**Final Thesis**

**Final Year Project**

# Project: Face Recognition

**Student: Martin Luhanjo**

**Student ID: 200759154**

**Course: Electronics and Computer Science**

**Supervisor: Dr. Mu**

**Former Supervisor: Prof. A.K Nandi**

**Assessor: Dr. W. Al-Nuaimy**

**Date: 29/04/2013**

## Abstract

This project involved developing a system implemented in MATLAB language that can recognize a person from a still or video image from images source by comparing features of face of unknown person and face images stored in face database. Face recognition system contained four main stages; face database creation, image pre-processing, features extraction and classification process. For image pre-processing, three different techniques were applied prior to recognition process; Convolution methods, Statistical methods and method combinations, where multiple methods were combined together and applied on the face images. For features extraction stage, two algorithms were implemented; Eigenface approach and Fisherface approach and for classification process, three classifier were used; Euclidean distance, Mahalanobis distance and K-nearest neighbours classifier. In face recognition system, image pre-processing stage is only used to enhance performances of the face recognition algorithms by reducing some of the limitations of the system. Two experiments sets were carried out using two different face database; Yale face database which consists of face images in large variation in lightning conditions and facial expressions and another database was AT&T face database which consists of face images in large variation in viewing direction and facial expressions. From the experimental results, it can be concluded that for Yale face database, the optimal combination for the best performance was the one composed of Fisherface algorithm and k-NN classifier, while for AT&T face database, optimal combination for the best performance was the one composed of Eigenface algorithm and Mahalanobis distance, with image pre-processing applied prior to recognition process in both experiments. Generally both face recognition algorithms responded better on face images with frontal views, facial expressions and small viewing angles than on face images with variation in lightning directions.

# Contents

## Table of figures

# 1. Introduction

Face recognition system is computer program application which automatically identifies a person from digital images or video frame from video source such as CCTV cameras [2]. This can be implemented using different approaches based on comparing significant facial features from the face image of unknown person and face database of known individuals. Last several decades research based on the face recognition system has been of the most interesting and popular area of research in computer vision and successful applications of image analysis and understanding due to its wide application in today's technology, hence not only computer science researchers were involved in this project, but also neuroscientists and psychologist expand the use of face recognition system applications [2] [6]. Face recognition system has numbers of significances in today's technology and huge developing impact in different environment due to its applications and services it offers. Face recognition system can be used as embedded system to provide security services, can be used to identify criminals and terrorists, can provide access control on different areas e.g. computers, whereby you can use your face as key, also has been used to prevent face identification fabrication [8]. Examples of real scenarios where face recognition systems has been used; At super Bowl XXXV in January 2001, police in Florida used visage facial recognition system to identify criminals and terrorists at the event [7], also was used in Mexico 2000 presidential election to prevent votes duplication.

Face recognition algorithms consists of four main stages which are; face database creation, image pre-processing, Feature extraction and classification of features extracted from images. Feature extraction is only stage that differentiate most of the algorithms since each approach consider its own facial features to use for variation between images such as, some algorithm use relative position or size of eyes, nose and lips. On this report will look in details on face recognition based on Eigenface algorithm and Fisherface algorithm.

Face recognition based on Eigenface algorithm is one of the first successful approaches that machine could carry out face identification in automated manner [4]. Features considered in this algorithm are Eigenfaces (principal components) which are generated by mathematical process called Principal Component Analysis (PCA) [9] [10]. PCA tends to convert high dimensional vector representation of each face images in training set into low dimensional vector (subspace) whose basis vectors accounts to maximum variation of face images in whole image space [9]. Recognition is achieved by constructing face space and projecting all face images into this face space where, it maximize the total scatter of all projected images. Obtaining unknown face image and project onto the face space and comparing the position of all images relative to the unknown image and matching the test image with the closest image from database on the face space.

Another successful approach to implement face recognition system was Fisherface algorithm [1]; which is based on both PCA and Fisher Linear Discriminant Analysis (FLDA). First PCA was applied to reduce face images to corresponding Eigenfaces in low dimensional vector and then applies FLDA to find linear projection which further reduces dimensions and maximizes total scatter between each class while minimizes total scatter within each class [1].

## 1.1. Related Works

Face recognition technology has been one of the popular researches on computer vision over last several decades due to its nature of problem and relevance to today's technology [2]. There are numbers of attempts in computer recognition of faces which focused on detecting individual features such as eyes, nose, lips and head orientation, other techniques involved defining relative position, size or shape [2]. Following are some of the attempts or related works from different people on the face recognition over last several decades.

In 1966a,b Blesdoe [11] [12] was first to attempt semi automated face recognition with hybrid human computer system, which classified faces on the basis of fiducially mark entered on face image by hands. Parameters on the classification were normalized distances and ratios from the marks such as eyes corners, lips corners and nose tip. Another attempt was by Golden, Harmon and Lesk in 1971 [13], they generated a vector of up to 21 features and recognition of faces was achieved by using standard pattern classification techniques. The significant features were largely subjective evaluations such as length of ears made by human subjects, each of which would be quite hard to automate.

Another approach to implement face recognition was Eigenfaces algorithm which is considered as first successful face recognition technology. The approach was developed by Sirovich and Kirby in 1987 and used by M. Turk and A. Pentland in face classification [3] [4] [5]. This approach used Eigenfaces (Eigen vectors) derived from the covariance matrix of the probability distribution of high dimensional vector space of faces of human beings. Eigenfaces were generated using mathematical process called Principal Component Analysis (PCA) which was applied on the face database consist of different individuals. PCA transform original face images from high dimensional vector space into corresponding Eigenfaces in low dimensional vector space and find best eigenvectors (principal components) with highest eigenvalue to account for maximum discrimination among the images [10]. Recognition was achieved by projecting face images onto feature space and classify by using Euclidean distance, whereby unknown face image was identified by matching with the individual closest to unknown image on the face space. Another successful approach to implement face recognition was Fisherface algorithm which was originally derived from idea suggested by R.A Fisher in 1936 [1]. This was a linear algorithm based on Principal

Component Analysis and Fisher Linear Discriminant Analysis (FLDA), first PCA was applied to reduce face images to corresponding Eigenfaces in low dimensional vector and then applies FLDA to find linear projection which further reduces dimensions and maximizes total scatter between each class while minimizes total scatter within each class [1]. Another approach to implement face recognition system; Independent Component Analysis (ICA) which minimizes both second order and higher order dependencies in the input data and attempts to find the basis along which the data is statistical independent [20].

## 2. Design Overview

This section describes the design of the project and procedures followed to achieve specifications of the project. Face recognition system contained four main stages; Database creation, Image pre-processing, Features extraction, Classification. Database creation is essential phase where images are randomly selected to form training set and testing set, for Image pre-processing stage involved two methods; Convolution and Statistical methods which were used as critical evaluations to improve the performances of recognition system. For features extraction stage; Eigenface and Fisherface algorithms were implemented to extract features from the face images and for classification; Distance classifiers and k-Nearest Neighbour classifier were used to classify features extracted from the face images. Flow of these stages can be shown by figure 1 below.

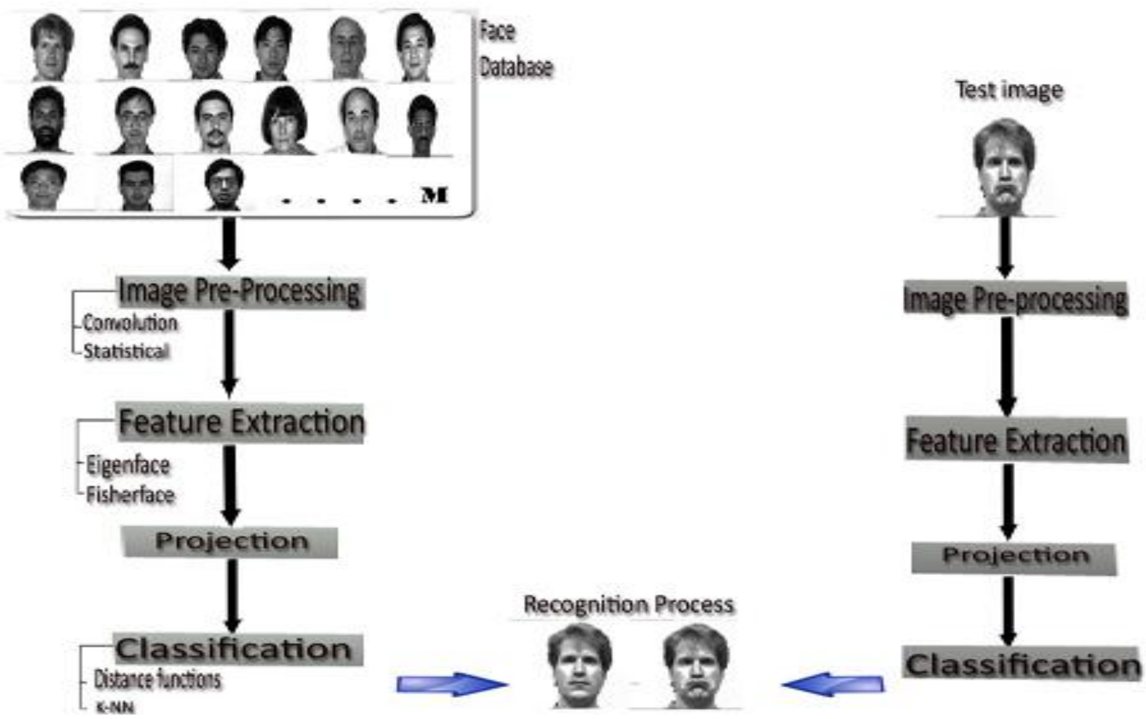



**Figure 1 :** Design Overview

## 2.1. Program design

Face recognition system was implemented using programming language MATLAB, which simulated all four main stages of the recognition system; face database creation, Image pre-processing, features extraction and classification. Program design is based on both recognition algorithms; Eigenface and Fisherface.

**Pseudo-English**

1. Start programme.
2. Load images to training set.
3. Apply image pre-processing methods on face images.
4. Features extraction methods; PCA and FLDA.
5. Load a test image
6. Apply image pre-processing methods on test image.
7. Extract features from test image
8. Project all images onto face space.
9. Classification methods; k-NN, Euclidean and Mahalanobis distance functions.
10. Recognition process.
11. End of the programme.

# 3. Face databases

This section discusses two face databases used to experiment face recognition system. One database was Yale face database [14], which consists of face images in large variation in lightning conditions and facial expressions. Another database used was AT&T face database [15] which consists of face images in large variation in viewing direction and facial expressions.

## 3.1. Yale face database

Yale face database [14] consists of 165 distinct face images of 15 individuals, where every person contains 11 samples which portray different facial expressions and lightning directions. This database assesses the performance of the system on the conditions portrayed by 11 samples of each individual on the database. Each face image has image resolution of 320 x 243 pixels in GIF format. Randomly images from each individual were selected to create a training set and testing set images, number of samples for each individual in training set were varied to observe how it affect the algorithm response.

**Figure 2**: Face images of all individuals in Yale face database.

**Figure 3:** 11 Face samples of one individual under different conditions from Yale face database.

## 3.2. AT&T face database

AT&T face database [15] consists of 400 distinct face images of 40 individuals, whereby each individual contains 10 samples which portray different facial expressions and viewing directions. This database used to assess the performance of the face recognition system under conditions with variation in facial expressions and viewing directions. Each face image in this database has image resolution 320x243 pixels in PGM format. Randomly images were selected to create a training set and testing set images, number of samples for each individual in both sets was varied to observe performance of face recognition algorithms under different conditions of face images.

**Figure 4**: Face images of some of individuals (15/40) in AT&T face database.

**Figure 5:** 10 Face samples of one individual under different conditions from AT&T database.

# 4. Theoretical Analysis

This section describes in general context about procedures used on both Eigenface and Fisherface algorithms to generate features which account for maximum variation between face images in the recognition system. For Eigenface approach, Principal Component analysis (PCA) was used to generate eigenfaces, while for Fisherfaces algorithm, both PCA and FLDA were used to generate fisherfaces.

## 4.1. Principal Component Analysis (PCA)

Principal Component analysis (PCA) [10] is mathematical analysis that applies orthogonal transformation to convert a set of observation of correlated variables into set of values of uncorrelated variables (principal components). Purpose of PCA is reducing high dimensionality of observed variables to lower dimensionality of feature space while maximizing the total scatter of the projected samples. Assume each face image has a size *w* by *h* pixels can be presented in 1-D as *N* where *N* is product of *w* and *h* (*w* x *h*). In case of face recognition can be performed as follows.

For given face database of *m* face images $\{x_1, x_2, x_3, \ldots, xm\}$ of a size *N* by *N* , whereby each face image in database belong to known class $\{X_1, X_2, X_3, \ldots, Xc\}$ (1)

Find mean image of the whole training set $\mu \in R^N$ is defined as

$$\mu = (1/m) \sum_{i=1}^{m} x_i \quad (2)$$

Then face images in training set are centered by subtracting mean image from each image vector

$$z_i = x_i - \mu; \quad (3)$$

$$\mathbf{A} = \{z_1, z_2, z_3 \ldots zm\} \in R^{NxN}$$

Covariance matrix of the training set $\mathbf{C} \in R^{NxN}$ can be defined by

$$\mathbf{C} = \sum_{i=1}^{m} (x_i - \mu)(x_i - \mu) T \quad (4)$$

Then find eigenvectors and eigenvalues of covariance matrix to get set of *m* orthogonal vectors. But finding eigenvector from **C** is enormous due to big size of **C** which is *N* by *N,* hence eigenvectors and eigenvalues can be resolved by $(x_i - \mu)^T (x_i - \mu)$ where the size of this matrix is of *m* by *m* (*m*<<*N*) and can be easily computed.

Then from eigenvectors of the surrogate matrix of covariance matrix can be used to find eigenvectors of **C** using linear algebra theory as follows.

Let $\mathbf{v_i}$ is eigenvectors and $\boldsymbol{\lambda_i}$ is eigenvalues

$$\mathbf{A^T A.v_i = \lambda_i.v_i} \quad (5)$$

Multiply by A on both sides of equation (5)

$$\mathbf{A\,A^T (A.v_i) = \lambda_i (A.v_i)} \quad (6)$$

From equation (6) it implies that $\mathbf{A.v_i}$ and $\boldsymbol{\lambda_i}$ are eigenvectors and eigenvalues of $\mathbf{A\,A^T}$ respectively whereby *m*-1 eigenvectors are selected to form linear combination of original images. Eigenvectors are sorted from high to low accordingly to the eigenvalues whereby eigenvectors with highest eigenvalues give maximum variation of different images.

## 4.2. Fisher Linear Discriminant Analysis (FLDA)

Fisher Linear Discriminant Analysis [21] [22] is a method that tends to find vectors in underlying space that best discriminate among classes whereby for all samples of all classes between-class scatter matrix and within-class scatter matrix are defined. FLDA [21] is an example of class specific method since it tends to shape the scatter in order to make it more reliable for classification. Purpose of Linear discriminant analysis is to maximize total discrimination between each class while minimizing total discrimination within each class, which tend to maximize ratio of between-class scatter matrix to within-class scatter matrix [23]. FLDA can be performed as follows.

Assume each face image in database has a size of *w* by *h* pixels can be represented in 1-Dimension as *N* where, *N* is product of *w* and *h.*

For a given face database of *m* face images $\{x_1,x_2,x_3,\ldots, x_m\}$ of size *N* by *N* whereby each face image belong to a known class $\{X_1,X_2,X_3,\ldots, Xc\}$ (1)

Find an overall mean face image of the whole training set, $\mu \in R^N$ can be defined as

$$\mu = (1/m) \sum^{m}_{i=1} x_i \qquad (2)$$

Then find mean face image $\mu_i \in R^N$ of each class, $X_i$; mean face image, $\mu_i$ can be defined as

$$\mu_i = (1/n) \sum^{c}_{i=1} x_i \quad \text{, where } \{i=1, 2\ldots c\} \text{ and} \qquad (3)$$

$n$=number of samples of each class

Then within-class scatter matrix of all the $m$ face images $S_w \in R^{NxN}$ can be defined as follows, whereby $k$ is number of samples in the class.

$$S_w = \sum^{c}_{i=1} \sum^{k}_{xi} (x_i-\mu_i)\,(x_i-\mu_i)^T \qquad (4)$$

Then the between-class scatter matrix of all the $m$ face images $S_B \in R^{NxN}$ can be defined as follows. $N_i$ is number of samples in each class.

$$S_B = \sum^{c}_{i=1} N_i\,(\mu_i-\mu_i)\,(\mu_i-\mu_i)^T \qquad (5)$$

From equation (4), if $S_w$ is nonsingular [22][23] the optimal projection $W_{opt}$ becomes matrix with orthonormal columns which maximizes the ratio of determinant of between-class scatter matrix of projected sample to the determinant of the within-class scatter matrix of projected samples. This can be defined as

$$W_{opt} = \text{argmax} \frac{|\,W^T S_B W\,|}{|\,W^T S_W W\,|} \qquad (6)$$

$$= \{w_1, w_2, .w_m\}$$

Where $\{w_i\,|\,i=1, 2 \ldots m\}$ is the set of generalized eigenvectors of $S_B$ and $S_w$ corresponding to the $m$ largest generalized eigenvalues $\{\boldsymbol{\lambda}_i\,|\,i=1, 2 \ldots m\}$. This can be defined as

$$S_B w_i = \boldsymbol{\lambda}_i\, S_w w_i \qquad (7)$$

It is noticed that there are at most c-1 non-zero generalized eigenvalues, where c is number of classes in the training set. Benefits of FLD include provision of class specific linear projection which construct low dimensional analogue to the classification in which the face images from each lass lie near a linear subspace.

## 5. Image Pre-processing

This section discusses three image pre-processing techniques used in this project to enhance performances of face recognition algorithms; First technique used was Convolution methods, which tend to enhance features, reduces noise and extract edges [26]. Second technique was Statistical methods which apply transformations to the image intensity values in order to make brightness adjust constant for all images [25] [26]. Third technique was Method combinations which tend to combine some of Convolution and Statistical methods to produce best improvement of recognition rate. In face recognition system Image pre-processing techniques were applied prior to recognition process on both training images and testing images.

### 5.1. Convolution Methods

Convolution methods tend to apply small template to a window, moved step by step over original mage to enhance or suppress features, reduce noises and extract edges [26]. Convolution methods used in this project include; Smoothing/Filtering face images using standard low-pass filtering, sharpening to reduce the blur in the face images and Edge detection to enhance the edges of face image. Following figures below show effect on face image after applying these methods.


**Figure 6:** Face image after applying Convolution methods

## 5.2. Statistical Methods

Statistical methods tend to apply transformations to the face image intensity values in order to make the brightness adjust constant for all face images [27]. Statistical methods are mostly useful on improving performances of recognition rate in images with large variation in lighting conditions since it tend to compensate variations in lightning across a face image by applying these methods to regions of the face [26]. It is noticed that Statistical methods do not only compensate difference in lightning conditions from one image to another, but also for different lightning conditions from one area of the face to another [27]. Statistical methods used in this project include; Brightness which provides global transformations of brightness to normalize intensity of face image and another method was local brightness which applies brightness method to individual local regions of an image [26]. Following figures show effect of these methods when applied on the face image.



**Figure 7:** Face image after applying Statistical methods

## 5.3. Methods Combinations

This image pre-processing techniques involve combination of multiple methods from Convolution and Statistical methods which produce best improvement in accuracy rate. Combinations method used in this project include; sharpening, local brightness and smoothing, whereby face image was sharpen followed by local brightness transformations, followed by smoothing. Following figures show effect of these combined methods when applied on face image.


**Figure 8:** Face image after applying Method combinations

## 6. Feature Extraction

This section provides details about two algorithms for features extraction stage of face recognition system. One approach used was Eigenface algorithm which uses eigenfaces (eigenvectors) as characteristics features generated by Principal Component Analysis to account for maximum variation between face images. Another approach was Fisherface algorithm which applies PCA and LDA to generate basis vectors defining subspace known as Fisherfaces.

### 6.1. Eigenface Algorithm

Eigenfaces [1] are set of eigenvectors used in computer vision of face recognition to account for maximum variation between face images. Eigenfaces are generated from covariance matrix of the probability distribution of high dimensional vector space of human faces using Principal Component Analysis which convert face images from high dimensional vector space to corresponding Eigenfaces in low dimensional vector space by finding only the best eigenvectors with high eigenvalues which can significantly discriminate face images. Each face image location tend to contributes more or less Eigenfaces, hence Eigenfaces can be considered as facial features which are independent from other facial features such as eyes, nose and lips, whereby face images can be represented as linear combination of their Eigenfaces or best Eigenfaces with highest eigenvalues.

#### 6.1.1. Eigenfaces Generation

Eigenfaces are generated using Principal Component Analysis from covariance matrix [10]. The following are the procedures to apply PCA on training set of images.

Given a set of $m$ training images $\{x_1, x_2, x_3 \ldots, xm\}$; each image has a size of $w$ by $h$ pixels. Images are converted from 2-Dimensional vector space to 1-Dimensional vectors space, whereby $w$ x $h= N$ and images can be represented as $x_i \in R^N$ ($i=1,2\ldots..m$) where $x_i$ is a member of samples of known individual which belong to known classes $\{X_1,X_2,X_3,\ldots, Xc\}$

**Images** = $\{x_1, x_2, x_3 \ldots, xm\}$, size of the matrix *N by N* (1)

Then mean face image $\boldsymbol{\mu} \in R^N$ of the whole training set can be defined as

$$\boldsymbol{\mu} = (1/m) \sum^{m}_{i=1} x_i \tag{2}$$

From overall mean **µ**, we can find deviation of all images in training set by subtracting mean face image from each face image in Image matrix.

$$s_i = x_i - \boldsymbol{\mu} \quad \text{where } i = 1, 2, 3 \ldots..M \tag{3}$$

Shifted image set is given by $\mathbf{A} = \{s_1, s_2 \ldots\ldots.s_m\} \in {}^{NxN}$

From shifted images matrix **A**, Covariance matrix **C** can be defined as

$$\mathbf{C} = \mathbf{A}.\mathbf{A}^T \quad \text{where the size of matrix is } N \text{ by } N \tag{4}$$

Determining eigenvectors and eigenvalues from covariance matrix $\mathbf{C} \in R^{NxN}$ is enormous due to large size of Covariance matrix which will give N eigenvectors and eigenvalues.

Hence eigenvectors and eigenvalues can first be obtained from $\mathbf{L}= \mathbf{A}^T.\mathbf{A}$, $\mathbf{L} \in R^{mxm}$ where $m$ is much smaller than **N** ($m<<N$) and computation becomes feasible and easily achieved. From **L**, eigenvectors of **C** can be obtained using linear algebra theory and only $m$-1 significant eigenvectors instead of $N$ eigenvectors will be selected to define variation between images.

Let eigenvectors set of **L** be $\mathbf{V}= \{v_1, v_2 \ldots\ldots v_m\}$, while $\boldsymbol{\lambda}= (\lambda_1, \lambda_2\ldots.. \lambda_m)$ are eigenvalues set of **L.** Then algebra theory can be used to obtain eigenvectors set and eigenvalues of Covariance matrix **C** as follows;

$$\mathbf{L}.\, v_i = \lambda_i.v_i;$$

$$(\mathbf{A}^T.\mathbf{A})\, v_i = \lambda_i\, .v_i \tag{5}$$

Multiply **A** on both sides of equation (5);

$$\mathbf{A}\,(\mathbf{A}^T\,\mathbf{A})\,v_i = \lambda_i\,.\mathbf{A}.\,v_i$$

$$\mathbf{A}\,\mathbf{A}^T\,(\mathbf{A}.v_i) = \lambda_i.\,(\mathbf{A}.v_i) \tag{6}$$

$$\mathbf{C}.\,(\mathbf{A}.v_i) = \lambda_i.\,(\mathbf{A}.\,v_i)$$

Hence eigenvectors and eigenvalues of **C** can be defined as $\mathbf{A}.v_i$ and $\lambda_i$ (i=1.2….m) respectively from equation (6).

Then use eigenvalues to sort eigenvectors from highest to the lowest to obtain best Eigenfaces to account for maximum variation between face images by omitting eigenvectors with zero eigenvalues.

Feature space with only useful eigenvectors can be defined as;

$$\mathbf{W} = \{w_1, w_2 \ldots\ldots w_m\} \in R^q; \text{ where q is number of best eigenvectors.} \tag{7}$$

Projection of images onto new feature space can be obtained as;

$$w_i = \mathbf{V}^T.\,\mathbf{A_i} \qquad \text{where V are eigenvectors and A is shifted images.} \tag{8}$$

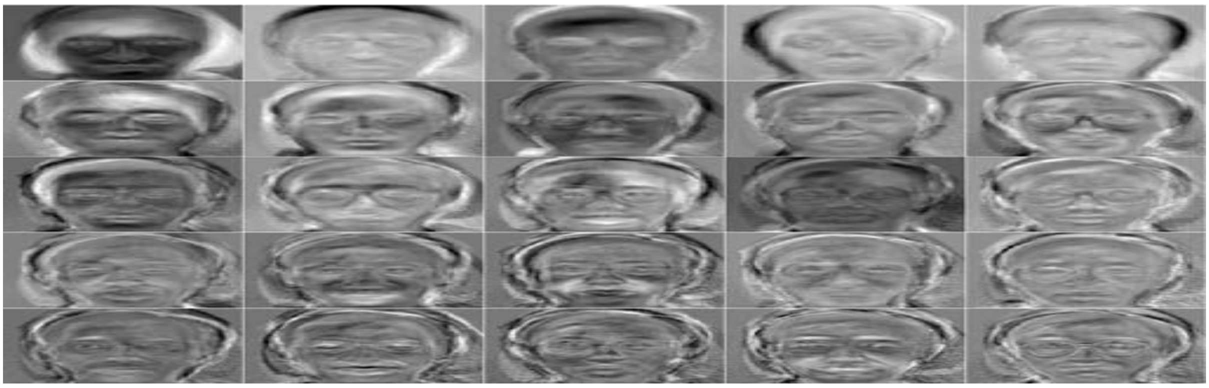


**Figure 9**: Example of Eigenfaces from Yale face database [19]

## 6.2. Fisherface Algorithm

Fisherface [21] [22] is a linear and class specific approach based on both PCA and LDA, whereby PCA is applied first to reduce a given image space to new subspace in lower dimensional, this yields to set of eigenfaces which account for maximum variation between face images, in other words it maximize total scatter between face images on the face space. Then FLDA is applied to find a linear projection which further reduce dimension of subspace and maximizes total discrimination between each class while minimizing total discrimination within each class. Fisherface algorithm achieves more class specific discrimination which allows samples from the same class to lie near each other on the feature space, this help to produce more reliable performance with even simple classification method, where classes can be linear separated. This approach in face recognition can be implemented using following procedures.

Let each face image has a size of $w$ by $h$ pixels can be represented in 1-Dimension as $N$ where, $N$ is product of $w$ and $h$.

For a given face database of $m$ face images $\{x_1, x_2, x_3 \ldots, x_m\}$ of size $N$ by $N$ whereby each face image belong to a known class $\{X_1, X_2, X_3 \ldots, Xc\}$ , where c is number of classes. (1)

Find an overall mean face image of the whole training set, $\mu \in R^N$ can be defined as

$$\mu = (1/m) \sum^{m}_{i=1} x_i \qquad (2)$$

From overall mean face image μ; find deviation of all images in training set by subtracting mean image from each face image.

$$s_i = x_i - \boldsymbol{\mu} \quad \text{where } i = 1, 2, 3 \ldots.. M \qquad (3)$$

Shifted image set is given by $\mathbf{A} = \{s_1, s_2 \ldots\ldots. s_m\} \in R^{NxN}$

Then apply PCA to obtain new eigenspace $V = \{v_1, v_2 \ldots v_m\} \in R^k$ from covariance matrix. PCA yields to eigenfaces which correspond to highest eigenvalues in low dimension feature space.

Then find mean face image $\mu_i \in R^N$ of each class, $X_i$; mean face image, $\mu_i$ can be defined as

$$\mu_i = (1/n) \sum_{i=1}^{c} x_i \quad \text{, where } \{i=1, 2 \ldots c\} \text{ and} \tag{4}$$

$n$=number of samples of each class

Then within-class scatter matrix of all the $m$ face images $S_w \in R^{NxN}$ can be defined as follows, whereby $k$ is number of samples in the class.

$$S_w = \sum_{i=1}^{c} \sum_{xi}^{k} (x_i-\mu_i)(x_i-\mu_i)^T \tag{5}$$

Then the between-class scatter matrix of all the $m$ face images $S_B \in R^{NxN}$ can be defined as follows. $N_i$ is number of samples in each class.

$$S_B = \sum_{i=1}^{c} N_i (\mu_i-\mu)(\mu_i-\mu)^T \tag{6}$$

From equation (5), it is noticed that, if $S_w$ is non-singular [22][23] the optimal projection $W_{opt}$ becomes matrix with orthonormal columns which maximizes the ratio of determinant of between-class scatter matrix of projected samples to determinant of the within-class scatter matrix of projected samples. This can be defined as.

$$W_{opt} = \text{argmax} \frac{| W^T S_B W |}{| W^T S_W W |} = \{w_1, w_2, .w_m\} \tag{7}$$

Where $\{w_i | i=1, 2 \ldots m\}$ is set of generalized eigenvectors of $S_B$ and $S_w$ corresponding to the $k$ largest generalized eigenvalues $\{\lambda_i | i=1, 2 \ldots m\}$, this can be defined as.

$$S_B w_i = \boldsymbol{\lambda}_i S_w w_i \tag{8}$$

Then only $k$ eigenvectors corresponding to $k$ largest eigenvalues are kept in matrix V, where $W= w_1, w_2 \ldots w_k\} \in R^k$, then new face space $P= \{p_1, p_2 \ldots p_m\}$ is defined as.

$$P= V.W \tag{9}$$

Where **V** are eigenfaces result from PCA.

Projection of images onto new feature space can be defined as.

$$T = P^T.A \quad (10)$$

Where **A** is shifted image matrix.

Projection of the test image onto the feature space can be defined as.

$$t = W^{-1}.V^{-1}.z \quad (11)$$

Where z is defined as z= x-μ.

## 7. Classification

This section provides detail about how feature extracted from training images can be classified in the feature space to identify unknown face images. Method used for classification included; Euclidean distance, Mahalanobis distance and K-NN classifier, which compare distance of weights of face images on the feature space. The following are initialization procedures.

a) Obtain unknown face image, $x \in R^N$, find deviation from mean face image of the whole training set, $\boldsymbol{\mu}$

   $s = x - \boldsymbol{\mu}$ where $\boldsymbol{\mu}$ is mean face image

b) Calculate set of weights based on unknown image and project onto the face space for classification.

   $w = \mathbf{V}^T.s$; where $w \in R^q$ and V are q best Eigenfaces or Fisherfaces.

c) Find if the unknown image is a face by calculating its position from the face space and check if it is sufficiently close to face space to present a face image.
d) If it is a face image, classify the weights as either known face image or unknown person from the face database.

## 7.1. Euclidean Distance

Euclidean distance [17] is one of mostly classification method used in different algorithms due to its simplicity to achieve recognition. Euclidean distance or metrics is an ordinary distance between two points which might be 1-D, 2-D or in more Dimensions. The following is implementation of this method in recognition process of unknown images.

For a given training set of *m* images $\{x_1, x_2 \ldots x_m\}$ and unknown image $x \in R^N$ , where $V \in R^q$ are best Eigen faces of face images as explained in Feature extraction section (5).

Projection of face images onto face space can be defined as;

$\mathbf{W} = \mathbf{V}^T.\mathbf{A}$ ; where **V** is a set of best q eigenvectors obtained by PCA

And **A** is set of shifted images. (1)

Projection of unknown image is defined by;

$w_x = V^T. S$; where S is deviation from mean image $(x-\boldsymbol{\mu})$ (2)

Euclidean distance is used to measure distance between position of projected images (training images and unknown image) onto feature space.

Distance between unknown image and each training image can be defined as;

$\mathbf{E_i^2} = || w_x - w_i ||^2$ ; where (i=1, 2.....m) (3)

Threshold distance $\mathbf{E_T}$ can be defined as half the largest distance between any two face images on the face space.

$\mathbf{E_T} = \frac{1}{2} \, \mathbf{max}_{i,j} (|| w_i - w_j ||^2)$ ; where i,j =1,.....m (4)

Hence for recognition process; compare minimum distance of individual in training set from unknown image to give different observations.

- ✓ If **min** ($\mathbf{E_i}$) > $\mathbf{E_T}$, implies that unknown image is not face image or cannot be recognized using this face database.
- ✓ If **min** ($\mathbf{E_i}$) < $\mathbf{E_T}$, implies that unknown image is recognized by the face of individual in the training set.

## 7.2. Mahalanobis Distance

Mahalanobis distance [28] is a distance measure based on correlations between variables by which different patterns can be identified and analysed. Mahalabonis distance provides better distance measure in recognition problems since it takes into account the covariance between the variables, this remove problems related to scale and correlation that inherent with Euclidean distance [29].

Mahalabonis distance function can be defined as.

$$D_i= \sqrt{(w_i-w_x)^T C^{-1} (w_i-w_x)}$$

Whereby $w_x$ is weight of the test image, $w_i$ is weight of known images from training set and C is covariance between the weights, {i=1, 2 … *m*}

.

Then minimum Mahalanobis distance $D_{min}$ can be defined as.

$$D_{min}= \min \{d_1, d_2 \ldots d_m\}.$$

Threshold distance $D_{th}$ can be defined as largest distance between any two face images on the feature space.

Hence for recognition process, test image x can be classified to the class which nearest sample belongs to only if $D_{min}$ is less than threshold distance, else test image is not a face image or cannot be recognized using this face database.

### 7.3. k-NN Classifier

K-Nearest Neighbor classifier [30] [31] is an algorithm for classifying objects based on closest training samples in the feature space. K-NN is a type of instance based learning where function is only approximated locally and all computation is deferred until classification [31]. Also this classifier is amongst the most used classification algorithm in pattern recognition; it can be implemented as follows.

For a given face database of *m* face images $\{x_1, x_2, x_3 \ldots x_m\}$ of size *N* by *N* whereby each face image belong to a known class $\{X_1, X_2, X_3 \ldots, Xc\}$ , where c is number of classes and test face image $x \in R^N$ on the feature space.

Distance function used was Euclidean distance between weight of the test image $w_x$ and the weight of the training samples $w_{xi}$ on the feature space, where {i=1, 2 … *m*} and *m* is number of images in the training set.

$$E_i^2 = (w_x - w_{xi})^2$$

Test image (unknown image) is classified by majority vote of its neighbors with the test image being assigned to the class most common amongst its k nearest neighbors measured, whereby k>1 and the value assigned was odd number which was greater than number of samples for each class in the training set to avoid tied votes.

## 8. Recognition Process

This section provides details on how unknown face images can be recognized from the feature space using distance classifiers; Euclidean and Mahalanobis distance functions and K-Nearest Neighbor classifier. Face images are randomly distributed on the feature space.

### 8.1. Recognition using Distance classifiers

Recognition process using distance classifiers; Euclidean and Mahalanobis distance functions, both of these distance functions were applied in the similar way to classify unknown images, where recognition was achieved by classifying test image to class which closest individual belongs, whereby minimum distance should be less than threshold distance $\mathbf{E_T}$ for image to be close to the face space and identified using stored face database. Two cases for recognition process using distance classifiers can be explained below.

### 8.1.1. When test image is a face image

In this case a test face image can be recognized if that individual is contained in the database, else image won't be recognized. For test image to be recognized minimum distance should be less than threshold distance, otherwise that face image is not contained in the database or is not a face image. Minimum distance being less than threshold distance means test image (input image) will be sufficiently close to the face space for the classifier to classify to corresponding class. Figure below show demonstration of this case, where input image is classified to class which closest individual belongs on feature space.

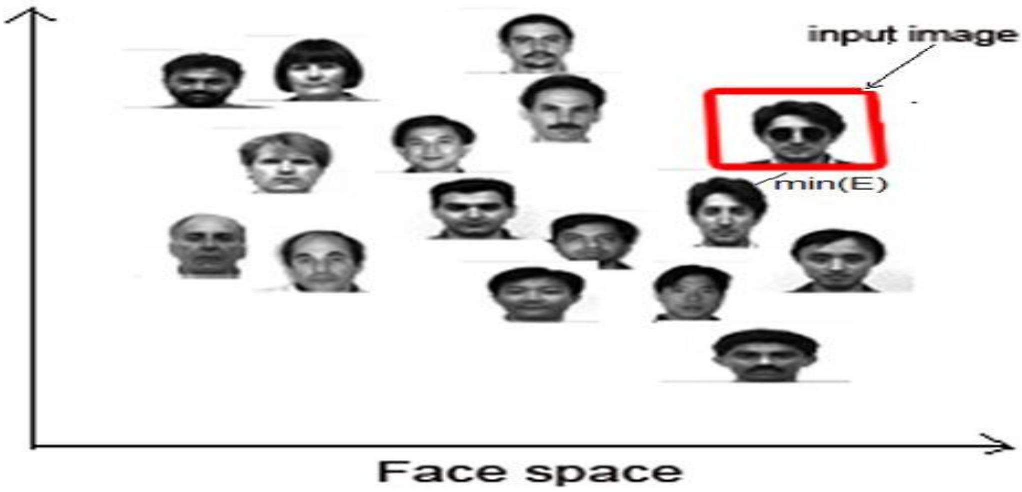



**Figure 10**: Recognition of the input face image which contained in database.

From the diagram above image closest to an input image has a minimum distance defined as **min (E),** which is then compared with threshold distance $\mathbf{E_T}$.

**Min (E) <** $\mathbf{E_T}$; hence input image can be matched to individual with minimum distance.

### 8.1.2. When test image is not face image

In this case an input image (test image) cannot be recognized, since the image doesn't have a face that can be matched using stored face database. For this condition minimum distance will be greater than threshold distance $\mathbf{E_T}$, since image will be projected further from the face space. Figure below show demonstration of recognition process when test image is not a face image.

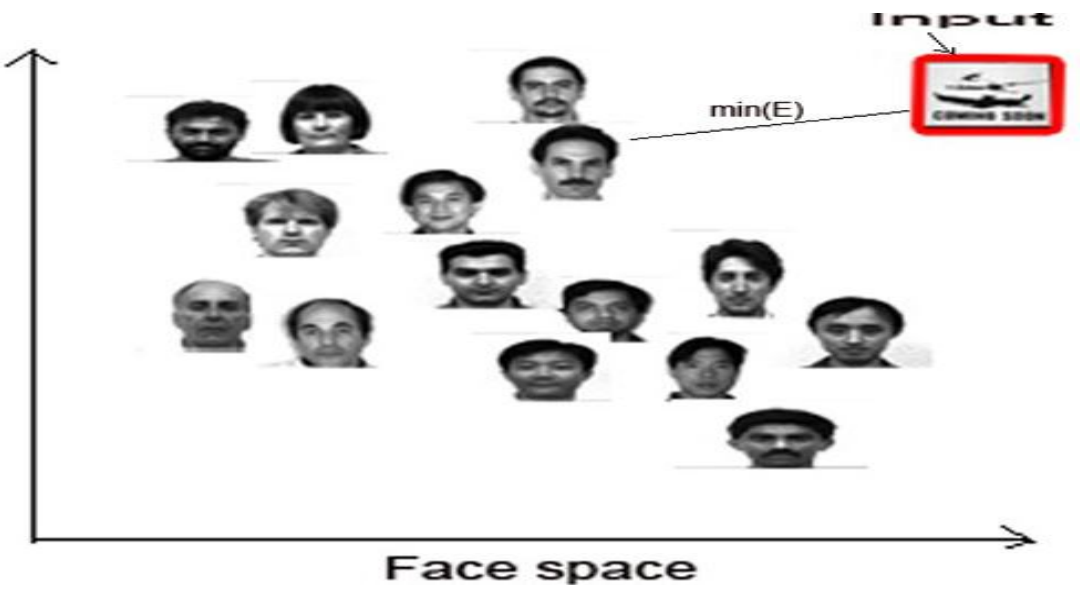



**Figure 11**: Recognition process when input image is not a face image.

From the diagram above, it shows that input image is projected further from the face space and minimum distance obtained from this part is greater than threshold distance, hence image can't be recognized.

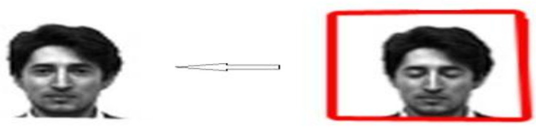
**Min (E) > $\mathbf{E_T}$**; hence image cannot be identified from this database.

## 8.2. Recognition using k-NN classifier

This part explains how k-NN classifier works to classify unknown face images using *k* nearest neighbours. In this case test images were classified by majority votes of its neighbours with test image being assigned to class most common amongst its *k*-nearest neighbours measured, where *k*>1. Number of neighbours used to classify was an odd number slightly greater than number of samples of each class in the training set to avoid tied votes. Figure below shows demonstration of how k-NN classifier can be used in the face recognition process, for this case 5 nearest neighbours were used to classify test image using majority votes.


**Figure 12** : Recognition process using K-NN classifier.

From the circle on the figure above cover 5 nearest neighbours which can be used to classify a test face image by majority votes, whereby image with red stroke is a test image. Hence test image will be classified to the class with three individuals in the circle.



## 9. Experiment

This section provides details of the how experiments were carried out and tools used to create simulation of a face recognition system and the main stages involved during the process. Tool used to program a recognition system was MATLAB R2012a programming language which included all four main stages to meet project specifications. Four main stages involved in this project were; face database creation, Image pre-processing stage, feature extraction and classification process. For image pre-processing stage, three different techniques were applied; first method was Convolution methods which tend to enhance features, reduce noises and extract edges [26]. Convolution methods involved; smoothing face images using low pass filter, sharpening to reduce blur on the face images and edge detection to enhance the edges of face images. Second image pre-processing technique was Statistical methods which tend to apply transformations to face images intensity values in order to make brightness adjust constant for all face images [27]. Third technique in image pre-processing stage was method combinations which tend to combine multiple methods from Convolution and Statistical methods to produce best improvement in accuracy rate. Summary of the image pre-processing stage is shown in the table [figure 13] below. Third stage was features extraction which included two algorithms; Eigenface and Fisherface algorithms. Features extracted by Eigenface algorithm were generated using mathematical procedure called Principal Component Analysis (PCA), which tend to obtain best eigenvectors with the highest eigenvalues which account for maximum discrimination between face images and maximizing total scatter of all projected face images onto feature space in low dimension [10]. Features extracted by Fisherface algorithm were generated by first applying PCA which reduces a given image space to new subspace in lower dimensional, then Fisher Linear Discriminant Analysis (FLDA) was applied to find linear projection which further reduces dimensional of a subspace and maximizes total scatter between each class while minimizes total scatter within each class [21] [22]. Summary of features extraction stage is shown in the table [figure 14] below. Fourth stage was classification process; three different approaches were implemented to classify test face images; Euclidean distance, Mahalanobis distance and K-Nearest Neighbour classifier. Summary of classification stage can be shown in the table [figure 15] below. Two experiments sets were carried out in this project; first experiment based on Yale face database which consists of face images with large variation in lightning conditions and facial expressions and other experiment was based on AT&T face database which consists of face images with large variation in viewing direction and facial expressions.

| Image Pre-processing | Convolution Methods | Statistical Methods | Method combinations |
|---|---|---|---|
| | Sharpening | Brightness | Convolution methods |
| | Edge detection | Local Brightness | Statistical methods |
| | Filtering/Smoothing | | |

**Figure 13:** Table of summary of image pre-processing stage.

| | |
|---|---|
| **Features Extraction** | Eigenface |
| | Fisherface |

**Figure 14**: Table of summary of feature extraction algorithms; Eigenface and Fisherface.

| | | | |
|---|---|---|---|
| **Classification** | K-NN classifier | | |
| | Distance classifiers | Euclidean distance | Mahalanobis distance |

**Figure 15**: Table of summary of classification stage.

## 10. Results and Discussions

This section provides an assessment of significance and reliability of results based on two experiments sets carried out; first experiment set was based on Yale face database and the other one was based on AT&T face database. Both face databases were experimented using different combinations of approaches from image pre-processing techniques, feature extraction methods and classification methods. Experiment based on Yale face database provided assessment of performance of face recognition system under variation in lightning conditions and different facial expressions while for the experiment based on AT&T face database provided assessment of performance of face recognition system under variation in viewing directions and different facial expressions. Both experimental results had two cases; with and without applying image pre-processing methods prior to recognition process. All the experimental results based on two databases are presented and discussed as follows.

## 10.1. Results based on Yale face database

This part provides results on the experiment based on Yale face database with both case; with and without applying image pre-processing techniques prior to recognition process. Yale face database assesses performance of face recognition system under variations in lightning conditions and facial expressions. In this experiment set, four different group of experiments were carried out including one without applying image pre-processing techniques and other three groups using three different image pre-processing methods; Convolution methods, Statistical methods and method combinations. Yale face database contained 165 face images, whereby each of 15 individual had 11 samples under different conditions. For each experiment, three combinations were used to randomly select images from database for training and testing sets; first combination was 7 images from each individual, totally 105 images were used as training set while remaining 60 images were used as testing set. Second combination was 6 images from each class, totally 90 images were used as training set while remaining 75 images were used as testing set and last combination was 5 images from each class, totally 75 images were used as training set while remaining 90 images were used as testing set. Experiments sets were repeated 20 times for each combination to obtain mean and standard deviation of each accuracy rate. For Eigenface algorithm, dimensionality of face space was set to be 75 while for Fisherface algorithm, dimensionality of face space was reduced to 14.

### 10.1.1. Without Image pre-processing methods

This part provides experimental results based on three different sets of experiments carried out without applying image pre-processing methods, whereby each set was carried out with each of the feature extraction approaches; Eigenface and Fisherface algorithms and all three classification methods; Euclidean distance, Mahalanobis distance and K-NN classifier. Summary of experimental results are shown in the table figure 16 and graph figure 17 below. From those results, it is observed that best three performances with the highest recognition rates were achieved by using; K-NN classifier to classify features extracted by Fisherface algorithm with mean of 86.83% accuracy rate and standard deviation of 11.04, second best performance was achieved by using K-NN classifier to classify features extracted by Eigenface algorithm with average of 83.37% accuracy rate and standard deviation of 11.46. Third best performance was obtained by using Mahalanobis distance function and Fisherface algorithm with mean of 83.17% and standard deviation of 11.47%.

| Yale face database | Accuracy rates % | | | |
|---|---|---|---|---|
| | Eigenface | | Fisherface | |
| | mean | std | mean | std |
| Euclidean distance | 79.23 | 9.86 | 80.23 | 10.16 |
| Mahalanobis distance | 80.67 | 10.16 | 83.17 | 11.47 |
| K-NN classifier | 83.37 | 11.46 | 86.83 | 11.04 |

**Figure 16**: Table of results without using image pre-processing on Yale face database.

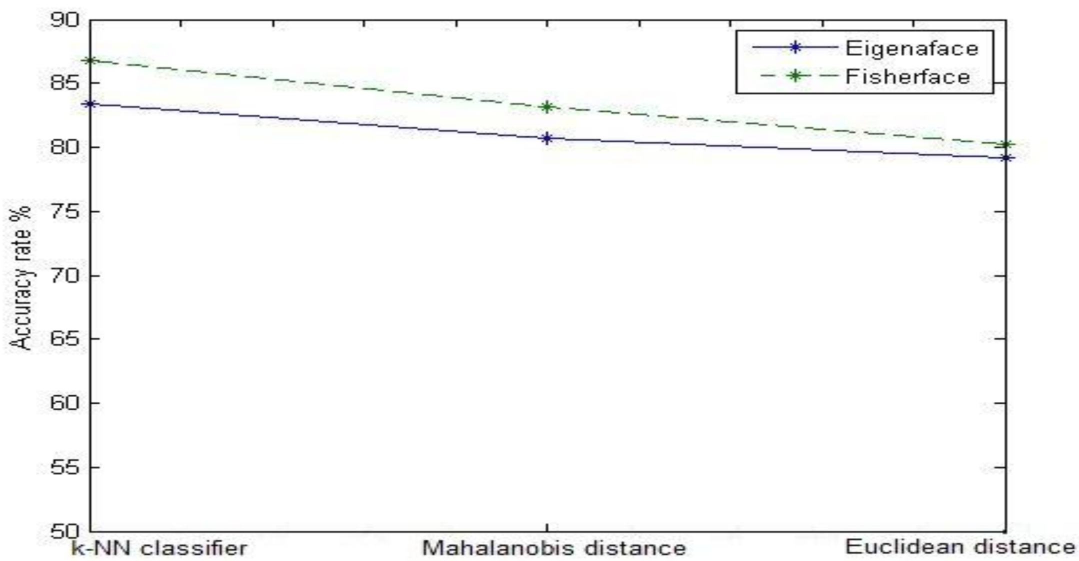



**Figure 17:** Graph of results with three classification methods; Euclidean, Mahalanobis and KNN.

### 10.1.2. With Convolution Methods

This part provides results of the experiments carried out with application of Convolution methods in the image pre-processing stage. Three sets of experiments were carried out including both of the features extraction algorithms; Eigenface and Fisherface with all three classification methods; K-NN classifier, Euclidean and Mahalanobis distance functions. Summary of these experimental results are shown in the table figure 18 and graph figure 19 below. From these results, best three performances with the highest recognition rates were achieved by using; K-nearest neighbour classifier to classify features extracted by Fisherface algorithm with average of 87.91% and standard deviation of 7.01, second best performance was obtained by using K-nearest neighbour classifier and Eigenface algorithm with average of 84.56% and standard deviation of 6.93 and the third best performance was achieved by using Mahalabonis distance function to classify features extracted by Fisherface algorithm with average of 83.61% and standard deviation of 6.73.

| Yale face database | Accuracy rates % | | | |
|---|---|---|---|---|
| | Eigenface | | Fisherface | |
| | mean | std | mean | std |
| Euclidean distance | 79.37 | 6.71 | 81.17 | 5.77 |
| Mahalanobis distance | 81.28 | 7.21 | 83.61 | 6.73 |
| K-NN classifier | 84.56 | 6.93 | 87.91 | 7.01 |

**Figure 18**: Table of results with Convolution methods on Yale face database.

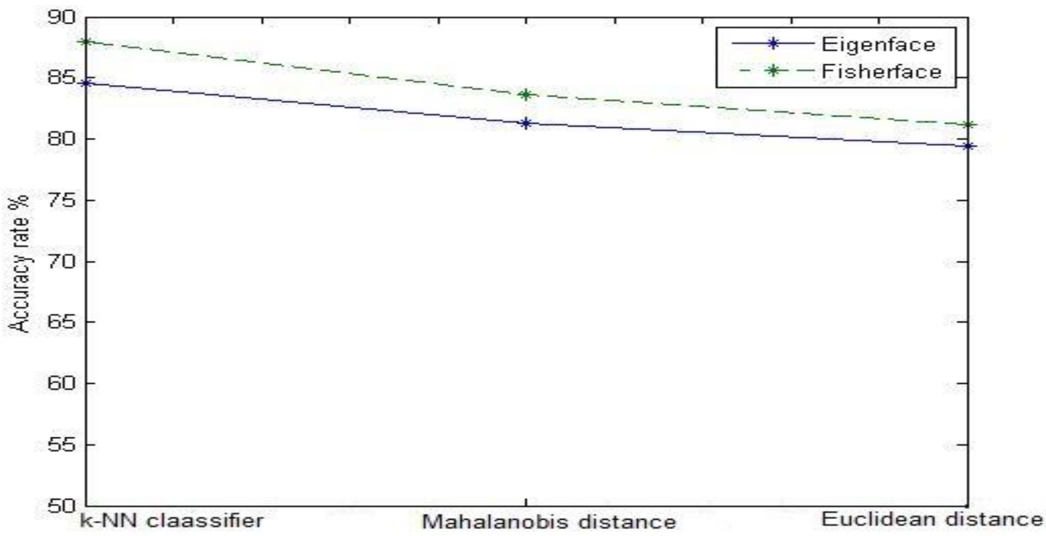



**Figure 19**: Graph of results with three classification methods; Euclidean, Mahalanobis and KNN.

### 10.1.3. With Statistical Methods

This part provides results of the experiments sets carried out with application of Statistical methods in the image pre-processing stage. Statistical methods included; brightness and local brightness transformations which tend to compensate variation in lightning conditions across the regions of the face [26]. Three sets of experiments were carried out including both of feature extraction algorithms; Eigenface and Fisherface with all three classification methods, Euclidean distance, Mahalanobis distance and k-NN classifier, where value for k was assigned to be odd number greater than number of samples in each class in the training set to avoid tied majority votes. Summary of experimental results are shown in the table figure 20 and graph figure 21 below. From those results, best three performances with the highest recognition rate were achieved using; k-NN classifier to classify features extracted by Fisherface algorithm with average of 87.03% accuracy rate and standard deviation of 7.01, second best performance was achieved by using k-NN classifier and Eigenface algorithm with average of 85.12% and standard deviation of 6.48 and third best performance was achieved by using Mahalabonis distance function and Fisherface algorithm with average of 83.77% and standard deviation of 7.42.

| **Yale face database** | Accuracy rates % | | | |
|---|---|---|---|---|
| | Eigenface | | Fisherface | |
| | mean | std | mean | std |
| Euclidean distance | 81.37 | 5.85 | 80.64 | 6.33 |
| Mahalanobis distance | 83.00 | 6.75 | 83.77 | 7.42 |
| K-NN classifier | 85.12 | 6.48 | 87.03 | 7.01 |

**Figure 20:** Table of results with Statistical methods.

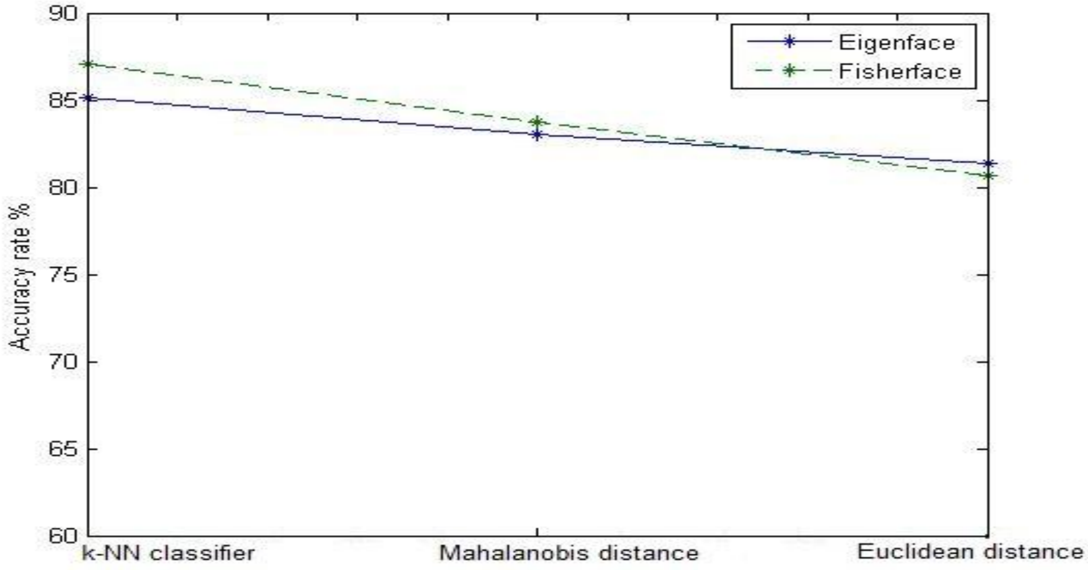



**Figure 21:** Graph of results with three classification methods; Euclidean, Mahalanobis and KNN.

#### 10.1.4. With Method Combinations

This part provides results of the experiments carried out with application of combined methods in the image pre-processing stage. This image pre-processing technique involved combination of multiple methods which produce best improvement in accuracy rate, combined methods included; sharpening, local brightness and smoothing, whereby face images were sharpen followed by local brightness transformations, followed by smoothing/filtering. Three sets of experiments were implemented using both feature extraction methods; Eigenface and Fisherface with all three classification methods; K-NN classifier, Euclidean and Mahalanobis distance functions. Summary of these experimental results are shown in the table figure 22 and graph figure 23 below. From these results, best three performances with highest accuracy rates were achieved using; K-nearest neighbour classifier and Fisherface algorithm with average of 90.07% and standard deviation of 3.13, second best performance was achieved by using K-nearest neighbour classifier and Eigenaface algorithm with average of 88.63% and standard deviation of 1.74, third best performance was achieved by using Mahalanobis distance function to classify features extracted by Fisherface algorithm with average of 87.04% and standard deviation of 3.47.

| **Yale face database** | Accuracy rates % | | | |
|---|---|---|---|---|
| | Eigenface | | Fisherface | |
| | mean | std | mean | std |
| Euclidean distance | 84.00 | 3.87 | 86.77 | 2.47 |
| Mahalanobis distance | 86.14 | 4.58 | 87.04 | 3.47 |
| K-NN classifier | 88.63 | 1.14 | 90.07 | 2.13 |

**Figure 22:** Table of results with method combinations on Yale face database.

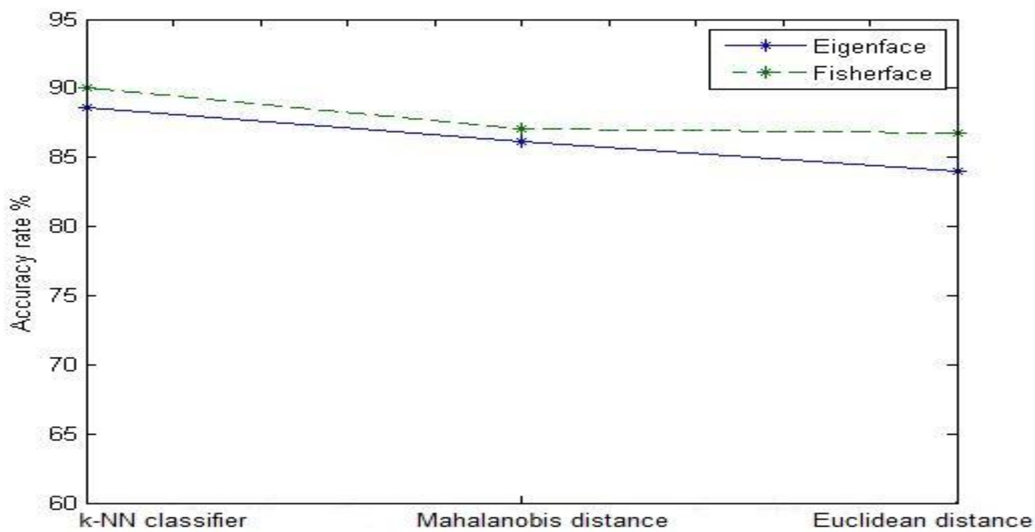



**Figure 23**: Graph of results with three classification methods; Euclidean, Mahalanobis and KNN.

### 10.1.5. Summary

From all four sets of experimental results based on Yale face database, it can be concluded that Fisherface algorithm provided better recognition rate compare to Eigenaface algorithm in the most of the experiments, whereby the optimal performance for this database was achieved by using Fisherface algorithm and k-NN classifier with average of 90.07% and standard deviation of 3.13. From comparison of the results of experiments carried out without and with image pre-processing stage, can be noticed that application of image pre-processing enhanced performance of the face recognition system with more stable and reliable accuracy rates with lower standard deviation under different experiments with different conditions of the test images. Also it is noticed that for both algorithms; Eigenface and Fisherface, they provided better performance in the testing images with large variation in facial expression than to variation in lightning conditions, since variations in lightning conditions tend to suppress features on the face images which makes it difficult to recognize the face. This problem was reduced by using image pre-processing techniques; Statistical methods which apply brightness transformations across the regions of the face to compensate variations in lightning conditions. Also from these experimental results It can be suggested that, face recognition system perform better when more samples for each individual located in training set.

## 10.2. Results based on AT&T face database

This section provides results of the experiments sets carried out based on AT&T face database with both case; with and without applying image pre-processing techniques prior to recognition process. AT&T face database contained face images with large variations in viewing direction and facial expressions. In this experiment set, four groups of experiments were carried out based on this face database; one was without image pre-processing and other three with image pre-processing; Convolutions methods, Statistical methods and method combinations. AT&T face database consists of 400 face images, whereby each of 40 individual contained 10 samples under different conditions to study strength and weakness of the system. For each experiment carried out, three combinations were used to randomly select images from database to form training and testing sets; first combination used 7 samples from each individual, totally 280 images for training set while remaining 120 images for testing set, second combination used 6 samples from each class, totally 240 images for training set while remaining 160 images used for testing set, third combination used 4 samples from each class, totally 160 images for training set while remaining 240 used for testing set. Each combination was repeated 20 times to obtain average and standard deviation of each accuracy rate. For Eigenface algorithm, dimensionality of the face space was reduced to 75 while for Fisherface algorithm, dimensionality of face space was further reduced to 39.

### 10.2.1. Without Image pre-processing methods

This part provides results based on three sets of experiments carried out without applying image pre-processing techniques. Each set of the experiment was carried out with both feature extraction approaches; Eigenface and Fisherface with all three classification methods; K-NN, Euclidean and Mahalanobis distance functions. Summary of the experimental results are shown in table figure 24 and graph figure 25 below. From those results, the best three performances were achieved by using; Mahalanobis distance function to classify features extracted by Eigenface algorithm with average of 93.87% and standard deviation of 1.57, second best performance was achieved by using Euclidean distance and Eigenface algorithm with average of 91.61% accuracy rate and standard deviation of 2.37, third best performance was obtained by using Fisherface algorithm and Mahalanobis distance function with average of 89.48% and standard deviation of 3.97.

| AT&T face database | Accuracy rates % | | | |
|---|---|---|---|---|
| | Eigenface | | Fisherface | |
| | mean | std | mean | std |
| Euclidean distance | 91.61 | 2.37 | 88.03 | 3.44 |
| Mahalanobis distance | 93.87 | 1.57 | 89.48 | 3.97 |
| K-NN classifier | 77.67 | 6.58 | 74.41 | 5.97 |

**Figure 24** : Table of results without image pre-processing on AT&T face database.

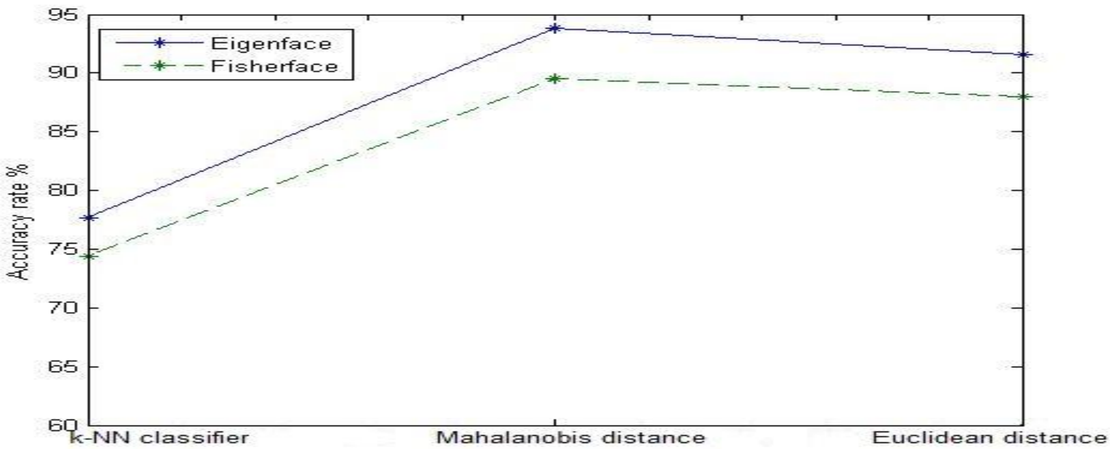



**Figure 25**: Graph of results with three classification methods; Euclidean, Mahalanobis and KNN**.**

### 10.2.2. With Convolution methods

This part provides results of the experiments carried out with application of Convolution methods in the image pre-processing stage. Three sets of experiments were carried out including both feature extraction algorithms; Eigenface and Fisherface with all three classification methods; K-Nearest Neighbour classifier, Euclidean and Mahalanobis distance functions. Summary of these experimental results are shown in the table figure 26 and graph figure 27 below. From those results, best three performances with the highest recognition rate were achieved by using Eigenface algorithm and Mahalanobis distance classifier with average of 93.01% recognition rate and standard deviation of 3.83, second best performance was obtained by using Eigenface algorithm and Euclidean distance classifier which achieved average of 91.98% accuracy rate and standard deviation of 2.17, third best performance was achieved by using Euclidean distance to classify features extracted by Fisherface algorithm with average of 88.54% and standard deviation of 3.14. From these results it can be noticed slight improvement of recognition rate with lower standard deviation compare to the experiments carried out without image pre-processing methods.

| **AT&T face database** | Accuracy rates % | | | |
|---|---|---|---|---|
| | Eigenface | | Fisherface | |
| | mean | std | mean | std |
| Euclidean distance | 91.98 | 2.17 | 88.54 | 3.14 |
| Mahalanobis distance | 93.01 | 3.83 | 88.04 | 3.11 |
| K-NN classifier | 78.23 | 4.41 | 74.87 | 6.01 |

**Figure 26:** Table of results with Convolution methods on AT&T face database.

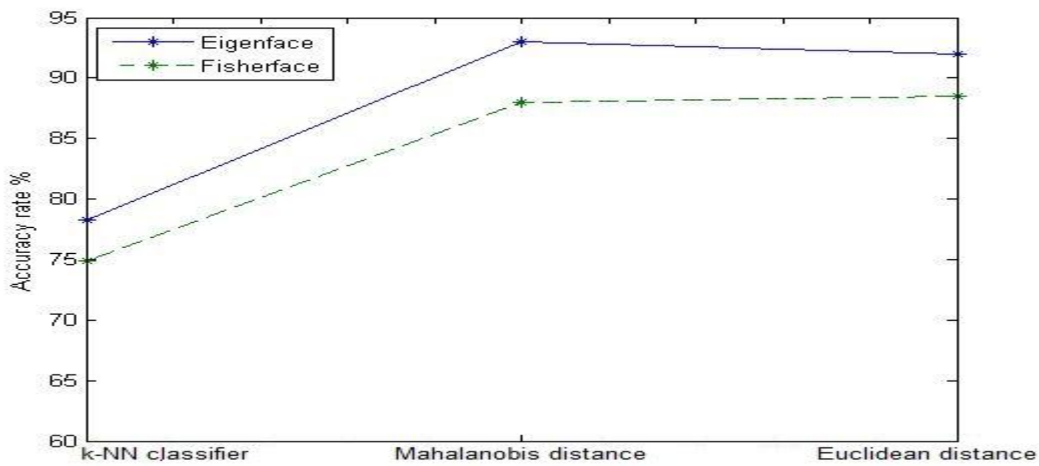



**Figure 27:** Graph of results with three classification methods; Euclidean, Mahalanobis and KNN.

### 10.2.3. With Statistical methods

This part provides results of the experiments carried out with application of Statistical methods in the image pre-processing stage. Statistical methods include; brightness and local brightness transformations. Three sets of experiments were implemented using two feature extraction algorithms; Eigenface and Fisherface with all three classification methods; K-Nearest Neighbour classifier, Euclidean and Mahalanobis distance functions. Summary of these experimental results can be shown in the table figure 28 and graph figure 29 below. From these results, best three performances with the highest accuracy rate were achieved using; Mahalanobis distance function and Eigenface algorithm which had average of 92.97% and standard deviation of 4.37, second best performance was obtained from the experiment included Euclidean distance as classifier and Eigenface algorithm for features extraction with average of 91.83% accuracy rate and standard deviation of 1.98, third best performance was achieved by using Fisherface algorithm and Euclidean distance with average of 89.61% and standard deviation of 2.48.

| **AT&T face database** | Accuracy rates % | | | |
|---|---|---|---|---|
| | Eigenface | | Fisherface | |
| | mean | std | mean | std |
| Euclidean distance | 91.83 | 1.98 | 89.61 | 2.48 |
| Mahalanobis distance | 92.97 | 4.37 | 89.06 | 3.81 |
| K-NN classifier | 77.91 | 5.01 | 74.21 | 4.67 |

**Figure 28**: Table of results with Statistical methods on AT&T face database.

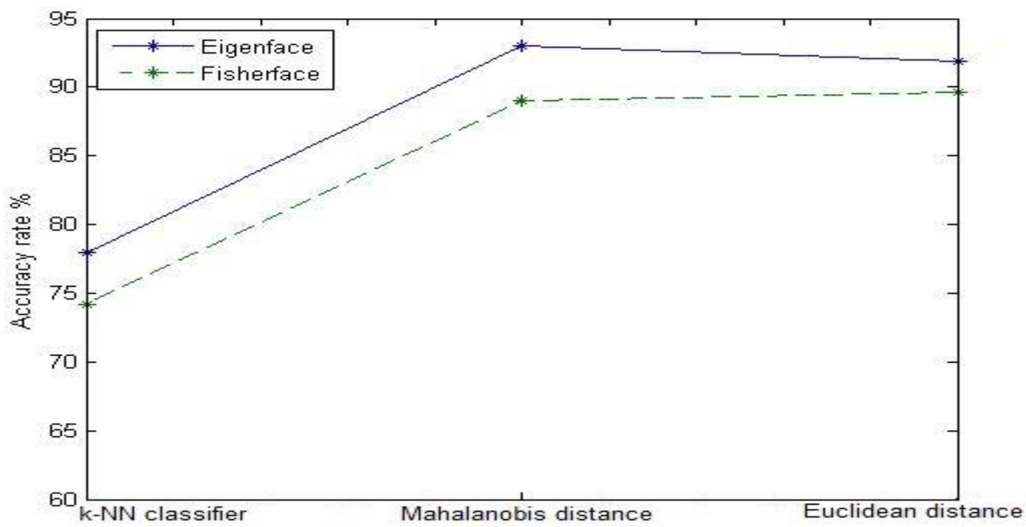



**Figure 29:** Graph of results with three classification methods; Euclidean, Mahalanobis and KNN.

#### 10.2.4. With Method Combinations

This part provides results of the experiment group carried out with application of method combinations of image pre-processing techniques. This method involved combination of multiple of methods used in Convolution and Statistical methods to produce best improvement in accuracy rate of face recognition system, methods combined included; sharpening followed by local brightness transformations, followed by filtering. Three sets of experiments were implemented using two features extraction approaches; Eigenface and Fisherface algorithms with all three classification methods; K-Nearest Neighbour classifier (k-NN), Euclidean and Mahalanobis distance functions. Summary of these experimental results are shown in the table figure 30 and graph figure 31 below. From these results, best three performances with the highest recognition rates were achieved by using; Mahalanobis distance functions to classify features extracted by Eigenface algorithm with average of 94.01% and standard deviation of 2.47, second best performance as achieved by using Euclidean distance and Eigenface algorithm with average of 92.77% accuracy rate and standard deviation of 1.64 and the third best performance was achieved by using Euclidean distance and Fisherface algorithm with average of 90.65% and standard deviation of 3.14.

| **AT&T face database** | Accuracy rates % | | | |
|---|---|---|---|---|
| | Eigenface | | Fisherface | |
| | mean | std | mean | std |
| Euclidean distance | 92.77 | 1.64 | 90.63 | 3.14 |
| Mahalanobis distance | 94.01 | 2.47 | 90.01 | 3.58 |
| K-NN classifier | 78.01 | 1.93 | 76.18 | 2.57 |

**Figure 30:** Table of results with method combinations on AT&T face database.

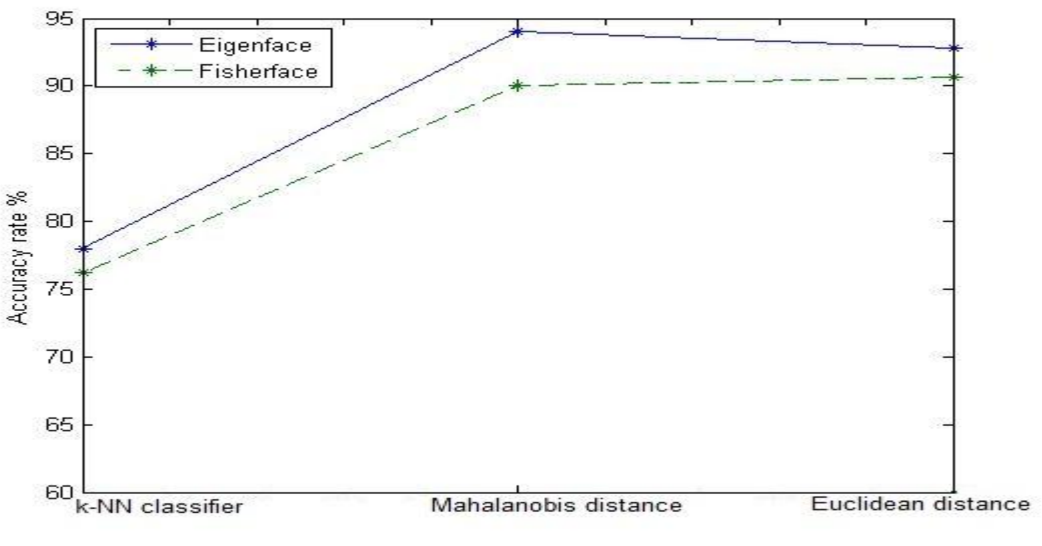



**Figure 31**: Graph of results with three classification methods; Euclidean, Mahalanobis and KNN.

#### 10.2.5. Summary

From all four sets of experimental results based on AT&T face database, it can be concluded that, Eigenface algorithm with distance classifiers provided better recognition rates than Fisherface algorithm, whereby optimal performance was achieved by using Eigenface algorithm and Mahalanobis distance function with average of 94.01% accuracy rate and standard deviation of 2.47, including image pre-processing stage prior to recognition process. Though image pre-processing methods seemed to produce more stable results with lower standard deviations, there have been small significance improvements of accuracy rates under different experiments based on AT&T face database, this is due to image pre-processing methods applied  improve face images with large variation in lightning conditions by compensating variation in lightning conditions across the regions of the face image, while AT&T face database contained only face images with large variation in viewing direction and facial expressions. One suggestion for this problem can be use of knowledge of 3-D geometry to compensate viewing angle for the face images having viewing direction with more than 20 degree viewing angle from which face recognition cannot produce positive results. Also from these experimental results can be noticed that, for the most experiments carried out based AT&T face database, k-NN classifier provided lower accuracy rates with average of 75.04%, there is one suggestion that may improve this classification method, varying the value of nearest neighbours to obtain the optimal performance. Generally, both of the face recognition algorithms; Eigenface and Fisherface approaches, provided reasonable high performances in this face database without image pre-processing stage, this suggest face recognition system can perform better on face images with full frontal faces and 20 degrees off in viewing directions.

### 10.3. Limitations of Face Recognition System

Though Eigenface and Fisherface algorithms were successful in the face recognition process, they have limitations which degrade the recognition rates. Followings are discussion of limitations and alternatives to reduce limitations in order to improve recognition system.

#### 10.3.1. Image Background

Face images with background features can affect the recognition performance, since Eigen faces and Fisherface algorithms can't distinguish a face from the background. Hence background should be eliminated before the image is inserted in the database in the system; this can be achieved or reduced by using 2-dimensional Gaussian centred at the face to reduce intensity of background.

### 10.3.2. Size of the face

Size of the face images also alter recognition performance since size of images must be close to the size of Eigenfaces and Fisherfaces for system to work better. This problem can be resolved by training face images using multi scales, and then use Eigenfaces and Fisherfaces at different scale to estimate size.

### 10.3.3. Head Orientation/Viewing Angle

Face recognition system works better at full frontal face images and 20 degrees off in viewing directions. Large viewing angle may degrade performance of the recognition process, since it is difficult for algorithm to extract significant features which account for maximum variations between face images. There is one suggestion that may improve recognition system under this condition, application of knowledge of 3-D geometry to determine tilt and rotate to standard orientation from which face recognition system can perform better.

### 10.3.4. Variations in Lightning Conditions

Variations in lightning conditions on the face images tend to create shadows on the face and suppress significance face features which account for variations between face images, this degrade performance of the recognition system. One alternative for this limitation is to introduce image pre-processing techniques to compensate variations of lightning conditions across the regions of the face, one of the image pre-processing methods which can be used is Statistical methods, this apply local brightness transformations on face images by adjusting the brightness constantly for all images.

### 10.3.5. Other Issues

Other limitations of face recognition system include; poor lightning conditions, poor resolutions of images, sunglasses, long hair. These conditions suppress significance features which account for variations between face images, hence lead to the mismatching of the images and difficult in learning classes for recognition algorithms. Alternatives for these limitations could be; improving resolutions of the images, introducing different image pre-processing methods, avoiding objects blocking the face image. From experiments carried out based on both databases; Yale and AT&T databases it can be suggested that, opting for bigger face database, whereby each class will contain more number of samples in the training set, may lead to better accuracy rates, though this may be good solution, it tend to increase memory cost which imply that, for the face recognition system huge memory location should be allocated for better performances. Other challenging part in this project was minimal knowledge of MATLAB language which limited trying more complex and better feature extraction approaches and different classification methods which could provide better learning of the classes.

## 11. Conclusion

This project involved designing of the system that can automatically identify a person from the image source using stored face database. The aims and objectives of the project specifications were achieved through four stages; face database creation, Image pre-processing, Feature extraction stage and classification process. Image pre-processing stage is optional for this system; was used for critical evaluations to enhance performances of recognition rates by eliminating and reducing some of the limitations of face recognition system. Three different image pre-processing techniques were implemented; Convolution methods, Statistical methods and combination of these two methods. For feature extraction stage, two algorithms were implemented; Fisherface and Eigenface algorithms, while for classification process, three classifiers were used; k-Nearest Neighbour classifier, Euclidean and Mahalanobis distance functions. Two experiments sets were carried out to study strengths and limitations of face recognition system; one experiment set was based on Yale face database which contained face images with large variation in lightning conditions and facial expressions, while another experiment was based on AT&T face database which contained face images with large variation in viewing directions and facial expressions. Several conclusions can be observed on these two experiments sets; for the experiment based on Yale face database, optimal performance was achieved by using k-NN classifier and Fisherface algorithm with average of 90.07% accuracy rate and standard deviation of 2.13, with image pre-processing stage prior to recognition process, this suggest that application of image pre-processing methods can yield to a better performances of the system by compensating some of the limitations e.g. variation in lightning conditions was compensate by using Statistical methods (Brightness transformations). For the experiment set based on AT&T face database, optimal performance was achieved by using Mahalanobis distance and Eigenface algorithm with average of 94.01% and standard deviation of 2.47. From this experiment can be concluded that; even without image pre-processing stage, system could perform well with reasonable optimal performances since the face database contained only face images with large variation in viewing directions and facial expressions from which algorithms can perform better without any critical evaluations. Though face recognition system was successfully implemented on these two databases, it still need to be improved to acquire more accuracy rates since this system is commonly used technology in critical areas e.g. security in today's technology. Suggestions for improvement for this system are explained below in the futures plan section.

## 11.1. Future Works

Though the main objectives of this project were highly achieved, still further improvements can be done to increase efficiency and study more strengths and limitations of this system. Future plans for this project are as follows.

- Experimental testing of feature extraction approaches; Eigenface and Fisherface algorithms, with another databases containing face images under different conditions to observe strengths and weakness of these algorithms.

- As for recognition systems, better classifications methods may improve the performance of the system, it can be suggested to use other different classification approaches; Neural network, Support Vector Machine (SVM) which can provide better learning of the classes.

- Trying different face recognition algorithm; 3-D face recognition which is able to compare surfaces independent of natural deformations resulting from facial expressions and compensating viewing angle of the head. Another approach may be Kernel methods which are more of generalized linear methods; this can be improvement to Eigenface and Fisherface algorithms.

- Linking together face detection and recognition systems to allow better performance under different conditions e.g. both of these systems can be used to identify human faces even from the image contain multiple faces on the same picture and also the system will be able to detect human face from any background feature before it can recognize.

## 12. References

[1] P. Belhumeur, J. Hespanha and D. Kriegman (1997), "Eigen faces vs Fisher faces : using class specific linear projection*", IEEE Transactions on pattern analysis and machine interrlligence* 19(7): 711, 1997.

[2] W. Zhao, R. Chellapa, A. Rosenfeld and P.J. Phillips," Face recognition: A literature survey", *ACM computing surveys*, pp. 3399-458, 2003.

[3] L. Sirovich and M. Kirby," Low dimensional procedure for the characterization of human face, "*Journal of the optical society of America*, A 4(3): 519-524, 1987.

[4] M. Turk and A. Pentland (1991), "Face recognition using Eigen faces", *Proc. IEEE conference on computer vision and pattern recognition,* pp. 586-591, 1991.

[5] M. Turk and A. Pentland 1991, "Eigen faces for recognition", *Journal of cognitive Neuroscience* 3(1) : 71-86, 1991.

[6] R. Brunelli and T. Poggio, "Face recognition features versus Template", *IEEE Trans. On PAM, 15(10),* 1042-1052, 1993.

[7] Kranse, Mike, "Is face recognition just high tech snake oil", ISSN 1488-1756, 2011.

[8] Schultz , Zac, "Facial recognition technology helps DMV prevent identity theft", 2007.

[9] M. Kirby and L. Sirovich, "Application of karhunen Loeve Procedure for the characterization of Human faces", *IEEE transactions on pattern analysis and machine intelligence,* vol. 12, no.1, pp. 103-108, 1990.

[10] H. Moon and P.J Phillips, " Computational and performances aspects of PCA based face recognition Algorithms", perception, vol. 30, pp. 303-321, 2001.

[11] Blesdoe, W.W (1966a), "The model method in facial recognition, Panoramic Research Inc, Pato Alto , CA," PRI:15, 1966.

[12] Blesdoe (1966b), “ Man : Machine facial recognition, Panoramic Research Inc. , Palo Alto, Rep”, PRI: 22, 1966.

[13] Goldstein, Harmon and Lesk (1971), “ Identification of human face proceeding”, IEEE, 59, 748, 1971.

[14] Yale face database; *http://cvc.yale.edu/projects/yalefaces/yalefaces.html*. (Accessed on 03 October 2012).

[15] AT&T database *http;//www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html*

(Accessed on 04 October 2012).

[16] A. Pentland, B. Moghaddam and T. Storner, “ View based and Modular Eigen spaces for face recognition ”, *proceedings of IEEE conference on computer vision and pattern recognition,* pp. 84-91, 1994.

[17] L. Wang, Y. Zhang and J. Feng, “ On the Euclidean distance of images”, *IEEE Transactions on pattern analysis and machine Intelligence,* vol. 27, no. 8, pp. 1334-1339, 2005.

[18] Y. Ling, X. Yin and S.M. Bhandarkar, “Fisherface: recognition using class specific linear projection*”, International conference on Image processing*, vol. 2, pp. III, 885-888, 2003.

[19] Face recognition using Eigen faces, http://www.cs.princeton.edu/~cdecoro/eigenfaces/

(Accessed 14 October 2012).

[20]. F.R. Bach, M.I. Jordan, Kernel Independent Component Analysis, Journal of Machine Learning Research, Vol. 3, 2002, pp. 1-48.

[21]. R.A. Fisher, “The use of multiple measures in Taxanomic Problems”, Ann Eugenics, vol. 7, pp. 179-188, 1936.

[22]. K. Etemad, R. Chellapa, Discriminant Analysis for recognition of human face image, Journal of the optical society of America, vol. 14, No.8, August 1997, pp. 1724-1733.

[23]. A.M. Martinez, A. C. Kak, PCA versus LDA, IEEE Trans. *on Pattern Analysis and Machine Intelligence*/vol. 23, No. 2, 2001, pp. 228-233.

[24]. J. Lu, K. N. Plataniotis, A. N. Venetsanopoulis, Face recognition using LDA-Based Algorithms, IEEE Trans. on Neural Network vol. 14, No. 1, January 2003, pp. 195-200.

[25]. G. Finlayson, G. Schaefer, “*Hue that is Invariant to brightness and Gamma*”, BMVC01, Session 3: Color & Systems, 2001.

[26]. W. Zhao, R. Chellapa, “*3D Model enhanced face recognition*”, In Proc. Int. conf. Image processing, Vancouver, 2000.

[27]. Y. Adini, Y. Moses, S. Ullman, Face recognition: the problem of compensating for changes in illumination direction, Department of applied mathematics and computer science, The Weizmann Institute of science.

[28]. Mahalnobis, P. Chandra, "*On the generalized distance in statistics*", Proceedings of the National Institute of sciences of India 2, 49-55, 1936.

[29]. De Maesschalk, Roy, J. Rimband, Delphine, Massart, The mahalanobis distance, chemo metrics and Intelligence laboratory systems.50; 1-8, 2000.

[30]. Cover TM, Hart PE (1967), "*Nearest Neighbor pattern classification*", IEEE Transactions on Information Theory, 13, 21-27, 1967.

[31]. D. Coomas, D.L massart, "Alternative K- nearest neighbor rules in supervised pattern recognition: Part1: k-nearest neighbor classification by using alternative voting rules", Analytical Chi mica Acta, 15-27, 1982

# 13. Appendices

## 1. Project Specification Report- Revised

### A. Project Description and Methodology

Face recognition system is computer application that can be used to automatically identify a person from a still or video image source using stored face database in the system. Objectives of this project are to create a program that can be used to recognize face images from a given database using simulation in MATLAB program language. Face recognition involve four main stages; face database creation, Image pre-processing, features extraction and classification process. For features extraction stage, two algorithms were implemented; Eigenface and Fisherface approaches, for classification process, three classifiers were used; k-NN, Euclidean distance and Mahalanobis distance functions. Two experiments were carried out; one was based on Yale face database and the other one was based on AT&T face database.

### B. Projects Tasks

- Learning MATLAB implantation for this project.
- Face database creation.
- Image pre-processing; Convolution and Statistical methods.
- Implementation of features extraction algorithm; Eigenface and Fisherface.
- Classification process.
- Testing program.

### C. Project Deliverables

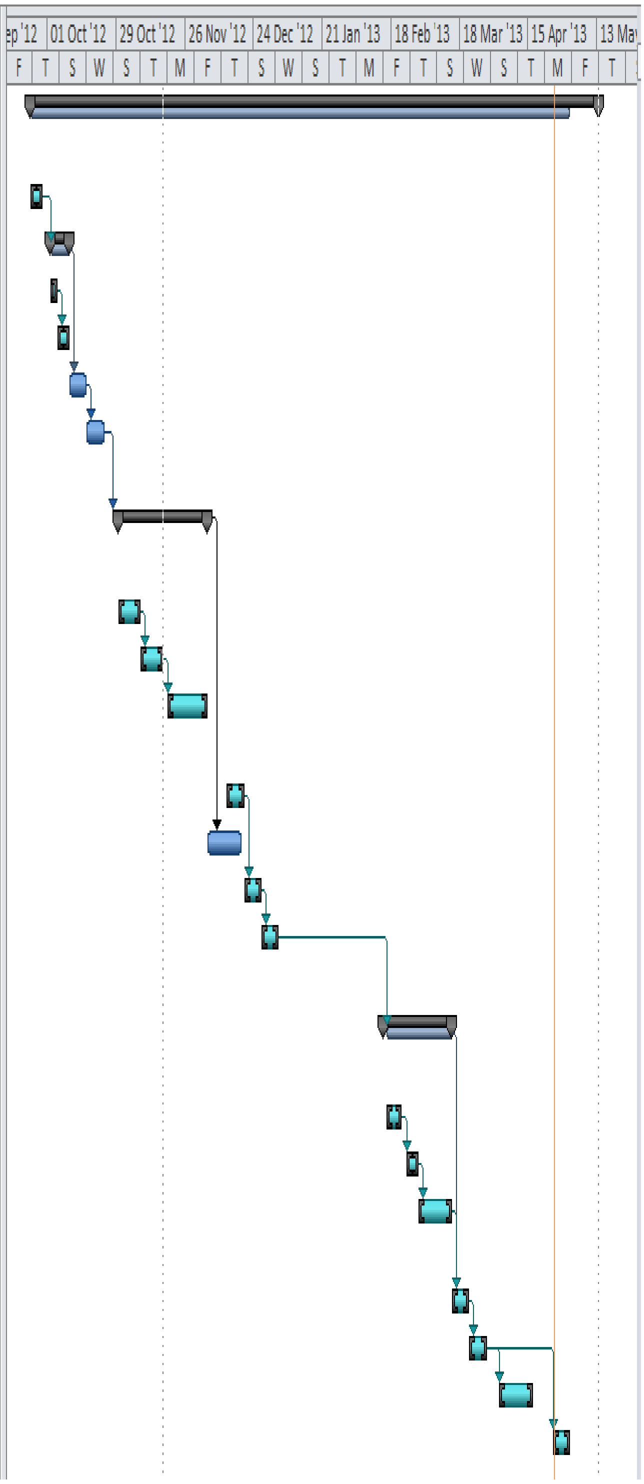
- Project specification report- Revised.
- Interim presentation.
- Interim report.
- Poster presentation/ Bench inspection.
- Summary of the report.
- Final thesis.

### D. Project Rationale and Industrial Relevance.

- This system can be used as embedded system in security cameras systems.
- Access control; using a face image as key.
- Can be used to identify criminals.
- Can be used to avoid fake ID fabrication.
- Illegal immigrant detection; passport.

## 2. Gantt Chart- Revised

| | Task Name | Duration | Start | Finish | Predecessors | Res |
|---|---|---|---|---|---|---|
| 1 | Project: Face Recognition | 166 days | Mon 24/09/12 | Mon 13/05/13 | | |
| 2 | Background Reading | 1 wk | Mon 24/09/12 | Fri 28/09/12 | | |
| 3 | Database Creation | 1.2 wks | Tue 02/10/12 | Tue 09/10/12 | 2 | |
| 4 | Yale database | 3 days | Tue 02/10/12 | Thu 04/10/12 | | |
| 5 | AT & T database | 3 days | Fri 05/10/12 | Tue 09/10/12 | 4 | |
| 6 | Learning Eigenfaces | 1 wk | Wed 10/10/12 | Tue 16/10/12 | 3 | |
| 7 | Program design based on Eigenfaces | 1 wk | Wed 17/10/12 | Tue 23/10/12 | 6 | |
| 8 | Eigenface Implementation | 26 days | Tue 30/10/12 | Tue 04/12/12 | 7 | |
| 9 | PCA | 7 days | Tue 30/10/12 | Wed 07/11/12 | | |
| 10 | Classification | 7 days | Thu 08/11/12 | Fri 16/11/12 | 9 | |
| 11 | Testing Eigenface Approach | 2.4 wks | Mon 19/11/12 | Tue 04/12/12 | 10 | |
| 12 | Interim Presentation | 1 wk | Thu 13/12/12 | Wed 19/12/12 | | |
| 13 | Interim Report | 2 wks | Wed 05/12/12 | Tue 18/12/12 | 8 | |
| 14 | Learning Fisherface | 1 wk | Thu 20/12/12 | Wed 26/12/12 | 12 | |
| 15 | Program design based on Fisherfaces | 1 wk | Thu 27/12/12 | Wed 02/01/13 | 14 | |
| 16 | Fisherface Implementation | 4 wks | Fri 15/02/13 | Thu 14/03/13 | 15 | |
| 17 | PCA and LDA | 1 wk | Sat 16/02/13 | Thu 21/02/13 | | |
| 18 | Classification | 1 wk | Sun 24/02/13 | Thu 28/02/13 | 17 | |
| 19 | Testing Fisherface approach | 2 wks | Fri 01/03/13 | Thu 14/03/13 | 18 | |
| 20 | Image Pre-processing | 1 wk | Fri 15/03/13 | Thu 21/03/13 | 19,16 | |
| 21 | Poster Presentation | 1 wk | Fri 22/03/13 | Thu 28/03/13 | 20 | |
| 22 | Final Thesis | 2 wks | Wed 03/04/13 | Tue 16/04/13 | 21 | |
| 23 | Summary | 1 wk | Thu 25/04/13 | Wed 01/05/13 | 21 | |



### 3. MATLAB Source codes

The following are source codes of the program created for face recognition system based on two algorithms; Eigenface and Fisherface algorithms on Yale face database. Codes were written in MATLAB language R2012a.

### I. Eigenface Algorithm - Source Code

```
%Author: MARTIN LUHANJO
%Date: 25th OCT 2012
%Program:Face recognition based on Eigenfaces algorithm and Euclidean
%distance.

%STEP 1:Loading images into 1 big training matrix

Images_directory = 'M:/TRIAL FINAL PROJECT/Fisherface/yalefaces';

image_dimensions = [66, 80];

filenames = dir(fullfile(Images_directory, '*.jpg'));

num_of_images = numel(filenames);

for n = 1:num_of_images
  filename = fullfile(Images_directory, filenames(n).name);

    load_img = imread(filename);
    t1=load_img(:,:,1);

    [w h]=size(t1);
   imagein=double (reshape(t1,w*h,1)); %convert from 2D to 1D vector

   if n == 1
     Images = zeros(prod(image_dimensions), num_of_images);

   end
    Images(:, n) = imagein(:,:);


end

%STEP 2:Calculating mean image face for the training set

mean_image=mean(Images,2);   %Obtaining a mean face image


%substract mean image from each face image in training

subct_image= Images-repmat(mean_image,1,num_of_images);


%STEP 3: Calculating Eigen vectors and Eigen values of covariaanc matrix
%Covariance matrix =A*A', let A=subct_image, but let use L=A'*A since is
```

```
%smaller matrix

L=subct_image'*subct_image;
[eig_matrix, eig_values]=eig(L); %Eigen values and vectors of L=A'*A

%Obtaining best eigenvectores with the highest eigenvalue

eig_vals_vec=diag(eig_values);
[sorted_eigvals, eigIndices]= sort(eig_vals_vec,'descend');
sorted_eigen_matrix=zeros(num_of_images);
for z=1:num_of_images
    sorted_eigen_matrix(:,z)=eig_matrix(:,eigIndices(z));
end

%Eigenfaces/ eigenvectors of Covariances matrix

eigenfaces=subct_image*sorted_eigen_matrix;

%STEP 4:Project the images in the face space to find weight for each image

Proj_trainImg=eigenfaces'*subct_image;

%STEP 5:Obtain test image

test_image='M:/TRIAL FINAL PROJECT/Fisherface/testimagejpg/';
test_imgfile=dir(fullfile(test_image,'*black background0000.jpg'));
imgname=fullfile(test_image,test_imgfile.name);
testImage=imread(imgname);

temp = testImage(:,:,1);
[w1 h1]=size(temp);
InImage = reshape(temp,w1*h1,1);
Difference = double(InImage)-mean_image; % Centered test image
Proj_testimg =eigenfaces'*Difference; % Test image feature vector


%STEP 6: CLASSIFICATION

%Euclidean distance

eucl_distance = arrayfun(@(n) sqrt(norm(Proj_trainImg(:,n) -
Proj_testimg)^2), 1:num_of_images);


%Calculating threshold distance
 for i=1:num_of_images
     maxdist=zeros(1,1);
     for k=1:num_of_images
         maxdist= max(Proj_trainImg(:,k)-Proj_trainImg(:,i));
     end
     if i==1
         maxdistarray=zeros(1,num_of_images);
     end
     maxdistarray(:,i)=maxdist(:,:);

 end
```

```
 ThreD=0.2 * max(maxdistarray); % Threshold distance.

%Recognition Process

[eucl_min, match_index] = min(eucl_distance);
if eucl_min <= ThreD

figure, imshow([temp reshape(Images(:,match_index),w,h)]);


title('Test image    Closest match');


else
    disp('ERROR:the test image cant be recognized or is not a face image')
end
```

## II. Fisherface algorithm- Source codes

```
%Author: MARTIN LUHANJO
%Date: 2nd feb 2013
%Program:Face recognition based on Fisherface algorithm

Image_directory = 'M:/TRIAL FINAL PROJECT/Fisherface/yalefaces';

image_dimensions = [66, 80];

filenames = dir(fullfile(Image_directory, '*.jpg'));

num_of_images = numel(filenames);

for n = 1:num_of_images
  filename = fullfile(Image_directory, filenames(n).name);

    load_img = imread(filename);
    t1=load_img(:,:,1);
    [w h]= size(t1);

   imagein=double (reshape(t1,w*h,1)); %convert from 2D to 1D vector

   if n == 1
     Train_images = zeros(w*h, num_of_images);

   end
    Train_images(:, n) = imagein(:,:);


end

%STEP 2:Calculating mean image face for the training set
```

```
mean_image=mean(Train_images,2);    %Obtaining a mean face image

%substract mean image from each face image in training

subct_img= Train_images-repmat(mean_image,1,num_of_images);

%Creating classes of persons from the training set

C1=Train_images(:,1:6);
C2=Train_images(:,7:12);
C3=Train_images(:,13:18);
C4=Train_images(:,19:24);
C5=Train_images(:,25:30);
C6=Train_images(:,31:36);
C7=Train_images(:,37:42);
C8=Train_images(:,43:48);
C9=Train_images(:,49:54);
C10=Train_images(:,55:60);
C11=Train_images(:,61:66);
C12=Train_images(:,67:72);
C13=Train_images(:,73:78);
C14=Train_images(:,79:84);
C15=Train_images(:,85:90);

%Calculating mean for each class

meanC1=mean(C1,2);
meanC2=mean(C2,2);
meanC3=mean(C3,2);
meanC4=mean(C4,2);
meanC5=mean(C5,2);
meanC6=mean(C6,2);
meanC7=mean(C7,2);
meanC8=mean(C8,2);
meanC9=mean(C9,2);
meanC10=mean(C10,2);
meanC11=mean(C11,2);
meanC12=mean(C12,2);
meanC13=mean(C13,2);
meanC14=mean(C14,2);
meanC15=mean(C15,2);

%Mean classes in one matrix
meanclasses=[meanC1 meanC2 meanC3 meanC4 meanC5 meanC6 meanC7 meanC8 meanC9
meanC10 meanC11 meanC12 meanC13 meanC14 meanC15];

%Number of classes and samples for each class

num_of_classes=15;      % number of classes
cp=6;     %number of samples in each class


%STEP 3: Calculating Eigen vectors and Eigen values of covariaanc matrix
%Covariance matrix =A*A', let A=subct_image, but let use L=A'*A since is
%smaller matrix

L=subct_img'*subct_img;
```

```
[eigenvector_matrix, eig_values]=eig(L); %Eigen values and vectors of
L=A'*A

%EIGENFACES-PCA IMPLEMENTATION
%Sort eigvalues to descending order to find the best eigenvectors

eig_vals_vector=diag(eig_values);
[sorted_eigenvals, eigIndices]= sort(eig_vals_vector,'descend');
sorted_eigenvector_matrix=zeros(num_of_images);
for z=1:num_of_images
    sorted_eigenvector_matrix(:,z)=eigenvector_matrix(:,eigIndices(z));
end

%Eigenfaces

eigenfaces=subct_img*sorted_eigenvector_matrix;

%Significant eigenfaces

eigenfaces=eigenfaces(:,1:num_of_images-num_of_classes);

%STEP 4:Project the images in the feature space for each column

Proj_trainImg=eigenfaces'*subct_img;   %find the weights

%Finding scatter matrix within classes

wh=w*h;
Sw=zeros(wh,wh);

for i=1:num_of_classes

    S=zeros(wh,wh);
    for j=((i-1)*cp +1): (i*cp)  %cp-number of samples
        S=S + (Train_images(:,j)-meanclasses(:,i))*(Train_images(:,j)-
meanclasses(:,i))';
    end
    Sw=Sw+S;
end

%Calculate scatter matrix between classes

sb=zeros(wh,wh);
for j=1:num_of_classes

    sb= sb + (meanclasses(:,j)-mean_image)*(meanclasses(:,j)-mean_image)';

end

%projection of between-class and within-class scatter matrices on PCA

SB=eigenfaces'*sb*eigenfaces;
SW=eigenfaces'*Sw*eigenfaces;

%eigenvectors on fisher space-low dimension
[eigen_fisher e_value]=eig(SW,SB);
```

```
eig_vals=diag(e_value);
lmda=eig_vals(1:(num_of_images-num_of_classes));

fisherfaces=eigenfaces * eigen_fisher;


%Projection to find the weights

projected_fisher=fisherfaces' * subct_img;

%STEP 5:Insert test image and project to the fisherspace

test_img='M:/TRIAL FINAL PROJECT/Fisherface/testimagejpg/';
test_imgfile=dir(fullfile(test_img,'*subject13centerlight.jpg'));
imgname=fullfile(test_img,test_imgfile.name);
testImage=imread(imgname);

temp = testImage(:,:,1);
[w1 h1]=size(temp);
InImage = reshape(temp,w1*h1,1);
Difference = double(InImage)-mean_image; % Centered test image
Projected_testImage = eigen_fisher'*eigenfaces'*Difference;
% Projection of the test image



 %STEP 6: CLASSIFICATION

%Euclidean distance

eucl_distance = arrayfun(@(n) sqrt(norm(projected_fisher(:,n) -
Projected_testImage )^2), 1:num_of_images);


%calculating threshold distance

 for i=1:num_of_images
     maxdist=zeros(1,1);
     for k=1:num_of_images
         maxdist= max(Proj_trainImg(:,k)-Proj_trainImg(:,i));
     end
     if i==1
         maxdistarray=zeros(1,num_of_images);
     end
     maxdistarray(:,i)=maxdist(:,:);

 end

 ThreD=0.5 * max(maxdistarray);

%Recognition

[eucl_min, match_ix] = min(eucl_distance);
if eucl_min <= ThreD
figure, imshow([temp reshape(Train_images(:,match_ix), w,h)]);
```

```
title('Test image   Closest match');

else
    disp('ERROR:the test image cannot be recognized or is not a face
image')
end
```

## III. Mahalanobis distance – Source Codes

```
%Calculating mahalanobis Distance

mahalDist=arrayfun(@(n)sqrt(norm((Proj_trainImg(:,n)-
Proj_testimg)'*((Proj_trainImg(:,n)-Proj_testimg)*(Proj_trainImg(:,n)-
Proj_testimg)')*(Proj_trainImg(:,n)-Proj_testimg))),1:num_of_images);
```

## IV. k-Nearest Neighbours classifier- Source Codes

```
Class= knnclassify (Proj_test', Proj_train',group,6,'euclidean','nearest');
```

## V. Image Pre-processing- Source codes

```
test_image='M:/TRIAL FINAL PROJECT/Fisherface/testimagejpg/';
test_imgfile=dir(fullfile(test_image,'*subject11leftlight0000.jpg'));
imgname=fullfile(test_image,test_imgfile.name);
testImage=imread(imgname);

temp = testImage(:,:,1);
%Filtering
H=ones(1,1)/0.9;
temp1=imfilter(temp,H);

%Fitering/Sharpening
lH=fspecial('unsharp');
temp2=imfilter(temp1,lH,'replicate');

%Contrast/Brightness
temp3=imadjust(temp2,[0  0.6],[0.1 0.9],0.8); %[a b][c d] ,a Up contrast ,c
Up increase Brightness

subplot (1,3,1)
imshow(temp); title('original Image');
subplot (1,3,2)
 imshow (temp2); title ('Brightness->sharpen');

 subplot(133)
 imshow(temp3);title('Sharpen->Brightness->filtering');
```