A MATLAB based Face Recognition using PCA with Back Propagation Neural network

Priyanka Dhoke[1], M.P. Parsai[2]

Dept. of Electronics and Communication, Jabalpur Engineering College, Jabalpur (M.P.), India[1]

Professor, Dept. of Electronics and Communication, Jabalpur Engineering College, Jabalpur (M.P.), India[2]

ABSTRACT: Automatic recognition of people is a challenging problem which has received much attention during recent years due to its many applications in different fields. Face recognition is one of those challenging problems and up to date, there is no technique that provides a robust solution to all situations. There are many techniques used for this purpose. Face recognition is an effective means of authenticating a person. In this paper, a face recognition system for personal identification and verification using Principal Component Analysis (PCA) with Back Propagation Neural Networks (BPNN) is proposed. The dimensionality of face image is reduced by the PCA and the recognition is done by the BPNN for face recognition. The system consists of a database of a set of facial patterns for each individual. The characteristic features of pca called 'eigenfaces' are extracted from the stored images, which is combine with Back Propagation Neural Network for subsequent recognition of new images.

KEYWORDS: Neural Networks, Principal Component Analysis, Eigen Values, Eigen Vector, Back Propagation Neural Network

I. INTRODUCTION

Face recognition has a large number of applications, including security, person verification, Internet communication, and computer entertainment. Although research in automatic face recognition has been conducted since the 1960s, this problem is still largely unsolved. Recent years have seen significant progress in this area owing to advances in face modelling and analysis techniques. Systems have been developed for face detection and tracking, but reliable face recognition still offers a great challenge to computer vision and pattern recognition researchers. There are several reasons for recent increased interest in face recognition, including rising public concern for security, the need for identity verification in the digital world, and the need for face analysis and modelling techniques in multimedia data management and computer entertainment. Recent advances in automated face analysis, pattern recognition, and machine learning have made it possible to develop automatic face recognition systems to address these applications. In this paper we proposed a mathematical model and computational model of face recognition which is fast, reasonably simple, and accurate in constrained environment. Face recognition using eigenface has been shown to be accurate and fast. When BPNN technique is combine with PCA non-linear face images can be recognised easily. [1][5]

II. WORKING MODEL

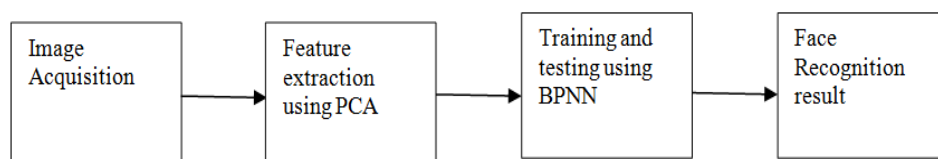The system involves three steps (Fig1):

Fig1: Generic representation of a face recognition system

The issues of the design and implementation of the Face Recognition System (FRS) can be subdivided into two main parts. The first part is image processing and the second part is recognition techniques. The image processing part consists of Face image acquisition techniques and the second part consists of the artificial intelligence which is composed by PCA and Back Propagation Neural Network. Face image acquired in the first step by web cam, digital camera or using scanner is fed as an input to PCA, which converts the input image to low dimensional image and calculates its Euclidian distance. This Euclidian distance is then fed as an input to Back-propagation Neural Network.

III. PROPOSED ALGORITHM

A. *PCA: Principal Component Analysis (PCA):*

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation. The PCA approach is used to reduce the dimension of the data by means of data compression basics and reveals the most effective low dimensional structure of facial patterns. This reduction in dimensions removes information that is not useful and precisely decomposes the face structure which involves transformation of number of possible correlated variables into a smaller number of orthogonal (uncorrelated) components known as Principal Components. Each face image may be represented as a weighted sum (feature vector) of the eigenfaces, which are stored in a 1D array. The test image can be constructed using these weighted sums of eigen faces. When a test image is given, the weights are computed by projecting the image upon eigen face vectors. The distance between the weighted vectors of the test image and that of the database images are then compared. Thus one can reconstruct original image with the help of eigen faces so that it matches the desired image.

Algorithm for PCA:

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

$$\Psi = 1/M \sum_{i=1}^{M} \Gamma i$$

Each face differs from the average by vector

$$\Phi i = \Gamma i - \Psi$$
Where i=1….M

The co- variance matrix is formed by
$$C = A.A^T$$

Where the matrix A is given by
$$A = [\Phi_1, \Phi_2, \ldots \Phi_M]$$

This set of large vectors is then subject to principal component analysis, which seeks a set of M orthonormal vectors. To obtain a weight vector W of contributions of individual eigen-faces to a facial image, the face image is transformed into its eigen-face components projected onto the face space by a simple operation.[8]

$$W_k = u^t_k \Phi$$

For k=1... M', where $M' \pounds M$ is the number of eigen-faces used for the recognition. The weights form vector W = [w1,w2,….wm] that describes the contribution of each Eigen-face in representing the face image, 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

$$\epsilon_k = ||(\Omega - \Omega_k)||^2$$

Where W_k is a weight vector describing the k_{th} face from the training set. It is this Euclidean distance that is given as an input to the neural networks.[9]

B. *BackPropagation Neural Network (BPNN):*

The backpropagation algorithm is a multi-layer network using a weight adjustment based on the sigmoid function, like the delta rule. According to the back-propagation Network (BPN) algorithm, is a fully feedforward network connection. The activation travels in a direction from input layer to the output layer and the units in one layer are all connected to every unit in the next layer. Basically, back-propagation algorithm consists of two sweeps of the network which are the forward sweep and the backward sweeps. Forward sweep defines the network from the input layer to the output layer, in which it propagates the input vectors through the network to provide outputs at the output layer in the end. During the forward sweep, the weights of the networks are all fixed. The backward sweep hence defines network from the output layer to the input layer, where it is similar to forward sweep except that the error values are propagated back through the network. This is done in order to determine how the weights are to be changed during the training, in which the weights are all adjusted in accordance of an error correction rule where the actual response of the network is subtracted from the target response to produce an error signal. [10]

In fig.2, the hidden units send activation to each output units and thus during backward sweep, this hidden unit will received an error signals from the output units. Basically, the number of processing elements in each layer will vary according to the applications verified.
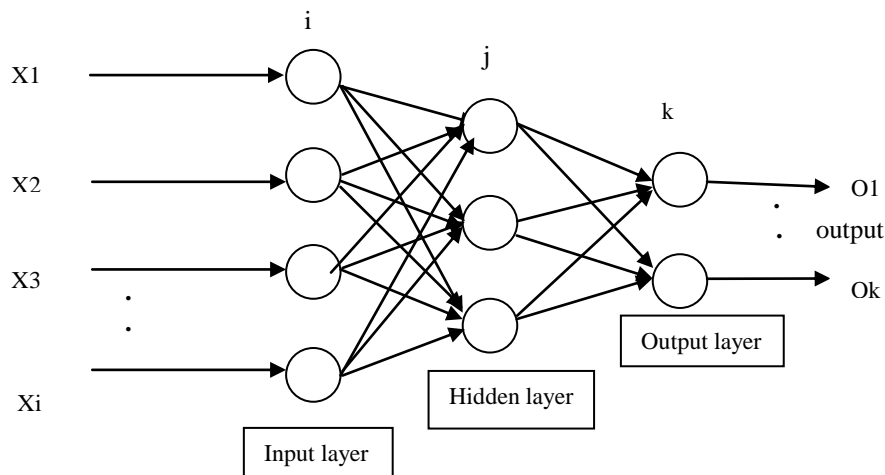

Fig 2: Multilayered neural network

Back-propagation algorithm used supervised learning approach, the target output vectors are defined earlier in the system. The learning process begins with random variables of an input pattern to the BPN. In which, the net total input is found using the standard summation of products as defined in Equation below:

$net_j = \sum_{i=0}^{i} . x_i . w_{ij}$
Where; x = input vectors w = weight vectors

Basically, units have a rule for calculating an output value that will be transmitted to other units, in which this rule is known as an activation function and the output value is referred to as the activation for the unit. Back-propagation algorithm used sigmoid function as the activation function, and represented by Equation below, where f (net$_j$) denoted the activation function for the hidden layer:
$f(net_j) = 1 \backslash [1+exp(-net_j)]$

In order to find the net total output, below Equation defined the appropriate formula:

$net_k = w_k.f(net_j)$

Then, the output at both hidden layer and output layer are each determined by Equation as defined below:

$O_j = f(net_j)$
$O_k = f(net_k)$

The input pattern is actually propagated through the entire network until the output pattern is produced. Basically, the BPN make used of generalized delta rule in order to determine the error. δ j denoted the error for all hidden layer units as below Equation and δ k denoted the error across all the output layer units as in below Equation:

$\delta j = f'(net_j)\sum_{k=1}^{k}.\delta_k.w_{kj}$
$\delta_k = (t_k\text{-}o_k)f'(net_k)$

Finally, each unit modifies its input connection weights slightly in a direction that reduces its error signal, and then the process is repeated for the next pattern. By applying a learning rate η, the weight change for a unit in hidden layer is determined by:

$\Delta w_{ij} = \eta\delta_j x_i$

While for output layer, the weight change could be determined from:

$\Delta w_{kj} = \eta\delta_k f(net_j)$

The very last step in back-propagation is to update the weight values in the system using the following equation.

$\Delta w_{ij}(n+1) = \eta\delta_j o_i + \alpha\Delta w_{ij}(n)$
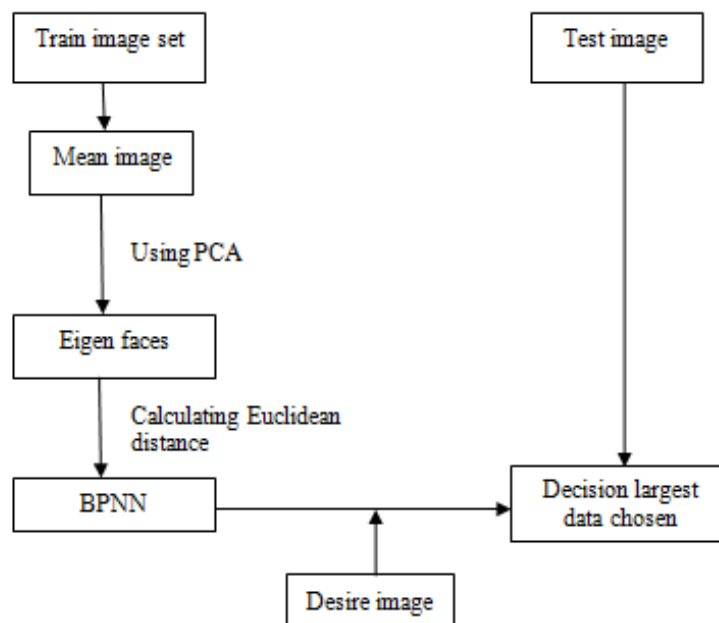
IV. FLOW CHAT FOR EXPERIMENTATION

Fig3: Complete process of PCA and BPNN face recognition system

V. SIMULATION RESULTS

TEST RESULTS OF FACE RECOGNITION USING PCA AND BPNN:

The simulation of the proposed approach was performed on MATLAB. The proposed method is tested on ORL face database. This database has more than one image of the individual's face with different conditions. The database is divided into two sets, which are, training database and the testing database. The network is trained on the training database and then one of the images from the testing database is fed as an input to test the network.
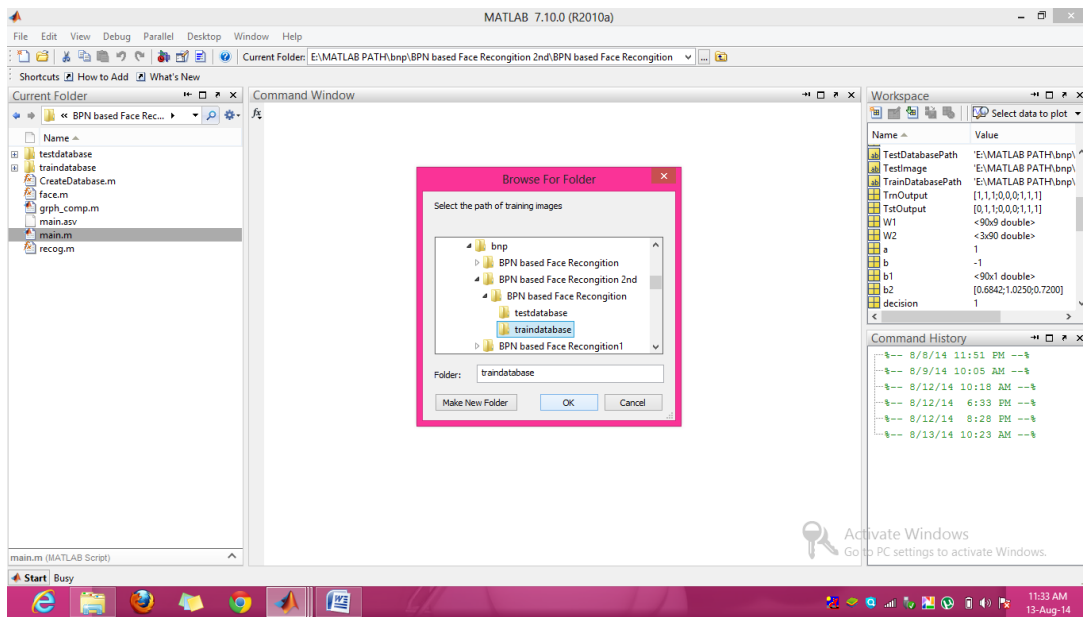
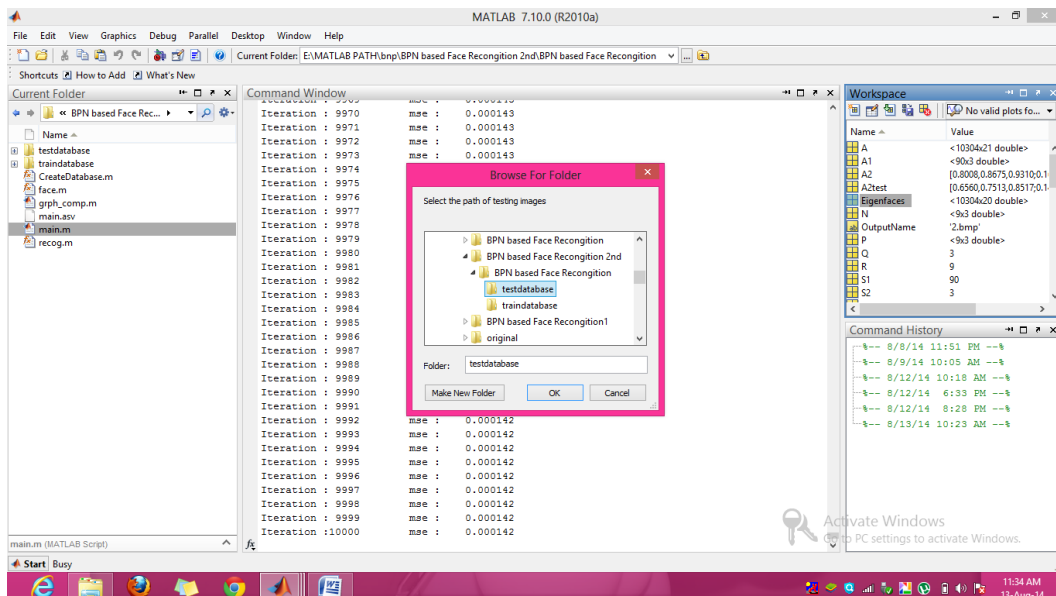Fig4: Training of Database Images

Fig5: Selecting the image to be tested
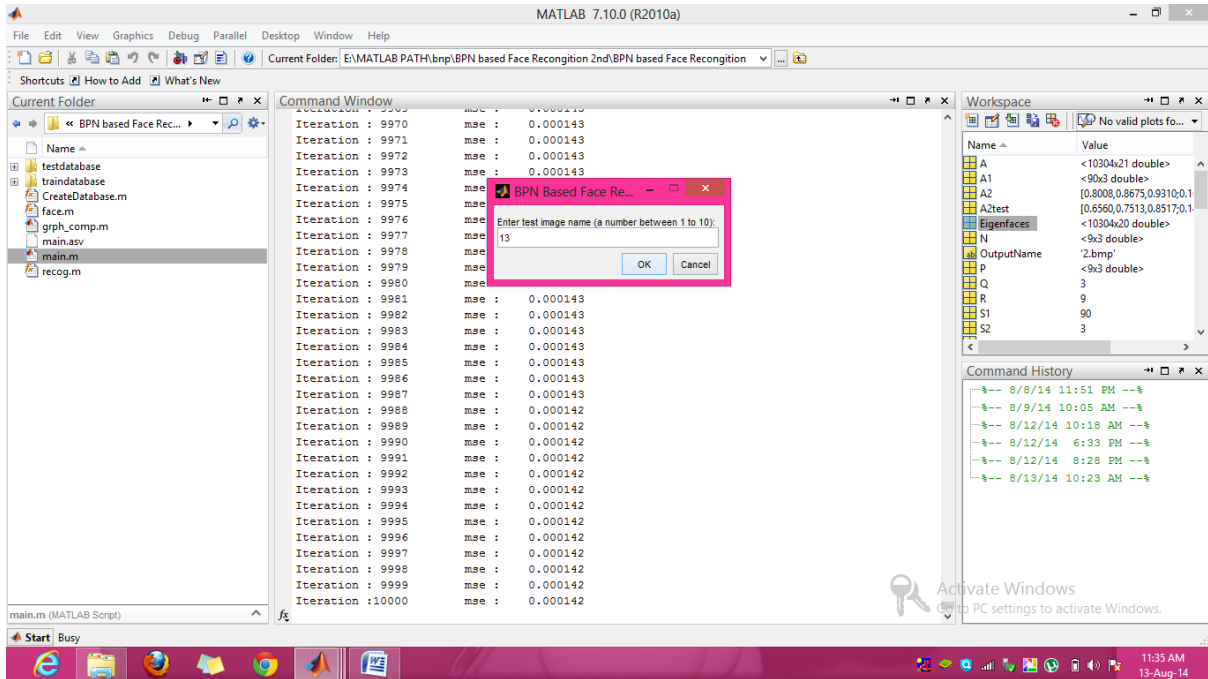
Fig6: Test Image selected
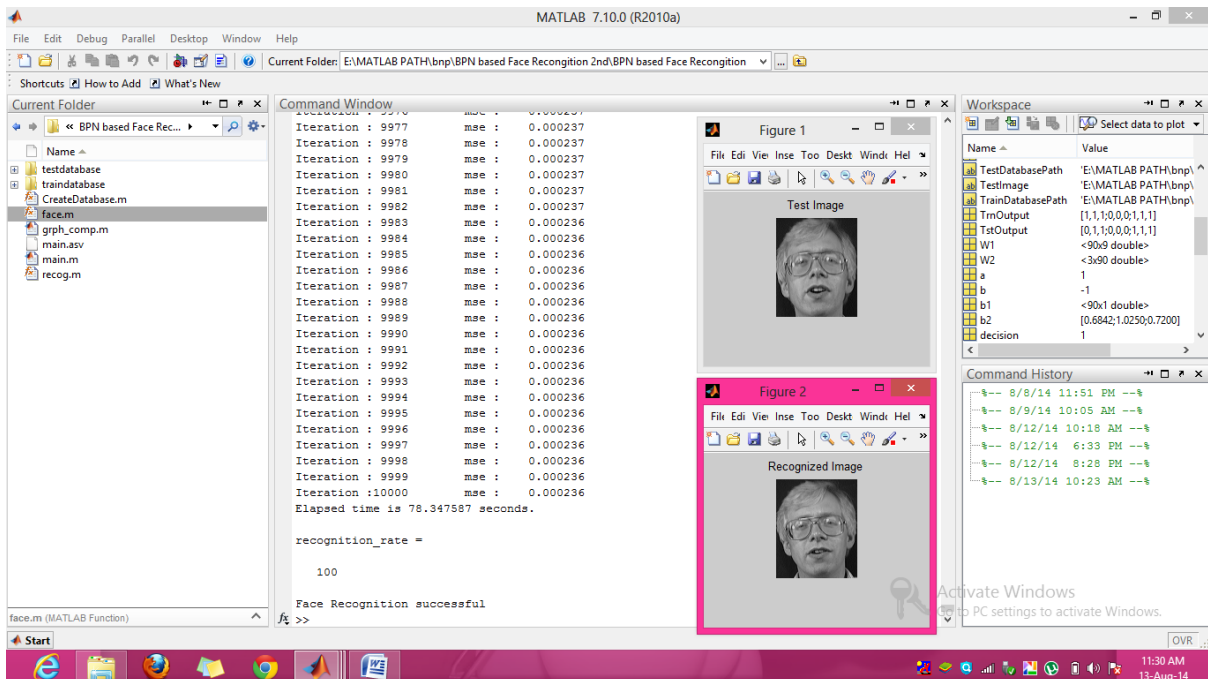
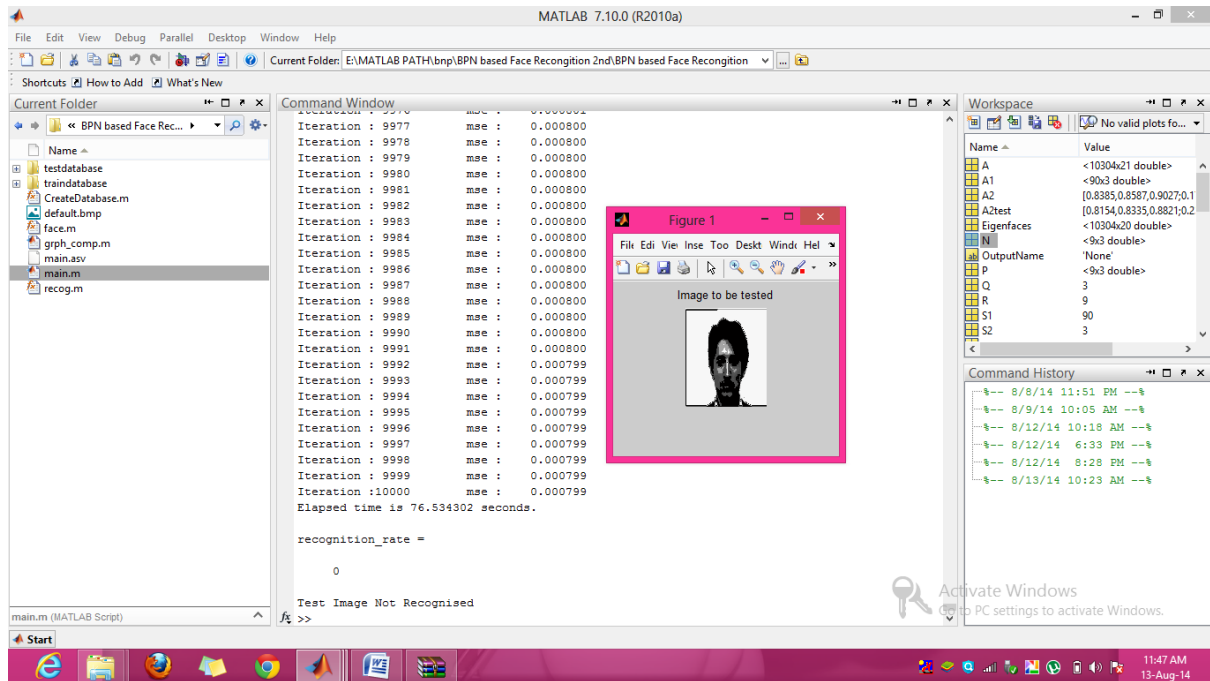Fig7: Output Result with image recognized from the database

Fig8: Output Result with image not recognized from the database

VI. CONCLUSION

The study shows that the face recognition system using PCA for feature extraction and BPNN for image classification and recognition provides a high accuracy rate and fast computation. By choosing PCA as the feature selection technique, the space dimension can be reduced. PCA combined with BPNN works better than the individual PCA, done on the basis of the performance of the system which is measured by varying the number of faces of each subject in the training and test faces. The recognition performance increases due to the increase in face images in the training set. This is because more sample images can characterize the classes of the subjects better in the face space. Hence it is concluded that this method has an acceptance ratio of more than 90% and the execution time of only a few seconds.

REFERENCES

1. http://murphylab.web.cmu.edu/publications/boland/boland_node17.html
2. R. Rojas(1996),"*Neural Network An Introductions*, Springer-Verlag, Berlin" IEEE Transactions of Neural Networks. vol.8, no.1,pp158-200
3. "Pattern Recognition and Neural Networks" by B.D. Ripley Cambridge University Press, 1996, ISBN 0-521-46086-7
4. Mohammod Abdul Kashem,Md. Nasim Akthar,Mshamim Ahmed and Md. Mahbub Alam "face recognition system based on principal component analysis(PCA) with back propagation neural network (BPNN)IJSER vol.2,issue 6,june-2011 ISSN 2229-5518
5. Jain, *Fundamentals of Digital Image Processing, Prentice-Hall Inc., 1982.*
6. Rafael C. Gonzalez and Richard E Woods, "DigitalImage Processing", Person Education Asia
7. A.Samal and P.A.Iyengar (1992): "Automatic recognition and analysis of human faces and facial expressions: A survey. Pattern Recognition".
8. M.A.Turk and A.P.Petland, (1991) "Eigenfaces for Recognition," Journal of Cognitive Neuroscience. vol. 3, pp.71-86.
9. K. Kim: Face Recognition using Principal Component Analysis, National Institute of Technology, Rourkela, India, 2008
10. S. Haykin, Neural Networks, A comprehensive foundation, Prentice Hall. Second Edition, 1999.