

Face Recognition using Neural Networks

P.Latha

*Selection .grade Lecturer,
Department of Electrical and Electronics Engineering,
Government College of Engineering,
Tirunelveli- 627007*

plathamuthuraj@gmail.com

Dr.L.Ganesan

*Assistant Professor,
Head of Computer Science & Engineering department,
Alagappa Chettiar College of Engineering & Technology,
Karaikudi- 630004*

Dr.S.Annadurai

*Additional Director, Directorate of Technical Education
Chennai-600025*

Abstract

Face recognition is one of biometric methods, to identify given face image using main features of face. In this paper, a neural based algorithm is presented, to detect frontal views of faces. The dimensionality of face image is reduced by the Principal component analysis (PCA) and the recognition is done by the Back propagation Neural Network (BPNN). Here 200 face images from Yale database is taken and some performance metrics like Acceptance ratio and Execution time are calculated. Neural based Face recognition is robust and has better performance of more than 90 % acceptance ratio.

Key words: Face recognition-Principal Component Analysis- Back Propagation Neural Network - Acceptance ratio-Execution time

1. INTRODUCTION

A face recognition system [6] is a computer vision and it automatically identifies a human face from database images. The face recognition problem is challenging as it needs to account for all possible appearance variation caused by change in illumination, facial features, occlusions, etc. This paper gives a Neural and PCA based algorithm for efficient and robust face recognition. Holistic approach, feature-based approach and hybrid approach are some of the approaches for face recognition. Here, a holistic approach is used in which the whole face region is taken into account as input data. This is based on principal component-analysis (PCA) technique, which is used to simplify a dataset into lower dimension while retaining the characteristics of dataset.

Pre-processing, Principal component analysis and Back Propagation Neural Algorithm are the major implementations of this paper. Pre-processing is done for two purposes

- (i) To reduce noise and possible convolute effects of interfering system,
- (ii) To transform the image into a different space where classification may prove easier by exploitation of certain features.

PCA is a common statistical technique for finding the patterns in high dimensional data's [1]. Feature extraction, also called Dimensionality Reduction, is done by PCA for a three main purposes like

- i) To reduce dimension of the data to more tractable limits

- ii) To capture salient class-specific features of the data,
- iii) To eliminate redundancy.

Here recognition is performed by both PCA and Back propagation Neural Networks [3]. BPNN mathematically models the behavior of the feature vectors by appropriate descriptions and then exploits the statistical behavior of the feature vectors to define decision regions corresponding to different classes. Any new pattern can be classified depending on which decision region it would be falling in. All these processes are implemented for Face Recognition, based on the basic block diagram as shown in fig 1.

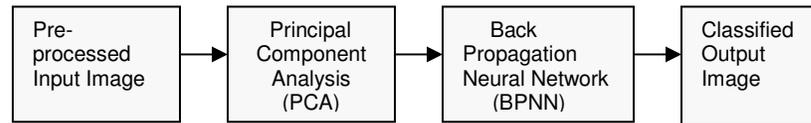


Fig. 1 Basic Block Diagram

The Algorithm for Face recognition using neural classifier is as follows:

- a) Pre-processing stage –Images are made zero-mean and unit-variance.
- b) Dimensionality Reduction stage: PCA - Input data is reduced to a lower dimension to facilitate classification.
- c) Classification stage - The reduced vectors from PCA are applied to train BPNN classifier to obtain the recognized image.

In this paper, Section 2 describes about Principal component analysis, Section 3 explains about Back Propagation Neural Networks, Section 4 demonstrates experimentation and results and subsequent chapters give conclusion and future development.

2. PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA) [2] involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. PCA is a popular technique, to derive a set of features for both face recognition.

Any particular face can be

- (i) Economically represented along the eigen pictures coordinate space, and
- (ii) Approximately reconstructed using a small collection of Eigen pictures

To do this, a face image is projected to several face templates called eigenfaces which can be considered as a set of features that characterize the variation between face images. Once a set of eigenfaces is computed, a face image can be approximately reconstructed using a weighted combination of the eigenfaces. The projection weights form a feature vector for face representation and recognition. When a new test image is given, the weights are computed by projecting the image onto the eigen-face vectors. The classification is then carried out by comparing the distances between the weight vectors of the test image and the images from the database. Conversely, using all of the eigenfaces extracted from the original images, one can reconstruct the original image from the eigenfaces so that it matches the original image exactly.

2.1 PCA Algorithm

The algorithm used for principal component analysis is as follows.

- (i) Acquire an initial set of M face images (the training set) & Calculate the eigen-faces from the training set, keeping only M' eigenfaces that correspond to the highest eigenvalue.
- (ii) Calculate the corresponding distribution in M' -dimensional weight space for each known individual, and calculate a set of weights based on the input image
- (iii) Classify the weight pattern as either a known person or as unknown, according to its distance to the closest weight vector of a known person.

Let the training set of images be $\Gamma_1, \Gamma_2, \dots, \Gamma_M$. The average face 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 vector

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

The co- variance matrix is formed by

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \cdot \Phi_n^T = A \cdot A^T \tag{3}$$

where the matrix $A = [\Phi_1, \Phi_2, \dots, \Phi_M]$.

This set of large vectors is then subject to principal component analysis, which seeks a set of M orthonormal vectors $u_1 \dots u_M$. To obtain a weight vector Ω of contributions of individual eigen-faces to a facial image Γ , the face image is transformed into its eigen-face components projected onto the face space by a simple operation

$$\omega_k = u_k^T (\Gamma - \Psi) \tag{4}$$

For $k=1, \dots, M'$, where $M' \leq M$ is the number of eigen-faces used for the recognition. The weights form vector $\Omega = [\omega_1, \omega_2, \dots, \omega_{M'}]$ that describes the contribution of each Eigen-face in representing the face image Γ , treating the eigen-faces as a basis set for face images. The simplest method for determining which face provides the best description of an unknown input facial image is to find the image k that minimizes the Euclidean distance ϵ_k .

$$\epsilon_k = \|(\Omega - \Omega_k)\|^2 \tag{5}$$

where Ω_k is a weight vector describing the k^{th} face from the training set. A face is classified as belonging to person k when the ' ϵ_k ' is below some chosen threshold Θ_ϵ otherwise, the face is classified as unknown.

The algorithm functions by projecting face images onto a feature space that spans the significant variations among known face images. The projection operation characterizes an individual face by a weighted sum of eigenfaces features, so to recognize a particular face, it is necessary only to compare these weights to those of known individuals. The input image is matched to the subject from the training set whose feature vector is the closest within acceptable thresholds.

Eigen faces have advantages over the other techniques available, such as speed and efficiency. For the system to work well in PCA, the faces must be seen from a frontal view under similar lighting.

3. NEURAL NETWORKS AND BACK PROPAGATION ALGORITHM

A successful face recognition methodology depends heavily on the particular choice of the features used by the pattern classifier. The Back-Propagation is the best known and widely used learning algorithm in training multilayer perceptrons (MLP) [5]. The MLP refer to the network consisting of a set of sensory units (source nodes) that constitute the input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes. The input signal propagates through the network in a forward direction, from left to right and on a layer-by-layer basis.

Back propagation is a multi-layer feed forward, supervised learning network based on gradient descent learning rule. This BPNN provides a computationally efficient method for changing the weights in feed forward network, with differentiable activation function units, to learn a training set

of input-output data. Being a gradient descent method it minimizes the total squared error of the output computed by the net. The aim is to train the network to achieve a balance between the ability to respond correctly to the input patterns that are used for training and the ability to provide good response to the input that are similar.

3.1 Back Propagation Neural Networks Algorithm

A typical back propagation network [4] with Multi-layer, feed-forward supervised learning is as shown in the figure. 2. Here learning process in Back propagation requires pairs of input and target vectors. The output vector 'o' is compared with target vector 't'. In case of difference of 'o' and 't' vectors, 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 mean square error between the output vector and the target vector.

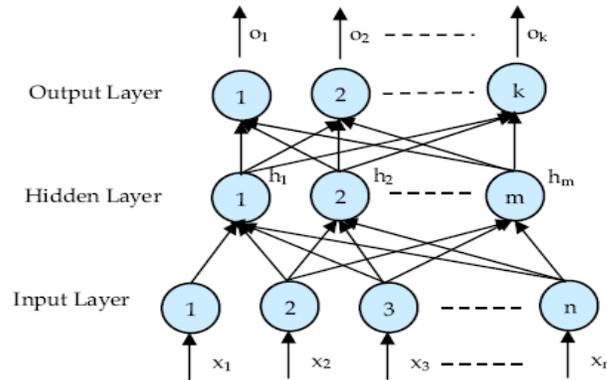


Fig. 2 Basic Block of Back propagation neural network

Input for hidden layer is given by

$$net_m = \sum_{z=1}^n x_z w_{mz} \quad \text{----- (6)}$$

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

$$h_m = \frac{1}{1 + \exp(-net_m)} \quad \text{----- (7)}$$

In same manner, input for output layer is given by

$$net_k = \sum_{z=1}^m h_z w_{kz} \quad \text{----- (8)}$$

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

$$o_k = \frac{1}{1 + \exp(-net_k)} \quad \text{----- (9)}$$

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

$$E = \frac{1}{2} \sum_{i=1}^k (o_i - t_i)^2 \quad \text{----- (10)}$$

o_i and t_i represents the real output and target output at neuron i in the output layer respectively. 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

$$\Delta w_{ij} = \alpha \delta_i h_j \quad \text{----- (11)}$$

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

$$\delta_i = (t_i - o_i) o_i (l - o_i) \quad \text{----- (12)}$$

Similarly, the change of the weights between hidden layer and output layer, is given by

$$\Delta w_{ij} = \beta \delta_{Hi} x_j \quad \text{----- (13)}$$

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

$$\delta_{Hi} = x_i (l - x_i) \sum_{j=1}^k \delta_j w_{ij} \quad \text{----- (14)}$$

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. After calculating the weight change in all layers, the weights can simply updated by

$$w_{ij} (new) = w_{ij} (old) + \Delta w_{ij} \quad \text{----- (15)}$$

This process is repeated, until the error reaches a minimum value

2.4.3 Selection of Training Parameters

For the efficient operation of the back propagation network it is necessary for the appropriate selection of the parameters used for training.

Initial Weights

This initial weight will influence whether the net reaches a global or local minima of the error and if so how rapidly it converges. To get the best result the initial weights are set to random numbers between -1 and 1.

Training a Net

The motivation for applying back propagation net is to achieve a balance between memorization and generalization; it is not necessarily advantageous to continue training until the error reaches a minimum value. The weight adjustments are based on the training patterns. As long as error the for validation decreases training continues. Whenever the error begins to increase, the net is starting to memorize the training patterns. At this point training is terminated.

Number of Hidden Units

If the activation function can vary with the function, then it can be seen that a n-input, m-output function requires at most $2n+1$ hidden units. If more number of hidden layers are present, then the calculation for the δ 's are repeated for each additional hidden layer present, summing all the δ 's for units present in the previous layer that is fed into the current layer for which δ is being calculated.

Learning rate

In BPN, the weight change is in a direction that is a combination of current gradient and the previous gradient. A small learning rate is used to avoid major disruption of the direction of learning when very unusual pair of training patterns is presented.

Various parameters assumed for this algorithm are as follows.

No.of Input unit	=	1 feature matrix
Accuracy	=	0.001
learning rate	=	0.4
No.of epochs	=	400
No. of hidden neurons	=	70
No.of output unit	=	1

Main advantage of this back propagation algorithm is that it can identify the given image as a face image or non face image and then recognizes the given input image .Thus the back propagation neural network classifies the input image as recognized image.

4. Experimentation and Results

In this paper for experimentation, 200 images from Yale database are taken and a sample of 20 face images is as shown in fig 3. One of the images as shown in fig 4a is taken as the Input image. The mean image and reconstructed output image by PCA, is as shown in fig 4b and 4c. In BPNN, a training set of 50 images is as shown in fig 5a and the Eigen faces and recognized output image are as shown in fig 5b and 5c.

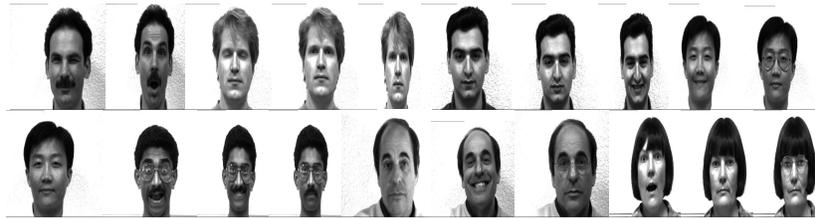
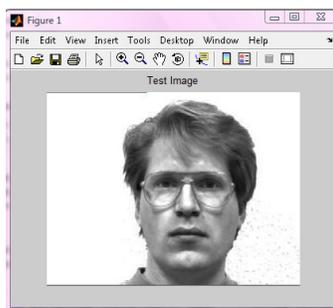
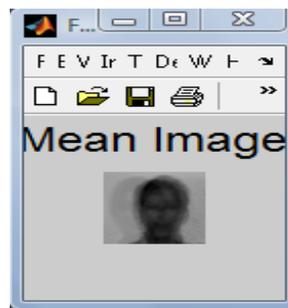


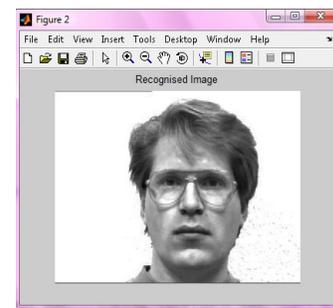
Fig 3. Sample Yale Database Images



4(a)

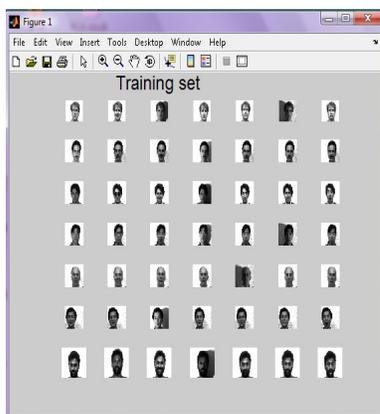


4(b)

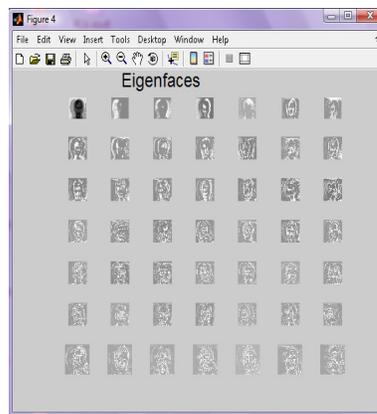


4 (c)

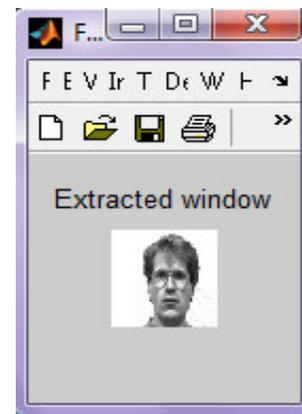
Fig 4.(a) Input Image , (b)Mean Image , (c) Recognized Image by PCA method



5(a)



5(b)



5(c)

Fig 5 (a) Training set, (b) Eigen faces , (c) Recognized Image by BPNN method

Table 1 shows the comparison of acceptance ratio and execution time values for 40, 80, 120,160 and 200 images of Yale database. Graphical analysis of the same is as shown in fig 6.

No .of	Acceptance ratio (%)	Execution Time (Seconds)
--------	----------------------	--------------------------

Images				
	PCA	PCA with BPNN	PCA	PCA with BPNN
40	92.4	96.5	38	36
60	90.6	94.3	46	43
120	87.9	92.8	55	50
160	85.7	90.2	67	58
200	83.5	87.1	74	67

Table 1 Comparison of acceptance ratio and execution time for Yale database images

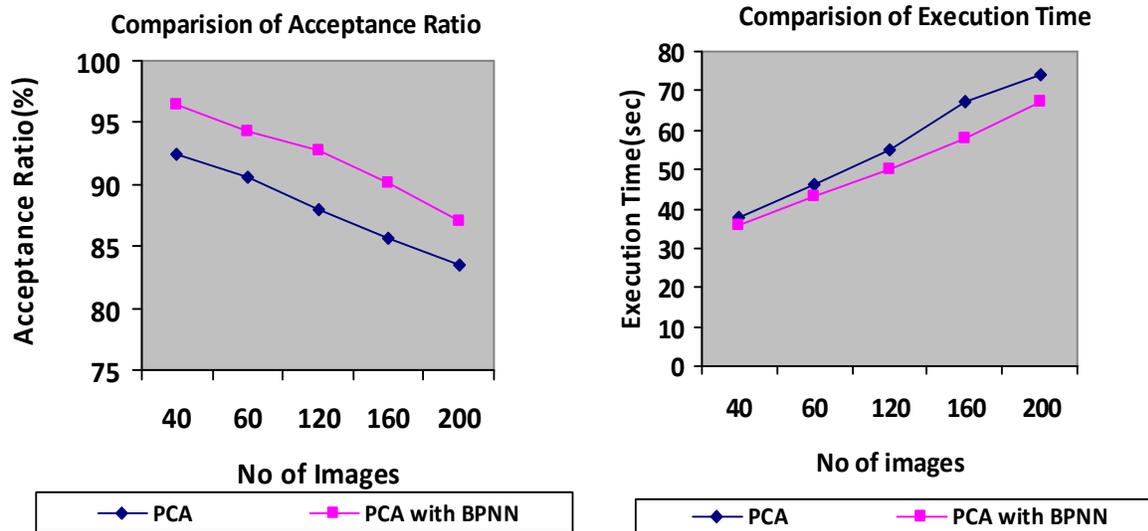


Fig.6: comparison of Acceptance ratio and execution time

5. CONCLUSION

Face recognition has received substantial attention from researches in biometrics, pattern recognition field and computer vision communities. In this paper, Face recognition using Eigen faces has been shown to be accurate and fast. When BPNN technique is combined with PCA, non linear face images can be recognized easily. Hence it is concluded that this method has the acceptance ratio is more than 90 % and execution time of only few seconds. Face recognition can be applied in Security measure at Air ports, Passport verification, Criminals list verification in police department, Visa processing , Verification of Electoral identification and Card Security measure at ATM's..

6. REFERENCES

- [1]. B.K.Gunturk,A.U.Batur, and Y.Altunbasak,(2003) "Eigenface-domain super-resolution for face recognition," IEEE Transactions of . Image Processing. vol.12, no.5.pp. 597-606.

- [2]. M.A.Turk and A.P.Petland, (1991) "Eigenfaces for Recognition," Journal of Cognitive Neuroscience. vol. 3, pp.71-86.
- [3]. T.Yahagi and H.Takano,(1994) "Face Recognition using neural networks with multiple combinations of categories," International Journal of Electronics Information and Communication Engineering., vol.J77-D-II, no.11, pp.2151-2159.
- [4]. S.Lawrence, C.L.Giles, A.C.Tsoi, and A.d.Back, (1993) "IEEE Transactions of Neural Networks. vol.8, no.1, pp.98-113.
- [5]. C.M.Bishop,(1995) "Neural Networks for Pattern Recognition" London, U.K.:Oxford University Press.
- [6]. Kailash J. Karande Sanjay N. Talbar "Independent Component Analysis of Edge Information for Face Recognition" International Journal of Image Processing Volume (3) : Issue (3) pp: 120 -131.