in the field of computer vision. The main reason behind this attention is the fact that the face is the conventional way people use to identify each others.
Over the last few decades, a lot of researchers gave up working in the face recognition problem due to the inefficiencies of the methods used to represent faces. The face representation was performed by using two categories. The First category is *global approach* or *appearance-based*, which uses holistic texture features and is applied to the face or specific region of it. The second category is *feature-based* or *component-based*, which uses the geometric relationship among the facial features like mouth, nose, and eyes. (Wiskott et al., 1997) implemented feature-based approach by a geometrical model of a face by 2-D elastic graph. Another example of feature-based was done by independently matching templates of three facial regions (eyes, mouth and nose) and the configuration of the features was unconstrained since the system didn't include geometrical model (Brunelli & Poggio, 1993).
Principal components analysis (PCA) method (Sirovich & Kirby, 1987; Kirby & Sirovich, 1990) which is also called eigenfaces (Turk & Pentland, 1991; Pentland & Moghaddam, 1994) is appearance-based technique used widely for the dimensionality reduction and recorded a great performance in face recognition. PCA based approaches typically include two phases: training and classification. In the training phase, an eigenspace is established from the training samples using PCA and the training face images are mapped to the eigenspace for classification. In the classification phase, an input face is projected to the same eigenspace and classified by an appropriate classifier. Contrasting the PCA which encodes information in an orthogonal linear space, the linear discriminant analysis (LDA) method (Belhumeur et al., 1997; Zhao et al., 1998) which also known as fisherfaces method is another example of appearance-based techniques which encodes discriminatory information in a linear separable space of which bases are not necessarily orthogonal.

Source: Face Recognition, Book edited by: Kresimir Delac and Mislav Grgic, ISBN 978-3-902613-03-5, pp.558, I-Tech, Vienna, Austria, June 2007

In this chapter, two face recognition systems, one based on the PCA followed by a feedforward neural network (FFNN) called PCA-NN, and the other based on LDA followed by a FFNN called LDA-NN, are explained. The two systems consist of two phases which are the PCA or LDA feature extraction phase, and the neural network classification phase. The introduced systems provide improvement on the recognition performances over the conventional LDA and PCA face recognition systems.
The neural networks are among the most successful decision making systems that can be trained to perform complex functions in various fields of applications including pattern recognition, optimization, identification, classification, speech, vision, and control systems. In FFNN the neurons are organized in the form of layers. The FFNN requires a training procedure where the weights connecting the neurons in consecutive layers are calculated based on the training samples and target classes. After generating the eigenvectors using PCA or LDA methods, the projection vectors of face images in the training set are calculated and then used to train the neural network. These architectures are called PCA-NN and LDA-NN for eigenfaces and fisherfaces methods respectively.
The first part of the chapter introduces PCA and LDA techniques which provide theoretical and practical implementation details of the systems. Both of the techniques are explained by using wide range of illustrations including graphs, flowcharts and face images. The second part of the chapter introduces neural networks in general and FFNN in particular. The training and test phases of FFNN are explained in detail. Finally the PCA-NN and LDA-NN face recognition systems are explained and the performances of the respective methods are compared with conventional PCA and LDA based face recognition systems.

2. Principal Component Analysis

Principal component analysis or *karhunen-loève transformation* (Papoulis, 2002) is standard technique used in statistical pattern recognition and signal processing for data reduction and Feature extraction (Haykin, 1999). As the pattern often contains redundant information, mapping it to a feature vector can get rid of this redundancy and yet preserve most of the intrinsic information content of the pattern. These extracted features have great role in distinguishing input patterns.
A face image in 2-dimension with size $N \times N$ can also be considered as one dimensional vector of dimension N^2. For example, face image from ORL (Olivetti Research Labs) database with size 112×92 can be considered as a vector of dimension 10,304, or equivalently a point in a 10,304 dimensional space. An ensemble of images maps to a collection of points in this huge space. Images of faces, being similar in overall configuration, will not be randomly distributed in this huge image space and thus can be described by a relatively low dimensional subspace. The main idea of the principle component is to find the vectors that best account for the distribution of face images within the entire image space. These vectors define the subspace of face images, which we call "face space". Each of these vectors is of length N^2, describes an $N \times N$ image, and is a linear combination of the original face images. Because these vectors are the eigenvectors of the covariance matrix corresponding to the original face images, and because they are face-like in appearance, we refer to them as "eigenfaces".

Let the training set of face images be $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$, then the average of the set is defined by

$$\Psi = \frac{1}{M} \sum_{n=1}^{M} \Gamma_n \tag{1}$$

Each face differs from the average by the vector

$$\Phi_i = \Gamma_i - \Psi \tag{2}$$

This set of very large vectors is then subject to principal component analysis, which seeks a set of M orthonormal vectors, U_m , which best describes the distribution of the data. The k[th] vector, U_k , is chosen such that

$$\lambda_k = \frac{1}{M} \sum_{n=1}^{M} \left(U_k^T \Phi_n \right)^2 \tag{3}$$

is a maximum, subject to

$$U_I^T U_k = \delta_{Ik} = \begin{cases} 1, & if\ I = k \\ 0, & otherwise \end{cases} \tag{4}$$

The vectors U_k and scalars λ_k are the eigenvectors and eigenvalues, respectively of the covariance matrix

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T \tag{5}$$

where the matrix A $=[\Phi_1\ \Phi_2 \ldots \Phi_M]$. The covariance matrix C, however is $N^2 \times N^2$ real symmetric matrix, and calculating the N^2 eigenvectors and eigenvalues is an intractable task for typical image sizes. We need a computationally feasible method to find these eigenvectors.

Consider the eigenvectors v_i of A^TA such that

$$A^T A v_i = \mu_i v_i \tag{6}$$

Premultiplying both sides by A, we have

$$AA^T A v_i = \mu_i A v_i \tag{7}$$

where we see that Av_i are the eigenvectors and μ_i are the eigenvalues of $C = A\,A^T$.

Following these analysis, we construct the $M \times M$ matrix $L = A^TA$, where $L_{mn} = \Phi_m^T \Phi_n$, and find the M eigenvectors, v_i , of L. These vectors determine linear combinations of the M training set face images to form the eigenfaces U_I .

$$U_I = \sum_{k=1}^{M} v_{Ik} \Phi_k \ , \quad I = 1, \ldots, M \tag{8}$$

With this analysis, the calculations are greatly reduced, from the order of the number of pixels in the images (N^2) to the order of the number of images in the training set (M). In practice, the training set of face images will be relatively small ($M << N^2$), and the calculations become quite manageable. The associated eigenvalues allow us to rank the eigenvectors according to their usefulness in characterizing the variation among the images. The eigenface images calculated from the eigenvectors of L span a basis set that can be used to describe face images. (Sirovich & Kirby, 1987, 1990) evaluated a limited version of this framework on an ensemble of 115 images (M = 115) images of Caucasian males digitized in a controlled manner, and found that 40 eigenfaces (M' = 40) were sufficient for a very good description of face images. In practice, a smaller M' can be sufficient for identification, since accurate reconstruction of the image is not a requirement. In the framework of face recognition, the operation is a pattern recognition task rather than image reconstruction. The eigenfaces span an M' dimensional subspace of the original N^2 image space and hence, the M' significant eigenvectors of the L matrix with the largest associated eigenvalues, are sufficient for reliable representation of the faces in the face space characterized by the eigenfaces. Examples of ORL face database and eigenfaces after applying the eigenfaces algorithm are shown in Figure 1 and Figure 2, respectively.

Figure 1. Samples face images from the ORL database

A new face image (Γ) is transformed into its eigenface components (projected onto "face space") by a simple operation,

$$w_k = U_k^T(\Gamma - \Psi) \tag{9}$$

for $k = 1,...,M'$. The weights form a projection vector,

$$\Omega^T = [w_1 \; w_2 .... w_{M'}] \tag{10}$$

describing the contribution of each eigenface in representing the input face image, treating the eigenfaces as a basis set for face images. The projection vector is then used in a standard pattern recognition algorithm to identify which of a number of predefined face classes, if any, best describes the face. The face class Ω_k can be calculated by averaging the results of the eigenface representation over a small number of face images of each individual. Classification is performed by comparing the projection vectors of the training face images with the projection vector of the input face image. This comparison is based on the Euclidean Distance between the face classes and the input face image. This is given in Eq. (11). The idea is to find the face class k that minimizes the Euclidean Distance. Figure 3 shows the testing phase of the PCA approach.

$$\varepsilon_k = \left\| (\Omega - \Omega_k) \right\| \tag{11}$$

Where Ω_k is a vector describing the k^{th} faces class.

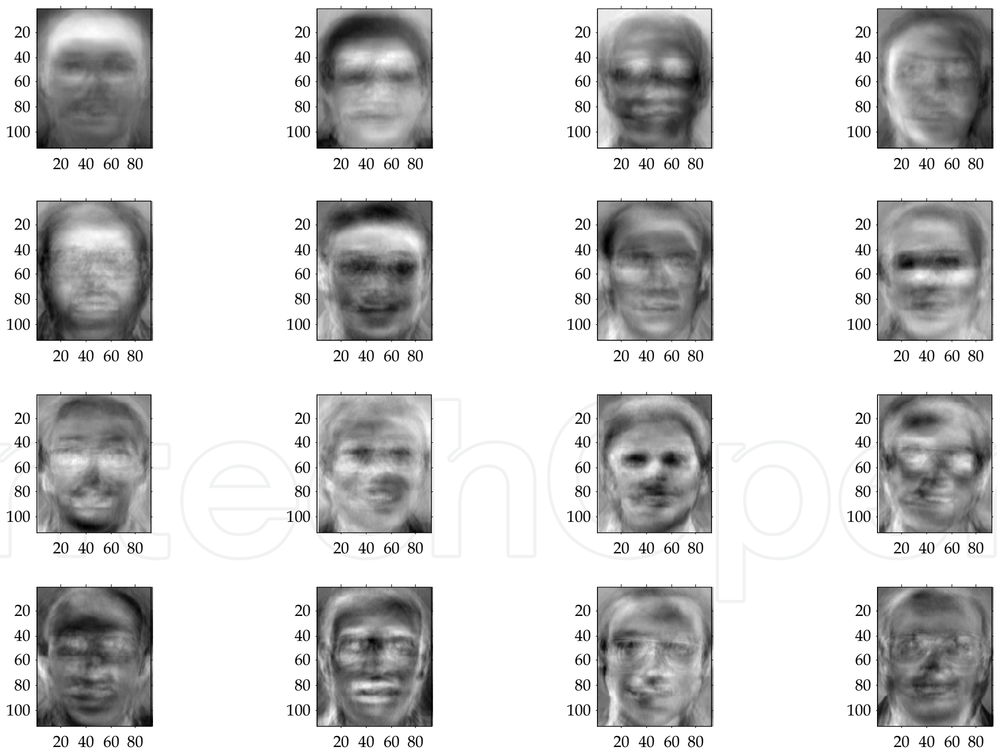

Figure 2. First 16 eigenfaces with highest eigenvalues

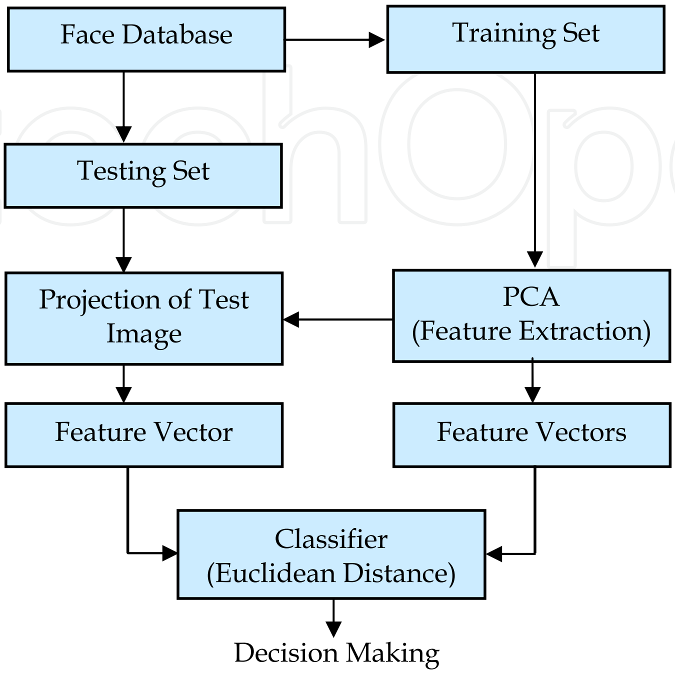

Figure 3. PCA approach for face recognition

3. Linear Discriminant Analysis

Linear Discriminant analysis or Fisherfaces method overcomes the limitations of eigenfaces method by applying the Fisher's linear discriminant criterion. This criterion tries to maximize the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples.

Fisher discriminants group images of the same class and separates images of different classes. Images are projected from N^2-dimensional space to C dimensional space (where C is the number of classes of images). For example, consider two sets of points in 2-dimensional space that are projected onto a single line. Depending on the direction of the line, the points can either be mixed together (Figure 4a) or separated (Figure 4b). Fisher discriminants find the line that best separates the points. To identify an input test image, the projected test image is compared to each projected training image, and the test image is identified as the closest training image.

As with eigenspace projection, training images are projected into a subspace. The test images are projected into the same subspace and identified using a similarity measure. What differs is how the subspace is calculated.

Unlike the PCA method that extracts features to best represent face images; the LDA method tries to find the subspace that best discriminates different face classes as shown in Figure 4. The within-class scatter matrix, also called intra-personal, represents variations in appearance of the same individual due to different lighting and face expression, while the between-class scatter matrix, also called the extra-personal, represents variations in appearance due to a difference in identity. By applying this method, we find the projection directions that on one hand maximize the distance between the face images of different classes on the other hand minimize the distance between the face images of the same class. In another words, maximizing the between-class scatter matrix S_b, while minimizing the within-class scatter matrix S_w in the projective subspace. Figure 5 shows good and bad class separation.

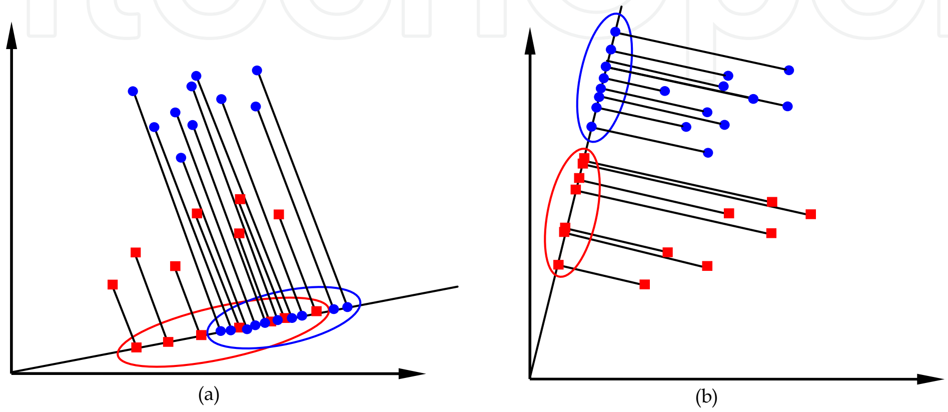

Figure 4. (a) Points mixed when projected onto a line. (b) Points separated when projected onto another line

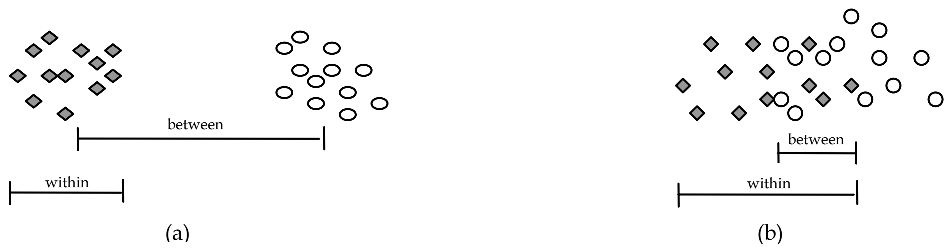

Figure 5. (a) Good class separation. (b) Bad class separation

The within-class scatter matrix S_w and the between-class scatter matrix S_b are defined as

$$S_w = \sum_{j=1}^{C} \sum_{i=1}^{N_j} (\Gamma_i^j - \mu_j)(\Gamma_i^j - \mu_j)^T \tag{12}$$

Where $\Gamma_i^{\,j}$ is the i^{th} sample of class j, μ_j is the mean of class j, C is the number of classes, N_j is the number of samples in class j.

$$S_b = \sum_{j=1}^{C} (\mu_j - \mu)(\mu_j - \mu)^T \tag{13}$$

where μ represents the mean of all classes. The subspace for LDA is spanned by a set of vectors $W = [W_1, W_2, \dots, W_d]$, satisfying

$$W = \arg\max = \left| \frac{W^T S_b W}{W^T S_W W} \right| \tag{14}$$

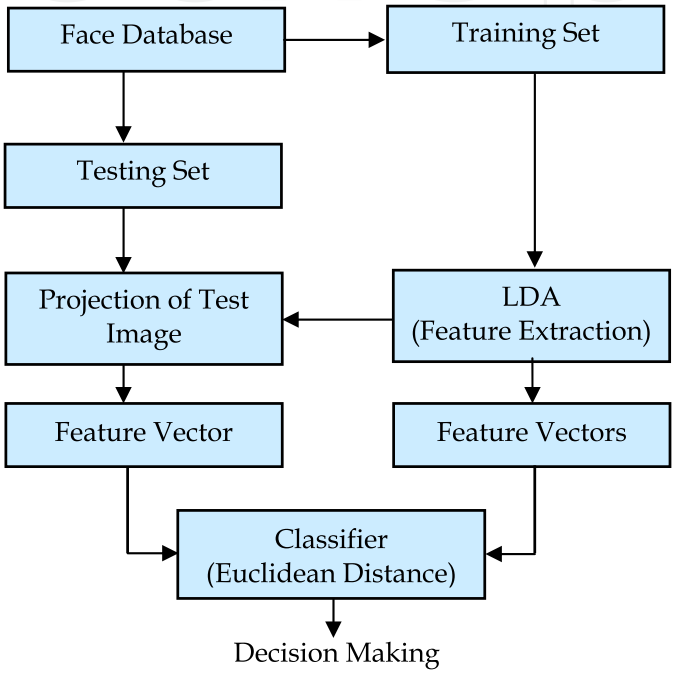

Figure 6. LDA approach for face recognition

The within class scatter matrix represents how face images are distributed closely within classes and between class scatter matrix describes how classes are separated from each other. When face images are projected into the discriminant vectors *W*, face images should be distributed closely within classes and should be separated between classes, as much as possible. In other words, these discriminant vectors minimize the denominator and maximize the numerator in Equation (14). *W* can therefore be constructed by the eigenvectors of $S_w^{-1} S_b$. Figure 7 shows the first 16 eigenvectors with highest associated eigenvalues of $S_w^{-1} S_b$. These eigenvectors are also referred to as the fisherfaces. There are various methods to solve the problem of LDA such as the pseudo inverse method, the subspace method, or the null space method.
The LDA approach is similar to the eigenface method, which makes use of projection of training images into a subspace. The test images are projected into the same subspace and identified using a similarity measure. The only difference is the method of calculating the subspace characterizing the face space. The face which has the minimum distance with the test face image is labelled with the identity of that image. The minimum distance can be

calculated using the Euclidian distance method as given earlier in Equation (11). Figure 6 shows the testing phase of the LDA approach.

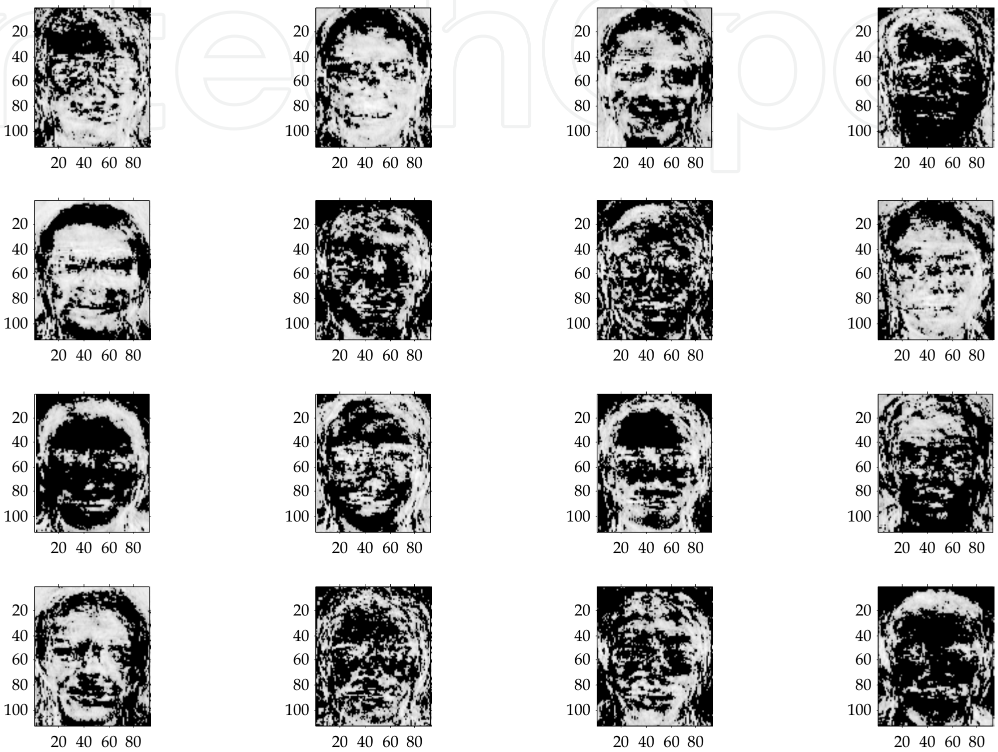

Figure 7. First 16 Fisherfaces with highest eigenvalues

4. Neural Networks

Neural networks, with massive parallelism in its structure and high computation rates, provide a great alternative to other conventional classifiers and decision making systems. Neural networks are powerful tools that can be trained to perform a complex and various functions in computer vision applications, such as preprocessing (boundary extraction, image restoration, image filtering), feature extraction (extract transformed domain features), associative memory (storing and retrieving information), and pattern recognition.

4.1 Feedforward Neural Networks (FFNN)

FFNN is suitable structure for nonlinear separable input data. In FFNN model the neurons are organized in the form of layers. The neurons in a layer get input from the previous layer and feed their output to the next layer. In this type of networks connections to the neurons in the same or previous layers are not permitted. Figure 8 shows the architecture of the system for face classification.

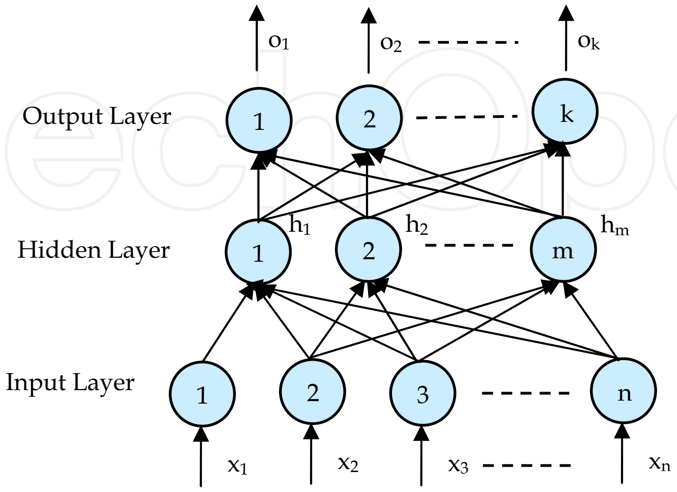

Figure 8. Architecture of FFNN for classification

4.2. Learning Algorithm (Backpropagation)

Learning process in Backpropagation requires providing pairs of input and target vectors. The output vector ***o*** of each input vector is compared with target vector ***t***. In case of difference the weights are adjusted to minimize the difference. Initially random weights and thresholds are assigned to the network. These weights are updated every iteration in order to minimize the cost function or the mean square error between the output vector and the target vector.

Input for hidden layer is given by

$$net_m = \sum_{z=1}^{n} x_z w_{mz} \tag{15}$$

The units of output vector of hidden layer after passing through the activation function are given by

$$h_m = \frac{1}{1+\exp(-net_m)} \tag{16}$$

In same manner, input for output layer is given by

$$net_k = \sum_{z=1}^{m} h_z w_{kz} \tag{17}$$

and the units of output vector of output layer are given by

$$o_k = \frac{1}{1+\exp(-net_k)} \tag{18}$$

For updating the weights, we need to calculate the error. This can be done by

$$E = \frac{1}{2}\sum_{i=1}^{k}(o_i - t_i)^2 \tag{19}$$

If the error is minimum than a predefined limit, training process will stop; otherwise weights need to be updated. For weights between hidden layer and output layer, the change in weights is given by

$$\Delta w_{ij} = \alpha \delta_i h_j \tag{20}$$

where α is a training rate coefficient that is restricted to the range [0.01,1.0], h_j is the output of neuron j in the hidden layer, and δ_i can be obtained by

$$\delta_i = (t_i - o_i)o_i(1 - o_i) \tag{21}$$

o_i and t_i represents the real output and target output at neuron i in the output layer respectively.
Similarly, the change of the weights between hidden layer and output layer, is given by

$$\Delta w_{ij} = \beta \delta_{Hi} x_j \tag{22}$$

where β is a training rate coefficient that is restricted to the range [0.01,1.0], x_j is the output of neuron j in the input layer, and δ_{Hi} can be obtained by

$$\delta_{Hi} = x_i(1 - x_i)\sum_{j=1}^{k}\delta_j w_{ij} \tag{23}$$

x_i is the output at neuron i in the input layer, and summation term represents the weighted sum of all δ_j values corresponding to neurons in output layer that obtained in equation (21).
After calculating the weight change in all layers, the weights can simply updated by

$$w_{ij}(new) = w_{ij}(old) + \Delta w_{ij} \tag{24}$$

5. Performance Analysis and Discussions

5.1. Training and Testing of Neural Networks

Two neural networks, one for PCA based classification and the other for LDA based classification are prepared. ORL face database is used for training and testing. The training is performed by n poses from each subject and the performance testing is performed by 10-n poses of the same subjects.
After calculating the eigenfaces using PCA the projection vectors are calculated for the training set and then used to train the neural network. This architecture is called PCA-NN. Similarly, after calculation of the fisherfaces using the LDA, projection vectors are calculated for the training set. Therefore, the second neural network is trained by these vectors. This architecture is called LDA-NN (Eleyan & Demirel, 2005, 2006). Figure 9 shows the schematic diagram for the neural network training phase.

When a new image from the test set is considered for recognition, the image is mapped to the eigenspace or fisherspace. Hence, the image is assigned to a feature vector. Each feature vector is fed to its respective neural network and the network outputs are compared.

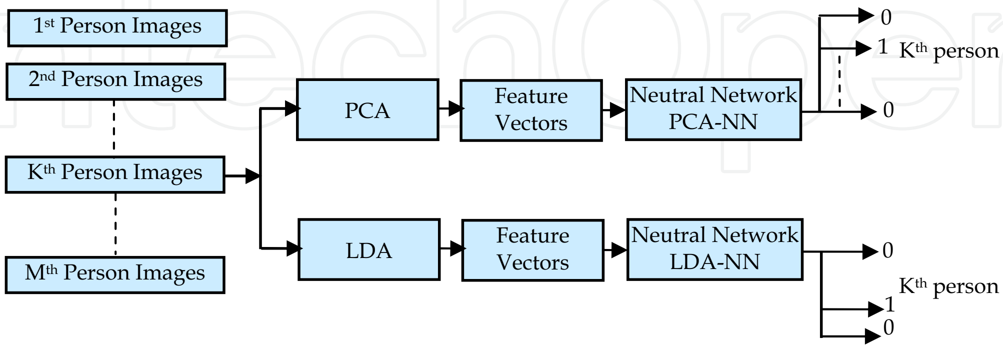

Figure 9. Training phase of both Neural Networks

5.2. System Performance

The performances of the proposed systems are measured by varying the number of faces of each subject in the training and test faces. Table 1 shows the performances of the proposed PCA-NN and LDA-NN methods based on the neural network classifiers as well as the performances of the conventional PCA and LDA based on the Euclidean Distance classifier. The recognition performances increase due to the increase in face images in the training set. This is obvious, because more sample images can characterize the classes of the subjects better in the face space. The results clearly shows that the proposed recognition systems, PCA-NN and LDA-NN, outperforms the conventional PCA and LDA based recognition systems. The LDA-NN shows the highest recognition performance, where this performance is obtained because of the fact that the LDA method discriminate the classes better than the PCA and neural network classifier is more optimal classifier than the Euclidean Distance based classifier. The performance improvement in PCA versus PCA-NN is higher than the LDA versus LDA-NN. For example, when there are 5 images for training and 5 images for testing, the improvement is 7% in PCA based approach and 4% in the LDA based approach. These results indicate that the superiority of LDA over PCA in class separation in the face space leaves less room for improvement to the neural network based classifier.

Training Images	Testing Images	PCA	PCA-NN	LDA	LDA-NN
2	8	71	75	78	80
3	7	73	76	82	84
4	6	77	80	87	89
5	5	78	85	87	91
6	4	89	90	93	93
7	3	92	94	95	95
8	2	94	95	96	97

Table 1. Performance of conventional PCA & LDA versus proposed PCA-NN & LDA-NN

6. Conclusions

In this chapter, two face recognition systems, the first system based on the PCA preprocessing followed by a FFNN based classifier (PCA-NN) and the second one based on the LDA preprocessing followed by another FFNN (LDA-NN) based classifier, are introduced. The feature projection vectors obtained through the PCA and LDA methods are used as the input vectors for the training and testing of both FFNN architectures. The proposed systems show improvement on the recognition rates over the conventional LDA and PCA face recognition systems that use Euclidean Distance based classifier. Additionally, the recognition performance of LDA-NN is higher than the PCA-NN among the proposed systems.

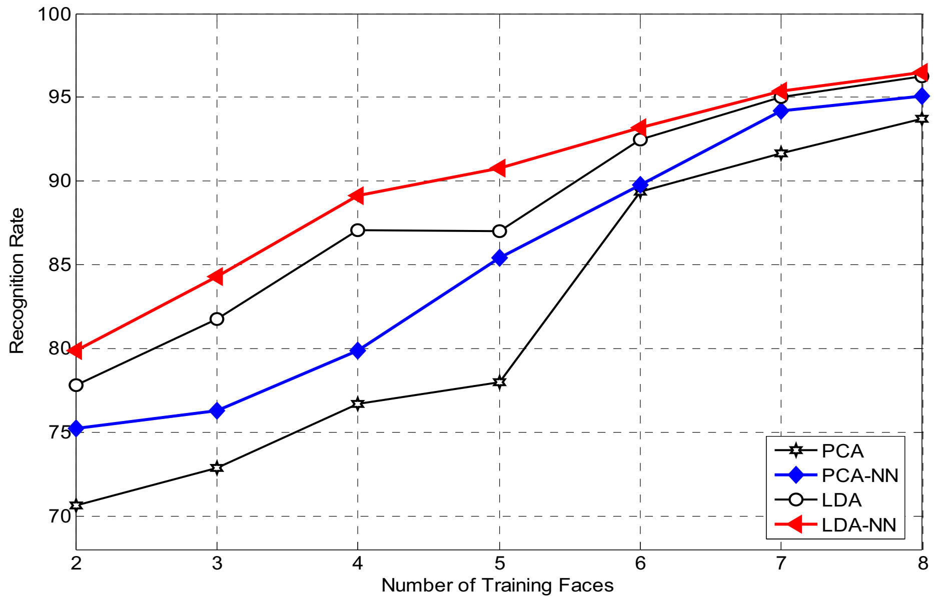

Figure 10. Recognition rate vs. number of training faces

7. References

Belhumeur, P.; Hespanha, J. & Kriegman, D. (1997). Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, (July 1997) 711-720, 0162-8828

Brunelli, R. & Poggio, T. (1993). Face recognition: Features versus Templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15 No. 10, (Octuber 1993) 1042-1052, 0162-8828

Eleyan, A. & Demirel, H. (2005). Face Recognition System based on PCA and Feedforward Neural Networks, *Proceedings of Computational Intelligence and Bioinspired Systems*, pp. 935-942, 978-3-540-26208-4, Spain, June 2005, Springer-Verlag, Barcelona

Eleyan, A. & Demirel, H. (2006). PCA and LDA Based Face Recognition Using Feedforward Neural Network Classifier, *Proceedings of Multimedia Content Representation, Classification and Security*, pp. 199-206, 978-3-540-39392-4, Turkey, September 2006, Springer-Verlag, Istanbul

Haykin, S. (1999). *Neural Networks: A comprehensive foundation,* Prentice Hall, 0-13-273350-1, New Jersey

Kirby, M. & Sirovich, L. (1990). Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12 No. 1, (January 1990) 103-108, 0162-8828

Moghaddam, B. and Pentland, A. (1997). Probabilistic visual learning for object recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19 No. 7, (July 1997) 696-710, 0162-8828

Papoulis, A. & Pillai, U. (2002). *Probability, random variables, and Stochastic Processes*, McGraw-Hill, 0073660116, New York

Pentland, A.; Moghaddam, B. & Starner, T. (1994). Viewbased and modular eigenspaces for face recognition, *In Proceedings of Computer Vision and Pattern Recognition*, pp. 84–91, 0-8186-5825-8, USA, June 1994. IEEE Computer Society, Seattle

Sirovich, L. & Kirby, M. (1987). Low-Dimensional Procedure for the Characterization of Human Faces, *Journal of the Optical Society of America A*, Vol. 4(3), (March 1987), 519-524, 1084-7529

Turk, M. & Pentland, A. (1991). Eigenfaces for Recognition, *Journal of Cognitive Neuroscience*, Vol. 3, (1991) 71-86. 0898-929X

Wiskott, L.; Fellous, J. Krüger, N. & Malsburg, V. (1997). Face Recognition by Elastic Brunch Graph Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19 No. 7, (July 1997) 775-779, 0162-8828

Zhao, W.; Chellappa, R. & Nandhakumarm, N. (1998). Empirical performance analysis of linear discriminant classfiers. *In Proceedings of Computer Vision and Pattern Recognition*, pp. 164–169, 0-8186-5825-8, Canada, June 1998. IEEE Computer Society, Santa Barbara

Face Recognition
Edited by Kresimir Delac and Mislav Grgic

ISBN 978-3-902613-03-5
Hard cover, 558 pages
Publisher I-Tech Education and Publishing
Published online 01, July, 2007
Published in print edition July, 2007

This book will serve as a handbook for students, researchers and practitioners in the area of automatic (computer) face recognition and inspire some future research ideas by identifying potential research directions. The book consists of 28 chapters, each focusing on a certain aspect of the problem. Within every chapter the reader will be given an overview of background information on the subject at hand and in many cases a description of the authors' original proposed solution. The chapters in this book are sorted alphabetically, according to the first author's surname. They should give the reader a general idea where the current research efforts are heading, both within the face recognition area itself and in interdisciplinary approaches.