FRAS: Face Recognition and Authentication System

Mayada.M.Salah H.E.Ibrahim Ramdan.Maowad

Maida_salah@yahoo.com hibrahim_eg@yahoo.com rammoawad@yahoo.com

ABSTRACT

In this paper, we present an automatic Face Recognition and Authentication (FRAS) System. The proposed System consists of three main phases, namely, Pre-processing, Feature Extraction, and Classification and Authentication phases. We use ORL faces database in the experiments. The most significant contribution of this work is using three face recognition methods; the Eigenface , the Fisherface and color histogram. The Eigenface is the first method considered as a successful technique of face recognition. The Eigenface method uses (PCA) to linearly project the image space to a low dimensional feature space. The Fisherface method is an enhancement of the Eigenface method that it uses (LDA) for the dimensionality reduction. The LDA maximizes the ratio of between-class scatter to that of within-class scatter; therefore, it works better than PCA for the purpose of discrimination. The color histogram based methods have proved simplicity and usefulness. Its idea was based on Color Histogram Quantization with 256 gray levels and using 24 quantization levels. We also use two classification methods (KNN) and (SVM). The proposed system has attained accuracy of 100% using color histogram features with KNN classifier and 95% using color histogram feature with SVM for ORL faces database of 40 persons with 10 image faces for each person.

1) <u>Introduction</u>

Face recognition [1] is one of the most popular problems in the field of image analysis and understanding. Face recognition is the identification of a person from a set, by comparing an image of that person's face with an image of every face in the set. It is a process effortlessly and accurately computed by human brains every day, but is extraordinarily hard to implement in a theoretical framework. Identifying a person from an unknown face is usually done by comparing the unknown face with the known faces from a face database. The interest of researchers and engineers in the face recognition problem has grown rapidly in recent years since there is a wide range of commercial and law enforcement applications on face recognition. The increasing need for surveillance-related applications, especially due to drug traffic and terrorist activities has a great impact on the growth of the interest in the field of face recognition.

Computational models of face recognition must address several difficult problems. These difficulties arise from the fact that faces must be represented in a way that best utilizes the available face information to distinguish a particular face from all other faces in the database. Face pose is a particularly difficult problem in this respect because all the faces look like each other, since they contain two eyes, a mouth, a nose, and other features located at about the same place.

The main computational stages of a face recognition system, the system starts with the first phase, captured images are sent through the Pre-processing phase to meet the standards required by a given recognition system. The Preprocessing [2] phase may perform tasks such as color-to-grayscale conversion, image resizing, and illumination and background removal in order to normalize the input image. Some of the images in the face database are used as the Training Set of the system and the rest will be the Test Set. The Feature Extraction [3][4] phase takes as input a normalized image and outputs a mathematical model of that input image that expresses the most important features in that image, thereby reducing its dimensionality. For example, techniques such as Principal Components Analysis (PCA) and the Linear Discriminate Analysis (LDA) can be used as feature extractors. Finally, the Feature matching [4] or a Classifier phase compares the feature vectors between a test image and all the training images and decides which training image is closest to that test image, For example, classifier such as K-nearest neighbor (K-NN) and support vector machine (SVM) are used on the proposed system.

This paper is structured as follows. In section 2 we briefly describe the stages of face recognition and authentication system. In section 3 we describe PCA, LDA and Color histogram quantization as feature extraction techniques. In section 4 we describe a K-NN and SVM that work as classifiers. In section 5 we present experimental results and in section 6 conclusions and future work.

2) Face Recognition and Authentication System

The proposed Face Recognition and Authentication System are composed of three main phases; preprocessing, feature extraction, and classification and authentication phases. Figure 1 describes the structure of the Face Recognition and Authentication System.

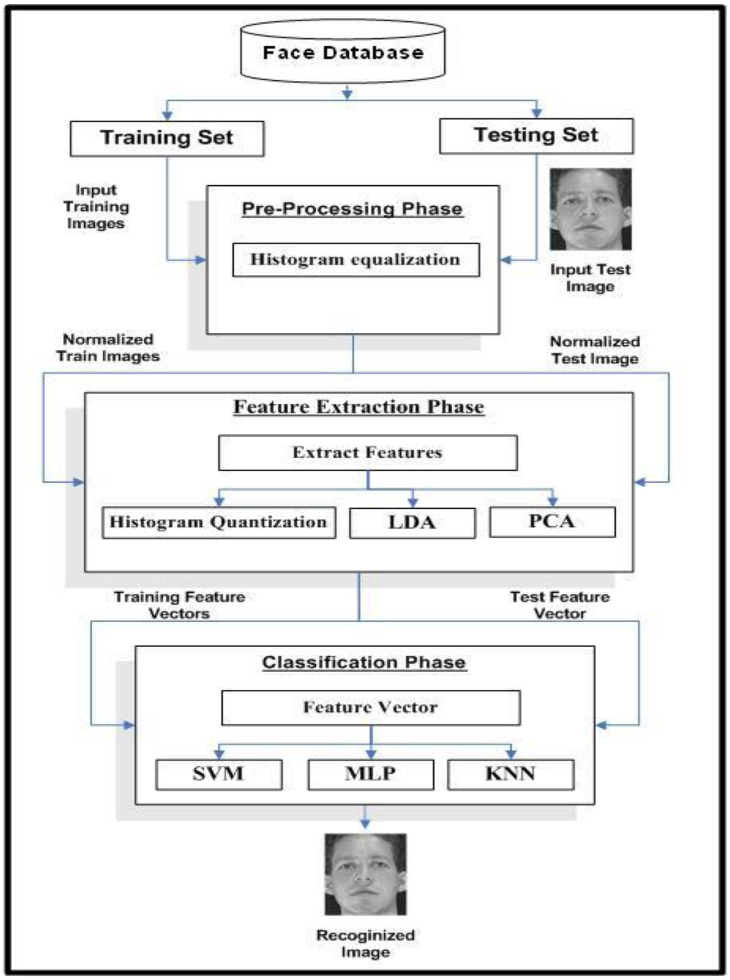

Figure 1: The Face Recognition and Authentication System General structure

2.1) Preprocessing Phase:-

By means of early vision techniques, face images are normalized and enhanced to improve the recognition performance of the system. The following preprocessing steps can be implemented in a face recognition system:

– **Image size normalization:** Because the Principal Components Analysis (PCA) and the Linear Discriminate Analysis (LDA) involve multiplication of arrays, it is important to normalize the size of all images. This is done by resizing all images to a default image size such as 112 x 92 pixels as in the ORL database we use in this work to guarantee that information about the eyes, nose, and mouth is not lost in potentially small versions of images.

– **Illumination normalization:** The general purpose of illumination normalization is to decrease lighting effect when the observed images are captured in different lighting environments. A common approach is to adjust observed images to approximate the ones captured under a standard lighting condition.

– Histogram equalization: Histogram equalization [5] is a process of adjusting the image so that each intensity level contains an equal number of pixels such that the appearance of the image is improved by balancing light and dark areas. Histogram equalization (HE) [5] can be used as a simple but very robust way to obtain light correction when applied to small regions such as faces. HE is to maximize the contrast of an input image, resulting in a histogram of the output image which is as close to a uniform histogram as possible. However, this does not remove the effect of a strong light source but maximizes the entropy of an image, thus reducing the effect of differences in illumination within the same “setup” of light sources. By doing so, HE makes facial recognition a somehow simpler task. Two examples of HE of images can be seen in Figure 2.

a) Before histogram Equalization b) After histogram Equalization

Figure.2. Histogram Equalization

2.2) Feature Extraction Phase:-

The aim of feature extraction [4] is to extract a compact set of interpersonal discriminating geometrical or/and photometrical features of the face. The main trend in feature extraction has been representing the data in a lower dimensional space computed through a linear or non-linear transformation satisfying certain properties. Statistical techniques have been widely used for face recognition and in facial analysis to extract the abstract features of the face patterns. Methods for feature extraction include: **PCA, LDA and Color Histogram Quantization** are described briefly in section 4.

2.3) Feature Matching Phase:-

Feature matching [4] is the actual recognition process. The feature vector obtained from the feature extraction is matched to classes (persons) of facial images already enrolled in a database. we will focus on two classifiers the: k-NN and SVM are described briefly in section 5.

3) <u>Feature Extraction Algorithms</u>

A number of methods have been proposed in the last decades [6]. In the field of face recognition, the dimension of the facial images is very high and require considerable amount of computing time for classification. The classification and subsequent recognition time [7] can be reduced by reducing dimension of the image data. Principal component analysis (PCA) [6] is one of the popular methods used for feature extraction and data representation. It not only reduces the dimensionality of the image, but also retains some of the variations in the image data and provides a compact representation of a face image. The key idea of the PCA method is to transform the face images into a small set of characteristics feature images, called eigenface, which are the principal components of the initial training set of the face images. PCA yields projection directions that maximize the total scatter across all classes, i.e., across all face images. In recognition process a test image is projected into the lower-dimension face space spanned by the eigenfaces and then classified either by using statistical theory or a classifier. The PCA method was developed in [6].

3.1) principal component analysis (PCA)

We implement a face recognition system using the Principal Component Analysis (PCA) [8] algorithm. Automatic face recognition systems try to find the identity of a given face image according to their memory. The memory of a face recognizer is generally simulated by a training set. In this paper, our training set consists of the features extracted from known face images of different persons. Thus, the task of the face recognizer is to find the most similar feature vector among the training set to the feature vector of a given test image. Here, we want to recognize the identity of a person where an image of that person (test image) is given to the system. You will use PCA as a feature extraction algorithm. We have 6 steps to perform a principal component analysis on a set of data.

1. Get face images.

$$X_i = [p_1 \ldots p_N]^T, i = 1, \ldots, M$$

2. Get the images mean.

$$m = \frac{1}{M}\sum_{i=1}^{M} x_i$$

3. Subtract each image from mean image

$$X_i = X_i - m$$

4. Calculate the covariance matrix.

$$\mathrm{cov}(X, Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

5. Calculate the eigenvectors and eigenvalues of the covariance matrix.

$$\lambda_i = \frac{1}{M}\sum_{n=1}^{M}\left(e_i^T w_n\right)^2$$

6. Choosing components and forming a feature vector

Our target to select the Eigenvectors with the largest Eigenvalues, one selects the dimensions along which the face images vary the most. Since the Eigenvectors are ordered high to low by the amount of variance found between images along each Eigenvector, the last Eigenvectors find the smallest amounts of variance. We set the minimum number of Eigenvectors to guarantee that energy e is greater than a threshold. A typical threshold is 0.95. If we define e_i as the energy of the i^{th} Eigenvector, it is the ratio of the sum of all Eigenvalues up to and including i over the sum of all the Eigenvalues.

$$e_i = \frac{\sum_{j=1}^{i} \lambda_j}{\sum_{j=1}^{k} \lambda_j} \qquad (1)$$

$$\text{Feature Vector} = (v_1\ v_2\ v_3 \ldots v_i)$$

In the training phase [9] [10], you should extract feature vectors for each image in the training set. Let Ω_A be a training image of person A which has a pixel resolution of M × N (M rows, N columns). In order to extract PCA features of Ω_A, you will first convert the image into a pixel vector ϕ_A by concatenating each of the M rows into a single vector. The length (or, dimensionality) of the vector ϕ_A will be M ×N. In this paper, you will use the PCA algorithm as a dimensionality reduction technique which transforms the vector ϕ_A to a vector ω_A which has a dimensionality d where d << M × N. For each training image Ω_i, you should calculate and store these feature vectors ω_i, as you see in figure 3.

In the recognition phase [10] (or, testing phase), you will be given a test image Ω_j of a known person. Let α_j be the identity (name) of this person. As in the training phase, you should compute the feature vector of this person using PCA and obtain ω_j . In order to identify Ω_j, you should compute the similarities between ω_j and the entire feature vectors ω_i's in the training set. The similarity between feature vectors can be computed using Euclidean distance. The identity of the most similar ω_i will be the output of our face recognizer. If i = j, it means that we have correctly identified the person j, otherwise if i ≠ j, it means that we have misclassified the person j.

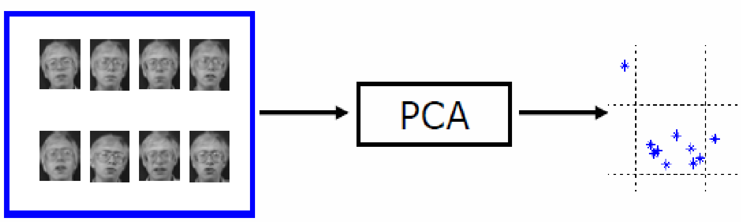

Figure 3. Reducing the dimensionality of the data, we speed up the computations, without losing too much information.

3.2) Linear discriminate analysis (LDA)

In [9], the PCA method is used for dimension reduction for linear discriminate analysis (LDA), generating a new paradigm, which called fisherface. The fisherface approach is more insensitive to variations of lighting, illumination and facial expressions. However, this approach is more computationally expensive than the PCA approach.

In face recognition, each face is represented by a large number of pixel values. Linear discriminate analysis is primarily used here to reduce the number of features to a more manageable number before classification. Each of the new dimensions is a linear combination of pixel values, which form a template. The linear combinations obtained using LDA finds an efficient way to represent the face vector space by exploiting the class information. It differentiates individual faces but recognizes faces of the same individual. [9] [11]. The images in the training set are divided into the corresponding classes. LDA is an example of a class specific method, in the sense that it tries to “shape” the scatter in order to make it more reliable for classification. This method selects W in such a way that the ratio of the between class scatter and the within- class scatter is maximized.

Given the data images U_i,
(Steps 1-4 compute the PCA eigenvectors; Step 5 computes the projected LDA data on those Eigenvectors. Steps 6-8 compute the LDA directions which separate the data.)

1. Subtract the sample mean from the data:

$$Y_i = U_i - \mu \qquad i = 1,2,\dots,n$$

2. Compute the Covariance Matrix S:

$$S = \sum_{i=1}^{n} Y_i Y_i^T$$

3. Compute eigenvectors { **v**1, **v**2 …**v**k } corresponding to the largest k eigenvalues of S.

4. Let **v**1, **v**2 …**v**k be the columns of eigenvector matrix A= [**v**1, **v**2 …**v**k].

5. The new projected LDA data are:

$$Z_i = A^T Y_i \qquad\qquad i = 1,2,\dots,n$$

6. Compute the sample mean μ_z of the LDA data and the sample mean μ_{zi} of each class.

7. Compute the class scatter matrix S_b and the within class scatter matrix S_w.

$$S_b = \sum_{i=1}^{c} n_i \left(\mu_{zi} - \mu_z\right)\left(\mu_{zi} - \mu_z\right)^T; \qquad S_w = \sum_{i=1}^{c} \sum_{\text{class K}} \left(Z_k - \mu_{zi}\right)\left(z_k - \mu_{zi}\right)^T$$

   n_i is the number of training samples in class i, c is the number of distinct classes, μ_{zi} is the mean vector of samples belonging to class i and Z_K represents the set of samples belonging to class k.

8. Solve:

$$S_b w = \lambda S_w w \;\Rightarrow [w_1, w_2, \dots, w_{c-1}]$$

3.3) Color Histogram Quantization

In last decade; the color histogram based methods have proved simplicity and usefulness. Initially, this idea was based on Color Histogram Quantization. [16]. we propose an algorithm for computing histogram of grayscale images with 256 gray levels are used and using 24 quantization levels.

In the proposed histogram, we get the frequency for every gray level value, so, we will have a vector nx1, where n indicates number of different gray levels in the image (as index) and its value is for the frequency of that gray level in the image. After that we quantize the gray level frequency vector into 24 quantization level with eleven gray level width ($256/11= 23.3 \approx 24$ quantization level).

For training phase, Firstly, the frequency of every gray level is computed and stored in vector for further processing. Secondly, we do a quantization by get the mean of consecutive eleven frequencies from the stored vectors is calculated and kept to use in testing phase. And we do the same previous processing for each testing image. So, we have a mean vector for each training images and mean vector for the test image to get its best face matching from training set. When we use the KNN classifier, it mentions on the next section in feature matching, we calculate the absolute differences among the mean of training set images and the test image mean vector, and finally, the minimum difference found identifies the matched class for the test image.

The recognition accuracy with KNN is of 100% and with SVM is 91.5% when we use 60% of faces images for training and 40% for testing, you can find all results in the experimental result section.

4) Feature Matching

Feature matching [4] is the actual recognition process. The feature vector obtained from the feature extraction is matched to classes (persons) of facial images already enrolled in a database.
The choice of the classifier [12] is the next challenging stage of our system. the election of the classifier is not as dependant of the application like feature selection, and in many cases it is done by the availability of the algorithm. We can find three main approaches in the design of the classifiers depending on the concept they are based on: similarity, geometry or probability.

The first type of classifiers is based on similarity; that is, similar patterns should correspond to the same class. One of the possible techniques is template matching; that is, the pattern to be recognized is matched against the stored templates and then a similarity measure like a correlation is estimated. Another option is the use of a minimum distance classifier, where we should select a metric (e.g. Euclidean) and a prototype for the class, for example the mean of the samples of every class. Then the class with a minimum distance between the sample and the prototype is chosen.

The second type is based on geometry. These classifiers try to estimate the decision boundaries; that is, the hyper planes that separates the classes, directly by optimizing some error criteria. Examples of this approach are support vector machines (SVM).

Finally, we have the probabilistic approach. Depending on the available information. If we do not have any information, we can only make a decision based on the a priori class probability, obtained from a large enough number of random samples, and then choose the class with the higher probability.

4.1 Classification

Now the problem is how to classify the feature vectors in different classes to help us on the recognition of the faces. One of the first ideas about this problem is the one commented in [13]. This simple method consists in finding the face class k that minimizes the Euclidian distance between the vector $\boldsymbol{x}$ and the vector $\boldsymbol{x}_k$ which describes the k^{th} class. The vectors $\boldsymbol{x}_k$ are calculated by averaging the feature vectors of the training faces of the same person. So, for each new face to be identified, the vector $\boldsymbol{x}$ is calculated and also the distances to each known class. If the minimum distance is lower than a fixed threshold, we could classify the face to that class, and if it is greater, it may be classified as unknown and optionally used to begin a new face class. Finally, the eigenfaces are recalculated to add the new faces classified as known to the model. We will focus on two classifiers the: k-NN and SVM.

4.1.1) K-nearest-neighbor algorithm (K-NN)

The first classifier we will taken about, the k-NN (k-nearest neighbor) [13] is one of the simplest classification techniques, it is based on similarity and it is based also on the idea of finding the classes of the k-nearest neighbor vectors. Then the most represented class on those k neighbors is found and it is assigned to the test sample.

In this algorithm the classification of a new object is based on attributes and training samples, the result of new instance query is classified based on majority of K-nearest neighbor category (K is predefined integer), given a query point, the algorithm find K number of objects or training points closest to the query point. Simply it works based on minimum distance from the searching query to the training one to determine the K-nearest neighbors, after we gather K-nearest neighbors we take simple majority of these K-nearest neighbors to be the prediction of the query instance.

We can compute the distance between query instance and each training data using some distance function $\mathrm{d}(\mathrm{x},\ \mathrm{y})$, where x, y are the samples composed of N features, such that $\mathrm{x} = \{\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_\mathrm{N}\}$, $\mathrm{y} = \{\mathrm{y}_1, \mathrm{y}_2, \dots, \mathrm{y}_\mathrm{N}\}$. We use Euclidean distance measuring as distance function.

$$\mathbf{d}(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{N}} \sqrt{\mathbf{x}_\mathbf{i}^\mathbf{2} - \mathbf{y}_\mathbf{i}^\mathbf{2}} \qquad (\mathbf{2})$$

- **Advantages of K-Nearest Neighbors;** Robust to noisy training data especially in Inverse Square of weighted distance and effective if the training data is large.
- **Disadvantages of KNN;** Need to determine value of parameter K (number of nearest neighbors), distance based learning is not clear which type of distance to use and which attribute to use to produce the best results, and computation cost is quite high because we need to compute distance of each query instance to all training samples.

4.1.2) Support Vector Machines (SVM)

The second classifier we will take about is the Support vector machines; SVM [14] is a binary classification method it is based on geometry. (SVMs) give us a supervised learning method for classifying both facial features and individuals based upon these. After a decent amount of training, an SVM [16] can predict whether an input falls into one of two categories. This is done by first constructing a hyper plane in a high dimensional space and then mapping input to points in this space. The input is determined to be in one category or the other by measuring its distance from the hyper plane. The strength of the category correspondence is then given by the magnitude of this distance.

- **Basic theory of support vector machine**

The basic idea of SVM [15] is to map the linear non-separable input vectors into some higher dimensional space such that a more suitable hyper plane can be found with minimal classification errors [9-11].
We start with training data,

$$D = \{(x_i, y_i)\}_{i=1}^{l}, \text{where } y_i \in \{-1,1\}, x_i \in R^N \qquad (3)$$

then map the training data into some other inner product space F via a nonlinear map,

$$\Phi : R^N \to F \qquad (4)$$

The separating hyper plane in the space F must satisfy the following constraints,

$$y_i\left(w^T z_i + b\right) \geq 1, \qquad z_i \in F, i = 1,2,\dots,l \qquad (5)$$

If the optimal hyper plane H is $w_0^T z + b_0 = 0$,

then the distance between the closest vector to the hyper plane H is,

$$\rho(w, b) = \min_{\{x|y=1\}} \frac{z^T w}{\|w\|} - \max_{\{x|y=-1\}} \frac{z^T w}{\|w\|} \qquad (6)$$

with its maximum,

$$\rho(w_0, b_0) = \frac{2}{\|w_0\|} = \frac{2}{\sqrt{w_0^T w_0}} \qquad (7)$$

So the optimal separating hyper plane is determined by the vector w, which minimizes the functional.

$$\Phi(w) = \frac{1}{2}\left(w^T w\right) \qquad (8)$$

Subject to $y_i\left(w^T z_i + b\right) \geq 1, \; i = 1,2,\dots,l$

It’s modified for the non-separable case to,

$$\Phi(w) = \frac{1}{2} w^T w + \gamma \sum_{i=1}^{l} \xi_i \qquad (9)$$

Where the ξ_i are measure of the misclassification error. In terms of Lagrange multipliers, w_0 can be written as

$$w_0 = \sum_{i=1}^{l} \lambda_i y_i z_i$$

So the decision function,

$$f = \operatorname{sgn}\left[\sum_{i=1}^{l} \lambda_i y_i \left(z^T z_i\right) + b\right] \qquad (10)$$

The theorem of functional analysis shows that a positive semi definite symmetrical function K(u, v) can solely define a Hilbert space H_k , K is the reproducing kernel of feature space H_k ,

$$K(u, v) = \sum_{k} \alpha_k \varphi_k(u) \varphi_k(v) \qquad (11)$$

which represents a inner product in the feature space,

$$z_i^T z = \phi(x_i)^T \phi(x) = K(x_i, x) \qquad (12)$$

The decision function can thus be written as

$$f = \operatorname{sgn}\left[\sum_{i=1}^{l} \lambda_i y_i K(x_i, x) + b\right] \qquad (13)$$

- **SVM in multi-class classification**

The formulation of SVM in previous section was based on a two-class problem; hence SVM is basically a binary classifier. Several different schemes can be applied to the basic SVM algorithm to handle the K-class pattern classification problem. The schemes which have been proposed in [15] for solving the multi-class problem are as listed below:

Using k one-to-rest classifiers is the simplest scheme, and it does give reasonable results. K classifiers will be constructed, one for each class. The K^{th} classifier will be trained to classify the training data of class k against all other training data. The decision function for each of the classifier will be combined to give the final classification decision on the K-class classification problem,

$$f(x) = \arg\max_k \sum_{i=1}^{l} \lambda_i^k y_i K^k(x_i, x) + b^k \qquad (14)$$

Using k(k-1)/2 pair wise classifiers with majority voting or pair wise coupling as the voting scheme. The schemes require a binary classifier for each possible pair of classes. The decision function of the SVM classifier for K^i -to-K^j and K^j -to-K^i has reflectional symmetry in the zero planes. Hence only one of these pairs of classifier is needed. The total number of classifiers for a K-class problem will then be K (K-1)/2. The training data for each classifier is a subset of the available training data, and it will only contain the data for the two involved classes. The data will be reliable accordingly, i.e. one will be labeled as +1 while the other as -1. These classifiers will then be combined with some voting scheme to give the final classification results, such as majority voting or pair wise coupling.

5) EXPERIMENTAL RESULTS

The ORL faces database is used in the experiments. The ORL faces database contains have 40 persons with 10 face images for each person with large illumination variation. We use a varying number of images as a training set and testing set, see the following experiment table:-.

Total Classes	**Training Percentage**	**Testing Percentage**	**Feature Extraction**	**Classifier**	**Total Correct**	**Total False**	**Accuracy**
40	60%	40%	PCA	KNN	146/160	14/160	91.25%
40	70%	30%	PCA	KNN	112/120	8/120	93.3%
40	80%	20%	PCA	KNN	75/80	5/80	93.75%
40	90%	10%	PCA	KNN	38/40	2/40	95%
40	60%	40%	PCA	SVM	136/160	24/160	85%
40	70%	30%	PCA	SVM	105/120	15/120	87.5%
40	80%	20%	PCA	SVM	75/80	5/80	93.75%
40	90%	10%	PCA	SVM	37/40	3/40	92.5%
40	60%	40%	LDA	KNN	151/160	9/160	94.4%
40	70%	30%	LDA	KNN	115/120	5/120	95.8%
40	80%	20%	LDA	KNN	76/80	4/80	95%
40	90%	10%	LDA	KNN	37/40	3/40	92.5%
40	60%	40%	LDA	SVM	123/160	37/160	77%
40	70%	30%	LDA	SVM	99/120	21/120	82.5%
40	80%	20%	LDA	SVM	70/80	10/80	87.5%
40	90%	10%	LDA	SVM	37/40	3/40	92.5%
40	60%	40%	Histogram Quantization	KNN	160/160	0/160	100%
40	70%	30%	Histogram Quantization	KNN	120/120	0/120	100%
40	80%	20%	Histogram Quantization	KNN	80/80	0/80	100%
40	90%	10%	Histogram Quantization	KNN	40/40	0/40	100%
40	60%	40%	Histogram Quantization	SVM	141/160	19/160	88.13%
40	70%	30%	Histogram Quantization	SVM	111/120	9/120	92.5%
40	80%	20%	Histogram Quantization	SVM	74/80	6/80	92.5%
40	90%	10%	Histogram Quantization	SVM	38/40	2/40	95%

6) CONCLUSION and FUTURE WORK

The Eigenface and Fisherface method were investigated and compared. The comparative experiment showed that the Fisherface method outperformed the Eigenface method. The usefulness of the Fisherface method under varying illumination was verified. The color histogram quantization gets the best result with KNN classifier. So, in the near future we want to try many dataset and enhance the PCA and LDA results.

7) References

1. Mazen Eleyan, “PCA- And LDA-based face recignition using region division with majority voting”, Master’s Thesis,Near East Unveristy,Nicosia 2008.

2. Nes, A., "Hybrid Systems For Face Recognition", master thesis, Faculty of Information Technology, Norwegian University of Science and Technology, June 2003.

3. MuhammadTariqMahmood,”Face Detection by Image Discriminating” Master thesis,Intelligent Software Systems,Blekinge Institute of Technology, Sweden,August 2006.

4. Jens Fagertun,” Face Recognition” , Master Thesis IMM-Thesis-2005-74,Kongens Lyngby 2005.

5. K.V. Krishna Kishore1, Dr G.P.S.Varma[2], “Hybrid Face Recognition System using Multi Feature Neural Network “,Journal of Computing, Volume 2, Issue 7, ISSN 2151-9617, July 2010.

6. Byung-Joo Oh ,” Face Recognition using Radial Basis Function Network based on LDA “,World Academy of Science, Engineering and Technology 7 2005.

7. S.thakur, J.k.sing, D.k.basu “face recognition using principal component analysis and rbf neural networks’’1-51/RID/EF(13),2008.

8. Erald VUÇINI,” Face Recognition Under Varing Illumination” master thesis, Isanbul Technical University, Institute Of Science and Technology, June 2006.

9. A. Hossein Sahoolizadeh, B. Zargham Heidari, “A New Face Recognition Method using PCA,LDA and Neural Network”, World Academy of Science, Engineering and Technology 41 2008.
10. Juwei Lu, Kostantinos N. Plataniotis, and Anastasios N. Venetsanopoulos, “Face Recognition Using LDABased Algorithms”, IEEE transactions on Neural Networks, Vol.14, No. 1, Jan 2003.

11. T. Giraud, A. Kodsi, T. Luong, R. Lauranson, C. Marcovici, K. Sadeghi, Reconnaissance de

formes par réseau de neurones, projet scientifique collectif, école polytechnique, Juillet 2002.

12. Jose Gomez Martil,”face recognition in controlled environments using multiple images”Master thesis,2009.

13. Iulia Chiriacescu,”Automatic Emotion Analysis Based on Speech’’,MSc THESIS, Faculty of Eng, Mathematics and CS,Delft University of Technology, July2009.

14. Ping Du, Yankun Zhang, Chongqing Liu,” Face Recognition using Multi-class SVM”, The 5th Asian Conference on Computer Vision, 23--25 January 2002, Melbourne, Australia.

15. Alex Bullard, William Martin, and Joe Redmon,” Component Based SVM Facial Recognition”, May 9, 2010.

16. Fazl-e-Basit, Younus Javed and Usman Qayyum, "Face Recognition using processed histogram and phase only correlation ", 3rd IEEE International Conference on Emerging Technology, pp. 238-242, NOV 2007.