Face Recognition using PCA and Multilayer Feedforward Neural Networks

Suman Saha
Department of Computer Science and Technology
University of Bedfordshire, United Kingdom.

*Corresponding author E-mail: sumansaha1981@gmail.com

Abstract

This paper presents a report on various phases of our proposed face recognition system. It covers the following topics. The description and structure of image data, analysis of image features, analysis of methods for feature processing and extraction, such as principal components analysis (PCA), independent component analysis (ICA), linear discriminant analysis (LDA), analysis of recognition methods, such as multi-class and pair-wise neural networks (MCNN & PWNN), implementation and designing of recognition methods. Finally the report demonstrates the various experimental results on our face recognition system and concludes on the basis of these experimental outcomes. As a part of future works an implementation of 3D face recognition system is suggested.

Keywords: Face Recognition, PCA, ICA, LDA, ANN, MCNN, PWNN, Gradient Descent, feed-forward, multi-layer, back-propagation, mse, Levenberg-Marquardt

I. Introduction

This paper presents a report on various phases of our IA-based system for face recognition (FR) task. MATLAB source-codes for face recognition system have been provided by our University as a supportive material for the task (Schetinin, 2013). First the MATLAB source codes have been studied and understood. Algorithms for different modules of face recognition system, such as image processing, features extraction, neural network configuration, training, testing (presented in Section-VI) have been prepared based on these codes. MATLAB-based face recognition application has been developed which allow users to input a face image and view the output of the recognition system (presented in Section-VI).

The main requirements for our face recognition system are mentioned as follows:-

1) It is capable of learning from the image dataset which is periodically updated (the first requirement).
2) It is able to recognize new person recently added to the data set (the second requirement).
3) It learns to recognize a person from his/her multiple images to enhance the accuracy of face recognition. The variation among these different images of same persons caused by different light, emotions, noise, etc (the third requirement).

The standard artificial neural network (ANN) techniques are used to solve the face recognition problem (BISHOP, 1995). Multi-class neural network (MCNN) and pair-wise neural network (PWNN) models are selected to design the recognition system (Uglov et al., 2008). For achieving higher accuracy it uses PWNN which gives better performance with the above mentioned requirements (Hastie&Tibshirani, 1998). To increase the performance of the FR system image data are normalized.

II. The Description and Structure of Image Data

2.1 The description of image data

Our FR system uses JPEG face images from Yale extended B dataset (Georghiades, Belhumeur & Kriegman, 2001). All the images are grey scale. Grey scale images are used for simplicity as it has RGB values between 0 to 255. The image database has face images of 38 different people. Each person has 60 different face images and these images are taken with different illumination conditions (Georghiades, Belhumeur & Kriegman, 2001). These 60 images per person are used to enhance the accuracy of the face recognition system and thus it fulfills the 3rd requirement.

2.2 The structure of image data

Every face image in the dataset has fixed dimension and it is 68x77 pixels. The 68x77 pixels image is resized to 32x32 pixels to fulfill the requirement of principal component analysis (PCA) technique which is used to reduce the dimensions of the image data (Chengjun Liu&Wechsler, 2000).

III. Analysis of image features

Image features can be categorized into two main sections, they are as follows:

3.1 Appearance-based

The following are the image features which produce variability in images of a same person:

3.1.1 Illumination or lighting

One person's images can vary due to different illumination conditions or variation in lighting (Georghiades, Belhumeur & Kriegman, 2001). Different illumination conditions can be described as the distance and angle between face and source or sources of light(s).

3.1.2 Distance between the face and the camera.

3.1.3 Viewpoint

The person's viewing direction or pose can produce variability in images (Georghiades, Belhumeur & Kriegman, 2001). In other words, the angle between the face and camera can produce difference in face images (Coppin, 2004).

3.2 Feature-based

Facial attributes, such as eyes, nose, mouth, chin, hairstyle, beards and relations among these attributes affect the variation in face images. As for example the distance and angle between these attributes produce different images. Apart from these attributes glasses, sun-glasses, hats, earrings, age, emotions etc. can significantly produce variations in face images (Coppin, 2004).

IV. Analysis of Methods for Feature Processing and Extraction

4.1 Feature processing methods

In the initial phase of feature processing the face images are converted to grey scale, resized and noise is eliminated (Yesu et al., 2012). Then normalization techniques are applied on image data. There are various normalization techniques exist. Discrete Wavelet Transform (DWT) and Histogram Equalization (HE) are two methods for normalisation task as mentioned in (Abbas et al., 2008).

The basic idea of feature processing is to reduce the variations in a face image so that the performance of the FR system can be enhanced. A variation in a face image happens due to illumination (Ranawade, 2010). In our face recognition system there are two methods used for feature processing, they are as follows. First normalise the grey image data within 0 to 255 values. Second method is applied after feature extraction process. It normalise the image data by mapping each row's mean to 0 and standard deviation to 1 (Inc. Mathworks, 2013).

4.2 Feature extraction methods

One of the significant phases of FR system is feature extraction. Features extraction can be defined as the process of reducing the dimensions of face image data and thus eliminating the redundant and noisy information from it. It is required because a face image has a very large dimensionality and it requires high computational cost to process this data. Dimensionality reduction using feature extraction methods enhance the performance of the FR system with respect to computational cost, speed and accuracy in recognition. Facial features are extracted using various techniques. Some of the techniques for features extraction are: principal components analysis (PCA) (Turk & Pentland, 1991), independent components analysis (ICA) (Leach, 2002), linear discriminant analysis (LDA) or fisher discriminant analysis (FDA) (Marcialis&Roli, 2002), discrete cosine transform (DCT) coefficients, DWT coefficients (K et al., 2010). In our face recognition system PCA or Karhunen–Loève expansion is used to extract features of face image data (Turk & Pentland, 1991).

4.2.1 Principal Components Analysis (PCA)

It is an important and very popular statistical method to identify the N main data items (features) among given M data items of an object (in our case the object is a face image and data items are the image pixel values) (Oravec et al., 01, April, 2010). PCA extracts the main data items or features which differ most from one object to another (that is one face image to another face). These features are then used to classify one object from all other objects and thus help in recognition process. The number of features extracted are smaller or equal to the number of total data items, that is N<=M. These main features are called principal components of a given object. Generally data processed with PCA are given as training data.

Our face recognition system uses PCA on a face image data vector of dimension 1024x1 pixel-values. After processing this set of data items, PCA returns a lower dimension vector of size 101x1. Among 1024 given data items (pixel-values) of a face image, PCA extracts the 101 principal components which are the independent features help to identify a face.

In other words PCA takes a column vector of size M as input and returns a lower dimension column vector of size N, where N<M. Thus, PCA is also termed as dimensionality reduction tool (Turk & Pentland, 1991). In real world problems, such as human face recognition system, the objects are usually multi-dimensional (2-D, 3-D or even more). When these objects are converted to vector space we need some tool to reduce the dimension of the vector space so that only the principal components are present and rest are eliminated. The basic idea is to extract the independent or uncorrelated variables which contribute to the total variation and remove the dependent variables whose contribution is negligible. The independent uncorrelated variables are ordered as per their contribution to total variation.

4.2.1.1 Mathematical representation of PCA

PCA remaps m-dimensional input data into a lower dimensional space in such a manner so that the components which contribute to very low variance are eliminated (Sengupta, 2013). A transformation matrix T is designed to perform this data elimination job. Input vector x is mapped into Tx vector where T is an mxm matrix. The basic idea is to find the T matrix. To find the T matrix , the input vector x is projected onto a set of m-dimensional input vectors qi and the solutions are obtained in the form of eigenvalues λi, which are the solution of the correlation matrix R and eigenvectors onto which the input vector x is projected.

Data processed with PCA are easily solvable by ANN (Sengupta, 2013). There are some real-time problems where the dimensionality of data is extremely large in size. The main problem with the large dimensional data is that it leads to an increased size of the neural networks.

Input vector x consist of x1, x2, x3, …, xm elements, where m is the size of the input vector x and it is a large number. T is a mxm matrix. Tx = x_new, here x_new is the transformed space of input vector x which is an m dimensional vector. Here x vector is pre-multiplied with T matrix. It is possible to get in the transformed space of x (that is x_new) that out of m dimensions, m0 dimensions are significant and rest are not so useful. PCA creates a transformation function T such that in the transformed space that is x_new, the low variance terms can be removed (Sengupta, 2013).

Let, x is an m-dimensional zero mean random vector. q is an m-dimensional unit vector, ‖ q ‖ = 1. Equation for the projection of x onto q is $A = x^Tq = q^Tx$ and from the variance equation, $\sigma^2 = E[A^2] = E[(q^Tx)(x^Tq)] = q^TE[xx^T]q = q^TRq$, where E[A] is the expectation of matrix A, R is the correlation matrix of dimension mxm and $R = E[xx^T]$. From the above equation the following equation can be derived:

Rq = λq … Equation No.1.

Given matrix R there are some combinations of λq with which the above equation can be solved. λ is the eigenvalue of correlation matrix R. As R is of dimension mxm and there are m such eigenvalues λ1, λ2, λ3, …, λm and associative eigenvectors of the correlation matrix R, q1, q2, q3, …, qm can be obtained. There will be m solutions for the equation no. 1 and thus it can be written as follows:

$Rq_j = \lambda_j q_j$, where j=1, 2, 3, …, m …Equation No. 2.

The eigenvalues are arranged in descending order, λ1>λ2> λ3>, …, > λm and the λ1 is the highest eigenvalue. Using the eigenvectors (q), matrix Q can be generated, Q = [q1, q2, q3, …, qm]. Now all the m equations in equation no.2 can be expressed in matrix form as follows:

RQ = Q Λ where Λ = diagonal elements of [λ1 λ2 λ3 … λm].

Q is an orthogonal matrix and satisfies the condition of orthonormal matrix. $q_i^Tq_j$ = { 1 when j=1 and 0 when j!=i} and $Q^TQ = I$, where I is identity matrix (Sengupta, 2013).

4.2.2 Independent Component Analysis (ICA)

ICA is an expansion of PCA concept. The principal components (independent variables) extracted from an input data (that is mixed random independent and dependent variables) by PCA is not completely independent and therefore the features extracted from the input data are not fully independent. In ICA a non-linear PCA approach is introduced (Leach, 2002). For high quality feature extraction the input data variables should be independent and the more they are independent from each other, the better is the quality of feature extraction. To identify the independent variables among a given set of input variables, the separation function should consider all higher order correlations. ICA takes into consideration of these higher order correlations and thus, it exhibits better quality feature extraction process (Leach, 2002).

4.2.2.1 Representation of ICA in mathematical terms

Let x1, x2, x3, …, xn are the random input variables. Matrix A is a mixing matrix consists of non-gaussian, independent components (Leach, 2002). Then, x=As …Equation-1. Estimation of s is done using input data x. There are two major conditions while processing feature extraction using ICA, they are as follows:

4.2.2.2 The independent variables must statistically independent.

4.2.2.3 The independent variables must have non-gaussian distribution.

The estimation of mixing matrix A is performed. Then the inverse of A is calculated. The inverted matrix is called the demixing matrix, and it is represented as:

$W = A^{-1}$…Equation-2.

This matrix W is used to find all the independent components, s = Wx…Equation-3 (Leach, 2002).

4.2.3 Linear Discriminant Analysis (LDA)

LDA or FLD is a class specific method (Belhumeur, Hespanha & Kriegman, 1997). The basic idea is to choose a transformation matrix W which maximizes the ratio of S_B and S_W. Where S_B is the between-class scatter and S_W is the within-class scatter matrix and these two matrices can be represented mathematically as mentioned below:-

$$S_B = \sum_{i=1}^{C} N_i(\mu_i - \mu)((\mu_i - \mu)^T$$

$$S_W = \sum_{i=1}^{C} \sum_{x_k \in X_i} (X_k - \mu_i)((X_k - \mu_i)^T$$

Where, μ_i = mean image of class X_i and N_i = number of samples in class X_i. If S_W is nonsingular matrix the optimal projection W_{opt} is a matrix has orthonormal columns. W_{opt} maximizes the ratio of $|W^TS_BW| / |W^TS_WW|$. Where, $|W^TS_BW|$ is the determinant of the between-class scatter matrix of the projected samples and $|W^TS_WW|$ is the determinant of the within-class scatter matrix of the projected samples (Belhumeur, Hespanha & Kriegman, 1997).

$$W_{opt} = \arg\max W \frac{|W^TS_BW|}{|W^TS_WW|} = [w1\ w2 \ldots w_m]$$

Where { w_i | I = 1, 2, …, m} = set of eigenvectors of S_B and S_W corresponding to eigenvalues {λ_i | I = 1, 2, …, m}, that is , $S_Bw_i = \lambda_iS_Ww_i$, i = 1, 2, …, m. C is the number of classes. LDA performs better than PCA as illustrated in (Belhumeur, Hespanha & Kriegman, 1997).

V. Analysis of Recognition Methods

There are various face recognition methods exists. Classical techniques for FR problem such as, Karhunen-Loeve transform based methods (PCA), LDA (Turk & Pentland1991; Marcialis & Roli, 2002; Martinez & Kak, 2001), singular value decomposition (Chellappa, Wilson & Sirohey, 1995). Recent approaches for FR are kernel methods (Wang, et al., 2008; Hotta, 2008; Wang et al., 2004; Yang, 2002; Yang et al., 2005) such as support vector machines (SVMs), kernel Fisher discriminant (KFD), and kernel principal component analysis (KPCA) and Artificial neural Networks (ANNs) (Chellappa, Wilson & Sirohey, 1995).

Face recognition is such a real world problem where input variables (face image data) are not linearly separable. Among these input variables some influence other variables and thus all are not independent. In other words, the change in the value of one variable may affect the values of other variables and thus some are dependent and some are independent (Dasgupta, 2013). Feature extraction process is used to extract the independent variables from the given input dataset. But recognition task requires a classifier function which is able to classify these input variables in different hyper-planes which are not linearly separable (Dasgupta, 2013). Multilayer feed-forward neural network with back-propagation technique is efficient in learning this kind of function (Dasgupta, 2013). Therefore it is often used to solve Face recognition task. Using the feed-forward technique the hidden layer forwards the output of the activation function. In the output layer the final output is generated and the error is calculated by comparing the ANN output with the target or actual output using root mean squared error (mse) technique (Yang Sai, Ren Jinxia & Li Zhongxia, 2009). Once the error is calculated a back-propagation technique is applied which readjust the weight values of neurons in the hidden layer. The main objective is to adjust the weights in such a way that with a new training sample and adjusted weights as input to the activation function, the output of the activation function should be closer to the target output with reduced error (Dasgupta, 2013). The weight learning function uses Gradient Descent or Steepest Descent based learning methods to learn the weights (Karayiannis, 1997). There are various kinds of activation functions exist and the selection of these functions depends upon the nature of the problem. Most commonly used activation functions are threshold and sigmoid functions (Dasgupta, 2013).

5.1 A mathematical representation of multi-layer feed-forward neural network with back-propagation

Let, a neuron-i in hidden-layer receives inputs from input-nodes Ij where j=1, 2, 3, …, n as shown in Figure 1. W_{ji} are the weights to the neuron-i for all inputs coming from I_j. Total input "in" to neuron-i can be defined as:

$$in = \sum_{j=1}^{n} W_{ji} I_j \ldots Eq - 1$$

The activation function of neuron-i can be defined as g(in) where "in" is the total input to neuron-i as shown in Equation-1 (Dasgupta, 2013). Activation function-g takes "in" as input and gives the output for neuron-i which is forwarded to the output-layer neurons. The output-layer gives the total output. The total output is defined as the collective outputs taken together from the neurons of the hidden-layer. The total output is measured with the target or actual expected output and the error is calculated. The error can be defined as E = (1/2)Err^2 and this is the root mean squared error (mse). Err = (y – h_w(x)), where y is the target or actual expected output, h_w(x) is the total output received from output-layer (Dasgupta, 2013).

$$Err = (y - h_w(x)) \equiv (y - g(in)) = (y - g(\sum_{j=1}^{n} W_j x_j)) \ldots Eq - 2$$

The change in the error E with respect to the change in weights Wj can be calculated by taking the partial differentiation as shown below (Dasgupta, 2013).

$$\frac{\partial E}{\partial W_j} = \frac{\partial}{\partial W_j}\left(\frac{1}{2} Err^2\right) = -Err\ x\ g'(in)\ x\ (x_j) \ldots Eq - 3$$

By putting the Err value from equation-2 in equation-3, the outcome of the change in error E with respect to the change in weights Wj is calculated. The term from equation-3 is added to the existing weights Wj to reduce the error in the output of the neural network for the next iteration with new training sample. The new weights Wj can be defined as:

$W_j = W_j + \alpha\ (-\ Err\ x\ g'(in)\ x\ (x_j)\ )$

The weight W_j is updated with some alpha (α) times Err times g'(in) times x_j. Where, α is the learning rate, Err is the error, g'(in) is the differential of the activation function and x_j are the inputs to hidden neurons (Dasgupta, 2013).

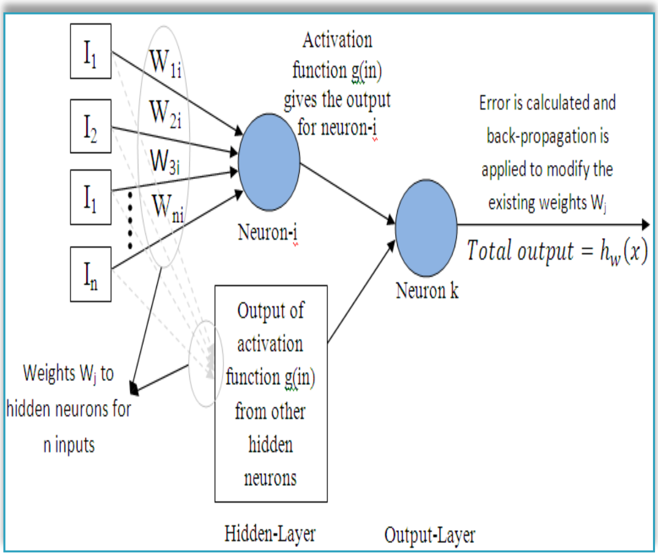

Figure 1. Diagramatic representation of multi-layer feed-forward neural network with back-propagation.

There are two models used alternatively for the face recognition task. They are as follows:-

5.2 Multi-Class Neural Network Model (MCNN).
5.3 Pair-Wise Neural Network Model (PWNN).

MCNN can learn from noisy and corrupted face image data (Uglov et al., 2008). It is also able to handle face images with variations such as different illumination conditions, poses etc. On the other hand, MCNN does not show high level performance with large face image dataset which consist of multiple face images per person (class). Multiple face images per person are required to make the FR system more robust. In a large image dataset the boundaries between classes are more dense and complex in nature as show in Figure 2 and due to this reason MCNN is not performing well in the recognition task (Uglov et al., 2008).

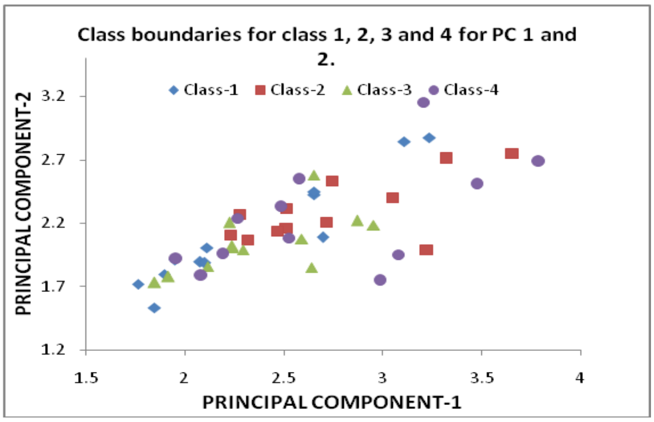

Figure 2. Class boundaries are shown for class 1, 2, 3 and against two highest principal components 1 and 2.

PWNN can handle this multi-class problem and perform efficiently with large image dataset (Uglov et al., 2008). PWNN transforms the multiclass problem into a set of binary classification problems. In a binary classification problem the class boundaries are simple in nature and the density of the training dataset is reduced as the number of classes is reduced from n-class problem to 2-class problem. Thus training is simpler in PWNN. Finally the output is generated by combining the outcomes of all the pair-wise classifiers using the “Winners-take-all” approach (Uglov et al., 2008).

Let there are N classes in a multi-class problem. All the possible combinations of N classes taken 2 at a time = Cx(C-1)/2 = Total number of binary classifiers = M. In a PWNN model M numbers of ANNs are trained to solve two-class problem (Uglov et al., 2008). From training, the ANNs learn M numbers of separating functions fi/j used to separate class i from j, where ij is all the possible combinations of N classes 2 taken at a time. It is assumed that function fi/j returns positive values for inputs related to class i and negative values for class j. New separating functions g1, g2, g3,…,gm are formed by combining the functions fi/j in such a way that: g1 = f1/2+f1/3+…+f1/N; g2 = f2/3+f2/4+…+f2/N - f1/2; g3 = f3/4+f3/5+…+f3/N - f1/3 - f2/3;…; g(N-1) = f(N-1)/N – f1/(N-1) – f2/(N-1) - … -f(N-2)/(N-1); g(N) = - f1/(N) – f2/(N) - … -f(N-1)/(N). Two-layer feed-forward ANN is used to form these separating functions g1, g2,…, gN. In this type of ANNs input neurons are fully connected to the hidden nodes. In the output layer of the ANN there are N numbers of output nodes. The final output is obtained by summing up all the outputs of M numbers ANNs (Uglov et al., 2008). As for example the PWNN model is illustrated in Figure 3 comprises of 4 input neurons. 6 ((4 x (4-1)) / 2) separating functions: f1/2, f1/3, f1/4, f2/3. F2/4, f3/4. The 4 output nodes (neurons) g1, g2, g3, g4 are connected to these networks. The separating functions in the output layer of the PWNN can be defined as: g1 = f1/2 + f1/3 + f1/4; g2 = f2/3 + f2/4 – f1/2; g3 = f3/4 – f1/3 – f2/3; g4 = -f1/4 – f2/4 – f3/4 and there corresponding weights are g1(+1 +1 +1), g2(-1 +1 +1), g3(-1 -1 +1) and g4(-1 -1 -1).

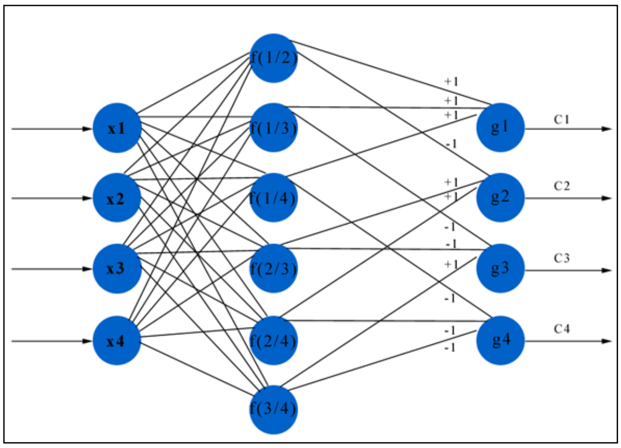

Figure 3. A pair-wise multi-layer neural network model for 4-class problem with four inputs x1, x2, x3 and x4. In the hidden layer the perceptrons are trained to learn the 6 (4x(4-1)/2) binary classifier functions f(i/j). The output layer has four perceptrons which classify the four classes using the four seperation functions g1, g2, g3 and g4.

VI. DESIGN A RECOGNITION METHOD

We have chosen the MCNN and PWNN models as explained in the previous section. There is an enormous large amount of mathematical computation involved in designing real time FR system. For this purpose MATLAB (Matrix Laboratory) software is selected as a suitable candidate solution. It is a high-level language. It provides an integrated development environment to solve mathematical problems faster than the traditional programming languages such as Java, C++ etc. MATLAB ANN Toolbox software provides the multi-layer feed-forward neural network which is used to model the MCNN and PWNN for our FR system. We have designed, trained and tested our ANN models using multi-layer feed-forward neural network. The neural networks for both MCNN and PWNN have two layers, one is hidden-layer and another is output-layer as shown in Figure 3. Every neuron in

hidden-layer is connected to every input node in the input layer. The main computations for the recognition task are carried out by the neurons present in the hidden-layer.
The following are the configuration settings for these two ANNs:

6.1 MCNN

The number of neurons in the hidden-layer is dependent on the problem. It ranges between 25 to 200. The activation function "purelin" is used. It is a linear activation function and performs well with linearly separable variables. To train the network training function “trainscg” is used. It uses the scaled conjugate gradient method to update the weight and bias values during the learning procedure.

6.2 PWNN

The number of hidden neurons is 1 to 2. The tangential sigmoid activation function (tansig) is used. It performs well with non-linearly separable variables. To train the network training function “trainlm” is used. It follows the Levenberg-Marquardt optimization technique to update weight and bias values. It is almost the fastest training function which uses the back-propagation algorithm to train the network.

6.3 Collect data

The data for our face recognition system is extracted from Yale extended B dataset (Georghiades, Belhumeur & Kriegman, 2001).The dataset consists of JPEG face images of 38 individuals (classes). Each person has 60 images. These 60 images are taken with different illumination conditions, so that the ANN can be trained more efficiently and the robustness of the FR system can be increased.

6.3.1 Division of sample data for training and testing

All 2280 (38 classes X 60 images per class) JPEG image files are stored in a folder named “yale_pic_sm” which is made as a subfolder of our FR system main folder fr_2011. Each image is a greyscale JPEG file with dimension 76x68 pixels. The FR system reshapes every image to 32x32 pixels and stores it as a vector of dimension <1024X1 double>. The 32X32 matrix is converted to a vector of 32X32=1024 pixel values. As there are total 2280 images present in the database, the main image dataset “data” is created with dimension <1024 X 2280 double>. 80% of the total sample records are used as training dataset that is (2280 X 80) / 100 = 1824. The neural network is trained with these 1824 sample records which consist of various face image data of 38 faces (classes) and the remaining 20% sample records that is 456 samples are used to test the neural network for testing the performance of the system. Among 1824 training records each class (or face image) has 1824/38 = 48 records. From test records each class has 456/38 = 12 records.

6.3.2 Data collection phase includes the following tasks

6.3.2.1 Reading the image file data and store them in MATLAB variable. MATLAB script “im_process.m” accomplishes this task.

6.3.2.2 Extract the features or principal components from this image data and normalize them. To perform this job “pw_xn.m” is designed.

Note: Within the algorithm section any word (string) within double quotes (“ ”) are assumed as a MATLAB script, binary file or variable.

6.4 Create, configure, initialize, train, test and validate the neural networks

To implement our FR system we have used MCNN and PWNN models. To implement MCNN the MATLAB script “mcnn.m” is developed and to create the PWNN, MATLAB scripts “pw_nn5.m”, “pw_nn.m” and “pw_test.m” are developed.

6.5 Use the network

After training, testing and validating, the network is ready to use for our recognition task. The following modules are developed for our recognition task:

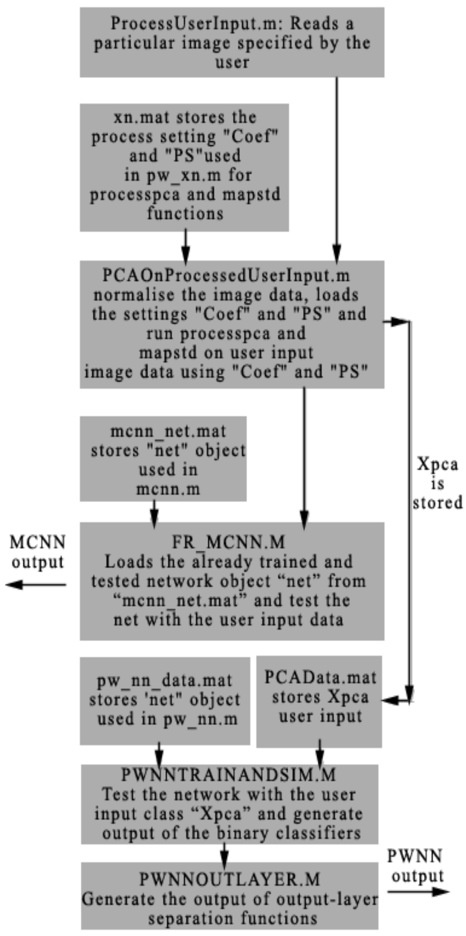

Figure 4. Pictorial representation of the face recognition uer interface developed in MATLAB where user can input a face image and view the recognition outcome.

VII. Experimental Results

7.1 *Performance Testing*

There are various sets of experiments conducted for the performance testing of MCNN and PWNN. The mean performance and the standard deviation in performance are measured against different settings such as number of principal components, hidden neurons and training samples per class. The mean performance shows the accuracy and standard deviation measures the stability of network predictions (or outcomes). The lower the standard deviation, the better is the stability of prediction of the recognition system.

7.1.1 Number of Principal Components Vs. Performance

Mean performance of MCNN and PWNN is measured by changing the number of principal components of a face image data. The experimental data and results are show in Table 1 and the performance and standard deviation graphs are shown in Figures 5 and 6 respectively.

Table 1. Maxfrac: Maximum fraction of variance for removed rows, PCs: no. of principal components, MP: mean performance, SD: standard deviation

Maxfrac	PCs	MCNN		PWNN	
		MP	*SD*	*MP*	*SD*
0.00001	272	0.942	0.048	0.962	0.018
0.000025	192	0.826	0.044	0.958	0.023
0.00005	142	0.858	0.032	0.96	0.022
0.00007	123	0.861	0.038	0.961	0.018
0.0001	101	0.849	0.034	0.963	0.013
0.0005	39	0.714	0.039	0.942	0.022
0.001	22	0.599	0.059	0.922	0.035

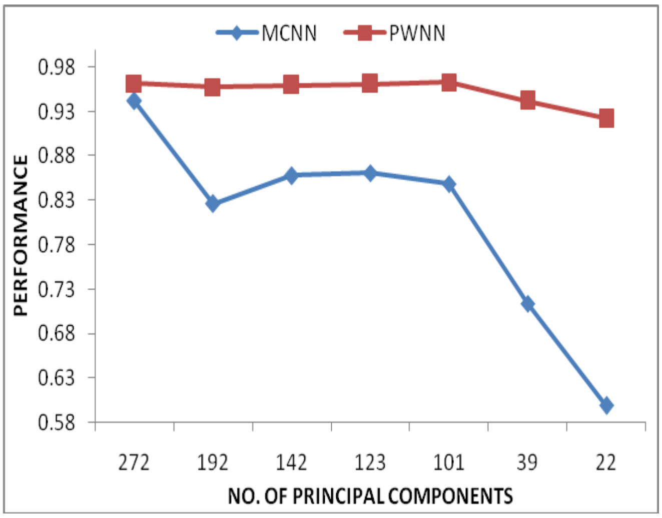

Figure 5. Mean performance of MCNN and PWNN outcomes are plotted against different numbers of principal components.

From Figure 5 it can be observed that PWNN shows better performance than MCNN with respect to predictive accuracy when the number of principal components are changed. On the other hand, MCNN's performance is decreased with the decrease in the number of PCs.

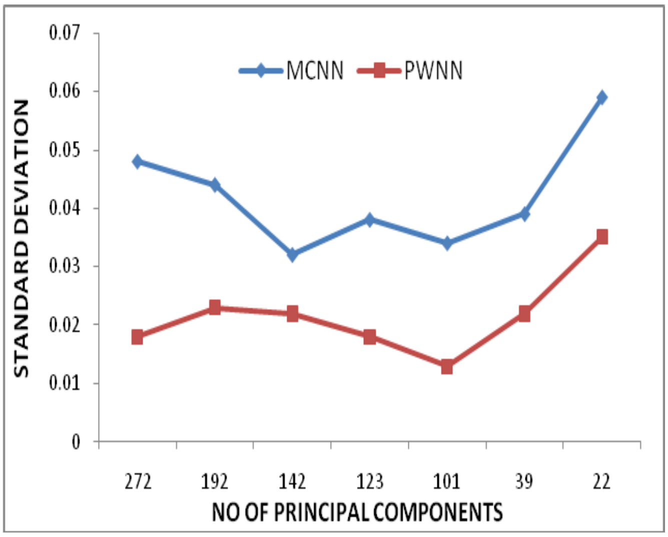

Figure 6. Standard deviation of MCNN and PWNN outcomes are plotted against different numbers of principal components

The standard deviation in performance is much lower and not much affected by the change in the no. of PCs. in case of PWN. MCNN has higher SDs than PWNN in all the cases as show in Figure 6.

7.1.2 Number of Hidden Neurons Vs. Performance

The performance is measured against different set of hidden neurons for both MCNN and PWNN. The experimental data and results are presented in Table 2 and the corresponding graphs are shown in Figure 7 and Figure 8. It can be observed from these results that with the change in the number of hidden neurons the performance of MCNN is slightly decreased. However, PWNN's performance is not affected by the change in the number of hidden neurons. One single neuron or 2 neurons are sufficient in case of PWNN to achieve a higher performance (Figure 8). MCNN shows best performance with 100 neurons as shown in the Figure 7.

7.1.3 Number of Training samples Vs. Performance

The number of training samples per class is varied and performance has been monitored for MCNN as well as PWNN. The testing input and output data are shown in Table 3 and graphs are plotted to analyse the outcomes of the test results as shown in Figures 9 and 10. It can be observed from Figure 9 that as the number of training samples per class increases the PWNN performs better than MCNN. Figure 10 shows that with the increase in the samples per class the standard deviation in performance for PWNN decreases rapidly as compared to MCNN.

Table 2. NOHN: no. of hidden neurons

MCNN			PWNN		
NOHN	***MP***	***SD***	***NOHN***	***MP***	***SD***
200	0.883	0.032	4	0.957	0.018
150	0.883	0.032	3	0.954	0.012
100	0.879	0.031	2	0.95	0.013
50	0.842	0.032	1	0.963	0.013

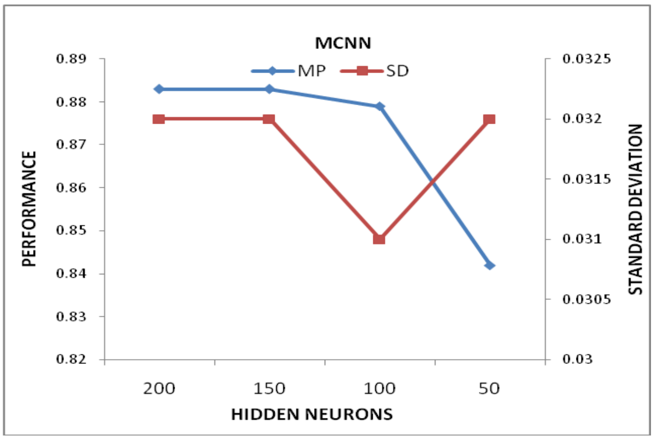

Figure 7. MCNN's performance is shown with different no of neurons in the hidden layer

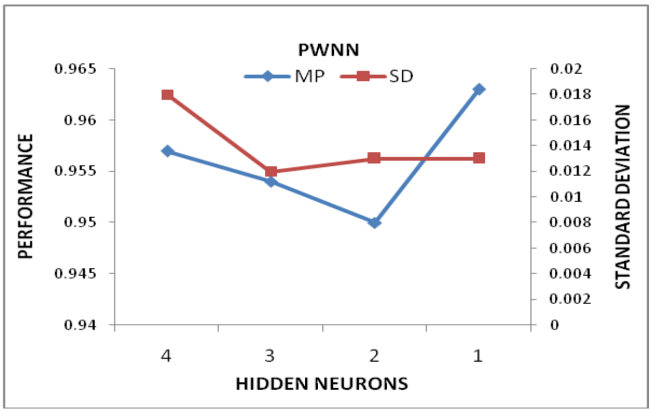

Figure 8. PWNN's performance is shown with different no of neurons in the hidden layer.

Table 3. SAMPLES: TRAINING SAMPLES PER CLASS

SAMPLES	PWNN		MCNN	
	MP	***SD***	***MP***	***SD***
15	0.889	0.132	0.87	0.164
20	0.917	0.073	0.857	0.127
30	0.924	0.06	0.828	0.099
40	0.952	0.035	0.88	0.04
50	0.951	0.032	0.889	0.047
60	0.963	0.013	0.849	0.034

Our face recognition system has three main requirements mentioned in section-I. Numerous testing have been conducted to check the robustness of the FR system against these requirements. The following are few test cases illustrated below.

7.2 Requirement-1

As per the first requirement, the FR system should learn from the face images of different individuals. We have tested our FR system with different person's images. The following are the two test cases shown in Table 4.

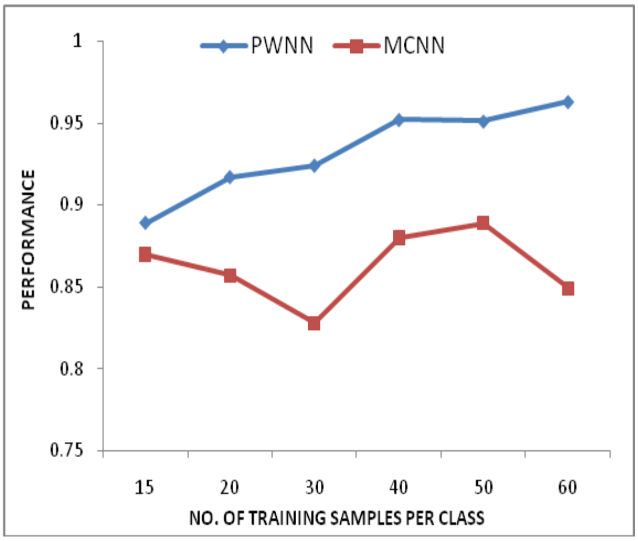

Figure 9. PWNN outperforms when the number of training samples per class gradually increases

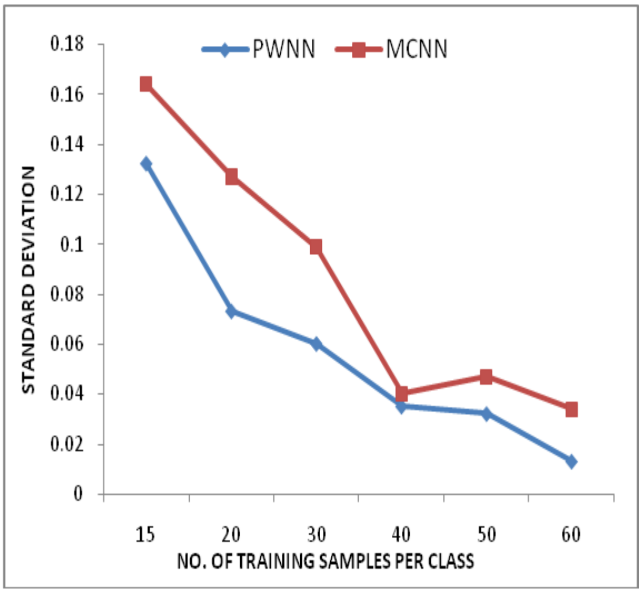

Figure 10. PWNN has much lower standard deviation in performance than MCNN and it decreases with the increase in samples per case

Table 4. Requirement-1

User Input	FR system output in MATLAB command window
Face Image of person 1	The person id is = 1, match rate = 0.495919
Face image of person 17	The person id is = 17, match rate = 0.358671

Results are extracted from MATLAB variable for the above two test cases and plotted as shown in Figure-11.

7.3 Requirement-2

To achieve the second requirement a new person's face images have been added to the image database. A colour passport size image is taken. It is converted to grey-scale and resized to 68x77 pixels using the image processing software Adobe Photoshop. The edited image is saved to 60 different JPEG image files of same size and these 60 images are saved in the "yale_pic_sm/" folder. The new person id is 39. The ANN is trained for the new updated database and then it is tested. The following is the input/output for the test case as shown in Table 5.

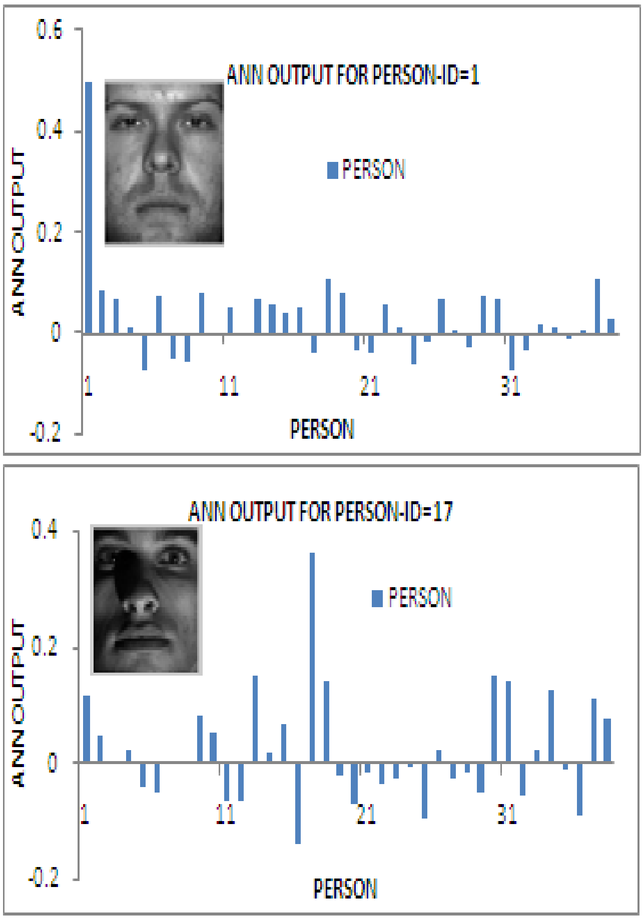

Figure 11. The recognition system identifies two persons havng id-1 and 17

Table 5. Requirement-2

User Input	FR system output in MATLAB command window
Face Image of person 39 (image no. 2281)	The person id is = 39, match rate = 0.952687

Results are extracted from MATLAB variable for the above test case and plotted as shown in Figure 12.

7.4 Requirement-3

To make the FR system more robust, the ANN is given a series of images (60 images per person) for each person, where images (of same person) varies with respect to different illumination conditions. Two different images (with varied illumination conditions) of a same person are given as input and results are tested for accuracy and correctness of the FR system. The following are the input/output details for the two test cases illustrated in Table 6.

Table 6. Requirement-3

User Input	FR system output in MATLAB command window
Face Image of person 38 (image no. 2265)	The person id is = 38, match rate = 0.781944
Face image of person 38 (image no. 2275)	The person id is = 38, match rate = 0.505899.

Results are extracted from MATLAB variable for the above two test cases and plotted as shown in Figure 13.

VIII. Contribution and Future Works

My contributions towards completion of this report are as follows. The description and structure of image data are presented. Analysis of image features, feature processing and extraction methods and recognition methods have been conducted. The recognition methods are designed and implemented. A detailed literature review has been undertaken for analysis of various feature extraction and recognition methods. Explanation and mathematical representation have been prepared for various techniques, such as PCA, ICA, LDA, multi-layer feed-forward neural networks, root mean squared error, gradient descent. A MATLAB application has been developed which takes user input and displays the output for the face recognition system. Various sets of experiments have been conducted to test the performance of PWNN and MCNN and results have been analyzed.

Fgure 12. The recognition system identifies a newly added person with person id-39

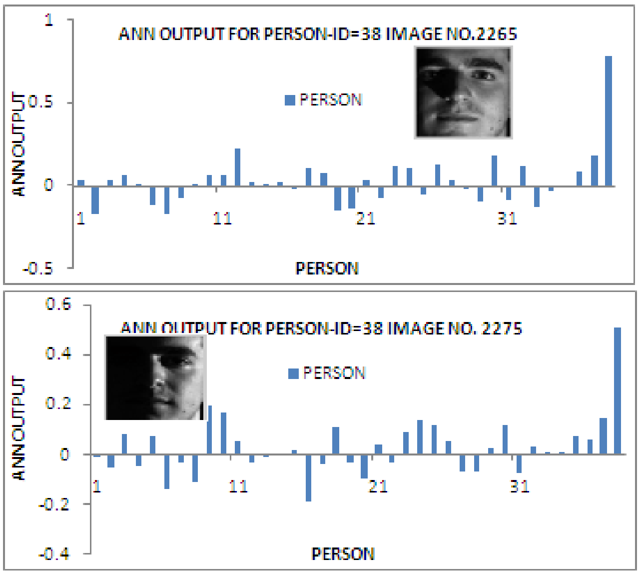

Figure 13. The recognition system identifies a person from two different images with varied illumination conditions.

IX. CONCLUSION

In this paper various phases of the assigned face recognition task have been discussed. The requirements of the recognition system have been clearly defined in Section-I. In Section-II the description and structure of image data is mentioned. An analysis of possible image features has been conducted in Section-III. In section-IV the methods for feature processing and extraction are briefly discussed. Principal components analysis (PCA), which has been applied to our face recognition system, has been explained. Besides, Independent Component Analysis (ICA) and Linear Discriminant Analysis (LDA) methods for feature extraction have been presented with mathematical representation. The methods used in face recognition problem are listed and a comparative analysis of two models (PWNN and MCNN) used in our face recognition system have been conducted. The complexity has been explained in recognition task by showing the complex class boundaries within classes. The advantage of PWNN model over MCNN has been discussed in section-V. In section-VI the actual implementation of face recognition system using MATLAB has been presented. The various phases of implementation, such as, face image data collection, feature processing and extraction, configuring, training and testing the neural networks have been presented with algorithms in section-VI. Finally in Section-VII the experimental results have been presented and observations have been discussed. It has been illustrated by various test results that PWNN is suitable as it satisfies all the requirements for our face recognition system. From the outcomes of several experimental tests it is concluded that PWNN performs better than MCNN when the number of classes (persons) are large, the number of samples per class is more and the variations in illumination conditions are higher. Furthermore, this 2D face recognition system can be extended to a 3D recognition model using ICA and ANN (Sahambi&Khorasani, 2003).

REFERENCES

[1] Abbas, A.; Khalil, M.I.; Abdel-Hay, S.; Fahmy, H.M. "Expression and illumination invariant preprocessing technique for face recognition. *International Conference on, Computer Engineering & Systems, ICCES*. **2008**, 59-64.

[2] Belhumeur, P.N;, Hespanha, J.P; Kriegman, D.J. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection', Pattern Analysis and Machine Intelligence. *IEEE Transactions* **1997,** *19*, 711-720.

[3] BISHOP, C.M. Neural networks for pattern recognition. CLARENDON PRESS, OXFORD **1995**.

[4] Chellappa, R.; Wilson, C.L; & Sirohey, S. Human and machine recognition of faces: A survey. *Proceedings of the IEEE*. **1995**, 83, 705-741.

[5] Chengjun Liu & Wechsler, H. Robust coding schemes for indexing and retrieval from large face databases, Image Processing. *IEEE Transactions* **2000**, *9*, 132-137.

[6] Coppin, B. Artifical intelligent illuminated. 1st ed. edn. sudbury Massachusetts: Jones and Bartlett Publishers **2004**.

[7] Dasgupta, P. (2013) Artificial intelligence. Available at: http://nptel.iitm.ac.in/video.php?subjectId=106105079 (Accessed: January 13, 2013).

[8] Georghiades, A.S.; Belhumeur, P.N.; Kriegman, D.J. From few to many: Illumination cone models for face recognition under variable lighting and pose', Pattern Analysis and Machine Intelligence. *IEEE Transactions*. **2001**, *23*, 643-660.

[9] Hastie, T.; Tibshirani, R. Classification by pairwise coupling **1998**, *26*, 451-471.

[10] Hotta, K. Robust Face Recognition under Partial Occlusion Based on Support Vector Machine with Local Gaussian Summation Kernel. *Image and Vision Computing*. **2008**, *26*, 1490-1498.

[11] Inc., Mathworks (2013) Documentation center. Available at: http://www.mathworks.co.uk/help/nnet/ref/mapstd.html (Accessed: January 7, 2013).

[12] K, R.; K.R.; V.; Patnaik, L.M; Raja, K.B. Feature extraction based face recognition, gender and age classification. *International Journal on Computer Science and Engineering*. **2010**, *2*, 14-23.

[13] Karayiannis, N.B. Gradient descent learning of radial basis neural networks. *International Conference on Neural Networks*. **1997**, *3*, 1815-1820.

[14] Leach, K.N. A survey paper on independent component analysis. *Proceedings of the Thirty-Fourth Southeastern Symposium on, System Theory*. **2002**, 239-242.

[15] Marcialis, G.L.; Roli, F. Fusion of LDA and PCA for face recognition. *Proc. of the Workshop on Machine Vision and Perception, 8th Meeting of the Italian Association of Artificial Intelligence (AI*IA), Siena, Italy*, September **2002**.

[16] Martinez, A. M.; Kak, A. C. PCA versus LDA. *IEEE Trans. Pattern Analysis and Machine Intelligence*. **2001**, *23*, 228-233.

[17] Oravec, M.; Mazanec, J.; Pavlovičová, J;, Eiben, P.; Lehocki, F. (01, April, 2010) Face recognition in ideal and noisy conditions using support vector machines, PCA and LDA. Available at: http://cdn.intechopen.com/pdfs/10206/InTech Face_recognition_in_ideal_and_noisy_conditions_using_support_vector_machines_pca_and_lda.pdf (Accessed: January 6, 2013).

[18] Ranawade, S.S. Face recognition and verification using artificial neural network. *International Journal of Computer Applications* **2010**, *1*, 23-30.

[19] Sahambi, H.S.; Khorasani, K. A neural-network appearance-based 3-D object recognition using independent component analysis. Neural Networks, *IEEE Transactions*. **2003**, *14*, 138-149.

[20] Sengupta, S. (2013) Introduction to principal components and analysis. Available at: http://www.nptel.iitm.ac.in/video.php?subjectId=117105084 (Accessed: January 6, 2013).

[21] Uglov, J.; Jakaite, L.; Schetinin, V.; Maple, C. Comparing robustness of pairwise and multiclass neural-network systems for face recognition, EURASIP. *J.Adv.Signal Process*. **2008,** 641-647.

[22] Wang, J.; Lin, Y.; Yang, W.; Yang, J. Kernel Maximum Scatter Difference Based Feature Extraction and its Application to Face Recognition. *Pattern Recognition Letters*. **2008**, *29*, 1832-1835.

[23] Wang, Y.; Jiar, Y.; Hu, C.; Turk, M. Face Recognition Based on Kernel Radial Basis Function Networks, *Asian Conference on Computer Vision*, Asian Federation of Computer Vision Soc, Korea, January **2004**, pp. 174-179.

[24] Yang, M. Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods. *IEEE International Conference on Automatic Face and Gesture Recognition*. **2002**, pp.215-220, Mountain View, California.

[25] Yang, M.; Frangi, A. F.; Yang, J. Y.; Zhang, D.; Jin, Z. KPCA Plus LDA: A Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition. ,*IEEE Transactions on Pattern Analysis and Machine Intelligence*. **2005**, *27*, 230-244.

[26] Yang Sai, Ren Jinxia, Li Zhongxia, Learning of neural networks based on weighted mean squares error function. Second International Symposium on, Computational Intelligence and Design, 2009. ISCID '09, p241-244.

[27] Yesu, K.; Chakravorty, H.J.; Bhuyan, P.; Hussain, R.; Bhattacharyya, K.. Hybrid features based face recognition method using artificial neural network. Second National Conference on, Computational Intelligence and Signal Processing (CISP), 2012, p40-46.