

# Face Recognition by using Feed Forward Back Propagation Neural Network

Tej Pal Singh , Assistant Professor , TIT Bhopal MP(INDIA)

## Abstract

Face recognition technology has seen dramatic improvements in performance over the past decade, and such systems are now widely used for security and commercial applications. In this paper an improved approach has been used in which feed forward back propagation neural network is implemented through matlab. The results obtained shows that the proposed approach somewhat improves the performance compared to the previous approaches.

**Keywords:** Pattern recognition, feed forward network , matlab.

## Introduction

Recognition of person by means of biometric characteristics is an emerging phenomenon in modern society. The face is our primary focus of attention in social life playing an important role in conveying identity and emotions. We can recognize a number of faces learned throughout our lifespan and identify faces at a glance even after years of separation. This skill is quite robust despite of large variations in visual stimulus due to changing condition, aging and distractions such as beard, glasses or changes in hairstyle. The research on fully automated face recognition has been started from 1960s, in which feature such as eyes, ears, nose, and mouth from face image was located. Various approaches to face detection are discussed in this section. For recent surveys on face detection, different approaches has been used such as principal component analysis (PCA), neural networks, machine learning, information theory, geometrical modeling, template matching, Hough transform, extraction of geometrical facial features, motion extraction, and color analysis.

The neural network-based [1][2] and the view based [3] approaches require a large number of face and non-face training examples, and are designed primarily to locate frontal faces in grayscale images. It is difficult to enumerate “non-face” examples for inclusion in the training databases.

Schneiderman and Kanade [4] extend their learning-based approach for the detection of frontal faces to profile views. A feature-based approach combining geometrical facial features with belief networks [5] provides face detection for non-frontal views. Geometrical facial templates and the

Hough transform were incorporated to detect grayscale frontal faces in real time applications [6].

Face detectors based on Markov random fields [7], [8] and Markov chains [9] make use of the spatial arrangement of pixel gray values. Model based approaches are widely used in tracking faces and often assume that the initial location of a face is known. For example, assuming that several facial features are located in the first frame of a video sequence, a 3D deformable face model was used to track human faces [10]. Motion and color are very useful cues for reducing search space in face detection algorithms. Motion information is usually combined with other information (e.g., face models and skin color) for face detection and tracking [11]. A method of combining a Hidden Markov Model (HMM) and motion for tracking was presented [12]. A combination of motion , color filters, and a neural network model was also proposed[13].

W. Zhao [14] provides an up-to-date survey on face recognition and discusses two types of face recognition task which is based on still images and video.

Turk [15] used standard linear algebra technique to calculate the eigenfaces. This method enabled real-time automated face recognition system.

L.Liao [16] Used fast fourier transform algorithms for the input face image before applying recognition. The FFT algorithm transform input face images into “Whitenedfaces” then perform face recognition using Principal Component Analysis (PCA) and Independent Component Analysis (ICA) coding strategies, respectively. Experimental result shows that the whitening processing will bring better recognition performances for both PCA and ICA based face recognition system.

G. Jarillo [17] used different image transformations (such as contrast enhancement and edge detection). The experimental results shown that such transformations can provide unique discriminatory information useful for face classification, namely PCA, LDA (Linear Discriminate Analysis. For experiment YALE and FERET database are used.

G.K.Lin [18] proposed Hausdroff distance in recognition step instead of Euclidian distance. Experiments showed satisfactory results in terms of recognition accuracies and processing speed.

A. Eleyan [19] used wavelet transform along with PCA. Two methods have been proposed first is Multi resolution Feature Concatenation (MFC), where PCA is used for dimensional reduction on each subband and resulting each subband together for performing classification, second is Multi resolution Majority Voting (MMV) where PCA and classification are done separately on each subband. Experimental result showed that both techniques promising the results and NMV approach outperforms the MFC approach.

Shermina.J [20] used DCT for minimizing the illumination problem in face recognition. DCT applied image later is used in recognition step and for this step PCA approach is used necessary, equation numbers should not have parts to them. E.g., instead of using Equation 1(a) and Equation 1(b), please number them as Equation (1) and Equation (2).

## PATTERN RECOGNITION

The research area of pattern recognition is to study the operation and design of systems that recognize patterns in data. Pattern recognition is used for classification or description of observations. It aims to classify patterns based on known input or on statistical information that is extracted from the input patterns. The classified patterns are groups of observations or measurements. In this work Pattern matching is used to compare two patterns in order to determine whether they match or not (i.e., are they same or they differ). Pattern matching is the act of checking some sequence of tokens for the presence of the constituents of some [pattern](#). In contrast to [pattern recognition](#), the match usually has to be exact. The patterns generally have the form of either [sequences](#) or [tree structures](#). Uses of pattern matching include outputting the locations of a pattern within a token sequence, to output some component of the matched pattern, and to substitute the matching pattern with some other token sequence. Pattern recognition is the assignment of a label to a given input value. Back Propagation Neural Network tool is used in this work to recognize the face as a pattern.

### 2.1 Back Propagation:

Back propagation is the generalization of the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities. Properly trained back propagation networks tend to give reasonable answers when presented with inputs that they have never seen. Typically, a

new input leads to an output similar to the correct output for input vectors used in training that are similar to the new input being presented. This generalization property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs.

In this work we are using feed forward neural network method to recognize each person through its facial image.

## METHODOLOGY

The complete work is done through different stages:

- The first stage is Image pre-processing in which facial image of a person is captured. The images are read into the matlab environment using `imread()` function, whose syntax is `imread('filename')`. Here file name is a string containing the complete name of the image file and is used to bring the image into the matlab environment. For example, the statement

```
f=imread('sample.jpg')
```

reads the image from the JPEG file sample into image array f. After reading the image, it is displayed on the matlab workspace using the function `imshow()`, by using syntax `imshow(f)`. Where, f is an image array. Logical function converts numeric values to logical. Logical (f) converts the elements of the array f into logical, thus returning an array that can be used for logical indexing or logical tests. Logical can have the values 0 and 1 corresponding to false and true, respectively. Any nonzero real element of input array f is converted to a logical 1 while zeros in f become logical 0.

- Converting an image into digital form: `Im2bw` convert image to binary image. It produces binary images from indexed, intensity or RGB images. To do this it converts the input image to grayscale format and then converts this image to binary. The output binary image BW has values of 1 (white) for all pixels in the input image with luminance greater than level and 0 (black) for all other pixels. `hist` function finds the histogram values of the image. For eg:

```
Xh= hist(x (:), n);
```

computes n-bin histogram. Where bin is the number of bins used to generate the histogram.

- After finding the histogram equivalent of each part of the facial image of the given binary numbers, these histogram values are taken as input and some random values are assigned as target values to train, test and validate each part individually using feed forward back propagation neural network tool. After training of each part we have taken all the four trained output as a single input and trained it in a unique value. This whole process described above is also implemented with the full facial image of a person without any segmentation.

The above discussion can be processed in the following ways:

1. Capturing of the image (Fig 1)
2. Division of every image into two equal parts (Fig 2)
3. Binary conversion of images
4. Finding out the histogram equivalent of given binary numbers
5. Training of each part individually using neural network tool through Matlab
6. Collect output of all the parts and apply it as input
7. All the above steps are used without dividing the image into different parts also
8. Results obtained from steps 6 and 7 are compared to check the accuracy of this approach.

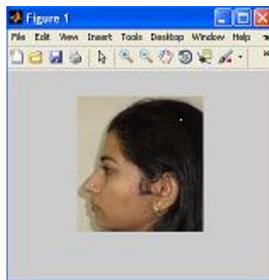


Fig 1: Sample Image

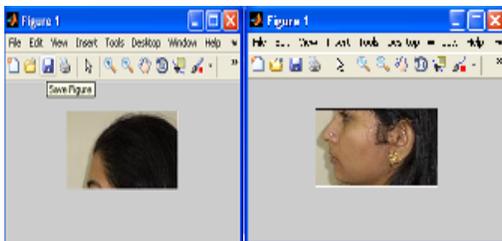


Fig 2: Segmented Image

A network is designed which contains neurons in the input layer in the hidden layer as well as in the output layer. The network is trained by applying different number of samples. Once the training phase is completed different samples are applied to the network to check its performance. Finally matching is used to check whether the network is able to recognize any existing image.

## DATA PREPARATION

In our work we have taken 3 sample facial images of each person. That is, the front image, left and the right image of each person and every sample is divided into 2 parts, each part is trained individually in Matlab's neural network tool. For simplicity the complete process of one person is shown in this paper. Each image is divided into two equal parts and each part is converted into the appropriate form by using different functions. To minimize the size of input data binary values are converted into the 4-bin histogram values. After finding the histogram equivalent of the value, training of each part is done in matlab's neural network tool. Table1 shows that the histogram values which we have found is taken as input data and some random values using 4 binary digits are taken as target values to train each part of every sample individually (All three samples of first person).

TABLE 1: Sample data of one person

S.NO.	PERSON	SAMPLE	PART	INPUT DATA			TARGET DATA	
1	1	1	1	10774	0	0	2762	1000
2	1	1	2	10632	0	0	2598	0011
3	1	2	1	19278	0	0	8190	1000
4	1	2	2	11339	0	0	9247	1001
5	1	3	1	21199	0	0	5100	1011
6	1	3	2	15144	0	0	5502	0110

In table 2 target data of each part of facial image is collected and is taken as a new input data for each sample of the same person. A new target data is set to map all the samples of the same person.

TABLE 2: Different sample Inputs and concerning Output for one person

S.NO.	PERSON	SAMPLE	INPUT DATA	TARGET DATA
1	1	1	10000011	01010101
2	1	2	10001001	
3	1	3	10110110	

Table3 shows 3 sample of the same person without dividing them into different parts.

TABLE 3: Different fingerprint samples of the same person without composition

S.NO.	PERSON	SAMPLE	INPUT DATA			TARGET DATA
1	1	1	21642	0	0	5574
2	1	2	31511	0	0	17995
3	1	2	37889	0	0	10936

## EXPERIMENTS AND RESULTS

All the experiments, results and discussion are shown in this section. A two layer feed-forward back propagation neural network has been used for complete experiment. Following diagram of the network is used for the complete experiment.

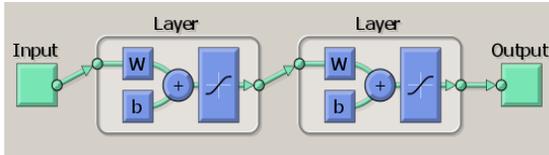


Fig 3: Sigmoid Model

Neural Network tool of Matlab 10 have been used for the complete work. Three different kinds of samples are applied on the network to perform different activities:

**Training:** These are presented to the network during training and the network is adjusted according to its error.

**Validation:** It is used to measure network generalization, and to halt training when generalization stops improving.

**Testing:** It is used only for testing the final solution in order to confirm the actual predictive power of the network. In our work we have conducted several training sessions. The training measures the performance on the basis of Mean Squared Error. It is the average squared difference between output

and target. Lower values of mean square errors are considered as better one while zero denotes no error.

### EXPERIMENT 1: Part 1 of First Sample of Person One

In this experiment we have taken 1000 epochs and the training was completed in 12 iterations. The network is trained through Feed-forward back propagation algorithm. Figure displays the training progress of the network.

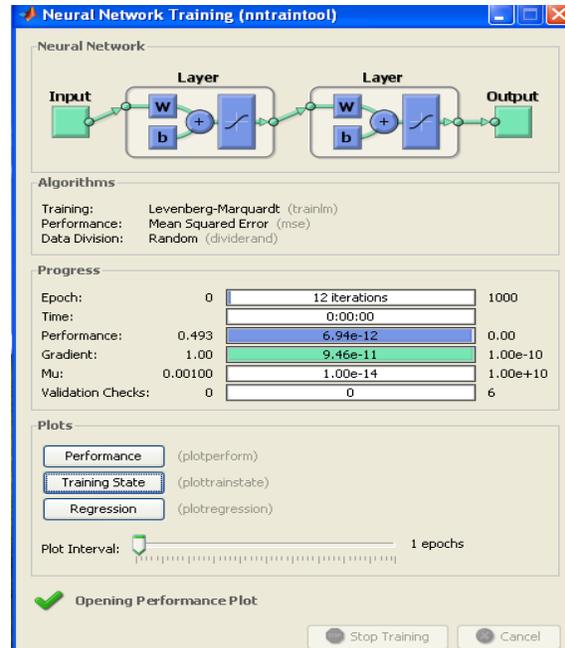


Fig 4: Network training result

From the above figure it seems clear that the total number of iterations to train the network is 12. The performance in terms of mean squared errors and the value of gradient and validation checks are also shown in the above figure.

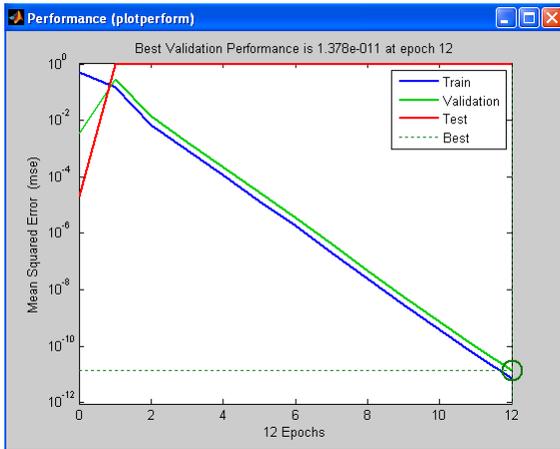


Fig 5: Performance curve produced by the network for the given sample

The above figure shows the performance curve produced while training, testing and validation of the network. We get the Best Validation Performance 1.378e-011 at 12 epochs.

**EXPERIMENT 2: Part 2 of First Sample of Person One**

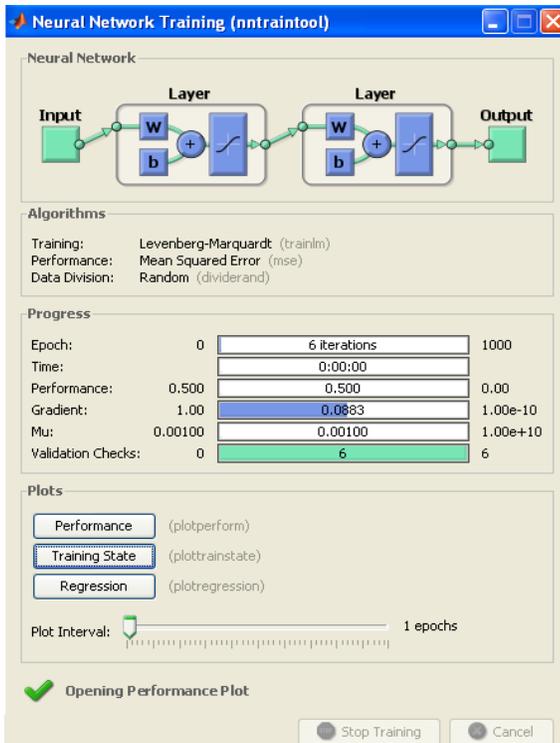


Fig 6: Network training result

From the above figure the total number of iterations to train the network is 6. The performance in terms of mean squared

errors, the value of gradient and validate checks is also shown in the figure.

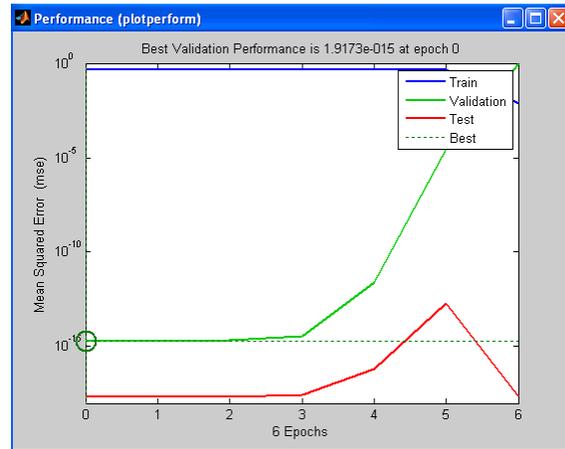


Fig 7: Performance curve produced by the network for the given sample

The above figure shows the performance curve produced while training, testing and validation of the network. We get the Best Validation Performance 1.9173e-015 at 0 epochs.

**EXPERIMENT 3: Collecting Output of Each Part and Setting it as Input to Recognize Each Person**

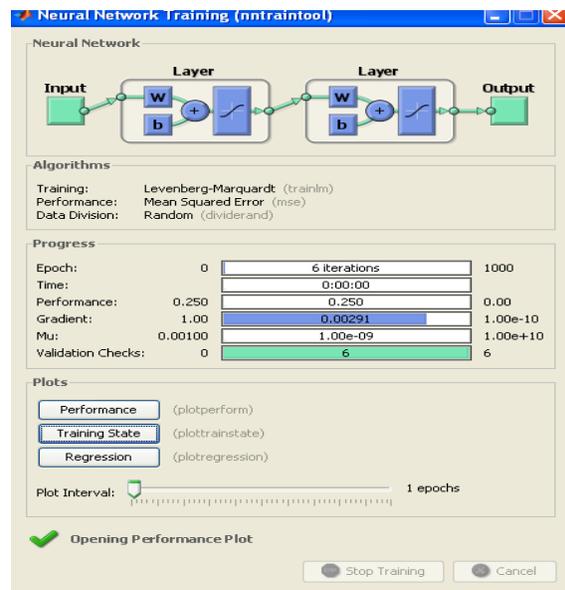


Fig 8: Network training result

From the above figure it seems clear that the total number of iterations to train the network is 6. The performance in terms of mean squared errors and the value of gradient and validate checks is also shown in the above figure.

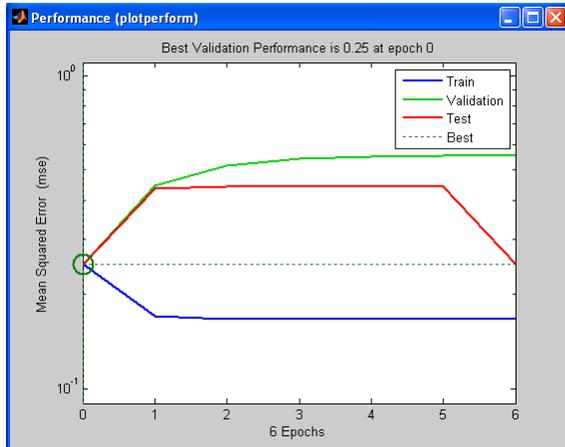


Fig 9: Performance curve produced by the network for the given sample

The above figure shows the performance curve produced while training, testing and validation of the network. We get the Best Validation Performance 0.25 at 0 epochs

**EXPERIMENT 4: First Person Fingerprint Sample without Dividing Into Parts**

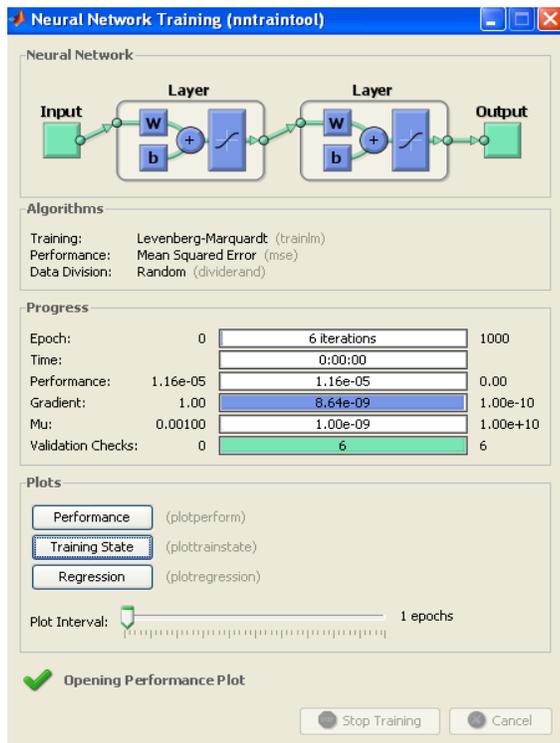


Fig 10: Network training result

From the above figure it seems clear that the total number of iterations to train the network is 6. The performance in terms

of mean squared errors and the value of gradient and validation checks is also shown in the above figure.

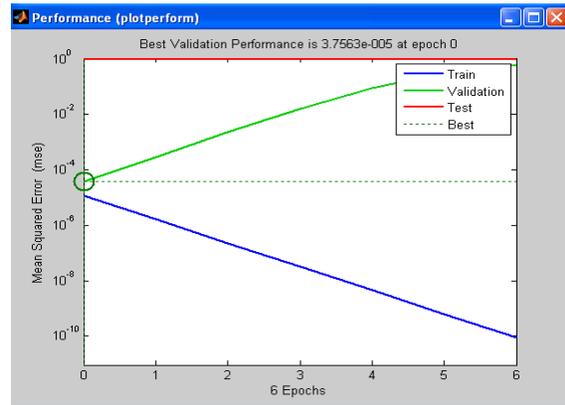


Fig 11: Performance curve produced by the network for the given sample

The above figure shows the performance curve produced while training, testing and validation of the network. We get the Best Validation Performance 3.7563e-005 at 0 epochs

**6. CONCLUSION**

In this work it has been shown that if a facial image of a person is given then the network can able to recognize the face of the person. The whole work is completed through the following steps:

1. Facial images of different persons have been collected by taking three different samples of each person for the experiment
2. Each image is divided into two equal parts.
3. Feed Forward Back Propagation neural network have been used to train, test and validate the network for each part of the image using MATLAB.
4. Output of all parts are collected and set as the new input and a new target is set to train, test and validate the entire parts of each sample.
5. Training of each sample is performed upto 8 to 10 times to minimize the error.
6. Facial image without dividing into parts have also been applied in the network.

Different unknown samples have also been applied in this network to obtain the result. And the accuracy level of this

network is approx 85%. Image division in this work has been done by using a matlab tool. Future work could be to develop a tool to automate some of the processes which have been done manually here.

## References

- [1] H.A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, Jan. 1998.
- [2] H.A. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 38–44, 1998.
- [3] K.K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, Jan. 1998.
- [4] H. Schneiderman and T. Kanade, "A statistical method for 3D object detection applied to faces and cars," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 746–751, June 2000.
- [5] K.C. Yow and R. Cipolla, "Feature-based human face detection," *Image and Vision Computing*, vol. 25, no. 9, pp. 713–735, Sept. 1997.
- [6] D. Maio and D. Maltoni, "Real-time face location on gray-scale static images," *Pattern Recognition*, vol. 33, no. 9, pp. 1525–1539, Sept. 2000.
- [7] M.S. Lew and N. Huijismans, "Information theory and face detection," *Proc. IEEE Int'l Conf. Pattern Recognition*, pp. 601–605, Aug. 1996.
- [8] S.C. Dass and A.K. Jain, "Markov face models," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 680–687, July 2001.
- [9] A.J. Colmenarez and T.S. Huang, "Face detection with information based maximum discrimination," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 782–787, June 1997.
- [10] D. DeCarlo and D. Metaxas, "Optical flow constraints on deformable models with applications to face tracking," *International Journal Computer Vision*, vol. 38, no. 2, pp. 99–127, July 2000.
- [11] V. Bakic and G. Stockman, "Menu selection by facial aspect," *Proc. Vision Interface, Canada*, pp. 203–209, May 1999.
- [12] A. Colmenarez, B. Frey, and T. Huang, "Detection and tracking of faces and facial features," *Proc. IEEE Int'l Conf. Image Processing*, pp. 657–661, Oct. 1999.
- [13] R. Feraud, O.J. Bernier, J.-E. Viallet, and M. Collobert, "A fast and accurate face detection based on neural network," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, pp. 1008–1011, Dec. 2001.
- [14] W. Zhao, R. Chellappa, P.J. Phillips and A. Rosenfeld, "Face Recognition: A literature Survey". *ACM Comput. Surv.*, 35(4): 399-458, 2003.
- [15] M.A. Turk and A.P. Pentland, "Face Recognition Using Eigenfaces", *IEEE conf. on Computer Vision and Pattern Recognition*, pp. 586-591, 1991.
- [16] Ling-Zhi Liao, Si-Wei Luo, and Mei Tian "Whitenedfaces" Recognition With PCA and ICA" *IEEE Signal Processing Letters*, Vol. 14, No. 12, pp. 1008-1011, Dec. 2007.
- [17] G. Jarillo, W. Pedrycz, M. Reformat "Aggregation of classifiers based on image transformations in biometric face recognition" *Machine Vision and Applications* (2008) Vol. 19, pp. 125-140, Springer-Verlag 2007.
- [18] Kwan-Ho Lin, Kin-Man Lam, and Wan-Chi Siu "A New Approach using Modified Hausdorff Distances with EigenFace for Human Face Recognition" *Seventh International Conference on Control, Automation, Robotics and Vision (ICARCV'02)*, Singapore, Dec 2002.
- [19] A. Eleyan and H. Demirel "Face Recognition using Multiresolution PCA" *IEEE International Symposium on Signal Processing and Information Technology-2007*.
- [20] Shermina.J "Illumination Invariant Face Recognition Using Discrete Cosine Transform And Principal Component Analysis" *International conference on Emerging Trends in Electrical and Computer Technology(ICETEECT)-2011*.
- [21] [http://en.wikipedia.org/wiki/Pattern\\_matching](http://en.wikipedia.org/wiki/Pattern_matching)
- [22] <http://homepage.inf.ed.ac.uk/rbf/IAPR/researchers/PPRPAGES/pprptut.htm>