

An Enhanced Empirical Method on Choosing the Highest Principal Features and the Number of Hidden Neurons in Principal Component Analysis-Artificial Neural Network Face Recognition based System

Yasmin Nabil Mousa
Teaching Assistant at
Electronic and Communication
Dept., Modern Academy for
Engineering and Technology
(MAM), Cairo, Egypt

Abd - Al- Halim Zekry
Professor Doctor at Electronic
and Communication Dept., Ain-
Shams University
(ASU), Cairo, Egypt

Nariman Abd - Alsalam
Doctor at Electronic and
Communication Dept.,
Canadian International College
(CIC), Cairo, Egypt

ABSTRACT

With fast evolving technology, it is necessary to design an efficient security system which can detect unauthorized access on any system. It's needed to implement an extremely secure, economic and perfect system for face recognition that can protect systems from unauthorized access. So, in this paper, a robust face recognition system approach is proposed for feature extraction using Principal Component Analysis, and recognition using Feed Forward Back Propagation Neural Network. The proposed approach gave better results in all aspects including recognition rate, training time, elapsed time and mean square error. The paper shows that when using 80% of the dataset for training, the proposed system can achieve up to 97.5% recognition rate if correct number of input principal features to the classifier, learning rate, momentum, and number of hidden neurons are used. This algorithm is implemented using Matlab software, and tolerable error value used is superiorly chosen as $MSE = 0.0004$.

Keywords

Face recognition, Principal component analysis (PCA), Eigenvector, Eigenface, Artificial Neural network (ANN).

1. INTRODUCTION

Human beings have a very good recognition capabilities of faces and complex patterns and nothing can affect this capability. The neurons of the human brain are trained by learning the face of a person and they can identify that face quickly even after several years. The main reason for this is the high degree of, acquisition skills, interconnectivity, abstraction capabilities and adaptive quality of the human nervous system. There are various highly correlated biological neurons in human brain which can outperform super computers in certain specific tasks. A human face can be identified completely and perfectly by any small child, but it is a difficult task for the computer. Thus, the main goal is to design systems which can compete with the ability of a small child to identify and recognize faces, so that computers will be as lively as humans can [1]. Brain's ability of learning and identifying is converted into machine systems using the Artificial Neural Networks. The basic function for the face recognition system is to compare the face of a person which is to be recognized with faces already trained in the Neural Networks and it recognized

the best matching face as an output even at different viewing conditions, lightening conditions and facial expressions [2] [3]. Face recognition systems can be used in fields that needs more security. For example on Military bases, Government offices, Air ports, accessing big firms, etc. A face recognition system is evaluated as a good system if it can extract the important features, without making the system complex, and can use those features for recognizing the unseen faces. For feature extraction, Principle Component Analysis is used, the method is based on an information theory approach that decomposes face images into a small set of characteristic feature images called 'eigenfaces', which are actually the principal components of the initial training set of face images. For recognition, feed forward back propagation neural network is used. [4] [5].

There are various successful methodologies in human face recognition that are purposed in past decades. Kirby and Sirovich [6] have shown that the face images can be represented in terms of a best coordinate system termed as "eigenfaces". These are the Eigen functions of the average covariance of the ensemble of faces. They also purposed that even for large number of faces, the small number of eigenfaces needed. Matthew A. Turk and Alex P. Pentland [7] proposed a face recognition method based on the eigenfaces representation of faces, they presented an approach for detection and identification of human faces and described a working, near-real-time face recognition system.. V.P. Kshirsagar, M.R.Baviskar, M.E.Gaikwad [5] proposed a face recognition system using eigenfaces, , they achieved a very good success rate up to 85%. Artificial Neural Networks (ANN) is a very robust and powerful classification technique. Nazish Jamil, Samreen Iqbal, Naveela Iqbal Fatima Jinnah [8] proposed a face recognition system based on Neural Network, Mayank Agarwal, Himanshu Agrawal, Nikunj Jain, and Manish Kumar [9] proposed a Face Recognition system Using Eigen Faces and Artificial Neural Network. P.Latha, L.Ganesan, and S.Annadurai [10] proposed a face recognition system using Neural Network and achieved recognition rate 96.5% with using $MSE = 0.001$. Navneet Jindal and Vikas Kumar [2] proposed an Enhanced Face Recognition Algorithm using PCA with Artificial Neural Networks and achieved 94.5% recognition rate with setting $MSE = 0.001$.

Face recognition techniques are divided into two groups based on the face representation used:

1. Feature-based-Geometric approach: This approach uses geometric facial features (eyes, mouth, eye's brows, cheeks etc), and geometric relationships between them [5]. Advantages of this approach include efficiency when achieving data reduction, simplicity, economy and insensitivity to variations in view points and illumination. However, such geometric properties alone are not sufficient for face recognition because rich information contained in the facial appearance or texture is discarded [11]. Therefore, statistical approaches are often used to determine sufficient features. In statistical approaches, the feature extraction is based on statistical attributes of the selected face images, thus, the final set of features are very dependent on the subset of faces that were used in the training process. These statistical techniques are known as appearance based approaches [12].

2. Appearance-based-holistic approach: In this approach, holistic texture features are extracted into a subspace, and applied to either whole-face or specific regions in a face image. This approach operates directly on an image-based representation. It extracts features into a subspace derived from training images. These approaches have significantly advanced face recognition techniques [11].

In this paper, 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.

The following sections are as follows: In section 2 feature extraction phase using PCA approach and its mathematical model will be explained, in section 3 the Artificial Neural Networks (ANN) based recognition system and its mathematical model will be explained, in section 4 experiments and simulation are clarified, and in section 5 discussions and conclusion will be given.

2. FEATURE EXTRACTION USING EIGENFACES APPROACH

Principal Component Analysis (PCA) is a method based on information theory concepts where information that best describes a face is derived from the entire face image [9]. It involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components [10] [13]. In the language of information theory, the relevant information in a face image is extracted, encoded as efficiently as possible, then it is compared with a database of models. The proposed method is independent of any judgment of features (facial expressions, open or closed eyes, with and without Glasses, etc.) [9] [14]. The face images are decomposed into a small set of characteristic feature images called "eigenfaces" (which contain the common features in a face), they are extracted from the original training set of images by means of principal component analysis. For this purpose, a several images from the same person is taken to obtain the variation. Mathematically, the algorithm calculates the eigenvectors of the covariance matrix of the set of face

images, each image from the set contribute to an eigenvector, these vectors characterize the variations between the images. When these eigenvectors are represented, they are called eigenfaces. Every face can be represented as a linear combination of the eigenfaces, each eigenface represents only certain features of the face, however, they can be reduced to the ones with greater values, to make the recognition process more efficient. The losses due to discarding some of the eigenfaces can be minimized by choosing only the most important features (eigenfaces) [13] [15] [16].

PCA is closely related to popular signal processing technique known as the Karhunen-Loeve transform (KLT). It can in fact be shown that under the assumption that the data is zero-mean, the formulations of PCA and KLT are identical [17]. Based on the Karhunen-Loeve expansion in pattern recognition, any particular face can be represented in terms of a best coordinate system termed as "eigenfaces". These are the eigen functions of the average covariance of the ensemble of faces [9].

2.1 Principal Component Analysis Mathematical Model

PCA algorithm steps are shown in Fig. 1, and the algorithm mathematically is described as follows:

1. Assume the M sample images contained in the database (the training set of images) denotes as:

$$\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_M \quad (1)$$

2. The next step is to create the eigenspace, this step is the initialization of the system, it includes the following:
 - a. Calculate the average face of the set to remove much redundant information in the face:

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (2)$$

The mean image is represented as a column vector where each scalar is the mean of all corresponding pixels of the training images, it has the same size as the sample image (x*y)*1.

- b. Center the data: each of the normalized training face image is mean centered, this is done by calculating how each face differs from the average by the following equation:

$$\Phi_i = \Gamma_i - \Psi \quad (3)$$

Where $i = \{1, 2, 3, \dots, M\}$

- c. Create data matrix: this is matrix of the training faces having the pixels arranged in rows and M faces of training set arranged in columns [20]. The training faces are combined into a data matrix of size N by M where N is the dimension of image, and M is the number of training images.

$$A = [\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_M] \quad (4)$$

The covariance matrix is formed by:

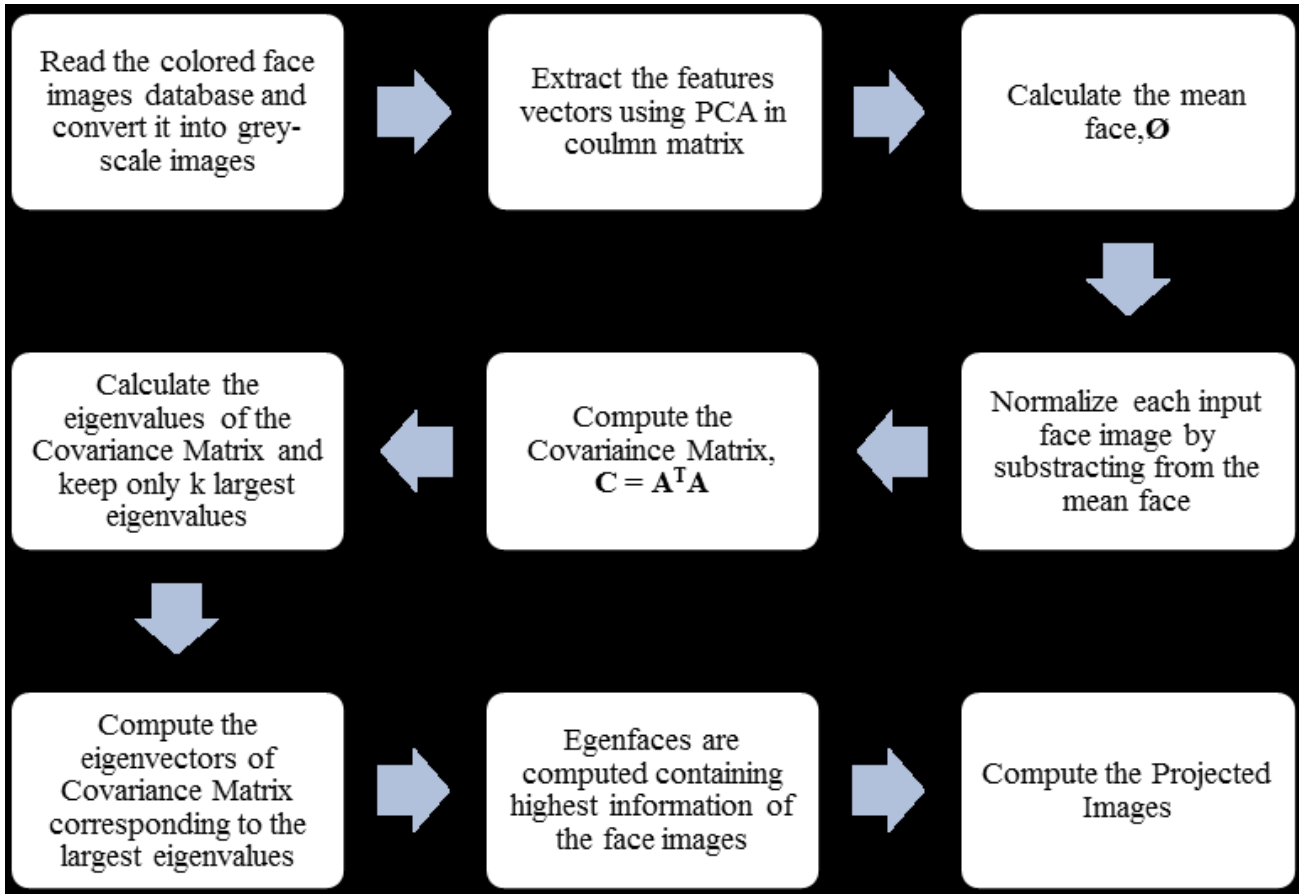


Fig.1: Features Extraction using PCA by computing the Eigenface Images

$$C = \frac{1}{M} \sum_{i=1}^M \Phi_i \cdot \Phi_i^T \quad (5)$$

$$= A \cdot A^T$$

- d. Calculate the eigenvalues of the covariance matrix C and keep only M' largest eigenvalues for dimensionality reduction:

The goal is to calculate the eigenvectors of the matrix $C = A \cdot A^T$, this can't be done directly because the size of $C = A \cdot A^T$ is N^2 by N^2 , this is very large even for small images, determining the N^2 eigenvectors and eigenvalues of C is a difficult and intractable task. Thus, a computationally feasible method is needed to find these eigenvectors to avoid this problem. If the number of images in the training set is less than the number of pixels in the image (i.e $M < N^2$), then the M by M matrix can be solved instead of solving a N^2 by N^2 matrix.. Finding the eigenvalues and eigenvectors of the matrix $L = A^T \cdot A$ instead of the matrix $C = A \cdot A^T$ could be done since the covariance matrix C is real and symmetric. Thus eigenvectors of L matrix, v_i can be calculated as follows:

$$A^T \cdot A v_i = \mu_i v_i \quad (6)$$

Where μ_i is the eigenvalues of L matrix, v_i is the corresponding eigenvector of L, these vectors determines linear combinations of the M training set face images to form the eigenfaces u_i .

The eigenvectors V (called eigenfaces) can be found by pre-multiplying equation 6 with the A matrix:

$$A A^T \cdot A v_i = \mu_i A v_i \quad (7)$$

From which it can be seen that the term $A v_i$ are the eigenvectors of the covariance matrix $A A^T$ and denoted by U where:

$$U = A v_i \quad (8)$$

$$U = \{u_1, u_2, u_3, \dots, u_M\} \quad (9)$$

$$v_i = \{v_1, v_2, v_3, \dots, v_M\} \quad (10)$$

$$u_i = \sum_{n=1}^M v_n \cdot \Phi_n, \quad (11)$$

Now the resulted eigenvectors can be sorted according to their corresponding eigenvalues from high to low with nonzero eigenvalues, according to their importance and principality in characterizing the variation among images. M' eigenfaces which have the largest associated eigenvalues are only selected and kept for dimensionality reduction as:

$$e_k = \sum_{n=1}^{M'} v_n \cdot \Phi_n, \quad (12)$$

Where $k = \{1, 2, 3, \dots, M'\}$

v_k are the largest selected eigenvectors

e_k are the selected eigenfaces.

These e_k selected eigenfaces now span a M' -dimensional face space instead of N^2 , and that is the new feature space for recognition. The matrix of the sorted eigenvectors forms the eigenspace E, where each column of E is one of the selected eigenvectors (eigenfaces). The principal components are the eigenspace E. With this analysis, the calculations are largely 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), as M eigenvectors are calculated instead of N^2 .

The efficiency of the PCA approach is high because all the M eigenvectors can be eliminated except the best M' ones (with the highest eigenvalues). Larger eigenvalue means that its corresponding eigenvector captures more of the data variance, thus contains more information about the face. Eliminating eigenvectors which are corresponding to small eigenvalues reduces the noise from the image and thus increase classifier's accuracy and its recognition capability.

3. Project the training images:

The projected training images y_i are computed based on the dot product of the centered training images Φ_i with each of the ordered eigenvectors, denoted as:

$$y_i = e_k^T \cdot \Phi_i \quad (13)$$

Where y_i is the projection of the centered training image Φ_i to the PCA eigenspace E defined by the reduced matrix of eigenvectors of the covariance matrix C [12].

The new vectors of projected images are the feature vectors of the training face images.

4. Recognition step by projecting testing images:

Given a new face of an individual t , it is transformed into its eigenface components by projecting it into the eigenspace. The pixels are concatenated the same way as the training images were, first, the vector of the testing face image t is mean centered by subtracting the mean image:

$$T = t - \Psi \quad (14)$$

Second, the feature vector of the testing image F is obtained by projecting the vector of the mean testing face image T into the eigenspace or the principal component:

$$F_k = e_k^T \cdot T \quad (15)$$

These calculated values of F together form a vector:

$$\Omega^T = [F_1, F_2, F_3, \dots, F_{M'}] \quad (16)$$

This vector describes the contribution of each eigenface in representing the input face image, treating the eigenfaces as a basis set for face images. In fact this is the projection of an unknown face into the eigenspace. Ω^T then is used to establish which of the predefined face classes best describes the new face in the classification phase [2] [5] [9] [10] [13] [14] [15] [17] [18] [19] [21].

3. ARTIFICIAL NEURAL NETWORK

Human brain consist of complex interconnected neurons to process different tasks, like solving noisy and complex data problems. These neurons are not depending on each other and they are working in an asynchronous manner. Artificial Neural Networks (ANN) idea is taken from the human biological nervous system for Face Recognition purposes, they are designed to learn the correlated patterns of input and target values. The Back-Propagation is the widely used and the best known learning algorithm in training multilayer perceptrons (MLP), this learning algorithm is supervised learning network based. MLP's are the networks consisting of a set of source nodes that form the input layer, one or more hidden layers of computation neurons, and an output layer of computation neurons [2] [13]. A typical back propagation network with Multi-layer, feed-forward supervised learning is as shown in the figure.2.

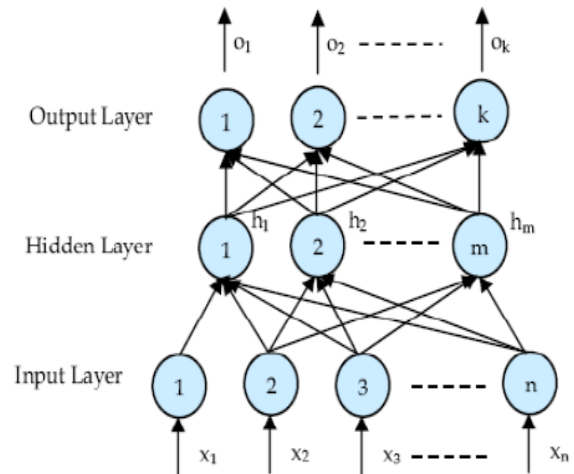


Fig.2: Artificial neural network

In face recognition system using ANN, the model works in the following frames:

Forward path: this path contains creating a feed forward network, initializing weights, simulation and training the network. Here the input layer processes the data to the hidden layer which computes the data further and passes it to the output layer [22]. Creating a feed forward network includes determining the number of neurons in the input layer, the number of hidden layers and their hidden neurons, and the number of neurons in the output layer. Input neurons receive the input pattern from the training data of face images, each input has its own weights.

Before training a feed forward network, the weight and biases must be initialized. Once the network weights and biases have been initialized, the network is ready for training.

The training process requires a set of inputs and target vectors. Data is processed by the input layer and passed it to the hidden layer which computes the data further and passes it to the output layer.

Backward path and adjust weights: The output layer compare the data propagated from the input layer in the forward path with the target value and obtain the error signals, errors are then sent back to adjust the weights of each layer inorder to minimize the network performance function [2] [3]. The default performance function for feed forward networks is mean square error, it is the average squared errors between the network outputs and the target output [23].

Target values are those values that the neural network must learn. These errors are sent back for adjusting the weights of each layer to minimize the error as shown in Fig.3.

Activation Transfer Function: the mathematical operation on the output signal are performed using an activation function. The functions can be threshold function, log-sigmoid and tan-sigmoid functions. If the output values of the function are the same as the output values of the tested face, the face is detected and recognized. Neural Networks recognize the input face which is similar to the image faces used for training process [2] [3].

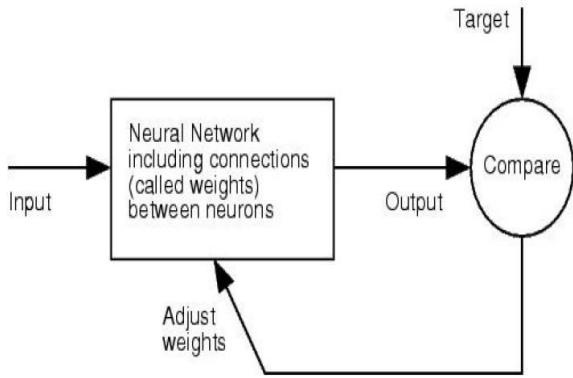


Fig.3: Back Propagation of multilayered ANN

3.1 Artificial Neural Network Training Mathematical Model:

There are many training algorithms for the feed forward networks, all these algorithms use the gradient of the performance function to determine how to adjust the weights in order to minimize the performance. The gradient is determined using the back propagation technique, which include making computational backwards through the neural network. In back propagation learning, network weights and biases are updated in the direction in which the performance function decreases more rapidly. One iteration of this algorithm can be as the following:

$$X_{k+1} = X_k - \alpha_k g_k \quad (17)$$

Where X_k = a vector of current weights and biases.

g_k = the current gradient

α_k = learning rate

This gradient descent algorithm can be implemented by two different modes: incremental mode and batch mode. In the incremental mode, each input is applied to the neural network, then the gradient is computed and the weights are updated. In the batch mode all inputs are applied to the neural network then the gradient is computed and the weights are updated. In this paper, Levenberg-Marquardt training (trainlm) is used. This algorithm operates in the batch mode and is invoked using training. Such the quasi-Newton methods, the

Levenberg-Marquardt training algorithm is designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as in training feed forward neural networks), then it is not needed to calculate Hessian matrix, it can be approximated by using the Jacobian matrix J with the following formula:

$$H = J^T J \quad (18)$$

And the gradient can be computed as

$$g = J^T e \quad (19)$$

Where J = the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases.

e = the network errors vector.

Computing the Jacobian matrix can be done through a standard back propagation technique which is much less complex than computing the Hessian matrix.

This approximation to the Hessian matrix is used by the Levenberg-Marquardt in the following Newton-like update:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \quad (20)$$

If the scalar μ is zero, then it approaches to Newton's method, using the approximate Hessian matrix. If the scalar μ is large, then this becomes gradient descent with a small step size. The aim is to shift towards Newton's method as quickly as possible because Newton's method is more accurate and faster near an error minimum. Thus, μ is decreased after each successful step, which means performance function reduction, and is increased only when an empirical step would increase the performance function. The performance function in this way will always be reduced at each iteration of the algorithm.

Levenberg-Marquardt training is more accurate and faster than the standard back propagation algorithm for learning such as delta rules, also it needs less input pattern to perform training than the other one [23] [24] [25] [26].

4. EXPERIMENTS AND SIMULATION

The flow diagram of the proposed system is shown in figure 4. This diagram consists of 5 steps: Preparing database, dimensionality reduction and feature extraction using PCA, network design, network training, network testing and classification.

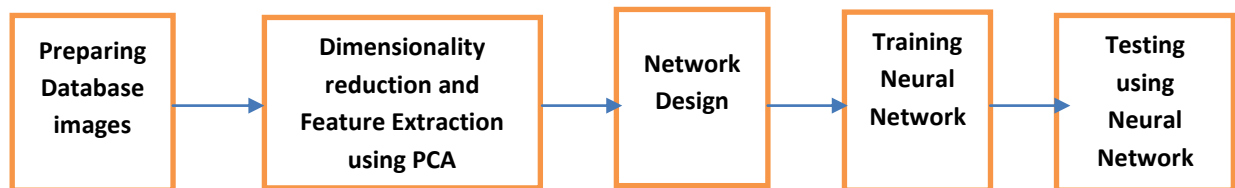


Fig.4: The proposed face recognition system

- **Preparing Database:** images in the data base are converted into gray scale, and reduced to $x \times y$ pixels (based on user request). The proposed method is tested on Olivetti and Oracle Research Laboratory (ORL) face database. In ORL there are ten different images each of 40 distinct subjects with different conditions. Each image has the size of 112×92 pixels with 256 levels of grey. For some subjects, the images were taken at different times, varying the facial expressions (open / closed eyes, smiling / not smiling), facial details (glasses / no glasses), and

lighting. All the images were taken against a homogeneous dark background with the subjects in an upright, frontal position (with tolerance for some side movement). A preview image of the database of faces was shown in Figure .5.

In this work, 8 input patterns with 80 gray scaled sample face images from ORL database are used, each pattern with 10 images, the original pictures of 112×92 pixels have been resized to 48×48 so that the input space has the reduced dimension of 2304 instead of 10,304.

- **Dimensionality reduction and Feature extraction using Principal Component Analysis:**

Implementing the PCA mathematical model described in section 2.1 briefly, and that clarified in figure 1. First images are made zero-mean and unit-variance. Second, going through steps and calculating the covariance matrix, will result M eigenvectors, the M' highest eigenvectors corresponding to the highest eigenvalues are chosen, and these are the eigenfaces of the training set. Eigenfaces of highest eigenvalue are actually needed to produce a complete basis for the face space. The choice of M' is depending on the ratio: Number of training images : Number of testing images.

Depending on how many training images are taken in a case, and correspondingly the number of eigenvalues and eigenvectors found, M' will be chosen. The chosen M' (top eigenfaces chosen) are now input to the neural network classifier. In experiments, for testing the whole database, the ratio between number of training and testing images was varied and the recognition rate results were reported, for 90 : 10, 80 : 20, 30 : 70, 40 : 60, and 50 : 50 cases, i.e training : testing ratio respectively.

In the proposed work, the number of M' largest eigenfaces is chosen by discarding the last 65% to 75% of the total number of the eigenvectors. The desired percentage is experimented through trial and error within this mentioned bound, and the best choice of M' means the choice which results in the highest recognition rate. For an example, in the 80 : 20 training : testing images case, the number of training images = 64, the number of testing images = 16, then the number of the resulted eigenvectors = 64, and the trial and error experiments to choose M' will be within values in the following empirical range formula:

$$\frac{\{\text{No. of training images} - (0.75 * \text{No. of training images})\}}{\{\text{No. of training images} - (0.65 * \text{No. of training images})\}} \quad (21)$$

i.e { 64 - (0.75 * 64) to 64 - (0.65 * 64) }, means the chosen M' are the values (16, 17, 18, 19, 20, 21, 22) eigenfaces. This procedure is done for the all resting training: testing ratio cases.

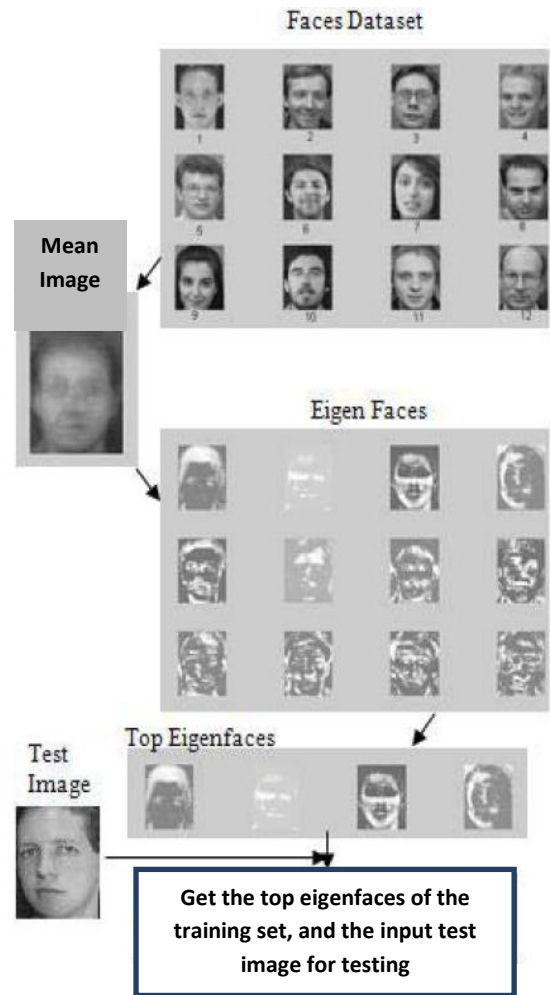


Fig.5: Face dataset, mean image of on the dataset, Eigenface and ANN based faced recognition system

- **Network design step:** The initial parameters of the Neural Network used in the experiment are given below:
 - Training type: Levenberg-Marquardt training.
 - Network type: Feed forward back propagation network.
 - Number of layers: 3 (input, one hidden, output layer)
 - Number of neurons in input layer: Number of features chosen (no. of eigenfaces to describe faces).
 - Number of neurons in hidden layer: depends on the proposed trial and error method in this work.
 - Number of neurons in output layer: 3
 - Transfer function of the hidden layer: tansig.
 - Transfer function of the output layer: logsig.
 - Training Function: trainlm.
 - Back propagation weight/bias learning function: learnqdm.
 - Number of epochs used in training: 1000.
 - Performance function: Mean square error.

Experiments showed that the training function Levenberg-Marquardt training (trainlm) is the most suitable training algorithm to train the feed forward neural network using back propagation algorithm being gradient decent rather than other training functions. Training occurs according to the trainlm training parameters, such as performance goal or error tolerance, minimum gradient, mu, mu decrement, mu increment, mu max, minimum gradient, maximum fail, ... etc, using defaults are more practical.

Another important parameter in designing the neural network is the number of hidden neurons. Deciding the number of hidden layer's neurons is a very important part in determining the overall neural network architecture. In spite of that these layers do not interact directly with the external environment, they have a great effect on the final network output. The hidden layer is responsible for internal representation of the data and the information transformation between input and output layers. If too few neurons are used in the hidden layer, the network may not contain sufficient degrees of freedom to form a representation, and this will lead to underfitting problem. If too many neurons are used, the network will be overtrained, and this problem is called overfitting. The two problems will affect the network output and the resulted recognition rate. It is clear that some compromise must be reached between too many and too few neurons in the hidden layers [24].

In the proposed work, a new empirical method for selecting the number of hidden neurons is achieved, based on the number of M' inputs introduced to the Neural Network. Ultimately the selection of an architecture of neural network will come down to a trial and error, by making experiments through the following empirical range formula:

$$\{ 1.5 * M' \text{ to } 3 * M' \} \quad (22)$$

At each experiment, depending on the number of training images, M' value ranges are determined as described in formula (21). Each time, one of these M' values is chosen and set, and a trial and error successive experiments in varying the number of hidden neurons through the range values described in the formula (22) is performed. After each run the recognition rate is observed, and the highest recognition rate will be recorded for that certain M' and the used number of hidden neurons. This described procedure will be repeated for all M' values. After finishing all trials for all M' range values, the highest recognition rate corresponds to M' value and the number of hidden neurons used at that experiment is chosen to be the recognition rate of the overall system at this case.

For an example, in the 80 : 20 training : testing images ratio case, the number of training images = 64, then the range of M' values will be between $\{ 64 - (0.75 * 64) \text{ to } 64 - (0.65 * 64) \}$,

means the chosen M' are the values (16, 17, 18, 19, 20, 21, 22). And through successive trials, it is found that at $M' = 21$, and when the number of hidden neurons are tried in the range $\{(1.5 * 21) \text{ to } (3 * 21)\}$, the highest recognition rate was recorded at 44 hidden neurons to be 97.5%. . The same procedure is performed for the rest training: testing ratio cases.

Training step: the designed network is trained with a network training tool which is available in the neural network toolbox in Matlab. Levenberg-Marquardt training is used, the input is propagated forward through the network to compute the output vector. The output vector is compared with the desired output, and errors are determined, the errors in the output layer are then sent back through the network from the output to input and weights of these layers are updated, which minimize the error. The momentum and learning rate parameters counts the updates from previous iterations and recalculates the new updated output. The iterations used is 1000, learning rate used value is 0.7, momentum used value is 0.9, and the errors are minimized to value MSE= 0.0004. The process is repeated until the errors being minimized to reach the desired tolerable performance function value.

• **Testing and Classification step, For MSE=0.0004:**

This proposed empirical methods for choosing M' eigenfaces and the number of hidden neurons achieves very high recognition rates for all cases:

- ✓ For 90 : 10 case, the achieved recognition rate is 98.75 % at $M' = 19$, 40 hidden neurons, 22 epochs, and training time 23 sec.
- ✓ For 80 : 20 case, the achieved recognition rate is 97.5 % at $M' = 21$, 44 hidden neurons, 32 epochs, and training time 18 sec.
- ✓ For 70 : 30 case, the recognition rate is 91.25 % at $M' = 16$, 42 hidden neurons, 15 epochs, and training time 3 sec.
- ✓ For 60 : 40 case, the achieved recognition rate is 86.25 % at $M' = 15$, 44 hidden neurons, 21 epochs, and training time 13 sec.
- ✓ And finally for the case 50 : 50 the achieved recognition rate is 80 % at $M' = 10$, 23 hidden neurons, 320 epochs and training time 15 sec.

All experiment results for all chosen M' values are recorded; i.e the number of hidden neurons, number of epochs, training time, elapsed time, and the recognition rate. The detailed results through all experiments are clarified in the following tables:

Table 1. For 90% Training, 10% Testing, Number of training images= 72 , Number of testing images=8

No. of eigenfaces input to the classifier	No. of neurons in the hidden layer	No. of epochs	Training time (seconds)	Elapsed time(seconds)	Recognition rate%
18	40	62	23	33.897715	97.5
19	40	22	23	14.650846	98.75
20	24	15	53	52.961025	97.5
21	57	15	12	13.504914	98.75

Table 2. For 80% Training, 20% Testing, Number of training images= 64 , Number of testing images=16

No. of eigenfaces input to the classifier	No. of neurons in the hidden layer	No. of epochs	Training time (seconds)	Elapsed time(seconds)	Recognition rate%
17	50	163	187	118.537535	95
18	50	17	19	20.454908	90
19	56	24	15	15.948398	85
20	39	22	6	7.720546	93.75
21	44	32	18	19.699123	97.5

Table 3. For 70% Training, 30% Testing, Number of training images= 56, Number of testing images= 24

No. of eigenfaces input to the classifier	No. of neurons in the hidden layer	No. of epochs	Training time (seconds)	Elapsed time(seconds)	Recognition rate%
14	39	281	113	114.317337	86.25
15	44	18	9	10.828556	88.75
16	42	15	3	4.172359	91.25
17	50	40	30	31.877310	85

Table 4. For 60% Training, 40% Testing, Number of training images=48, Number of testing images= 32

No. of eigenfaces input to the classifier	No. of neurons in the hidden layer	No. of epochs	Training time (seconds)	Elapsed time(seconds)	Recognition rate%
12	31	22	2	2.933207	78.75
13	31	25	2	3.131336	83.75
14	22	18	11	12.211408	82.5
15	44	21	13	13.865089	86.25
16	29	35	31	46.643075	81.25
17	32	32	34	36.945159	83.75

Table 5. For 50% Training, 50% Testing, Number of training images=40, Number of testing images=40

No. of eigenfaces input to the classifier	No. of neurons in the hidden layer	No. of epochs	Training time (seconds)	Elapsed time(seconds)	Recognition rate%
10	23	320	15	15.666146	80
11	16	27	19	20.752887	76.25
12	27	33	25	26.628254	75
13	17	36	25	26.824671	68.75

4. DISCUSSION AND CONCLUSION

- All experiments are achieved setting the mean square error to 0.0004 which is a contribution in this work over all previous works that set the MSE to 0.001 and 0.01, means a higher accuracy algorithm is presented.
- Larger eigenfaces does not mean more recognition capabilities. If too many eigenfaces are selected to be introduced to the classifier, then they will be considered as noisy images and that will affect the performance and the accuracy of the classifier.
- In this paper, back propagation feed forward Artificial Neural Networks with features extraction using PCA is purposed for face recognition. The purposed face recognition system works with high accuracy and provides better recognition rates. The proposed technique is analyzed by using an empirical method based on varying the number of eigenfaces used for feature extraction and an empirical method for determining number of hidden neurons in the hidden layer-based on the number of features input introduced to the network- for optimum network design to achieve high recognition rates.
- The proposed approach gave better results in all aspects including recognition rate, training time and elapsed time which are reduced to few seconds, at mean square error 0.0004 as a tolerance error level, if a correct learning rate, momentum, a suitable adequate number of hidden neurons are assigned to the network, and a right choice of the number of principal features M' that will be introduced to the neural network are used.
- This paper showed also back propagation neural network is more sensitive to structuring parameter (hidden neurons) and learning parameters (learning rate). The paper also gave some comparative analysis like:
 - ✓ Effect of various the number of training images on the recognition rate.
 - ✓ Effect of numbers of hidden neurons on training time and elapsed time.
 - ✓ Effect of numbers of hidden neurons on recognition rate.
 - ✓ Effect of number of eigenfaces as a principal features introduced to the Neural Network classifier on recognition rate.

5. REFERENCES

- [1] Prachi Agarwal, Naveen Prakash “An Efficient Back Propagation Neural Network Based Face Recognition System Using Haar Wavelet Transform and PCA”, International Journal of Computer Science and Mobile Computing (IJCSMC), Vol. 2, Issue. 5, pg.386 – 395, May 2013.
- [2] Navneet Jindal, Vikas Kumar, “Enhanced Face Recognition Algorithm using PCA with Artificial Neural Networks”, Electronics and Communication Engineering Department Samalkha Group of Institutes, IJARCSSE, pp 864-872, 2013.
- [3] Jyotshana Kanti, Shubha Sharm, “Automated Attendance using Face Recognition based on PCA with Artificial Neural Network”, International journal of science and research IJSR, 2012.
- [4] Raman Bhati, Sarika Jain, Nilesh Maltare, Durgesh Kumar Mishra, “A Comparative Analysis of Different Neural Networks for Face Recognition Using Principal Component Analysis, Wavelets and Efficient Variable Learning Rate”, Int'l Conf. on Computer & Communication Technology, IEEE, pp. 526-531 2010.
- [5] Prof. V.P. Kshirsagar, M.R. Baviskar, M.E. Gaikwad, “Face Recognition Using Eigenfaces”, Dept. of CSE, Govt. Engineering College, Aurangabad (MS), India, IEEE pp 302-306, 2011.
- [6] Kirby, M., and Sirovich, L., "Application of the Karhunen-Loeve procedure for the characterization of human faces", IEEE PAMI, Vol. 12, pp. 103-108, (1990).
- [7] Matthew A. Turk and Alex P. Pentland “Face Recognition Using Eigenfaces”, Vision and Modeling Group, The Media Laboratory Massachusetts Institute of Technology IEEE pp.586-591, (1991).
- [8] Nazish Jamil, Samreen Iqbal, Naveela Iqbal, Fatima Jinnah, “Face Recognition Using Neural Networks”, Women University, Rawalpindi, Pakistan, IEEE pp. 277-281, 2001.
- [9] Mayank Agarwal, Nikunj Jain, Mr. Manish Kumar and Himanshu Agrawal, “Face Recognition Using Eigen Faces and Artificial Neural Network”, International Journal of Computer Theory and Engineering, pp 624-629 August, 2010.
- [10] P.Latha, L.Ganesan, and S.Annadurai, “Face Recognition using Neural Networks”, Signal Processing: An International Journal (SPIJ), pp 153-160.
- [11] Borut Batagelj and Franc Solina, “Face Recognition in Different Subspaces: A Comparative Study”, University of Ljubljana, Faculty of Computer and Information Science.
- [12] Simon Gangl, Domen Mongus, and Borut Z'alik, “Comparison of face recognition algorithms in terms of the learning set selection”, Laboratory for Geometric Modelling and Multimedia Algorithms, SICESCG, the Central European Seminar on Computer Graphics, 2010.
- [13] f.Ujval Chaudhary, Chakoli Mateen Mubarak, Abdul Rehman, Ansari Riyaz, and Shaikh Mazhar, “Face Recognition Using PCA-BPNN Algorithm”, International Journal of Modern Engineering Research (IJMER), pp.1366-1370, Vol.2, Issue.3, May-June 2012.
- [14] Vinodpuri Rampuri Gosavi, G.S. Mandal's, “A New Era of Face Recognition”, International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 8, October – 2012.
- [15] Pavan k. Yalamanchili and Bhanu Durga Paladugu, “ECE847 : Digital Image Processing, Comparative Study of Face Recognition Algorithms”, Department of Electrical Engineering, Clemson University, South Carolina, USA.
- [16] Jorge Orts, “ECE533 – Image Processing Project on Face Recognition Techniques”, the university of Wisconsin Madison.

- [17] Kresimir Delac , Mislav Grgic , Panos Liatsis, “Appearance-based Statistical Methods for Face Recognition”, International Symposium ELMAR, pp 151-158, June 2005.
- [18] Shahrin Azuan Nazeer and Marzuki Khalid, “PCA-ANN Face Recognition System based on Photometric Normalization Techniques” Telekom Research & Development Sdn. Bhd., University Technology, Malaysia, Source: State of the Art in Face Recognition, Book edited by: Dr. Mario I. Chacon M., ISBN -3-902613-42-4, pp. 250, , I-Tech, Vienna, Austria, Jan 2009.
- [19] A.Swathi1 , Dr.R.Pugazendi, and , K.S.Rangasamy, “Face Recognition Using Multi-Support Vector Machines”, International Journal of Computer Trends and Technology (IJCTT) – volume 4 Issue 8pp. 2556-2560– August 2013.
- [20] Fábio Abrantes Diniz, Francisco Milton Mendes Neto, Francisco das Chagas Lima Júnior, and Laysa Mabel de Oliveira Fontes, “A Facial Recognition System Based on Techniques of Principal Component Analysis and Autofaces with K-NN, K-Star and Random Forest Classifiers”, Research Notes in Information Science (RNIS), Volume12, pp. 7-14, April 2013.
- [21] Matthew A. Turk and Alex P. Pentland “ Eigenfaces for recognition”, Vision and Modeling Group, The Media Laboratory Massachusetts Institute of Technology, journal of cognitive neuro science, pp 71-86, 1991.
- [22] Jamshid Nazari, Implementation of back-propagation neural networks with Matlab., ECE Technical Reports. Paper 275, <http://docs.lib.purdue.edu/ecetr/275>, 1992.
- [23] E. Hosseini Aria, J. Amini, M.R.Saradjian, “Back Propagation Neural Network for Classification of IRS-1D Satellite Images, Department of geomantics, Faculty of Engineering, Tehran University, Iran, Jamebozorg G., National Cartographic Center (NCC), Tehran, Iran.
- [24] Mohamed Elwakeel, Ziyad Shaaban, “ Face Recognition based on Haar wavelet transform and Principal Component analysis via Levenberg-Marquardt back propagation neural network”, European Journal of scientific Research, pp 25-31, 2010.
- [25] <http://www.rohan.sdsu.edu/doc/matlab/toolbox/nnet/backpr54.html>.
- [26] <http://in.mathworks.com/help/nnet/ref/trainlm.html>.