ISSN 2047-3338

Face Recognition Using PCA and Feed Forward Neural Networks

M.S.R.S. Prasad[1], S.S. Panda[2], G. Deepthi[3] and V. Anisha[4]
[1,2]Department of CSE, Regency Institute of Technology, Yanam, India
[3,4]Department of IT, Regency Institute of Technology, Yanam, India
prasad_merla@yahoo.com, shekhar_regency@yahoo.com

***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 Feed forward Neural Network (FFNN). Here 50 face images from the database are 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.**

***Index Terms*– Face Recognition, Principal Component Analysis, Feed Forward Neural Network and Acceptance Ratio**

I. INTRODUCTION

A face recognition system [1] 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 Feed Forward 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, and (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 [2]. 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 Feed Forward Neural Networks [3]. FFNN 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 as: input image→PCA→FFNN→output image

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 FFNN classifier to obtain the recognized image.

In this paper, Principal Component Analysis, and Feed Forward Neural Networks, has been demonstrated by different experiments and subsequent results are obtained.

II. PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA) [4] 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 [7]. 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 eigen-faces. 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 eigen-faces extracted from the original images, one can reconstruct the original image from the eigen-faces so that it matches the original image exactly.

A. 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′* eigen-faces that correspond to the highest eigen-value.

(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 $\Gamma_1, \Gamma_2, \ldots . \Gamma_M$, 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

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

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

This set of large vectors is then subject to principal component analysis, which seeks a set of *M* orthonormal vectors *u1….um*. 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… *M′*, where $M' \leq M$ is the number of eigen-faces used for the recognition. The weights form vector $\Omega = \omega 1, \omega 2, \ldots\ldots \omega M$ 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

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

where ΩK 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 eigen-faces 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.

B. Multi-layers Perceptron

The MLP neural network [5] has feed forward architecture within input layer, a hidden layer, and an output layer. The input layer of this network has N units for an N dimensional input vector. The input units are fully connected to the I hidden layer units, which are in turn, connected to the J output layers units, where J is the number of output classes. A Multi-Layers Perceptron (MLP) is a particular of artificial neural network [6]. We will assume that we have access to a training dataset of l pairs (xi, yi) where x i is a vector containing the pattern, while yi is the class of the corresponding pattern. In our case a 2-class task, yi can be coded 1 and -1.

We considered a MLP (Multi-Layers Perceptron) with a 3 layers, the input layer is a vector constituted by n2 units of neurons (n x n pixel input images). The hidden layer has n neurons, and the output layer is a single neuron which is active to 1 if the face is presented and to otherwise the activity of a particular neuron j in the hidden layer is written by [8].

$$S_j = \sum_{i \in input} w_{ji}\, x_i \;, X_i = f(s_j)\ (1),\ f$$

a sigmoid function, where W1i is the set of weights of neuron i, b1(i) is the threshold and xi is an input of the neuron.

Similarly the output layer activity is:

Sj = Σ w ji x i

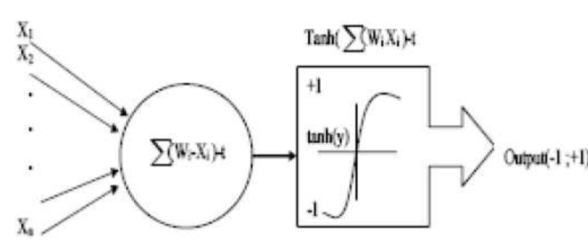

Fig. 1. The neuron of supervised training

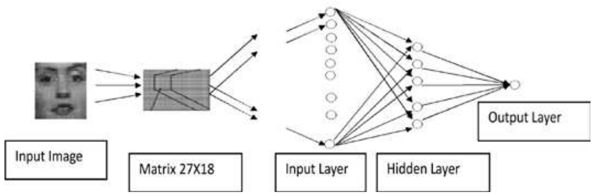

Fig. 2(a). Architecture of proposed System

In our system, the dimension of the retina is 27x18 pixels represent human faces and non-face, the input vector is constituted by 2160 neurons, the hidden layer has 100 neurons.

We are designing a feed forward neural network with one hundred neurons in the hidden layer and one neuron in the output layer which prepares images for training phase. All data form both "face" and "non-face" folders will be gathered in a large cell array. Each column will represent the features of an image, which could be a face, or not. Rows are as follows:

Row 1: File name

Row 2: Desired output of the network corresponded to the feature vector.

Row 3: Prepared vector for the training phase.

We will adjust the histogram of the image for better contrast. Then the image will be convolved with Gabor filters by multiplying the image by Gabor filters in frequency domain. To save time they have been saved in frequency domain before Features is a cell array contains the result of the convolution of the image with each of the forty Gabor filters. These matrices have been concated to form a bif 135x144 matrix of complex numbers.

We only need the magnitude of the result. That is why "abs" is used.135x144 has 10,400 pixels. It means that the input vector of the network would have 19,400 values, which means a large amount of computation. So we have reduced the matrix size to one-third of its original size by deleting some rows and columns. Deleting is not the best way but it save more time compare to other methods like PCA. We should optimize this function as possible as we can.

First training the neural network and then it will return the trained network. The examples were taken from the Internet database. The MLP will be trained on 500 face and 200 nonface examples [7], [8].

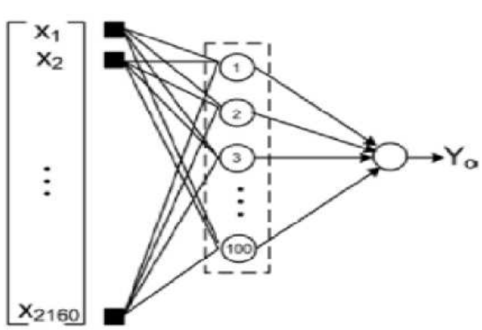

Fig. 2(b). Architecture of proposed system

C. Training Algorithm

For the efficient operation of the feed forward neural network it is necessary for the appropriate selection of the parameters used for training.

The 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.

The motivation for applying feed forward 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 along 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 an 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 Ω's are repeated for each additional hidden layer present, summing all the Ω's for units present in the +previous layer that is fed into the current layer for which Ω is being calculated.

Learning Rate- in FFNN, 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 [8].

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 feed forward neural network classifies the input image as recognized image.

D. 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. 4 is taken as the Input image. The mean image and reconstructed output image by PCA, is as shown in Fig. 4. In FFNN, a training set of 50 images is as shown in Fig. 5 and the Eigen faces and recognized output image are as shown in Fig. 3.

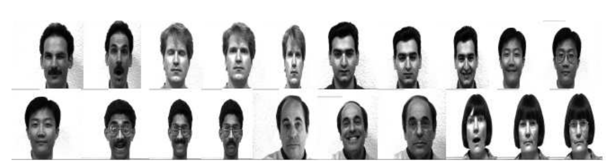

Fig. 3: Sample database

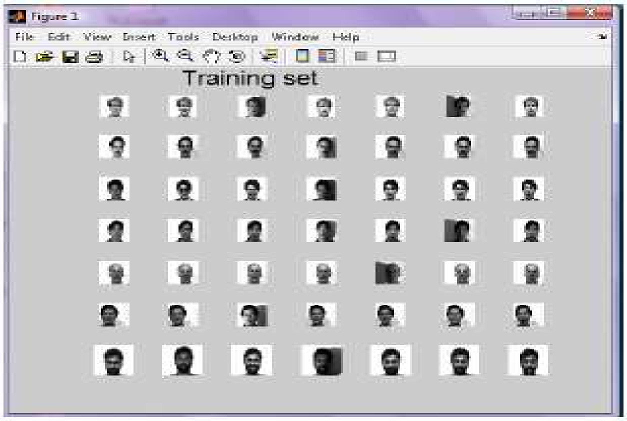

Fig. 4. Training set

Fig. 5. Recognised by FFNN method

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

Images	PCA	PCA with FFNN
40	92.4	96.5
60	90.6	94.3
120	87.9	92.8
160	85.7	90.2
2{x}	83.5	87.1

III. 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 FFNN 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**.**

REFERENCES

[1] 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.

[2] 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.

[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. 177-D-II, no. 11, pp. 2151-2159.

[4] M.A. Turk and A.P. Petland, (1991) "Eigenfaces for Recognition," Journal of Cognitive Neuroscience. vol. 3, pp.71-86.

[5] Yang M.H., Kriegman D.J., and Ahuja N., *"Detecting Faces in Images: A Survey"*, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 24, No.1, January 2002.

[6] Theocharides T., Link G., Narayanan V., Irwin M.J., *"Embedded Hardware Face Detection"*, 17th International Conference on VLSI Design, Mumbai, India, January 5-9, 2004.

[7] P. Latha, Dr. L. Ganesan, Dr. S. Annadurai, " Face Recognition using Neural Networks", Signal Processing: An International Journal (SPIJ) Vol. 3, Issue 5.

[8] Avinash Kaushal, J. P. S. Raina, "Face Detection using Neural Network & Gabor Wavelet Transform", IJCST, Volume 1, Issue 1, September 2010.