KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY, KUMASI

INSTITUTE OF DISTANCE LEARNING DEPARTMENT OF COMPUTER SCIENCE

COMPARATIVE STUDY ON FACE RECOGNITION TECHNIQUES: PRINCIPAL COMPONENT ANALYSIS AND LINEAR DISCRIMINANT ANALYSIS

FRANK PEPRAH
ACCRA CENTRE

SUPERVISOR: DR. MICHAEL ASANTE

A THESIS SUBMITTED TO THE INSTITUTE OF DISTANCE LEARNING, KNUST, IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF MASTER OF SCIENCE DEGREE IN INFORMATION TECHNOLOGY

NOVEMBER 2015

DECLARATION

I hereby declare that this submission is my own work towards MSc. in Information Technology and that, to the best of my knowledge, it contains no material previously published by another person nor material which has been accepted for the award of any other degree of the University, except where due acknowledgement has been made in the text.

FRANK PEPRAH …………………………. ………………………

PG8310012 SIGNATURE DATE

Certified by:

DR. MICHAEL ASANTE ……………………………. ………………………

SUPERVISOR SIGNATUREDATE

Certified by:

DR. J.B. HAYFRON-ACQUAH……………………………………………

HEAD OF DEPARTMENT SIGNATURE DATE

ABSTRACT

Face Recognition System employs a variety of feature extraction (projection) techniques which are grouped into Appearance-Based and Feature-Based. In a vast majority of the studies undertaken in the field of Face Recognition special attention is given to the Appearance-Based Methods which represent the dominant and most popular feature extraction technique used. Even though a number of comparative studies exist, researchers have not reached consensus within the scientific community regarding the relative ranking of the efficiency of the appearance-based methods (LDA, PCA etc) for face recognition task.

This paper studied two appearance-based methods (LDA, PCA) separately with three (3) distance metrics (similarity measures) such as Euclidean distance, City Block & Cosine to ascertain which projection-metric combination was relatively more efficient in terms of time it takes to recognise a face. The study considered the effect of varying the image data size in a training database on all the projection-metric methods implemented. LDA-Cosine Distance Metric was consequently ascertained to be the most efficient when tested with two separate standard databases (AT & T Face Database and Indian Face Database). It was also concluded that LDA outperformed PCA.

ACKNOWLEDGEMENT

My heartfelt gratitude is first and foremost expressed to the Almighty God for giving me wisdom, knowledge, understanding and strength to undertake and accomplish this research work.

I owe God much appreciation for his immense providence and I dedicate this thesis to Him. Secondly, I extend my profound and sincere gratitude to my supervisor, Dr. Asante who availed himself despite his busy schedule to have fruitful meetings with me to completion of this work. I am also equally indebted to the numerous authors whose books, journals were used as references in my bid to have the project work completed.

My special thanks also go to my family particularly my lovely wife Mrs Rose Abena Peprah and my children for their support and encouragement to the completion of this masters program.

Again, to AT & T Laboratories at Cambridge and Indian Institute of Technology, Kanpur I am very much grateful for making public their standard face image database for me also to use for my experiments. And finally I extend my gratitude to Mr Amir Hossein Omidvarmia, whose MATLAB sample codes on the Internet greatly helped me in my work.

DEDICATION

This is dedicated to God, my source of livelihood, and my family particularly my lovely wife Mrs Rose Abena Peprah and my children for their immense support and encouragement.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
ICA	Independent Component Analysis
PGM	Portable Gray Map

JPEG	Joint Photographic Expert Group

CHAPTER ONE

INTRODUCTION

Face recognition has been one of the most relevant applications of image analysis. It is challenged to build an automated system which has the ability to recognise faces as human beings. In spite of the fact that human beings are quite good at identifying known faces, we are limited when we have to deal with large amount of unknown faces. Computers, with an almost limitless memory and computational speed should overcome human limitations. It is therefore not surprising that computer-based face recognition has been an active research area over three decades. For instance a lot of scientists from different branches have delved into this area.

The term face recognition can be referred to as classifying or identifying, by computational algorithm, an unknown face image. This operation compares the unknown face image with the known face images stored in a database.

The input of a face recognition system is always an image or video stream. The output is an identification or verification of the subject or subjects that appear in the image or video.

Facial identification consists of assigning input face image to one person of a known group.

Facial verification consists of validating the previously detected person’s identity.

Figure 1-1 indicate the generic representation of face recognition system

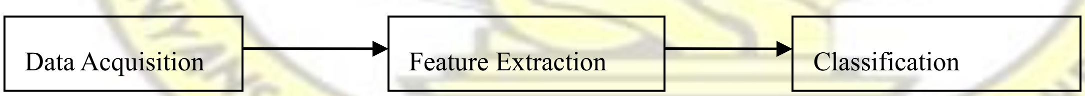


Figure 1-1: Generic Face Recognition System

There are a number of techniques that Face recognition systems can employ for feature extraction.

Generally face recognition techniques can be divided into two groups based on the face representation they use:

i. Appearance-based: With this method, the entire face image is used to derive the most important information that describes a face best. Principal component analysis, Linear discriminant analysis, Independent component analysis are a few examples of appearance-based methods.

ii. Feature-based: This method uses geometric facial features (mouth, eyes, brows, cheeks etc) and geometric relationships between them. It is based on extracting feature vectors from the basic parts of a face such as eyes, nose, mouth, and chin.

A great number of studies in face recognition had paid special attention to the appearance based methods, which has been the most popular feature extraction techniques used in the field of face recognition. Often time researchers reported contradictory results concerning the relative performance of the appearance based techniques. Thus, no consent has been reached within the scientific community regarding the relative ranking of the efficiency of appearance based methods for the face recognition task.

1.1 Problem Statement

In spite of the number of comparative studies that have been done on the topic understudied very often a contest between the abilities of the research groups rather than a comparison between methods was performed. Again, not all possible implementations were considered. Often times other studies had focused on the feature extraction techniques of face recognition (projection methods). That is, the projection methods (such as PCA, LDA etc) were seldom combined with a number of similarity criteria to determine which projection-metric combination was relatively efficient.In most cases only one similarity measure (distance metric) was combined with the projection methods and in some cases using nonstandard databases.

More often than not, where more than one distance metric classifiers were used variability of data size and its effect on their results were not considered. In the case where variability of data size was considered one distance metric classifier (i.e. Euclidean distance most commonly used) was used. This study considered more than one distance metric classifiers and the variability of data size.

Again the findings of other research groups were often contradictory on the subject as indicated earlier and that was another important reason for performing a study of this kind. Thus, the relative performance of the two techniques is an open question.

This thesis studied two appearance-based techniques combined with three distance metrics. Projection methods studied were: Principal Component Analysis (PCA), Linear Discriminant Analysis. The Similarity Matching Methods (Distance metrics) used were City Block, Euclidean, and Cosine Metrics.

1.2 Objective

The thesis aimed at studying and ascertaining which projection–metric combination was more efficient using MATLAB. Specifically the objectives were as follow:

- To study and understand the two appearance-based methods i.e. PCA and LDA as applied in face recognition.
- To study and understand the three similarity measures (distance metrics) i.e. Euclidean distance, City Block and Cosine as applied in face recognition
- To combine each of the projection method with each similarity measure.
- To design and implement algorithm of each projection-metric method with varying data

size.

- To analyze the results and report the most efficient among the combined projectionmetric methods.

1.3 Methodology

The methodology of this thesis was based upon information gathered and processed during the study and research phase of the course. The method applied for the design and implementation of the face recognition system was as follows:

Data Formation phase: This phase involved the acquisition of face images from standard face image database through the Internet and the pre-processing of such images using MATLAB. The pre-processed images were stored into Training database.

Training Phase: In this phase, those face images used in the training set were chosen from the entire training database. After obtaining the training set (consisting of converted vectors of the face images), feature vectors (eigenfaces or fisherfaces)were formed by applying the appearancebased techniques (feature extraction technique) employed in this study i.e. PCA and LDA and stored for later use.

Recognition Phase: Having obtained feature vectors, recognition process began in this phase. At this stage, probe /test image was selected and used as input image to initiate the recognition process. Here determination was made whether input image was similar to any of the images (or has a match) in the training set using the similarity measure.

Similarity Matching Methods*:* They define a value that allows the comparison of feature vectors of test image and those from the training set. This helps to find equivalent match in the database for an input(test) image.. In this study, similarity matching methods employed were Euclidean and City Block distance, and Cosine distance.

Performance evaluation phase: This phase showed the comparative analysis of the outcome of face recognition techniques employed. This was based on the accuracy (number of correct match), time of execution of algorithm. The generalization ability of the combined projectionmetric methods was also ascertained.

The generalization ability of an algorithm has several meanings some of which are; It is an ability to maintain a recognition rate when reducing the number of images in the training set. Alternatively, when considering the algorithms that use more than one image per class in the training database, the ability of an algorithm to maintain a recognition rate when the number of images per class used in training is reduced (Navarrete and Ruiz-del-Solar, 2002).

1.4 Some applications of face recognition

Face recognition has been applied in many areas such as Law enforcement, Airport Secuirty, Access Control and Driver's Licences and Passport etc to help in the following;

- to help governments to stay ahead of the world's ever-advancing terrorists,
- To enhance security efforts that already underway at most airports and other major transportation hubs (seaports, train stations, etc.),
- To enhance security efforts considerably in organisations,
- To leverage the existing identification infrastructures in driver's licence and passport.

1.5 Research Questions

Which projection-metric combination algorithm yields higher recognition rate (accuracy)? Which projection-metric combination algorithm is more efficient in terms of time taken to recognise a face?

Does the varying of the number of training image affect the performance of projection metric algorithm?

What will be the effect of varying the number of training images in a class on projection-metric algorithm?

1.6 Significance of the study

This study seeks to deepen the search for most efficient and accurate technique for a face recognition system. Again, to determine which projection-metric can withstand increased data size of face recognition system. This, as a result , will help to design a face recognition system for Voter Registration and Verification, Driver's Licenses, Immigration activities and host of others which deals with large data and requires little amount of time to identify or verify ones image.

1.7 Organization of the Study

The study is organized into five chapters.

- **Chapter One:**The Chapter presents the Research Proposal, comprising of Introduction to the study, the Problem Statement, Objectives of the Study, Methodology, Some application of Face Recognition, Challenges to Face Recognition, Research Questions and Significance of the study.
- **Chapter Two:** This presents a comprehensive review of relevant literature in order to position the study in an appropriate theoretical framework. Thus it deals with historical facts about Face Recognition, related works of the study and some findings, and theoretical framework of the two techniques under review.

- **Chapter Three:** This chapter also discusses the methodology employed for the study coupled with the algorithms of the two techniques and the standard face image database used. It also presents the results of the experiments undertaken in this thesis.
- **Chapter Four:** This chapter analyses results of the experiments implemented.
- **Chapter Five:** This chapter presents conclusion for the study and future work.

CHPATER TWO

LITERATURE REVIEW

2.1 History of Face Recognition

Studies in the field of face recognition dated back to 19th century where Darwin (1872) worked on different facial expression due to different emotional state, and Galton (1888) worked on facial profile. Thereafter, there were some attempts to develop semi-automated facial recognition system in the late 1960s and early 1970s based on geometrical information(such as eyes, nose mouth etc and their geometric relationship). For instance, Goldstein et al (1971) created a system of 21 subjective marks such as hair colour and lip thickness but was very difficult to automate. Fischler and Elschanger(1973) measured the facial features using templates of single facial features and mapped onto global template. However, Kenade (1973) developed the first fully automated face recognition system, whose algorithm extracted 16 facial parameters automatically and compared to human or manual extraction which showed only small difference. From the above historical facts, it is obvious that early part of face recognition focused on automatic detection of individual facial features. This approach had advantages of being insensitive to illumination and that there was intuitive understanding of the extracted features. However, according to Cox et al (1996) , and Li and Jain (2005)facial feature detection and measurement techniques were not reliable enough for the geometric feature-based recognition of a face and geometric properties alone were inadequate for face recognition. As a result of this setback geometric feature-based technique had gradually been abandoned and an effort had been made in researching holistic (appearance-based) techniques, which provided better results.

Sirovich and Kirby (1987) were the first to employ Eigenface technique which was based on

PCA to recognise an image in a lower dimension without losing much information and then reconstructing it. Turk and Pentland (1991) enhanced this coarse method.

Since the 1990s, face recognition has received a lot of attention with a noticeable increase in a number of publications resulting in other holistic techniques like Fisher Discriminant Analsys, Independent component Analysis etc.

2.2 RELATED WORKS

As shown in the figure 1-1 feature extraction techniques plays important role in every face recognition system particularly on the performance of such systems. It is therefore imperative to have a detailed knowledge of the extraction techniques employed for effective design of recognition schemes and then for construction of robust face recognition systems. A lot of studies have been conducted by Researchers to compare various feature extraction techniques and their robustness to facial appearance changes. Most of these studies paid special attention to appearance-based methods, which were the most popular feature extraction techniques used in the field of face recognition.

However, more often than not researchers reported contradictory result of the comparative studies conducted with respect to the performance of the appearance-based techniques. For example, Beveridge et al(2001) reported that in their experiments PCA systematically outperformed LDA, whereas Belhumeur et al (1997) claimed that LDA performs better than PCA in all of their tests. According to Delac et al (2006) the performance of the appearancebased techniques largely depended on the similarity measure employed. As result, with the right combination of the technique and distance no claim regarding the superiority of any of the three techniques i.e., PCA, LDA, ICA, could be made.

It is quite obvious from the discussion that even within the scientific community different results had been found regarding the relative ranking of the appearance-based methods for the face recognition task.

2.3FINDINGS

Having studied literature and journals on face recognition it was observed that appearancebased image approaches seemed to dominate up to now in face recognition systems mainly because of the strong prior knowledge that all face images belong to face class.

Again, it is important to note that some differences and similarities between PCA and LDA were found. PCA tries to keep as much structure of the features (variance). It de-correlates the feature space and orders the dimension with decreasing variance. In order to reduce dimensionality and choose the first N dimension the most structure of the data is kept.LDA rather focus on dimension that separates classes and orders dimension according to class separability. LDA explicitly attempts to model the difference between classes of data whereas PCA on the other hand does not take into account any difference in class. Whereas LDA seeks directions that are efficient for discriminating data, PCA seeks the directions that are efficient for representing data. There are some characteristics which are common to both PCA and LDA. First and foremost, they produce spatially global feature vectors. In other words, they produce basis vectors which are non-zero for almost all dimensions.This implies that a change to a single input pixel will change every dimension of its subspace projection. Again, they both look for linear combinations of variables which best explain the data.

Database: At the early stage of face recognition every individual or research group collected their own database of images. Subsequently, it became necessary to have a uniform benchmark database and thus FERET database was collected at NIST (National Institute of Standards and Technology), AT and T database at AT and T Laboratories Cambridge (2002), Indian Face Database at Indian Institute of Technology, Kanpur etc for testing face recognition algorithms. In this study, the standard databases used were **AT and T database and Indian Face Database** for testing face recognition algorithms.

Another observation was that, the most commonly used parameters for face recognition were as follows:

i. Accuracy

ii. Variability in data size iii. Blurriness in test data

iv. Image size.

It is important to note , however that where more than one distance metric classifiers were used variability of data size and its effect on their results were not considered. In the case where variability of data size was considered one distance metric classifier (i.e. Euclidean distance most commonly used) was used.

However, this study considered more than one distance metric classifiers and the variability of data size.

2.4 FACE SPACE

Generally, a two dimensional image *I(x,y)* of size *m*-by-*n* pixels can be viewed as a vector (or a point) in high dimensional space, where *m* is the number of rows of pixels and *n*, the number of columns of pixels.

An image space can be referred to as a space having dimensions equal to the number of pixels making up the image and having values in the range of the pixels values. Thus, for example with a grey scale image of size (*m* x*n*), the dimension of the image space is P, where P =m x n. In respect of gray scale images, the image could have a dimension with a value in between 0 and 255.

A face image can be referred to as a point in the image space by converting the image to a long vector by concatenating each column of the image one after the other as illustrated in the Figure 2-1.

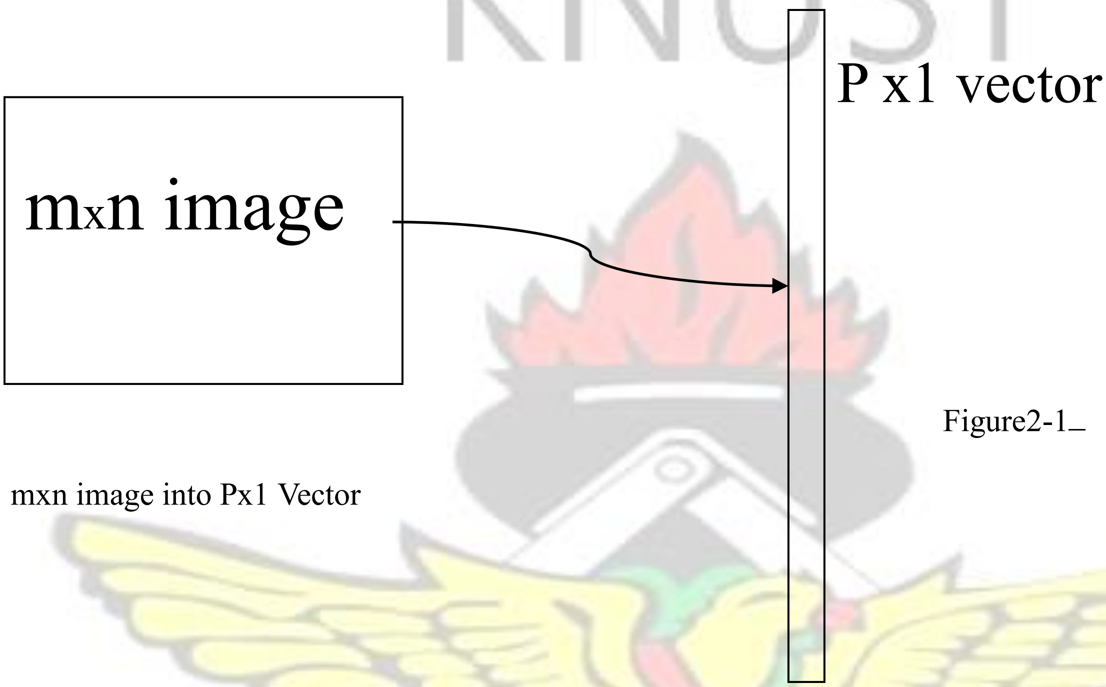


Figure2-1_ mapping mxn image into Px1 Vector

When all the face images are converted into vectors, they will group at a certain location in the image space as they have similar structure, having eyes, nose and mouth in common and their relative position correlated as illustrated in Figure 2-2.Therefore, all the face vectors are located in a very narrow cluster in the image space as face space.

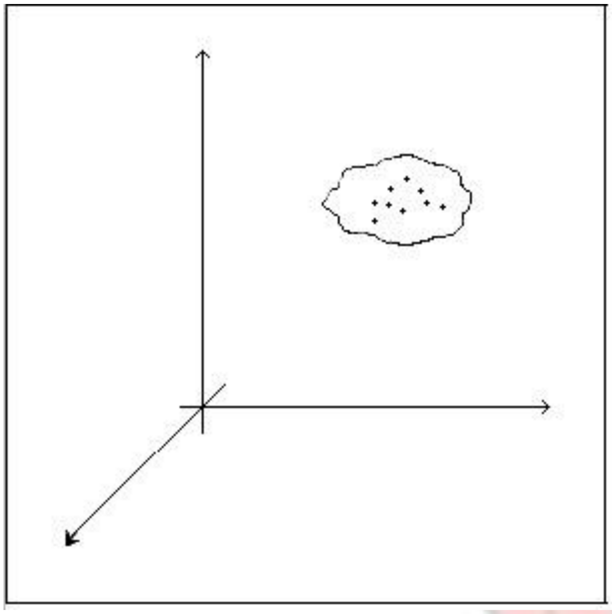


Figure 2-2 Image Space and face cluster

2.5 Principle Component Analysis (PCA)

Introduction

PCA is a standard technique used in statistical pattern recognition and signal processing for data reduction and Feature extraction. In this method, patterns are mapped onto feature vectors to remove redundant information which they normally contain while preserving most of their intrinsic information content. These extracted features play very important role when input patterns have to be distinguished.

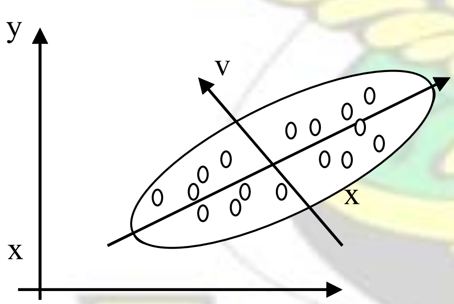
It can also be described as a method of transforming a number of correlated variables into a smaller number of uncorrelated variables. PCA behaves in much the same way as Fourier analysis in that they both decompose signals into a set of additive orthogonal basis vector(in the case of PCA) or sinusoids of varying frequencies (in the case of Fourier analysis).There is one important difference between them. That is whereas Fourier analysis uses a fixed set of basis function, the PCA basis vectors are learnt from the data set via unsupervised training.

How PCA Works

PCA focuses on explaining the covariance structure of a set of variables. Particularly, it helps in identifying the principal directions in which the data varies.

Take for instance, two variable data set in the X-Y coordinate system as illustrated in figure 2-3.The U axis and the V axis which are orthogonal to each other shows the principal direction in which the data varies. If the U- V axis system is placed at the mean of the data it will give a compact representation. When each (X, Y) coordinates is transformed into its corresponding (U, V) value it will de-correlate the data. This means that the co-variance between the U and V variables becomes zero.

For a given set of data, PCA finds the axis system which is defined by the principal directions of variance (i.e. the U -V axis system in figure 2-3). The directions U and V are called the principal components.



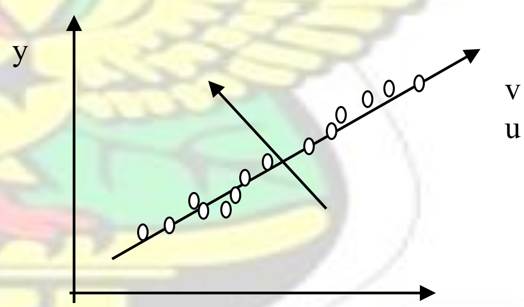


Figure 2-3: PCA for Data Representation Figure 2-4: PCA for Dimension Reduction In a situation where natural property or experimental error causes variation in dataset, such dataset maybe expected to be normally distributed. The nominal extent of the normal distribution is shown by hyper-ellipse in figure 2-3. This hyper –ellipse encloses data points that may be considered to belong to a class, hence it is thought of as class boundary.

Where some other relationship causes variation in data PCA will result in reducing the dimensionality of dataset. From figure 2-4 the two variables are nearly related linearly. From the figure2-3, the principal direction in which the data varies is shown by the U axis and the secondary direction by the V axis. However, all the V coordinates are very close to zero. Assuming they are only non-zero because of experimental noise, then the U-V axis system can represent the data set with one variable U and discard V. Thus the dimensionality of the problem has been reduced by 1.

Computing the Principal Components

The principal components can be computed by calculating the eigenvectors and eigenvalues of the data covariance matrix ($\sum$). Below is a standard technique to avoid computational complexities.

Let $\sum \; = X = BB^T$, n x n matrix

Where B is nxp matrix,n>p

This covariance matrix is very hard to work with due to its huge dimension causing computational complexity.

To prove that the eigenvectors of B^TB (p x p) matrix can be used instead of BB^T (nxn),

Let $Y=B^TB$ .... (1) which is of size (p x p).

Then, the eigenvectors δ and the eigenvalues Λ of Y are obtained as,

$$Y\delta = \Lambda\delta \quad \text{.....(2)}$$

i.e $$B^T B\delta = \Lambda\delta \quad \text{..... (3)}$$

On left multiplying B both sides of (3) ,

$$BB^T B\delta = B\Lambda\delta \quad \text{.....(4)}$$

Since Λ is a scalar, Equation (4) can also be written as

$BB^T B\delta = \Lambda B\delta$……………………………………………………(5)

Substituting $X=BB^T$ into (5)

$XB\delta= \Lambda B\delta$……………………………………………………….. (6)

If $B\delta$is substituted by $\upsilon=B\delta$Then

$\upsilon=B\delta$……………………………………………………………………(7)

is one of the eigenvectors of $X=BB^T$ and its size is (n x1) .

It therefore implies that the eigenvectors of X can be deduced using the eigenvectors of Y. Hence a matrix of size (p x p) is utilized instead of a matrix of size (n x n). This formulation brings substantial computational efficiency.

From the figure 2-3 the matrix of eigenvectors u will represent linear transformation, which transforms data points [X,Y] axis systems into the [U,V] axis systems. Generally, the linear transformation given by υ transforms the data points into a data set where the variables are uncorrelated. The correlation matrix of the data in the new coordinate system is Λwhich has zeros in all the off diagonal elements.

PCA in practice

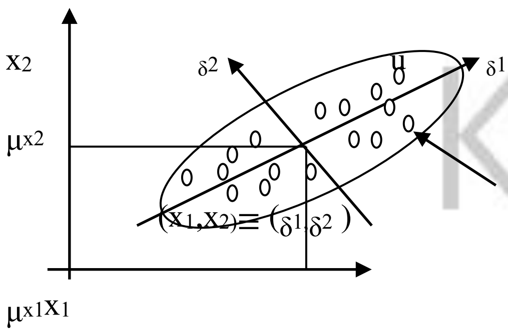


Figure 2-5: The PCA Transformation

Figure 2-5 gives a geometric illustration of the process in two dimensions. Using all the data points, the mean values of the variables (μ_{x1}; μ_{x2}) and the covariance matrix ($\sum$) which is a 2 x2 matrix in this case are found.

Calculating the eigenvectors of the co-variance matrix gives the direction vectors indicated by δ_1, and δ_2. Putting the two eigenvectors as columns in the matrix $\delta = [\delta_1\ \delta_2]$ creates a transformation matrix which takes the data points from the [x_1; x_2] axis system to the axis [δ_1,δ_2] system with the equation:

$$p_\delta = (p_x - \mu_x).\delta \qquad (8)$$

where p_x is any point in the [x_1; x_2] axis system, $\mu_x = (\mu_{x1}; \mu_{x2})$ is the data mean, and p_δ is the coordinate of the point in the [δ_1;δ_2] axis system.

Dimension Reduction

PCA as dimension reducer can be explained by considering an application having M images with n pixels each. Then the entire data set can be written as an n x M data matrix E with each column of E representing one image of the data set.

The standard technique is employed to reduce dimension to MxM matrix which would otherwise have been nxn matrix when computing covariance matrix of the dataset. This consequently results to M eigenvectors and their corresponding eigenvalues of the covariance matrix of data set.

Size reduction can further be achieved by choosing to represent the data with fewer dimensions. Normally the set of m ($m < M < n$) eigenvectors of the covariance matrix($\sum$) which have the m largest eigenvalues can be chosen. Typically for face recognition system m will be quite small. These can be composed in an n x m matrix $\delta pca = [\delta_1;\delta_2;\delta_3 \ldots \delta_m]$ which performs the PCA projection. For any given image $p_x = (i_1, i_2, i_3, .. i_n)$ a corresponding point in the PCA space can be found by computing

$$p_\delta = (p_x - \mu_x).\delta_{pca} \ldots \quad (9)$$

The m-dimension vector $p_\delta$is all that is needed to represent the image. This is a massive reduction in data size since typically n will be at least 1600 and m varies between twenty and a few hundred proportionate to the number of training images. All the data base images can be stored in the PCA space and can easily search the data base to find the closest match to a test image.

PCA has an important feature of reconstructing original image from eigenfaces. This will require a computation of weight vector of each mean-centred image vector. Each weight vector of the face to be reconstructed is then multiplied by the selected eigenvectors. The result is added to mean image vector to get an approximation of the original image .However, if all the eigenvectors are used then it is mostly to have the image as exactly as the original image..

Reconstructing any image with the inverse transform:

$px = p_{\delta.}\delta^{T}_{pca} + \mu_x$ …………………………………………………………..(10)

Eigenface Method

Eigenface method is the implementation of PCA over images. The Eigenface method eliminates variance resulting from non-face images to find a lower dimensional space for the representation of the face images.Take for instance a 2D situation where an input image is compared with a set of data base images to find the best match. And assume that all the images have the same resolution and are equivalently framed. Each pixel can be considered a variable thus having a very high dimensional problem which can be simplified by PCA.

In this method, the features of the studied images are obtained by the following;

- Convert each member of the training images into image vectors(Γ) by adding or appending each column one after the other .Each image vector may have a size, say Px1.
- Form Training set matrix containing all the image vectors :($\Gamma_1, \Gamma_2, \Gamma_3,,, \Gamma_M$)
- Calculate the mean vector of the entire image vector. That is arithmetic average of the training image vectors at each pixel point: $(\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i)$…………………………..(11)
- Subtract mean vector from, every image vector in the a training image to look for the maximum deviation of each image from the mean image. $\Phi = \Gamma - \Psi$……………(12)
- Form matrix with mean-centred image vectors. $A=[\Phi_1,\Phi_2,\Phi_3 \ldots \Phi_M]$……......(13)
- Calculate the covariance matrix of the images. $Z=A.A^{T} = \frac{1}{M}\sum_{i=1}^{M}\Phi_i \Phi_i^{T}$ …………..(14)

- Obtain eigenvectors and their corresponding eigenvalues of the covariance matrix of the images.
- Eigenvectors are sorted in descending order by eigenvalues and eliminate eigenvectors with corresponding small eigenvalue.
- The eigenface space is obtained by applying the eigenvectors on the mean-centred

images. $\omega = A.\ \upsilon_k$, ……………………………………………………………….(15)

Where υk eigenvectors of $k=1,2,3...M'$(M' number of selected eigenvectors. Weight matrix $\omega=[\ \omega_1, \omega_2, \omega_3 .... \omega_{M'}]$

- Later, the training images are projected into the eigenface space $\Omega = \omega'.\ A$ …………..(16)
- Each image in the image space is transformed into eigenspace. In image space each image has a size of Px1 whereas in the eigenspace image has a size of M'x 1.The dimension is thus reduced as $M'<P$.

2.6 Linear Discriminant Analysis (LDA)

Introduction

Linear Discriminant Analysis is a popular classification technique developed by Roland Fisher. It is sometimes called Fisher Discriminant Analysis(FDA). The main objective of LDA is to separate samples of distinct groups. Essentially, it transforms data to a different space which optimally distinguishes classes.This was recognised and applied on face recognition by Belhumeur et al (1996).

The LDA basically finds a linear transformation in order that feature clusters are most separable after the transformation. And this can be achieved through scatter matrix analysis.

It maximizes the ratio of Between-classes scatter to Within-classes scatter in order to find the combination of features that separate best between classes. The Between-class (also known as extra-personal) scatter represents variations in appearance as a result of difference in identity. Within-class (intra-personal) scatter represents variations in appearance of the same individual due to different lighting and face expression. In other words, it finds the projection directions that on one hand maximize the distance between the face images of different classes and on the other hand minimize the distance between the face images of the same class. Consequently, images of the same class (or person) are grouped together whereas images of different class(or persons) are separated.

Take an instance, two sets of points in 2-dimensional space which are projected onto a single line. Depending on the direction of the line, the points can either be mixed together (as in the case of Figure 2-6a) or separated (Figure 2-6b). LDA find s the line that best separates the points.

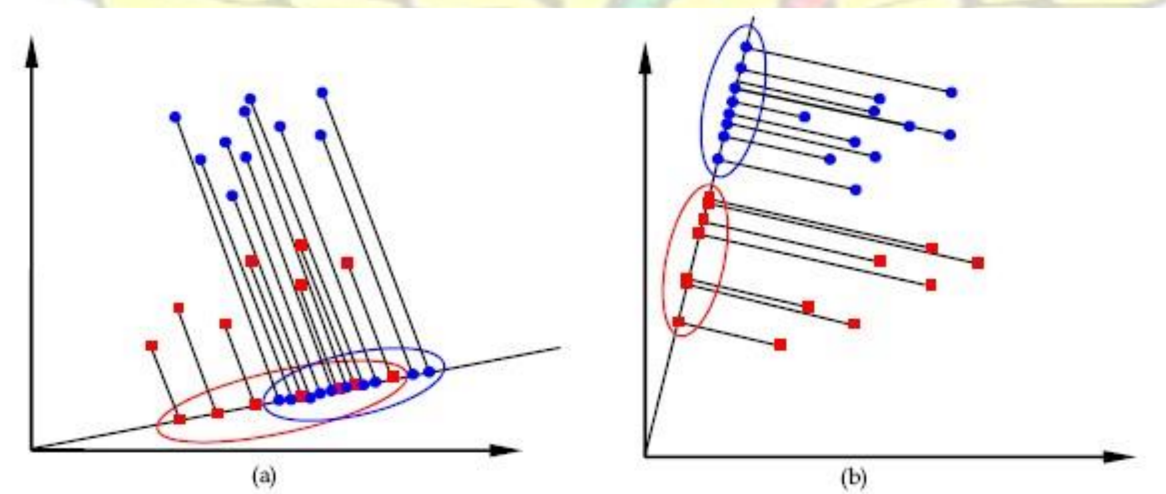


Figure 2-6 (a) Points mixed when projected onto a line. (b) Points separated when projected onto another line.

Figure 2-7 shows good and bad class separation.

Figure 2-7(a) Good class separation. (b) Bad class separation.

The steps involved in LDA are as follow;

- Find the two scatter matrices referred to as the "between class" and "within class" .
- finds a set of vectors W_{LDA} such that Fisher Discriminant criterion is maximised.

$$W_{LDA} = \arg\max_{W} \frac{W^T.S_B.W}{W^T.S_W.W} \quad \ldots\ldots(17)$$

Where W is transformation matrix and W^T is transpose matrix of W.

$$S_B=\sum_{i=1}^{c} N_i(\pi_i-\mu).(\pi_i-\mu)^T \quad \ldots\ldots(18)$$

$$S_W=\sum_{i=1}^{c}\sum_{j=1}^{Ni} (x^i_j-\pi).(x^i_j-\pi)^T \quad \ldots\ldots(19)$$

Where N_i is the number of training samples in class i,

c is the number of distinct images(classes), π is the

class (group) mean of class i,

μ is overall mean, and x^i_j represents the set of samples

belonging to class i.

S_W= the scatter of features around the mean of each class and

S_B= the scatter of features around the overall mean for all face classes.

The solution of the maximization problem is the solution of generalized eigensystem;

$S_B.V=\Lambda.S_W..V$ ……………………………………………………………….(19)

Where V is the eigenvector (Fisherface) matrix and Λ are corresponding eigenvalues of the within class and between –class matrices.

This system can be easily solved by

$S_W^{-1}.S_B.V=\Lambda.V$ ………………………………………………………………..(20)

However, this approach can produce the following problems;

i. This eigensystem does not have orthogonal eigenvectors because $S_W^{-1}.S_B$is, in general not symmetric.

ii. The matrices $S_W,.S_B$ are usually too big iii. The number of training images available in the field of face recognition is undoubtedly significantly smaller than the images' dimension. This leads to the Within-class scatter matrix S_W being singular and non-invertible, Jain (1991).

In attempt to overcome the problems enumerated above a modified LDA was recommended in Belhumeur et al (1997).The following were the recommendations;

- project all images into the PCA subspace. This is to reduce their dimensionality and consequently ensures that the matrix S_W is invertible)
- perform LDA in the reduced space to produce transformation matrix W comprising of eigenvectors. At this point the eigenvectors can be referred to as fisherface or fisherface pattern. Struc and Pavesic (2008), Delac et al (2005).

Algorithmic Description of LDA

To begin with, all images are projected into the PCA subspace instead of using the pixel values of the images. The eigenface projection of PCA transformation is initially applied in the Subspace LDA method. Equations 11-16 referred.

- Convert each member of the training images into image vectors (Γ) by adding or appending each column one after the other.
- Form Training set matrix containing all the image vectors :($\Gamma_1, \Gamma_2, \Gamma_3,,, \Gamma_M$) □ Calculate the mean vector of the entire image vector. $(\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i)$
- Subtract mean vector from, every image vector in the a training image. $\Phi = \Gamma - \Psi$
- Form matrix with mean-centred image vectors. A=[$\Phi_1, \Phi_2, \Phi_3 \ldots Q_M$]
- Calculate the covariance matrix of the images. $Z = A.A^T = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\,\Phi_i^T$
- Obtain eigenvectors and their corresponding eigenvalues of the covariance matrix of all the images.
- Sort eigenvectors in descending order by eigenvalues and eliminate those with corresponding small eigenvalues.
- Apply the eigenvectors on the mean-centred image vectors to obtain eigenface space. $\omega = A.\,\upsilon_k$, where υk eigenvectors of k=1,2,3...M'(M' number of selected eigenvectors.

  Weight matrix ω=[$\omega_1, \omega_2, \omega_3 \ldots \omega_{M'}$]
- Later, the training images are projected into the eigenface space. $\Omega = \omega'.\,A$

Use the eigenspace to compute the between-class and within-class scatter matrices

- Compute the arithmetic average of the eigenface projected training image vectors corresponding to the same individual class. $m_i = \frac{1}{qi}\sum_{k=1}^{qi}\Omega_k$, q^i = 1, 2, …C and size of each eigenface class mean is (M′ x 1). ……………………………………………(21)

- Compute the arithmetic average of all the eigenface projected training image vectors.

$m^0 = \frac{1}{M}\sum_{k=1}^{M'} \Omega_k$…………………………………………………………………………(22)

.

- Compute the ***Within Class Scatter Matrix:*** - It represents the average scattering of the projection matrix Ω in the eigenface space of different individuals C_i around their respective class means m_i.
- Compute the ***Between-Class Scatter Matrix:*** - It also represents the scatter of each projection classes mean m_i around the overall mean vector m_o.
- Find the projection say W which maximizes between-class scatter and minimizes withinclass scatter. W can be obtained by solving the generalized eigenvalue problem;

$S_b W = S_w W \lambda_w$……………………………………………………………(23)

Here too the vectors W are selected based on their corresponding eigenvalues, thus eliminating the vectors with smallest eigenvalues. This implies that the selected number of eigenvectors (fisherfaces), F will be less than the number of vectors in the eigenspace (M') that LDA started with.

- Project the eigenface projections of the training image vectors to the fisherface space by the dot product of optimum projection, W and eigenface projection matrix, Ω.

$g(\Omega) = W^T . \Omega$,It is of size (F x M'). ………………………………………………(24)

2.7 CLASSIFICATION (RECOGNITION)

Classification: -It is the problem of identifying which set of categories a new observation belongs to, based on a training set of data containing observation whose category membership is known. The distance measure between data points is an important component of a classification algorithm.

The most popular distance measures used are as follow:

A. Euclidean distance function

In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points .This is calculated by the Pythagorean formula.

B. City Block distance function

The City Block distance (also referred to as Manhattan distance)between two items is the sum of the differences of their corresponding components.

The distance between two points, *a* and *b*, with *k* dimensions is calculated as:

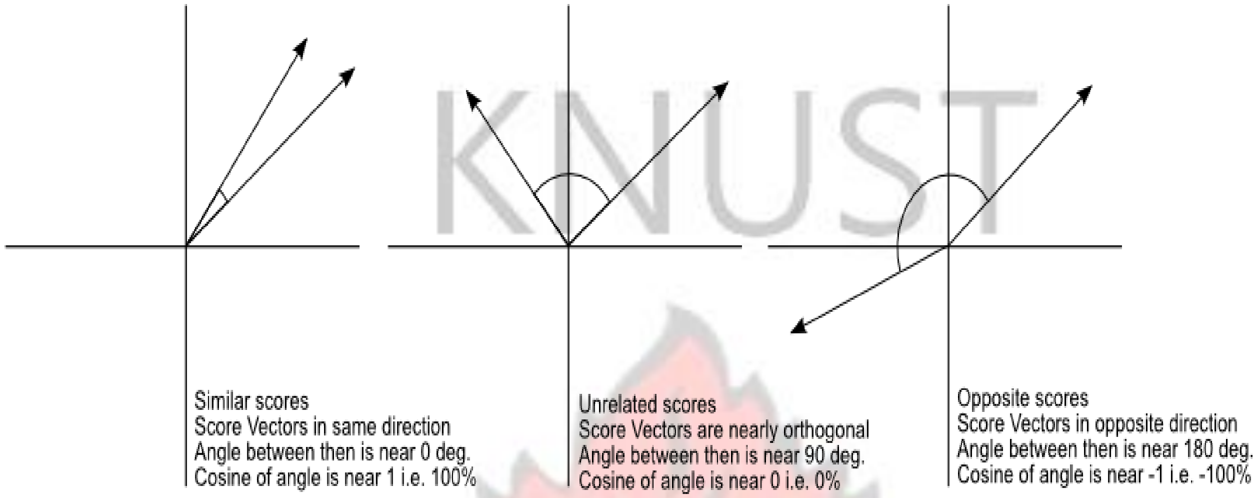
$$\sum_{j=1}^{k} |a_j - b_j| \qquad \text{.........(25)}$$

It is always greater than or equal to zero. If the measurement is zero then they are identical points whereas high measurement will indicate little similarity

C. Cosine Similarity

The Cosine similarity between two vectors is a measure that calculates the cosine of the angle between them. This metric is a measurement of orientation and not magnitude .It can be seen as a comparison between vectors on a normalized space since it does not only take into consideration the magnitude of each vector, but the angle between them. It is solved by finding the dot product for the angles between the vectors. (i.e. $\cos\theta$):

Cosine Similarity generates a metric using angles to show the relationship between two vectors, as can be seen in figure 2-8:



The Cosine Similarity values for different vectors are 1 (same direction), 0 (90 degrees.), -1 (opposite directions). Figure2-8 Cosine Similarity metrics

Given two vectors of attributes, *A* and *B*, the cosine similarity, *cos(θ)*, is represented using a dot product and magnitude as

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad \text{........ (26)}$$

The result ranges between -1 and 1. The -1 means the vectors are exactly opposite whereas 1 is exactly the same. The result 0 indicates orthogonality (de-correlation)whiles the in-between values means intermediate similarity or dissimilarity.

With regard to face recognition, classification is employed to perform similarity measure. This measure defines the distance or angle between the projected Test image and each image in projection matrix depending on the similarity measure used.

Once the face images have been projected into the eigenspace or fisherface, the similarity between any pair of face images can be calculated by finding distance or angle between their corresponding feature vectors. The smaller the distance(in respect of Euclidean and city block distance measures)or angle (in respect of Cosine Similarity) between the features, the more similar the faces.

Algorithmic Description of Classification

- Convert the test (probe) image into a single column vector , Γ_T
- Substract mean image vector from test image. $\Phi_T=\Gamma_T-\Psi$ ……………………………(27)
- The mean-centered image is projected over eigenspaces / fisherfaces. $\Omega_{T}=\omega'.\ \Phi_T$........28)
- Compute the similarity distance metrics (Euclidean distance, city block and cosine) using the projected vector of the test image and that of the training set.
- The projected image vector of the training set with the least distance metric (in the case of Euclidean distance, city block) or maximum cosine value (in the case of cosine similarity measure) compare to test image becomes the equivalent image of the tested image.

CHAPTER THREE

METHODOLOGY

The aim of this study was to perform the comparative analysis of PCA and LDA as applied on face recognition. First of all the algorithms of the PCA and LDA were implemented and then evaluated their performances under various parameters.

In this thesis, a framework of facial biometric was designed based on the two feature extraction subspace methods studied.. Both PCA and LDA features were presented to the same similarity measures separately such as Euclidean distance, City Block and Cosine Distance measurement.

The algorithms were tested with standard face database which were AT and T Database and Indian Face Database.

The framework had four phases viz: Data Formation Phase, Training Phase, Recognition Phase and Performance Phase.

3.1 DATA FORMATION PHASE

For consistency with other studies and the need for uniform benchmark database, this study used two different standard set of face images from AT& T Database which had been collected by AT & T Laboratories at Cambrigde and Indian Face Database.

AT & T Database contained 400 images with 10 different images of 40 distinct subjects (persons). The images were of portable gray map (.pgm) format. They varied in terms of lighting, facial expressions including open/closed eyes, facial details such as glass/without glasses and different time of snapping pictures. All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position. The Figure3-1 is an example of AT & T face images.

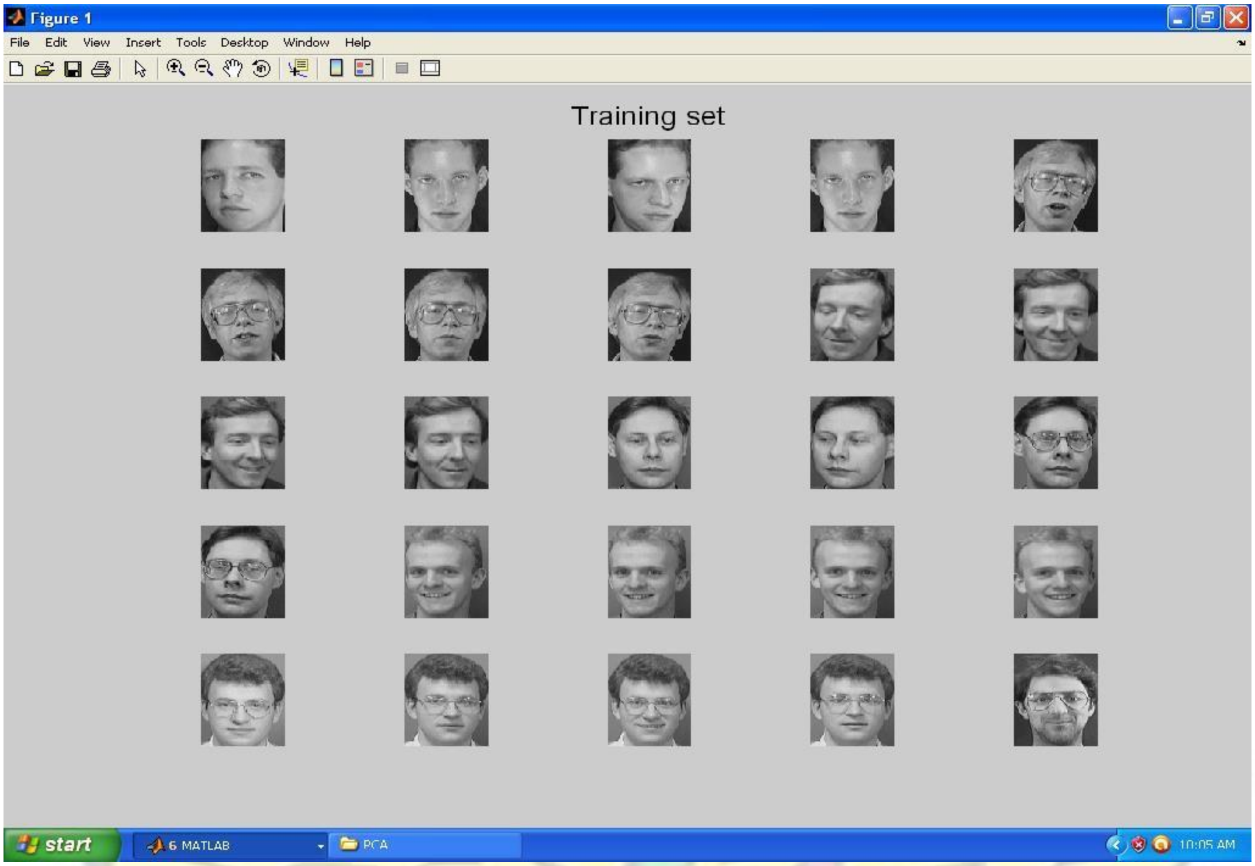


Figure 3-1;Sample images in AT & T Database

The Indian Face Database (IFD) contained 429 male images and 242 female images from Indian Institute of Technology, Kanpur Campus. The male group had averagely 11 different images of 39 subjects (persons) whereas female contained 11 different images of 22 subjects (persons).IFD contained images of 61 distinct subjects with eleven different poses for each individual. However, in the study 600 images were used. The images had bright homogeneous background and the subjects were in an upright, frontal left, looking right, looking up, looking up towards left, looking up towards right, looking down. In addition to the variation in pose, images with four emotions - neutral, smile, laughter, sad/disgust - were also included for every individual.The images were of joint photographic expert group (.jpg or .jpeg) format. Figure 3-2a,b,c d are samples of original Indian Face Images.

Figure. 3-2a

Figure 3-2c

Figure 3-2b

Figure 3-2d

Having obtained images to test the algorithms, the images were pre-processed for effective test. The pre-processing was done to perform image size conversions (i.e from original of 640x480 pixels to 60x50 pixels) and enhancements on face images. Again, as part of the pre-processing activities, the dynamic range of face images were modified (histogram equalization) in order to improve face recognition performance. That is, image pixel values were histogram equalized to the range of values from 0 to 255.

For example the original Indian face images (figures 3-2a ,3-2b) which were coloured but not cropped were pre-processed as follows;

a. Changed the images to grey(as can be seen in figure3-3a&3-3b).
b. Cropped them to 300(height) x 280 (width)to eliminate as much background as possible.(as shown in figure3-3a &3-3b).
c. Resized to 60x 50 pixels (as shown figure3-4a and 3-4b).
d. Histogram equalised (as shown in figure3-4a and 3-4b).

The figures 3-3a and 3-3b are samples of the images grey scaled and cropped

Figure3-3a

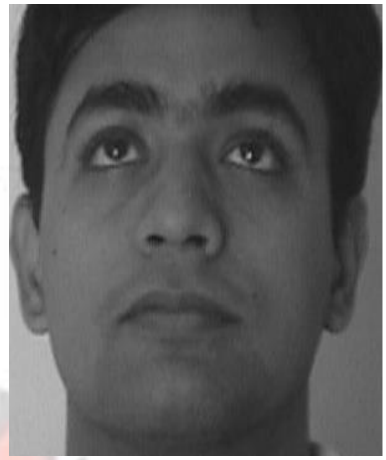


Figure 3-3b

The figures 3-4a and 3-4b are samples of the images resized and histogram equalised

Figure3-4a

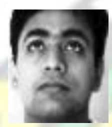


Figure3-4b

As pre-processing steps, the file format of AT& T face images (.pgm) were converted into .jpg for uniformity. The images which were already cropped and in greyscale (but not coloured) were resized to 60 x 50 pixels .Again, they were histogram equalized as shown in Figure 3-5a, b, c. The codes in appendix A was used to perform it.

Figure 3-5 a **Figure 3-5b** **Figure 3-5c**

Figure a=Original Images (112x92 pixels) Figure b =Resized images (60x50 pixels) Figure c=Histogram equalised Images

The pre-processed images were stored into Training database and Probe database.

The study categorised the Training and Probe database into three separate datasets for each of the standard database used namely Dataset1 (TrainDatabase1 and ProbeDatabase1), Dataset2 (TrainDatabase2 and ProbeDatabase2), and Dataset 3 (TrainDatabase3and ProbeDatabase3). The TrainDatabases 1,2 and 3 contained three, five and ten images per person(class) respectively whereas the ProbeDatabases 1,2 and 3 contained one selected image per person from their respective TrainDatabases. Hence the datasets which were formed from AT &T Database and Indian Database were as follows;

AT & T DATABASE

DATASETS	NO. OF IMAGES
Dataset 1:	
TrainDatabase1	120 (3 images per person)
ProbeDatabase1	40
Dataset 2:	
TrainDatabase2	200 (5 images per person)
ProbeDatabase2	40
Dataset 3:	
TrainDatabase3	400 (10 images person)
ProbeDatabase3	40

INDIAN FACE DATABASE

DATASETS	NO. OF IMAGES
Dataset 1:	
TrainDatabase1	180 (3 images per person)
ProbeDatabase1	50
Dataset 2:	
TrainDatabase2	300 (5 images per person)
ProbeDatabase2	50
Dataset 3:	
TrainDatabase3	600(10 images per person)
ProbeDatabase3	50

3.2 TRAINING PHASE

In this phase the algorithms of PCA and LDA were employed. After adding face images to the initially empty training databases, the PCA or LDA algorithm was initialized by feeding it a set of training image of faces. The resized images (60x 50 pixels) were each reshaped into 1 D column vector (3000x1) by concatenating each column pixel of each 2D image one to another. These 1 D vectors were put together into 2 D matrix “T” called the Training set .The Training set, containing all training image vectors, was used to define the face space . Matlab functions called EigenfaceCore in the case of PCA algorithm and Fisherface Core for LDA algorithm were created to extract the intrinsic features of the images.

The EigenfaceCore performed the following activities;

- Mean face calculation (arithmetic average of the training image vectors at each pixel point),
- difference matrix /mean centred (where mean face vector was subtracted from each training vector),
- Covariance matrix (dot product of the transpose of difference matrix and the difference matrix itself). Then used *eig function* in MATlAB to produce eigenvectors and their corresponding eigenvalues.These were sorted eliminating vectors with less than or equal to zero eigenvalues.
- Eigenface space (dot product of the eigenvector of the covariance matrix and the difference matrix)

- Projection matrix (The matrix of mean centred images were then projected onto the eigenfacespace). That is, the dot product of the eigenfaces and the difference matrix. The output of the preceding activity served as input for the activity that followed.

The output of EigenfaceCorefunction (the mean vector of the training database, matrix of mean centred image vectors and Eigen vectors of the covariance matrix of the trainingdatabase became the input for recognition function. The sample patterns or images of the output are shown in figures 3-6, 3-7, and 3-8.

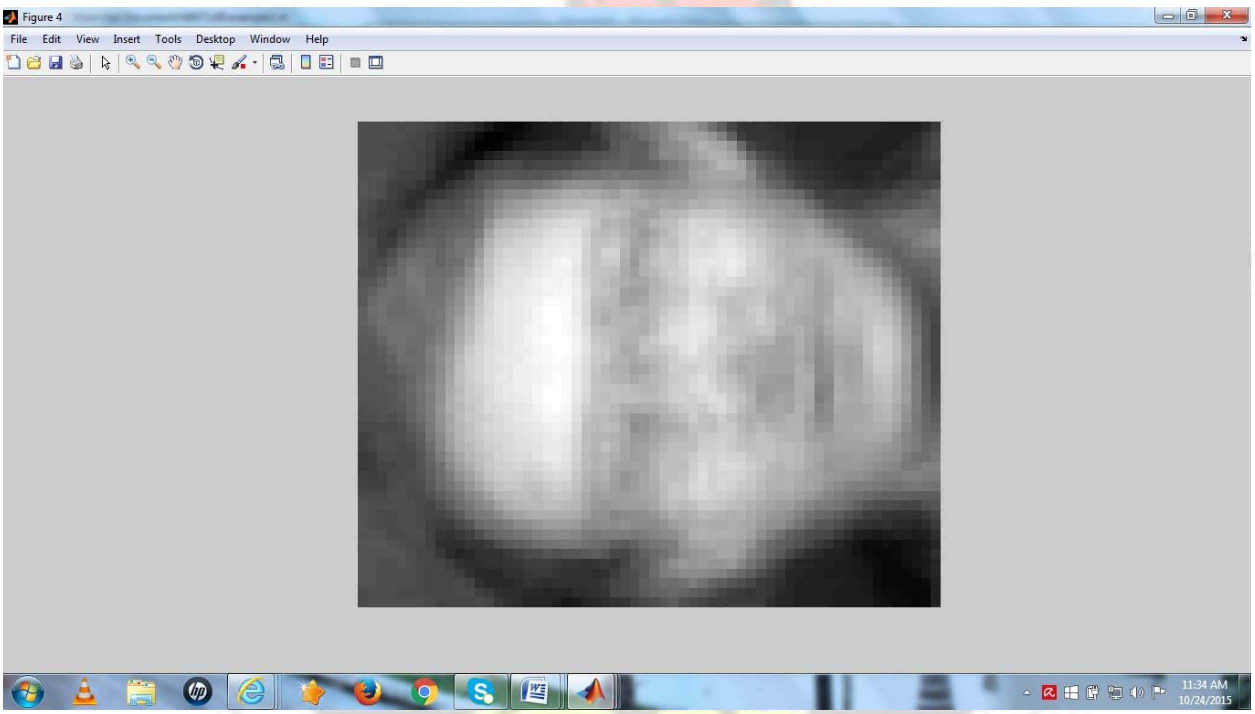


Figure 3-6*Mean image(pattern)*

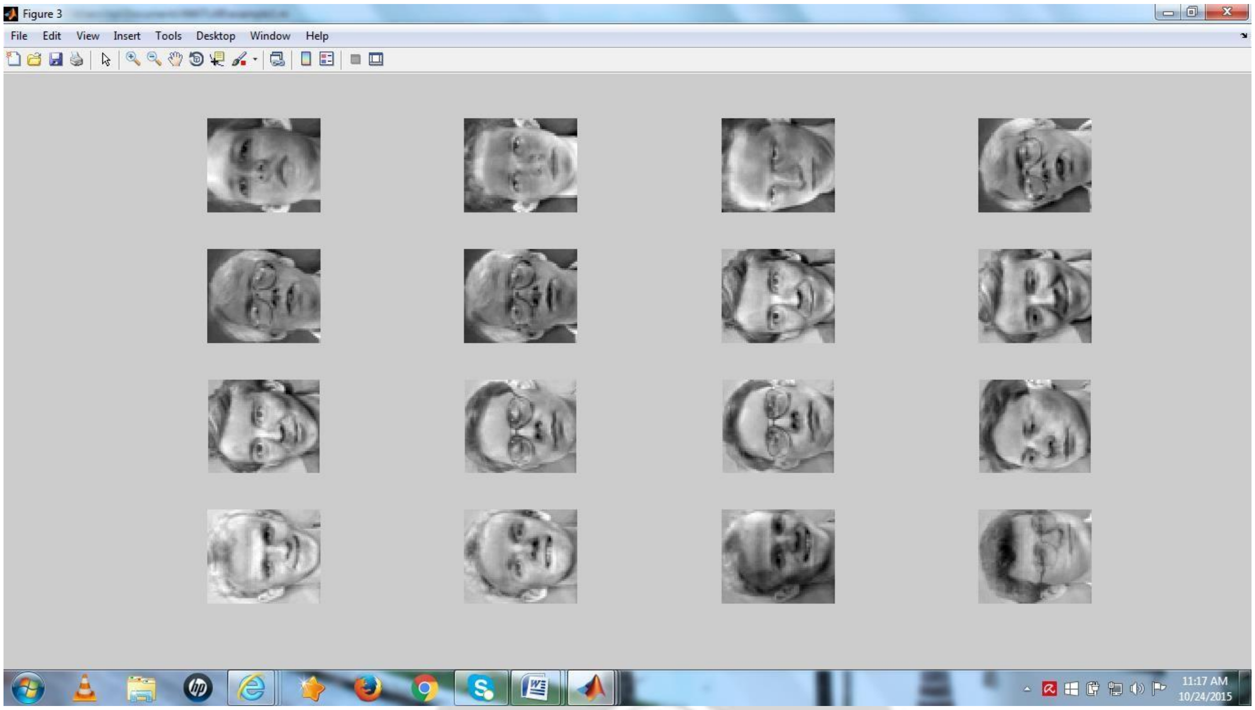


Figure 3-7*Sample of Mean Centered Image (pattern)*

Figure 3-8 Samples of *Eigenfaces(eigen pattern)*

The Fisherface Core

The eigenface projections of PCA transformation were used in the Subspace LDA method instead of using the pixel values of the images.

- At first, centred images were mapped onto a (P-C) linear subspace as was the case in EigenfaceCore. Here it was P-C linear subspace in that the eigenvectors were sorted by their corresponding eigenvalues and those with small eigenvalues were eliminated leaving up to (P-C) number of eigenvectors to form the eigenspace.
- The projected images were then projected onto a (C-1) linear subspace, so that images of the same class (or person) move closer together and images of difference classes move further apart.
    - Calculated the mean of each class in the eigenspace (projection class mean) o Calculated the total mean in the eigenspace o Performed the within scatter matrix by subtracting the eigenfaces class mean from each of the projected images of PCA within a class. Then did a dot product of the result and its transpose.
    - Performed the between scatter matrix by subtracting the eigenfaces mean(overall mean) from projection class mean and did dot product of the result and its transpose.
    - Calculated fisher discriminant basis (linear fisher space) by maximising the between class while minimising the within class. That was done by using *eig function* in Matlab which produced eigenvectors and their corresponding eigenvalues.
    - The eigenvectors with zero eigen values were eliminated leaving C-1 number of eigenvectors to form a matrix.

  - Projected the projected images of PCA onto the linear fisher subspace by doing a dot product of the transpose of the fisherface (matrix) and the projected images of PCA.

- These activities produced four outputs(the Mean of the Training set, Eigen vectors of the covariance matrix of the training database, Fisherface space and projected fisherface) which were used in recognition phase as input. Figures 3-9,3-10, and 3-11 represent sample patterns or images of some the outputs.

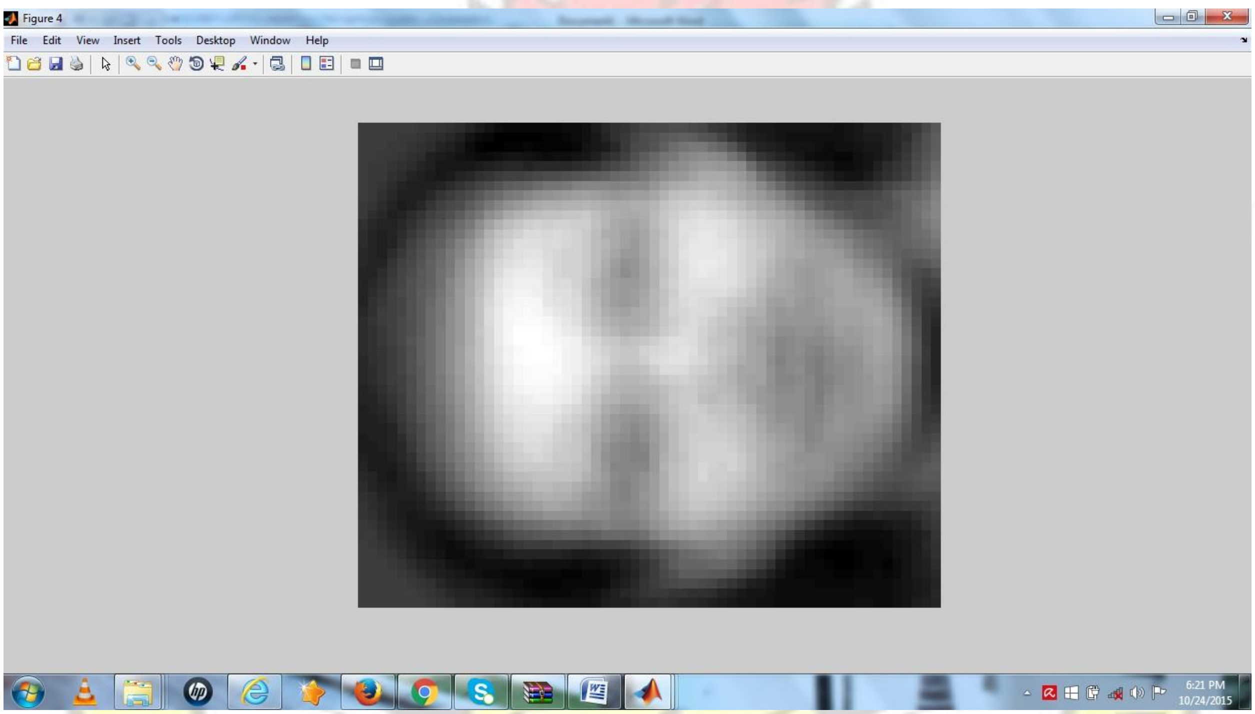


Figure 3-9*Mean image (parttern) of the training set*

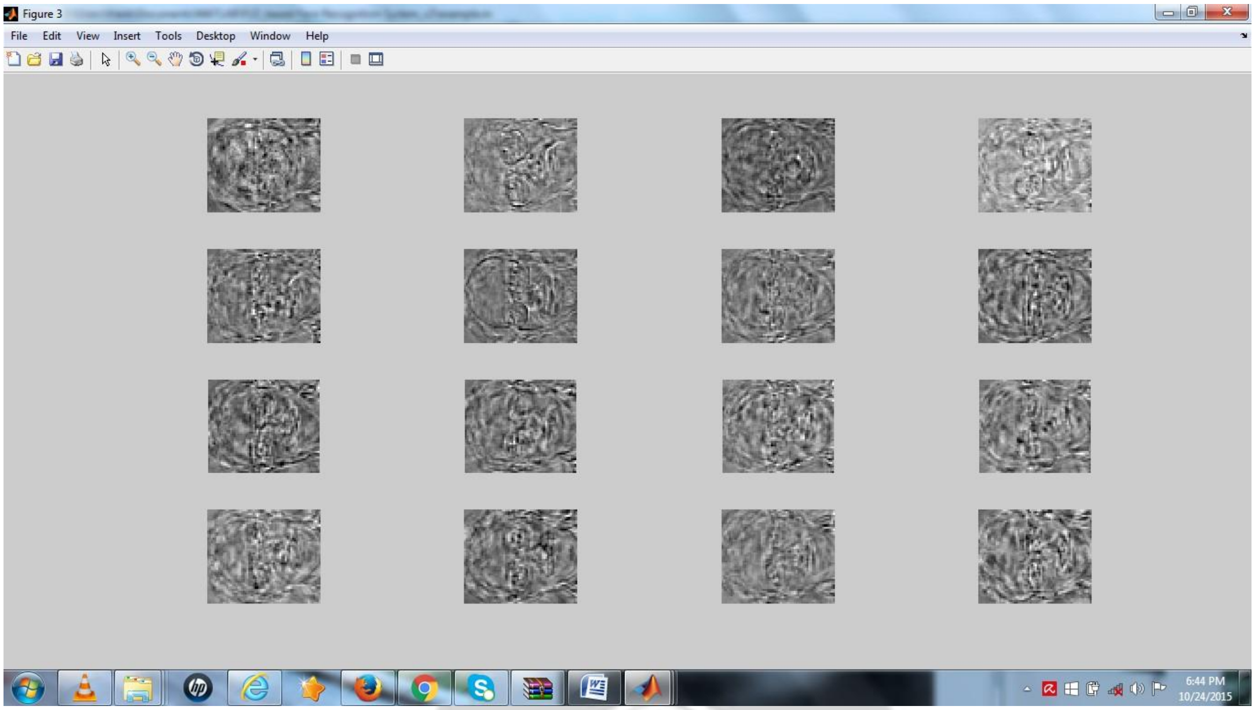


Figure 3-10*sample of Eigenface of the covariance of the training set*

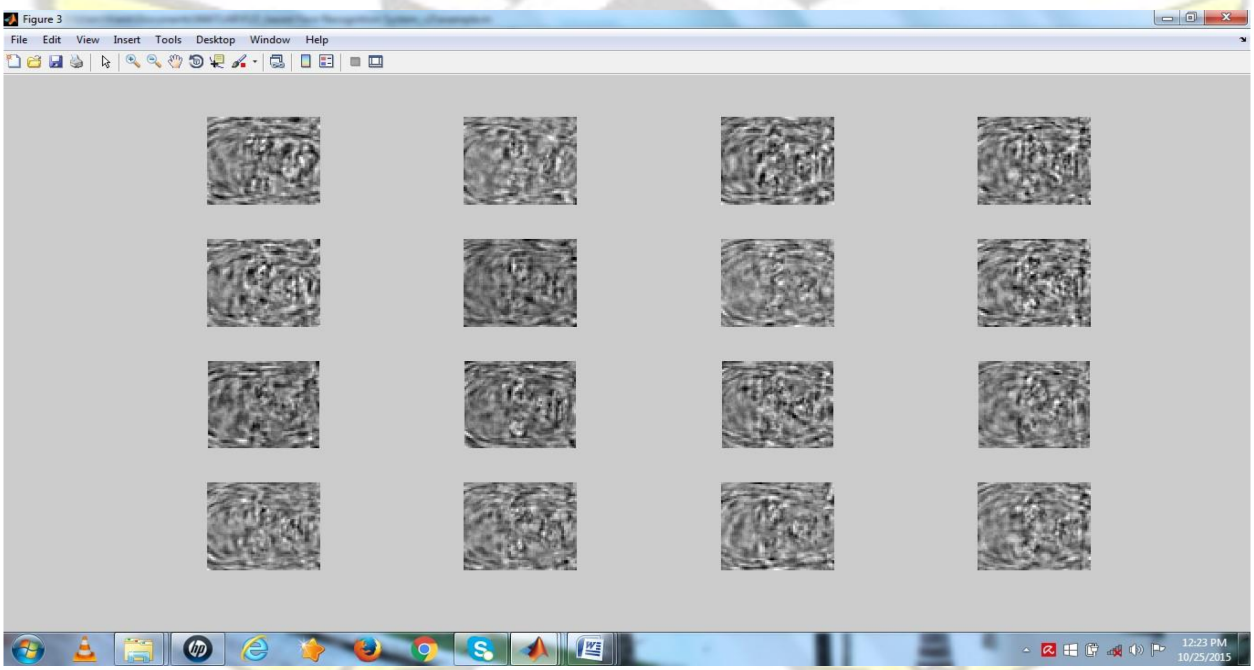


Figure 3-11*Fisherface(parttern)*

3.3 RECOGNITION (CLASSIFICATION) PHASE

At this stage, the algorithm performed identification of unknown image (test/probe image). The test (probe) image was also projected over eigenface space (in the case of PCA) or fisherface space (in the case of LDA) like that of training images in the training phase.

- The test image with same size of 60x 50 pixels was reshaped into column vector
- It's column vector was mean subtracted and projected onto the eigenface space or fisherface space
- The projected test image was then classified (recognized). That is, the distance between the projected test image and all the training images in eigenface space or fisherface space using any of the three distance or similarity measures (Euclidean distance, City Block, Cosine). Recognition function was created in MATLAB to compare test image and all the images in the training set to obtain equivalent image in the training database. The figure 3-12 and figure 3-13 are the illustrations of the result of the recognition phase. They showed the time it took the projection-metric combination algorithm being used to recognise a face or identify a match of the input image in the database. They also showed the similarity measure or metric used and the filename of the equivalent image.

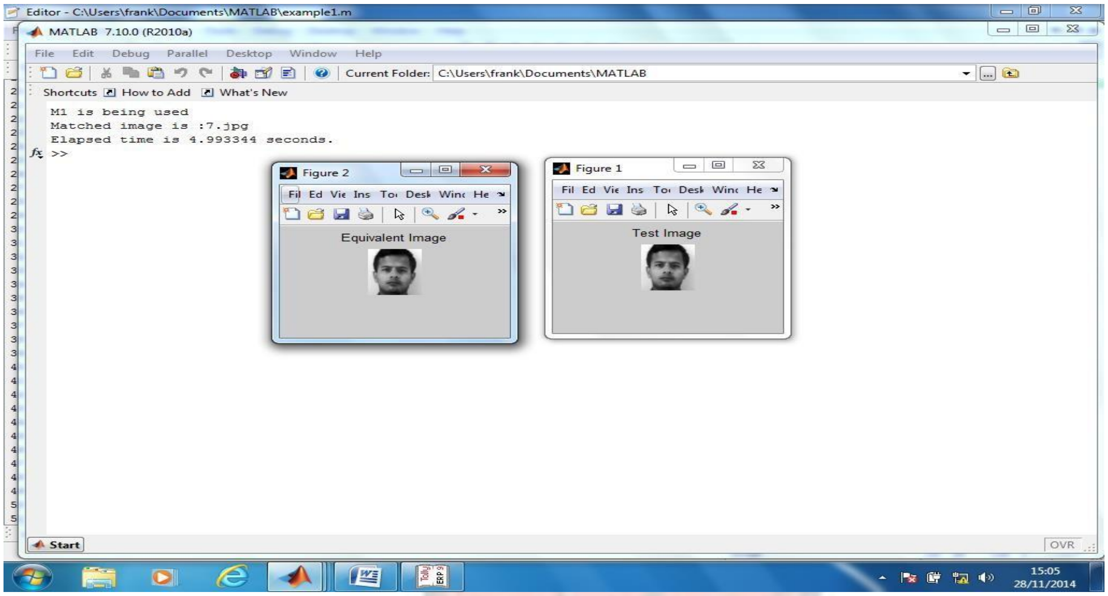


Figure 3-12 : A sample of results of Indian Face tested

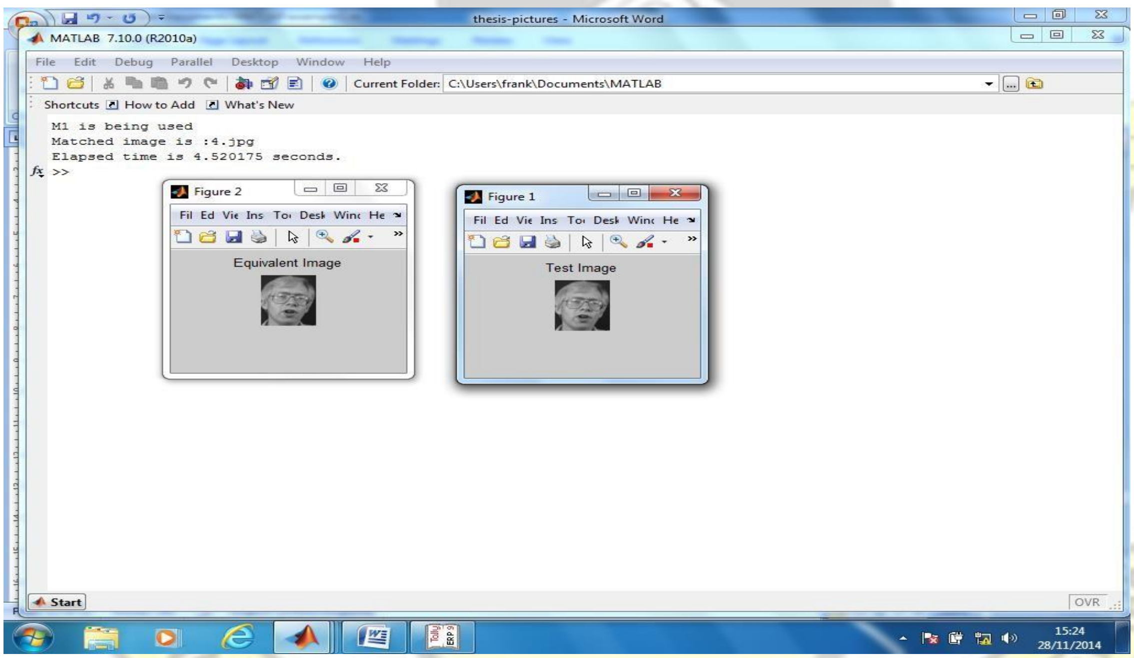


Figure3-13: A sample of results of AT& T face image tested.

3.4 PERFORMANCE EVALUATION

At this stage the performance of all projection-metric combined algorithms were assessed to ascertain the most efficient.

The analysis was done based on the following parameters:-

- Accuracy(Recognition rate)= $\frac{\text{Number of correctly matched images}}{\text{Total number of probe image}}\times 100$
- Execution time = time taken (in seconds) for execution of algorithm
- Average Execution Time of an algorithm=Sum of Execution Time/ Total number of probe image
- Effect of varying number of training image, and number of images of person (class) which will include the generalization ability of the projection-metric combination.

The face recognition experiments were done with six different projection-metric algorithms viz; PCA-Euclidean distance metric, PCA-City Block metric, PCA- Cosine metric, LDA-Euclidean distance, LDA-City Block and LDA-Cosine metric. Each of the algorithms was executed using the three separate datasets of eachof the face Database. Thus, 18experiments (scenarios) were performed with each of the two face database;

- PCA-Euclidean distance metric using dataset1
- PCA-Euclidean distance metric using dataset2
- PCA-Euclidean distance metric using dataset3
- PCA-City Block metric using dataset1
- PCA-City Block metric using dataset2
- PCA-City Block metric using dataset 3

- PCA- Cosine metric using dataset 1
- PCA- Cosine metric using dataset 2
- PCA- Cosine metric using dataset 3
- LDA-Euclidean distance metric using dataset1
- LDA-Euclidean distance metric using dataset2
- LDA-Euclidean distance metric using dataset3
- LDA-City Block metric using dataset1
- LDA-City Block metric using dataset2
- LDA-City Block metric using dataset3
- LDA- Cosine metric using dataset 1
- LDA- Cosine metric using dataset 2
- LDA- Cosine metric using dataset 3

The purpose was to observe the behaviour of each projection-metric as the sample size changes. The results of the experiments (scenarios) indicated in the tables 3-1,3-2,3-3,3-4,3-5,3-6,3-7,38,3-9,3-10,3-11,and 3-12 showed the time taken for each algorithm to recognise a probe(input) image and also indicated whether there was a true match or not. Each of the tables had three sub- tables showing the results when the datasets 1, 2, and 3 were used .Tables 3-1 to 3-6 contained results when AT& T Database was used whereas Tables 3-7 to 3-12 had outcome of Indian Face Database.

TABLE 3-1: **PCA-Euclidean Distance Metric with AT & T Database (ORCL)**

A. DATASET1:(TrainDatabase1 and ProbeDatabase1)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	3.387809	TRUE
2	3.742150	TRUE
3	3.079408	TRUE
4	3.624291	TRUE
5	3.797418	TRUE
6	2.850317	TRUE
7	3.275675	TRUE
8	3.172545	TRUE
9	3.447010	TRUE
10	2.950193	TRUE
11	3.120583	TRUE
12	3.120158	TRUE
13	3.088432	TRUE
14	3.155916	TRUE
15	3.041090	TRUE
16	3.12578	TRUE
17	3.495705	TRUE
18	3.323802	TRUE
19	3.015211	TRUE
20	3.091216	TRUE
21	3.005699	TRUE
22	2.864488	TRUE
23	2.947509	TRUE
24	3.277394	TRUE
25	3.102272	TRUE
26	3.145623	TRUE
27	3.125023	TRUE
28	3.041468	TRUE

B. DATASET 2:(TrainDatabase2 and ProbeDatabase2)

Probe Image No	Time Taken for execution(s econds)	Accurate Match?
1	6.199392	TRUE
2	4.882568	TRUE
3	4.005939	TRUE
4	4.229203	TRUE
5	4.22299	TRUE
6	4.289686	TRUE
7	4.145401	TRUE
8	4.263738	TRUE
9	4.311963	TRUE
10	4.652786	TRUE
11	4.349059	TRUE
12	4.670744	TRUE
13	4.602386	TRUE
14	4.350037	TRUE
15	4.211216	TRUE
16	4.373455	TRUE
17	4.512604	TRUE
18	4.332659	TRUE
19	4.256888	TRUE
20	4.379846	TRUE
21	4.125967	TRUE
22	4.782809	TRUE
23	4.23728	TRUE
24	4.554918	TRUE
25	4.376593	TRUE
26	4.166089	TRUE
27	4.066083	TRUE
28	4.623079	TRUE

C. DATASET 3:(TrainDatabase3 and ProbeDatabase3)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	10.747025	TRUE
2	10.217329	TRUE
3	10.83966	TRUE
4	11.809159	TRUE
5	10.464620	TRUE
6	10.547158	TRUE
7	10.801520	TRUE
8	11.211181	TRUE
9	10.372905	TRUE
10	10.411739	TRUE
11	10.622265	TRUE
12	10.522938	TRUE
13	10.620901	TRUE
14	10.578699	TRUE
15	10.226738	TRUE
16	10.595028	TRUE
17	10.474972	TRUE
18	10.713437	TRUE
19	10.485615	TRUE
20	10.890577	TRUE
21	10.506244	TRUE
22	10.544416	TRUE
23	10.462235	TRUE
24	10.920184	TRUE
25	10.368547	TRUE
26	10.566980	TRUE
27	10.166680	TRUE
28	11.312205	TRUE

29	3.014945	TRUE
30	2.878306	TRUE
31	3.037604	TRUE
32	2.877320	TRUE
33	3.674658	TRUE
34	2.956951	TRUE
35	3.020544	TRUE
36	3.430904	TRUE
37	3.431960	TRUE
38	3.483968	TRUE
39	3.164425	TRUE
40	3.512282	TRUE

29	4.421554	TRUE
30	4.283858	TRUE
31	4.317274	TRUE
32	4.589235	TRUE
33	4.231435	TRUE
34	4.502592	TRUE
35	4.689929	TRUE
36	4.239229	TRUE
37	4.632180	TRUE
38	4.332776	TRUE
39	4.779560	TRUE
40	4.277536	TRUE

29	10.255051	TRUE
30	10.561249	TRUE
31	10.604536	TRUE
32	10.509406	TRUE
33	10.388799	TRUE
34	10.141598	TRUE
35	10.869940	TRUE
36	10.878715	TRUE
37	10.683394	TRUE
38	10.702813	TRUE
39	10.580622	TRUE
40	10.797230	TRUE

TABLE 3-2 : PCA-City Block Metric with AT & T Database (ORCL)

A. DATASET 1:(TrainDatabse1 ProbeDatabase1)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	3.815805	TRUE
2	3.220672	TRUE
3	3.004956	TRUE
4	2.983798	TRUE
5	3.294745	TRUE
6	3.612958	TRUE
7	3.364692	TRUE
8	3.201083	TRUE
9	3.093549	TRUE
10	3.145258	TRUE
11	2.805174	TRUE
12	3.203232	TRUE
13	2.996518	TRUE
14	3.193465	TRUE
15	2.869927	TRUE
16	4.965532	TRUE
17	3.066402	TRUE
18	2.937208	TRUE
19	3.083524	TRUE
20	3.012861	TRUE
21	2.994545	TRUE
22	2.231299	TRUE
23	3.158518	TRUE
24	3.699593	TRUE

B. DATASET 2:(TrainDatabase2 and ProbeDatabase2)

Probe Image No	Time Taken for execution (seconds)	Accurate Match?
1	4.792229	TRUE
2	4.462534	TRUE
3	4.494986	TRUE
4	4.605092	TRUE
5	4.624663	TRUE
6	5.063578	TRUE
7	4.203702	TRUE
8	4.247873	TRUE
9	4.606905	TRUE
10	4.162258	TRUE
11	4.534263	TRUE
12	4.849747	TRUE
13	4.755029	TRUE
14	5.091634	TRUE
15	4.231955	TRUE
16	4.236752	TRUE
17	4.477539	TRUE
18	4.349985	TRUE
19	4.744701	TRUE
20	4.555601	TRUE
21	4.399288	TRUE
22	4.269119	TRUE
23	4.628683	TRUE
24	4.218123	TRUE

C. DATASET 3:(TrainDatabase3 and and ProbeDatabase3)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	10.6999	TRUE
2	10.518788	TRUE
3	10.434775	TRUE
4	10.647362	TRUE
5	10.848006	TRUE
6	10.267886	TRUE
7	10.650416	TRUE
8	10.536174	TRUE
9	10.426495	TRUE
10	10.532789	TRUE
11	10.933835	TRUE
12	10.442807	TRUE
13	10.854353	TRUE
14	11.657375	TRUE
15	10.613492	TRUE
16	10.747744	TRUE
17	10.467031	TRUE
18	10.739662	TRUE
19	19.622385	TRUE
20	10.397150	TRUE
21	10.309721	TRUE
22	10.552889	TRUE
23	10.423017	TRUE
24	10.448617	TRUE

25	3.024532	TRUE	25	4.689941	TRUE	25	10.714099	TRUE
26	3.306026	TRUE	26	4.148882	TRUE	26	10.470229	TRUE
27	3.083005	TRUE	27	4.631362	TRUE	27	11.244761	TRUE
28	3.140326	TRUE	28	4.218329	TRUE	28	10.649156	TRUE
29	2.793348	TRUE	29	4.402027	TRUE	29	10.679245	TRUE
30	3.212198	TRUE	30	5.409675	TRUE	30	10.573037	TRUE
31	3.111070	TRUE	31	4.532874	TRUE	31	10.691213	TRUE
32	3.131102	TRUE	32	4.434809	TRUE	32	11.052534	TRUE
33	3.017982	TRUE	33	4.276814	TRUE	33	10.362203	TRUE
34	3.203795	TRUE	34	4.349428	TRUE	34	10.634145	TRUE
35	3.96982	TRUE	35	4.437285	TRUE	35	10.297944	TRUE
36	3.457809	TRUE	36	5.006930	TRUE	36	11.039128	TRUE
37	3.217827	TRUE	37	4.225123	TRUE	37	10.938179	TRUE
38	3.32164	TRUE	38	4.436989	TRUE	38	10.836574	TRUE
39	2.946091	TRUE	39	4.543573	TRUE	39	11.066268	TRUE
40	3.077716	TRUE	40	4.311780	TRUE	40	10.735413	TRUE

TABLE 3-3: PCA-Cosine Metric with AT & T Database(ORCL)

DATASET 1:(TrainDatabse1 and ProbeDatabase1)			DATASET 2:(TrainDatabase2 and ProbeDatabase2)			DATASET 3:(TrainDatabase3 and ProbeDatabase3)		
Probe Image No	**Time Taken for execution (seconds)**	**Accurate Match?**	**Probe Image No**	**Time Taken for execution (seconds)**	**Accurat e Match?**	**Probe Image No**	**Time Taken for execution(s econds)**	**Accurate Match?**
1	3.149788	TRUE	1	4.784104	TRUE	1	10.78498	TRUE
2	3.087177	TRUE	2	4.481632	TRUE	2	10.725957	TRUE
3	2.963325	TRUE	3	4.815212	TRUE	3	10.49135	TRUE
4	3.126824	TRUE	4	4.410216	TRUE	4	11.194527	TRUE
5	2.972944	TRUE	5	4.503234	TRUE	5	10.772121	TRUE
6	2.823131	TRUE	6	4.665665	TRUE	6	10.412511	TRUE
7	2.950871	TRUE	7	5.069794	TRUE	7	11.149374	TRUE

8	2.997732	TRUE
9	3.037472	TRUE
10	3.260838	TRUE
11	2.924637	TRUE
12	2.836883	TRUE
13	2.994644	TRUE
14	3.030633	TRUE
15	3.147532	TRUE
16	3.454107	TRUE
17	2.861964	TRUE
18	3.104392	TRUE
19	3.0522	TRUE
20	3.251692	TRUE
21	3.046548	TRUE
22	3.143808	TRUE
23	2.9993	TRUE
24	3.087922	TRUE
25	3.090558	TRUE
26	3.081468	TRUE
27	2.897062	TRUE
28	3.056792	TRUE
29	2.941583	TRUE
30	2.847241	TRUE
31	3.021364	TRUE
32	2.884107	TRUE
33	2.955912	TRUE
34	3.407669	TRUE
35	3.199618	TRUE
36	3.004693	TRUE
37	2.933879	TRUE
38	3.750519	TRUE
39	3.104055	TRUE

8	4.185536	TRUE
9	4.575336	TRUE
10	4.362409	TRUE
11	4.227024	TRUE
12	4.998078	TRUE
13	4.312065	TRUE
14	4.395274	TRUE
15	4.401022	TRUE
16	4.338421	TRUE
17	4.197966	TRUE
18	4.430459	TRUE
19	4.184457	TRUE
20	4.434734	TRUE
21	4.275735	TRUE
22	4.511059	TRUE
23	4.323197	TRUE
24	4.198576	TRUE
25	4.356409	TRUE
26	5.399184	TRUE
27	4.488412	TRUE
28	4.510312	TRUE
29	4.305142	TRUE
30	4.103614	TRUE
31	4.254114	TRUE
32	4.366362	TRUE
33	4.151751	TRUE
34	4.4019	TRUE
35	4.760127	TRUE
36	4.85864	TRUE
37	4.570181	TRUE
38	5.095261	TRUE
39	4.337537	TRUE

8	10.808878	TRUE
9	10.700226	TRUE
10	11.072855	TRUE
11	10.306871	TRUE
12	10.634222	TRUE
13	10.817018	TRUE
14	10.717712	TRUE
15	10.576304	TRUE
16	10.742547	TRUE
17	11.143422	TRUE
18	10.587206	TRUE
19	10.515557	TRUE
20	10.561156	TRUE
21	10.500028	TRUE
22	10.207747	TRUE
23	10.408219	TRUE
24	10.577213	TRUE
25	11.633234	TRUE
26	10.65942	TRUE
27	10.3967	TRUE
28	10.522918	TRUE
29	10.87223	TRUE
30	11.65033	TRUE
31	10.457521	TRUE
32	10.930499	TRUE
33	10.884046	TRUE
34	10.507061	TRUE
35	10.546309	TRUE
36	10.304872	TRUE
37	10.719682	TRUE
38	10.716691	TRUE
39	10.535086	TRUE

40	3.121596	TRUE

40	4.22444	TRUE

40	10.525399	TRUE

TABLE 3-4: LDA-Euclidean Distance Metric with AT & T DATABASE (ORCL)

A. DATASET 1:(TrainDatabse1 and ProbeDatabase 1)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	2.616401	TRUE
2	2.688212	TRUE
3	2.554928	TRUE
4	2.547041	TRUE
5	2.445618	TRUE
6	2.704497	TRUE
7	2.547633	TRUE
8	2.572079	TRUE
9	2.669552	TRUE
10	2.578299	TRUE
11	2.264397	TRUE
12	2.461774	TRUE
13	2.71105	TRUE
14	2.502796	TRUE
15	2.301462	TRUE
16	2.489078	TRUE
17	2.555657	TRUE
18	3.15319	TRUE
19	2.601186	TRUE
20	2.526701	TRUE
21	2.614893	TRUE
22	2.619951	TRUE
23	2.699686	TRUE

B. DATASET 2:(TrainDatabase2 and ProbeDatabase2)

Probe Image No	Time Taken for execution(s econds)	Accurate Match?
1	3.643723	TRUE
2	3.467293	TRUE
3	3.507806	TRUE
4	3.508931	TRUE
5	3.556659	TRUE
6	3.635238	TRUE
7	3.446174	TRUE
8	3.344759	TRUE
9	3.386641	TRUE
10	3.264331	TRUE
11	3.443756	TRUE
12	3.517448	TRUE
13	3.434154	TRUE
14	3.507045	TRUE
15	3.467744	TRUE
16	3.26615	TRUE
17	3.428693	TRUE
18	3.355305	TRUE
19	3.407223	TRUE
20	3.479969	TRUE
21	3.933312	TRUE
22	3.705266	TRUE
23	3.594477	TRUE

C. DATASET 3:(TrainDatabase3 and ProbeDatabase and ProbeDatabase3)

Probe Image No	Time Taken for execution(sec onds)	Accurate Match?
1	8.676843	TRUE
2	8.79752	TRUE
3	8.659363	TRUE
4	8.67988	TRUE
5	8.510325	TRUE
6	8.810351	TRUE
7	8.749874	TRUE
8	8.583128	TRUE
9	8.425483	TRUE
10	8.822075	TRUE
11	8.786645	TRUE
12	8.650401	TRUE
13	8.458793	TRUE
14	8.854728	TRUE
15	8.850784	TRUE
16	8.791175	TRUE
17	8.597961	TRUE
18	8.86352	TRUE
19	8.739849	TRUE
20	8.405774	TRUE
21	8.615037	TRUE
22	8.847655	TRUE
23	8.649679	TRUE

24	2.502599	TRUE
25	2.392634	TRUE
26	2.721057	TRUE
27	2.718404	TRUE
28	2.626726	TRUE
29	2.627311	TRUE
30	2.682098	TRUE
31	2.568213	TRUE
32	2.568266	TRUE
33	2.733643	TRUE
34	2.456988	TRUE
35	2.696131	TRUE
36	2.657109	TRUE
37	2.542772	TRUE
38	2.55825	TRUE
39	2.538965	TRUE
40	2.644345	TRUE

24	3.486311	TRUE
25	3.47026	TRUE
26	3.418194	TRUE
27	3.32057	TRUE
28	3.40714	TRUE
29	3.856139	TRUE
30	3.558373	TRUE
31	3.929814	TRUE
32	3.674359	TRUE
33	3.660752	TRUE
34	3.386646	TRUE
35	3.609328	TRUE
36	3.298353	TRUE
37	3.504017	TRUE
38	3.469441	TRUE
39	3.708737	TRUE
40	3.566227	TRUE

24	8.841214	TRUE
25	8.556384	TRUE
26	8.733357	TRUE
27	8.440549	TRUE
28	8.599011	TRUE
29	8.72144	TRUE
30	8.730488	TRUE
31	8.840073	TRUE
32	8.74111	TRUE
33	8.582724	TRUE
34	8.635012	TRUE
35	8.994335	TRUE
36	8.752872	TRUE
37	8.780226	TRUE
38	8.737237	TRUE
39	8.558103	TRUE
40	8.715839	TRUE

TABLE 3-5: LDA-City Block Metric with AT & T DATABASE (ORCL)

A. DATASET 1:(TrainDatabse1

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	2.683808	TRUE
2	2.623955	TRUE
3	2.456868	TRUE
4	2.654625	TRUE
5	2.338997	TRUE
6	2.38201	TRUE
7	2.620032	TRUE
8	2.617033	TRUE
9	3.027645	TRUE
10	2.459814	TRUE
11	2.550304	TRUE
12	2.575161	TRUE
13	2.644464	TRUE
14	2.632398	TRUE
15	2.302639	TRUE
16	2.956964	TRUE
17	2.479205	TRUE
18	2.660442	TRUE
19	2.470531	TRUE
20	2.307654	TRUE
21	2.367121	TRUE
22	2.618509	TRUE
23	2.859683	TRUE
24	2.584013	TRUE
25	2.531743	TRUE

B. DATASET 2:(TrainDatabase2 and ProbeDatabase2)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	3.471813	TRUE
2	3.26532	TRUE
3	3.466388	TRUE
4	3.256301	TRUE
5	3.426299	TRUE
6	3.536434	TRUE
7	3.798512	TRUE
8	3.31358	TRUE
9	3.402614	TRUE
10	3.43864	TRUE
11	3.610426	TRUE
12	3.440175	TRUE
13	3.41405	TRUE
14	3.474463	TRUE
15	3.279259	TRUE
16	3.469554	TRUE
17	3.450654	TRUE
18	3.558337	TRUE
19	3.457527	TRUE
20	3.914527	TRUE
21	3.894266	TRUE
22	3.326095	TRUE
23	3.225233	TRUE
24	3.309113	TRUE
25	3.367712	TRUE

C. DATASET 3:(TrainDatabase3 and ProbeDatabase 1) and ProbeDatabase3)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	9.952712	TRUE
2	9.621063	TRUE
3	9.919887	TRUE
4	9.203231	TRUE
5	9.468395	TRUE
6	9.40687	TRUE
7	9.007414	TRUE
8	8.716742	TRUE
9	9.124135	TRUE
10	9.193127	TRUE
11	9.269927	TRUE
12	9.768988	TRUE
13	8.699175	TRUE
14	9.062414	TRUE
15	9.588157	TRUE
16	8.849826	TRUE
17	9.273945	TRUE
18	9.264054	TRUE
19	9.351256	TRUE
20	9.28597	TRUE
21	9.941014	TRUE
22	9.443023	TRUE
23	9.463625	TRUE
24	9.521923	TRUE
25	9.168985	TRUE

26	2.341038	TRUE
27	2.66253	TRUE
28	2.637909	TRUE
29	2.557023	TRUE
30	2.793261	TRUE
31	2.932615	TRUE
32	2.547181	TRUE
33	2.446462	TRUE
34	2.6951	TRUE
35	2.624119	TRUE
36	2.533864	TRUE
37	2.518162	TRUE
38	2.610422	TRUE
39	2.469275	TRUE
40	2.349165	TRUE

26	3.738258	TRUE
27	3.427917	TRUE
28	3.517387	TRUE
29	3.141424	TRUE
30	3.273737	TRUE
31	3.517637	TRUE
32	3.522941	TRUE
33	3.342824	TRUE
34	3.501278	TRUE
35	3.242454	TRUE
36	3.538406	TRUE
37	3.189021	TRUE
38	3.376476	TRUE
39	3.453794	TRUE
40	3.208606	TRUE

26	9.403219	TRUE
27	9.431243	TRUE
28	9.214155	TRUE
29	8.979578	TRUE
30	8.969958	TRUE
31	9.227252	TRUE
32	9.406117	TRUE
33	8.880846	TRUE
34	9.107096	TRUE
35	9.459821	TRUE
36	8.951551	TRUE
37	8.94249	TRUE
38	9.330184	TRUE
39	8.843963	TRUE
40	9.326814	TRUE

TABLE 3-6:LDA-Cosine Metric with AT & T Database (ORCL)

A. DATASET 1:(TrainDatabse1 and ProbeDatabase 1)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	2.646833	TRUE
2	2.529745	TRUE
3	2.747280	TRUE
4	2.582507	TRUE
5	2.466938	TRUE
6	2.417554	TRUE
7	2.402778	TRUE
8	2.443574	TRUE

B. DATASET 2:(TrainDatabase2 and ProbeDatabase2)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	3.469847	TRUE
2	3.527578	TRUE
3	3.562090	TRUE
4	3.448387	TRUE
5	3.304540	TRUE
6	3.474156	TRUE
7	3.400341	TRUE
8	4.076774	TRUE

C. DATASET 3:(TrainDatabase3 and ProbeDatabase3)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	8.988299	TRUE
2	8.403586	TRUE
3	8.416943	TRUE
4	8.559919	TRUE
5	8.694248	TRUE
6	8.369293	TRUE
7	8.655166	TRUE
8	8.423217	TRUE

9	2.516932	TRUE	9	4.200393	TRUE	9	8.432000	TRUE
10	2.518293	TRUE	10	3.494763	TRUE	10	9.052079	TRUE
11	2.473127	TRUE	11	3.152525	TRUE	11	8.622524	TRUE
12	2.639395	TRUE	12	3.177151	TRUE	12	8.444848	TRUE
13	2.915481	TRUE	13	3.862672	TRUE	13	8.399831	TRUE
14	2.701481	TRUE	14	3.744191	TRUE	14	8.686765	TRUE
15	2.476059	TRUE	15	3.851988	TRUE	15	8.618898	TRUE
16	2.805214	TRUE	16	3.546483	TRUE	16	8.537377	TRUE
17	2.554217	TRUE	17	3.551382	TRUE	17	8.471883	TRUE
18	2.448394	TRUE	18	3.746355	TRUE	18	8.683656	TRUE
19	2.402995	TRUE	19	3.407273	TRUE	19	8.427342	TRUE
20	2.521781	TRUE	20	3.438719	TRUE	20	8.665033	TRUE
21	2.940825	TRUE	21	3.884675	TRUE	21	8.621569	TRUE
22	2.579409	TRUE	22	3.464799	TRUE	22	8.634842	TRUE
23	2.958157	TRUE	23	3.714693	TRUE	23	8.680259	TRUE
24	2.503974	TRUE	24	3.611615	TRUE	24	8.821395	TRUE
25	2.646904	TRUE	25	3.390774	TRUE	25	8.211373	TRUE
26	2.353053	TRUE	26	3.372391	TRUE	26	8.411496	TRUE
27	2.476928	TRUE	27	3.407282	TRUE	27	8.623107	TRUE
28	2.454916	TRUE	28	3.453018	TRUE	28	8.609473	TRUE
29	2.584033	TRUE	29	3.940491	TRUE	29	8.906372	TRUE
30	2.330673	TRUE	30	3.431175	TRUE	30	8.382765	TRUE
31	2.464837	TRUE	31	3.561772	TRUE	31	8.580778	TRUE
32	2.613539	TRUE	32	3.602517	TRUE	32	8.624877	TRUE
33	2.403421	TRUE	33	3.773680	TRUE	33	8.381591	TRUE
34	2.655523	TRUE	34	3.472985	TRUE	34	8.319010	TRUE
35	2.775549	TRUE	35	3.375096	TRUE	35	8.581494	TRUE
36	2.666952	TRUE	36	3.444505	TRUE	36	8.541000	TRUE
37	2.408470	TRUE	37	3.756154	TRUE	37	8.353537	TRUE
38	2.605531	TRUE	38	3.607921	TRUE	38	8.532827	TRUE
39	2.455309	TRUE	39	3.378203	TRUE	39	8.713640	TRUE
40	2.631668	TRUE	40	3.300552	TRUE	40	8.817077	TRUE

TABLE 3-7: PCA-Euclidean Distance Metric with Indian Face Database

A. DATASET 1:(TrainDatabse1 and ProbeDatabase 1)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	9.055789	TRUE
2	4.919944	TRUE
3	5.712822	TRUE
4	5.115144	TRUE
5	5.251785	TRUE
6	5.098103	TRUE
7	5.076524	TRUE
8	5.381432	TRUE
9	6.729868	TRUE
10	5.046638	TRUE
11	4.582832	TRUE
12	4.576623	TRUE
13	4.871127	TRUE
14	5.842417	TRUE
15	6.024947	TRUE
16	4.991218	TRUE
17	4.66208	TRUE
18	5.134562	TRUE
19	5.064189	TRUE
20	5.048743	TRUE
21	4.918722	TRUE
22	5.690503	TRUE
23	4.848837	TRUE
24	5.062089	TRUE
25	6.212823	TRUE
26	5.730524	TRUE
27	5.081054	TRUE
28	5.802429	TRUE

B. DATASET 2:(TrainDatabase2 and ProbeDatabase2)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	8.167578	TRUE
2	8.784309	TRUE
3	8.234217	TRUE
4	8.564145	TRUE
5	8.040968	TRUE
6	7.913332	TRUE
7	8.290156	TRUE
8	8.393963	TRUE
9	8.060211	TRUE
10	8.215021	TRUE
11	7.736514	TRUE
12	8.362056	TRUE
13	9.279398	TRUE
14	8.276736	TRUE
15	8.069547	TRUE
16	8.285966	TRUE
17	8.677019	TRUE
18	8.026703	TRUE
19	8.166565	TRUE
20	8.322504	TRUE
21	8.160507	TRUE
22	9.374235	TRUE
23	8.083951	TRUE
24	8.028531	TRUE
25	9.251687	TRUE
26	7.614028	TRUE
27	7.90765	TRUE
28	8.263884	TRUE

C. DATASET 3:(TrainDatabase3 and ProbeDatabase3)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	27.357678	TRUE
2	23.027876	TRUE
3	23.801713	TRUE
4	23.726957	TRUE
5	23.418895	TRUE
6	22.972093	TRUE
7	23.616248	TRUE
8	23.450879	TRUE
9	23.588566	TRUE
10	25.877517	TRUE
11	23.324046	TRUE
12	23.947616	TRUE
13	23.327946	TRUE
14	23.227100	TRUE
15	27.116226	TRUE
16	24.705872	TRUE
17	25.332152	TRUE
18	23.795496	TRUE
19	23.417289	TRUE
20	24.099282	TRUE
21	23.415438	TRUE
22	24.31267	TRUE
23	22.959444	TRUE
24	23.453818	TRUE
25	23.889237	TRUE
26	22.771548	TRUE
27	25.796284	TRUE
28	23.313982	TRUE

29	6.318863	TRUE
30	5.961623	TRUE
31	5.083447	TRUE
32	5.381045	TRUE
33	7.570106	TRUE
34	5.353218	TRUE
35	5.015201	TRUE
36	5.167504	TRUE
37	4.723653	TRUE
38	6.93272	TRUE
39	5.293865	TRUE
40	5.182063	TRUE
41	5.580933	TRUE
42	5.506122	TRUE
43	5.33428	TRUE
44	6.014866	TRUE
45	5.673073	TRUE
46	5.030111	TRUE
47	6.621479	TRUE
48	6.621479	TRUE
49	6.284601	TRUE
50	4.841132	TRUE

29	8.299643	TRUE
30	8.117871	TRUE
31	8.230657	TRUE
32	8.308575	TRUE
33	7.966032	TRUE
34	8.277724	TRUE
35	8.539304	TRUE
36	10.01138	TRUE
37	8.142595	TRUE
38	8.426368	TRUE
39	8.842141	TRUE
40	8.983145	TRUE
41	9.985866	TRUE
42	7.989839	TRUE
43	8.944784	TRUE
44	8.072376	TRUE
45	7.918121	TRUE
46	8.228573	TRUE
47	9.615423	TRUE
48	9.321602	TRUE
49	8.181096	TRUE
50	8.050188	TRUE

29	23.413656	TRUE
30	23.191801	TRUE
31	23.729959	TRUE
32	23.443946	TRUE
33	23.323598	TRUE
34	23.245952	TRUE
35	23.436695	TRUE
36	22.854487	TRUE
37	23.296024	TRUE
38	23.082316	TRUE
39	23.251613	TRUE
40	23.51434	TRUE
41	23.241686	TRUE
42	23.022331	TRUE
43	23.200434	TRUE
44	24.130721	TRUE
45	23.354992	TRUE
46	23.287407	TRUE
47	23.612320	TRUE
48	23.040962	TRUE
49	23.407542	TRUE
50	23.130413	TRUE

TABLE 3-8: PCA-City Block WITH INDIAN FACE DATABASE

A. DATASET 1:(TrainDatabse1 ProbeDatabase 1)   B. DATASET 2:(TrainDatabase2 and ProbeDatabase2)   C. DATASET 3:(TrainDatabase3 and and ProbeDatabase3)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	5.034048	TRUE
2	5.102078	TRUE
3	4.770357	TRUE
4	5.231947	TRUE
5	5.331882	TRUE
6	4.902579	TRUE
7	5.199488	TRUE
8	5.104200	TRUE
9	4.482216	TRUE
10	5.000980	TRUE
11	5.466893	TRUE
12	4.763561	TRUE
13	5.132988	TRUE
14	5.090517	TRUE
15	4.996867	TRUE
16	5.449627	TRUE
17	4.997434	TRUE
18	6.691953	TRUE
19	5.250067	TRUE
20	4.858151	TRUE
21	4.988161	TRUE
22	5.075205	TRUE
23	5.588373	TRUE
24	4.964360	TRUE
25	5.041664	TRUE
26	5.443604	TRUE
27	5.015703	TRUE
28	4.800810	TRUE
29	4.821900	TRUE
30	5.825521	TRUE
31	5.053765	TRUE
32	5.237170	TRUE
33	4.958848	TRUE
34	5.125614	TRUE
35	5.137136	TRUE

Probe Image No	Time Taken for execution(s econds)	Accurate Match?
1	9.532687	TRUE
2	8.207971	TRUE
3	8.547910	TRUE
4	8.181615	TRUE
5	7.952099	TRUE
6	8.963999	TRUE
7	8.626435	TRUE
8	7.957595	TRUE
9	8.177972	TRUE
10	8.227901	TRUE
11	8.248921	TRUE
12	8.297438	TRUE
13	8.513603	TRUE
14	8.330373	TRUE
15	8.368319	TRUE
16	9.121850	TRUE
17	8.335362	TRUE
18	8.524498	TRUE
19	8.361445	TRUE
20	8.566860	TRUE
21	8.275572	TRUE
22	8.457325	TRUE
23	8.459307	TRUE
24	8.384663	TRUE
25	8.112874	TRUE
26	8.215717	TRUE
27	8.008798	TRUE
28	8.249755	TRUE
29	8.450635	TRUE
30	7.835477	TRUE
31	9.275202	TRUE
32	8.265623	TRUE
33	8.170341	TRUE
34	8.404136	TRUE
35	8.014429	TRUE

Probe Image No	Time Taken for execution(sec onds)	Accurat e Match?
1	23.588977	TRUE
2	23.511279	TRUE
3	23.595586	TRUE
4	23.762069	TRUE
5	23.20308	TRUE
6	25.149882	TRUE
7	23.1833105	TRUE
8	23.521564	TRUE
9	23.478413	TRUE
10	23.258127	TRUE
11	23.442758	TRUE
12	23.531960	TRUE
13	23.468220	TRUE
14	23.217661	TRUE
15	23.528141	TRUE
16	23.463995	TRUE
17	23.539582	TRUE
18	23.415347	TRUE
19	23.561116	TRUE
20	23.626214	TRUE
21	23.421267	TRUE
22	23.787781	TRUE
23	23.395292	TRUE
24	24.997288	TRUE
25	23.570493	TRUE
26	23.434840	TRUE
27	23.652763	TRUE
28	23.897559	TRUE
29	23.665581	TRUE
30	24.421675	TRUE
31	23.375915	TRUE
32	23.573699	TRUE
33	23.947154	TRUE
34	23.838114	TRUE
35	23.427506	TRUE

36	5.020175	TRUE
37	5.326930	TRUE
38	5.071906	TRUE
39	4.992703	TRUE
40	5.413910	TRUE
41	4.986394	TRUE
42	5.072130	TRUE
43	4.940785	TRUE
44	4.822907	TRUE
45	5.243164	TRUE
46	5.012147	TRUE
47	5.023271	TRUE
48	5.000673	TRUE
49	5.880890	TRUE
50	4.099049	TRUE

36	8.169252	TRUE
37	8.297488	TRUE
38	7.992423	TRUE
39	8.275101	TRUE
40	8.200113	TRUE
41	8.396287	TRUE
42	8.121100	TRUE
43	8.356482	TRUE
44	8.080448	TRUE
45	8.131186	TRUE
46	8.146268	TRUE
47	8.073123	TRUE
48	7.905945	TRUE
49	8.240457	TRUE
50	8.181066	TRUE

36	23.756936	TRUE
37	23.319388	TRUE
38	23.279740	TRUE
39	23.539847	TRUE
40	24.242287	TRUE
41	23.474568	TRUE
42	23.534938	TRUE
43	23.745334	TRUE
44	23.514195	TRUE
45	23.779126	TRUE
46	23.334810	TRUE
47	23.658107	TRUE
48	23.237014	TRUE
49	23.648444	TRUE
50	23.337297	TRUE

TABLE 3-9: PCA-Cosine Metric WITH INDIAN FACE DATABASE

DATASET 1:(TrainDatabse1 and ProbeDatabase 1)

DATASET 2:(TrainDatabase2 and ProbeDatabase2)

DATASET 3:(TrainDatabase3 and ProbeDatabase3)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	5.273198	TRUE
2	4.961446	TRUE
3	4.919715	TRUE
4	4.973841	TRUE
5	4.807652	TRUE
6	5.269447	TRUE
7	4.915279	TRUE
8	5.020833	TRUE
9	4.866726	TRUE
10	4.794655	TRUE
11	5.213846	TRUE
12	5.086748	TRUE
13	5.896713	TRUE
14	4.878095	TRUE
15	4.960775	TRUE
16	4.981248	TRUE
17	4.712726	TRUE
18	4.61173	TRUE
19	5.102445	TRUE
20	4.860045	TRUE
21	4.650766	TRUE
22	4.809874	TRUE
23	5.276201	TRUE
24	4.751238	TRUE
25	4.506937	TRUE
26	4.894377	TRUE
27	5.203822	TRUE
28	4.873869	TRUE
29	5.172674	TRUE
30	4.837791	TRUE
31	5.120371	TRUE

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	8.746602	TRUE
2	8.116897	TRUE
3	7.87925	TRUE
4	8.250455	TRUE
5	7.905801	TRUE
6	8.120295	TRUE
7	9.375859	TRUE
8	8.295827	TRUE
9	8.017154	TRUE
10	8.380923	TRUE
11	8.032791	TRUE
12	8.80972	TRUE
13	8.491758	TRUE
14	7.642525	TRUE
15	8.3781	TRUE
16	7.757845	TRUE
17	7.390168	TRUE
18	7.804938	TRUE
19	7.524306	TRUE
20	8.022344	TRUE
21	8.128619	TRUE
22	7.750908	TRUE
23	8.068358	TRUE
24	8.628162	TRUE
25	8.356158	TRUE
26	8.104488	TRUE
27	8.061809	TRUE
28	8.792513	TRUE
29	8.092259	TRUE
30	8.000098	TRUE
31	8.239028	TRUE

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	23.110943	TRUE
2	23.085206	TRUE
3	23.659864	TRUE
4	23.1428	TRUE
5	23.448125	TRUE
6	22.940626	TRUE
7	23.532454	TRUE
8	23.261106	TRUE
9	23.149936	TRUE
10	23.035462	TRUE
11	23.122303	TRUE
12	23.245392	TRUE
13	24.191728	TRUE
14	23.740495	TRUE
15	23.068342	TRUE
16	23.280237	TRUE
17	23.295092	TRUE
18	23.27249	TRUE
19	23.242095	TRUE
20	23.387788	TRUE
21	23.514623	TRUE
22	23.276244	TRUE
23	23.43664	TRUE
24	23.363118	TRUE
25	23.421662	TRUE
26	23.144891	TRUE
27	23.361249	TRUE
28	23.340585	TRUE
29	23.154671	TRUE
30	23.359964	TRUE
31	23.221552	TRUE

32	5.037218	TRUE
33	5.138695	TRUE
34	4.900039	TRUE
35	5.075533	TRUE
36	4.757094	TRUE
37	4.962635	TRUE
38	4.758446	TRUE
39	5.00356	TRUE
40	4.843732	TRUE
41	4.761729	TRUE
42	4.860056	TRUE
43	4.821022	TRUE
44	4.596413	TRUE
45	5.006755	TRUE
46	4.753811	TRUE
47	5.535497	TRUE
48	4.817607	TRUE
49	4.92334	TRUE
50	5.3202	TRUE

32	7.699615	TRUE
33	8.645001	TRUE
34	8.521038	TRUE
35	7.631631	TRUE
36	7.973001	TRUE
37	7.901725	TRUE
38	8.110843	TRUE
39	7.933534	TRUE
40	8.518718	TRUE
41	8.208355	TRUE
42	7.722571	TRUE
43	8.204028	TRUE
44	8.09031	TRUE
45	7.962907	TRUE
46	7.990697	TRUE
47	7.937589	TRUE
48	8.071879	TRUE
49	7.671658	TRUE
50	8.006246	TRUE

32	23.546203	TRUE
33	23.273958	TRUE
34	24.426958	TRUE
35	23.106897	TRUE
36	24.466663	TRUE
37	23.549342	TRUE
38	22.976395	TRUE
39	23.10554	TRUE
40	23.660199	TRUE
41	22.871875	TRUE
42	23.412243	TRUE
43	23.349822	TRUE
44	23.457375	TRUE
45	23.429386	TRUE
46	24.101175	TRUE
47	24.024918	TRUE
48	23.198907	TRUE
49	23.447424	TRUE
50	24.030246	TRUE

TABLE 3-10: LDA-Euclidean Distance Metric WITH INDIAN FACE DATABASE

A. DATASET 1:(TrainDatabse1 and ProbeDatabase 1)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	4.597194	TRUE
2	4.177149	TRUE
3	3.923358	TRUE

B. DATASET 2:(TrainDatabase2 and ProbeDatabase2)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	6.782193	TRUE
2	6.584499	TRUE
3	6.673115	TRUE

C. DATASET 3:(TrainDatabase3 and ProbeDatabase3)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	22.68624	TRUE
2	22.836168	TRUE
3	22.724100	TRUE

4	4.254115	TRUE
5	3.594049	TRUE
6	4.399549	TRUE
7	4.002119	TRUE
8	4.208288	TRUE
9	4.286118	TRUE
10	3.717664	TRUE
11	3.938124	TRUE
12	3.889836	TRUE
13	4.063755	TRUE
14	4.066435	TRUE
15	3.818763	TRUE
16	3.902611	TRUE
17	3.901726	TRUE
18	4.112843	TRUE
19	3.983045	TRUE
20	3.900084	TRUE
21	3.796358	TRUE
22	3.769793	TRUE
23	3.814335	TRUE
24	3.459938	TRUE
25	3.668325	TRUE
26	4.930558	TRUE
27	4.150082	TRUE
28	4.113808	TRUE
29	4.013617	TRUE
30	4.309346	TRUE
31	4.147351	TRUE
32	3.761242	TRUE
33	4.407601	TRUE
34	3.912946	TRUE

4	6.433304	TRUE
5	6.371827	TRUE
6	6.433212	TRUE
7	6.882341	TRUE
8	6.691803	TRUE
9	7.718873	TRUE
10	6.360976	TRUE
11	6.19209	TRUE
12	6.357381	TRUE
13	6.051684	TRUE
14	6.444108	TRUE
15	6.398233	TRUE
16	6.179745	TRUE
17	6.331272	TRUE
18	6.331131	TRUE
19	5.986397	TRUE
20	6.177273	TRUE
21	6.179888	TRUE
22	6.493228	TRUE
23	6.206117	TRUE
24	6.33943	TRUE
25	6.160464	TRUE
26	6.227033	TRUE
27	6.061467	TRUE
28	6.319202	TRUE
29	6.100985	TRUE
30	6.142863	TRUE
31	6.154601	TRUE
32	6.451334	TRUE
33	6.312018	TRUE
34	6.173028	TRUE

4	22.934528	TRUE
5	23.020943	TRUE
6	22.823133	TRUE
7	22.698569	TRUE
8	22.804598	TRUE
9	22.791353	TRUE
10	23.007478	TRUE
11	22.982655	TRUE
12	22.674594	TRUE
13	22.956234	TRUE
14	22.583199	TRUE
15	22.915473	TRUE
16	23.084040	TRUE
17	23.133128	TRUE
18	22.687797	TRUE
19	23.096683	TRUE
20	22.846917	TRUE
21	22.901346	TRUE
22	22.883098	TRUE
23	22.733809	TRUE
24	23.074457	TRUE
25	22.903086	TRUE
26	22.956718	TRUE
27	23.015755	TRUE
28	22.63672	TRUE
29	22.949745	TRUE
30	22.829912	TRUE
31	23.181404	TRUE
32	22.760286	TRUE
33	22.963409	TRUE
34	22.628816	TRUE

35	3.871925	TRUE
36	3.756114	TRUE
37	4.246649	TRUE
38	4.052227	TRUE
39	3.974197	TRUE
40	3.974197	TRUE
41	3.593778	TRUE
42	3.840111	TRUE
43	4.186216	TRUE
44	3.822252	TRUE
45	3.967682	TRUE
46	3.757517	TRUE
47	3.715371	TRUE
48	3.936657	TRUE
49	3.771116	TRUE
50	3.934841	TRUE

35	6.625452	TRUE
36	6.046678	TRUE
37	6.331318	TRUE
38	6.180516	TRUE
39	6.176527	TRUE
40	6.055209	TRUE
41	6.271009	TRUE
42	5.957007	TRUE
43	5.982626	TRUE
44	6.034935	TRUE
45	5.780486	TRUE
46	5.979164	TRUE
47	6.374207	TRUE
48	6.009984	TRUE
49	6.863191	TRUE
50	6.066779	TRUE

35	23.022803	TRUE
36	22.733985	TRUE
37	22.609228	TRUE
38	22.652603	TRUE
39	22.674347	TRUE
40	23.136512	TRUE
41	22.637879	TRUE
42	22.930839	TRUE
43	22.604071	TRUE
44	22.892008	TRUE
45	22.966376	TRUE
46	22.838602	TRUE
47	23.056733	TRUE
48	22.695009	TRUE
49	22.776353	TRUE
50	22.765952	TRUE

TABLE 3-11:LDA-City Block Metric WITH INDIAN DATABASE

DATASET 1:(TrainDatabse1 and ProbeDatabase 1)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	3.855598	TRUE
2	3.914812	TRUE
3	3.896327	TRUE
4	3.752086	TRUE
5	4.820862	TRUE
6	3.896239	TRUE
7	3.599303	TRUE
8	3.676036	TRUE
9	3.907367	TRUE
10	3.553624	TRUE

DATASET 2:(TrainDatabase2 and ProbeDatabase2)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	6.579361	TRUE
2	6.437409	TRUE
3	6.336096	TRUE
4	5.996782	TRUE
5	5.882168	TRUE
6	5.993171	TRUE
7	5.961663	TRUE
8	6.049901	TRUE
9	6.302388	TRUE
10	6.245661	TRUE

DATASET 3:(TrainDatabase3 and ProbeDatabase3)

Probe Image No	Time Taken for execution(seconds)	Accurate Match?
1	22.922382	TRUE
2	22.906877	TRUE
3	22.969162	TRUE
4	23.013158	TRUE
5	22.918651	TRUE
6	22.899897	TRUE
7	22.891047	TRUE
8	22.950199	TRUE
9	23.111277	TRUE
10	23.256401	TRUE

11	3.842594	TRUE
12	4.198442	TRUE
13	3.83703	TRUE
14	4.138871	TRUE
15	3.769131	TRUE
16	3.709890	TRUE
17	3.75465	TRUE
18	3.981362	TRUE
19	3.835491	TRUE
20	3.71077	TRUE
21	3.690818	TRUE
22	3.662753	TRUE
23	3.856714	TRUE
24	3.742889	TRUE
25	3.745227	TRUE
26	3.638170	TRUE
27	3.706287	TRUE
28	3.640457	TRUE
29	3.77219	TRUE
30	3.859966	TRUE
31	3.89554	TRUE
32	3.679581	TRUE
33	3.834379	TRUE
34	3.581804	TRUE
35	3.824996	TRUE
36	4.196955	TRUE
37	4.079211	TRUE
38	3.817725	TRUE
39	3.675603	TRUE
40	3.49639	TRUE
41	3.490831	TRUE
42	3.501656	TRUE
43	4.225644	TRUE

11	6.012537	TRUE
12	6.344128	TRUE
13	6.092731	TRUE
14	6.544193	TRUE
15	6.023194	TRUE
16	6.050317	TRUE
17	5.987423	TRUE
18	6.105172	TRUE
19	6.386773	TRUE
20	5.893105	TRUE
21	5.940337	TRUE
22	6.006651	TRUE
23	6.217301	TRUE
24	5.945883	TRUE
25	6.415511	TRUE
26	6.018196	TRUE
27	6.017075	TRUE
28	5.985846	TRUE
29	5.977821	TRUE
30	6.153259	TRUE
31	6.198939	TRUE
32	5.910600	TRUE
33	5.914710	TRUE
34	5.976349	TRUE
35	5.869222	TRUE
36	6.066070	TRUE
37	6.400524	TRUE
38	6.110840	TRUE
39	6.208351	TRUE
40	5.865734	TRUE
41	5.967133	TRUE
42	6.227629	TRUE
43	5.971247	TRUE

11	23.170768	TRUE
12	22.81815	TRUE
13	23.326093	TRUE
14	23.215874	TRUE
15	22.825108	TRUE
16	23.438625	TRUE
17	23.461557	TRUE
18	22.808055	TRUE
19	23.139022	TRUE
20	22.775193	TRUE
21	23.087632	TRUE
22	22.866844	TRUE
23	22.999687	TRUE
24	23.183883	TRUE
25	22.936486	TRUE
26	22.781667	TRUE
27	23.259517	TRUE
28	23.148956	TRUE
29	22.980481	TRUE
30	23.295712	TRUE
31	23.061266	TRUE
32	22.736348	TRUE
33	23.054725	TRUE
34	22.886680	TRUE
35	22.857499	TRUE
36	22.911286	TRUE
37	22.908870	TRUE
38	22.810305	TRUE
39	23.044055	TRUE
40	22.768315	TRUE
41	22.851249	TRUE
42	22.744088	TRUE
43	22.839162	TRUE

44	3.658777	TRUE	44	5.917221	TRUE	44	22.724326	TRUE
45	3.921247	TRUE	45	5.938462	TRUE	45	23.350449	TRUE
46	4.193737	TRUE	46	6.408950	TRUE	46	23.213480	TRUE
47	3.359598	TRUE	47	6.099973	TRUE	47	23.038650	TRUE
48	3.741679	TRUE	48	6.110480	TRUE	48	22.769230	TRUE
49	3.501340	TRUE	49	6.054591	TRUE	49	22.823164	TRUE
50	3.922033	TRUE	50	6.442405	TRUE	50	23.390217	TRUE

TABLE 3-12:LDA-Cosine Metric WITH INDIAN DATABASE

A. DATASET 1 :(TrainDatabse1 and ProbeDatabase 1) B. DATASET 2 :(TrainDatabase2 and ProbeDatabase2) C. DATASET 3 :(TrainDatabase3 and and ProbeDatabase3)

Probe Image No	Time Taken for execution(seconds)	Accurat e Match?	Probe Image No	Time Taken for execution(seconds)	Accurate Match?	Probe Image No	Time Taken for execution(s econds)	Accurate Match?
1	4.059146	TRUE	1	6.197623	TRUE	1	22.841533	TRUE
2	3.589910	TRUE	2	5.994401	TRUE	2	23.242509	TRUE
3	3.749141	TRUE	3	6.027504	TRUE	3	22.658288	TRUE
4	4.018381	TRUE	4	5.915736	TRUE	4	23.052604	TRUE
5	3.506776	TRUE	5	6.102279	TRUE	5	22.824451	TRUE
6	3.665036	TRUE	6	5.927925	TRUE	6	22.387742	TRUE
7	3.589215	TRUE	7	6.082260	TRUE	7	22.524073	TRUE
8	3.769617	TRUE	8	6.087507	TRUE	8	22.547457	TRUE
9	4.000320	TRUE	9	6.149877	TRUE	9	22.754950	TRUE
10	3.507181	TRUE	10	5.814110	TRUE	10	22.484599	TRUE
11	4.138541	TRUE	11	5.899006	TRUE	11	23.491178	TRUE
12	3.519982	TRUE	12	5.799846	TRUE	12	22.581634	TRUE
13	3.824206	TRUE	13	5.840573	TRUE	13	23.364114	TRUE
14	3.549683	TRUE	14	6.028006	TRUE	14	22.552429	TRUE
15	3.854881	TRUE	15	6.157091	TRUE	15	22.973657	TRUE

16	3.942426	TRUE
17	3.728720	TRUE
18	3.434515	TRUE
19	4.174112	TRUE
20	3.631307	TRUE
21	3.760883	TRUE
22	3.523527	TRUE
23	3.893753	TRUE
24	3.520495	TRUE
25	3.470706	TRUE
26	3.712140	TRUE
27	3.427398	TRUE
28	3.774115	TRUE
29	3.912027	TRUE
30	3.789327	TRUE
31	3.743038	TRUE
32	3.784616	TRUE
33	3.620391	TRUE
34	3.733857	TRUE
35	3.601189	TRUE
36	4.005014	TRUE
37	3.723826	TRUE
38	4.161938	TRUE
39	3.835608	TRUE
40	3.730653	TRUE
41	3.602885	TRUE
42	3.851916	TRUE
43	3.599697	TRUE
44	4.309204	TRUE
45	3.63319	TRUE
46	4.172247	TRUE
47	3.709823	TRUE
48	3.530235	TRUE
49	3.844573	TRUE
50	4.497079	TRUE

16	5.915664	TRUE
17	6.057597	TRUE
18	6.176219	TRUE
19	5.951036	TRUE
20	5.844391	TRUE
21	6.048307	TRUE
22	5.916653	TRUE
23	5.738954	TRUE
24	5.785199	TRUE
25	6.221003	TRUE
26	5.928315	TRUE
27	5.819240	TRUE
28	6.186523	TRUE
29	6.359665	TRUE
30	5.764515	TRUE
31	5.894921	TRUE
32	5.830610	TRUE
33	5.998161	TRUE
34	5.766051	TRUE
35	5.848567	TRUE
36	6.028291	TRUE
37	6.690308	TRUE
38	6.055313	TRUE
39	5.767466	TRUE
40	5.858383	TRUE
41	5.825342	TRUE
42	5.897915	TRUE
43	5.936645	TRUE
44	6.277547	TRUE
45	6.035588	TRUE
46	5.582908	TRUE
47	6.242195	TRUE
48	6.109972	TRUE
49	5.722345	TRUE
50	6.064593	TRUE

16	22.662087	TRUE
17	22.549967	TRUE
18	22.619303	TRUE
19	22.344715	TRUE
20	22.589517	TRUE
21	22.509926	TRUE
22	22.395815	TRUE
23	22.477649	TRUE
24	22.557474	TRUE
25	22.598872	TRUE
26	22.691066	TRUE
27	22.437897	TRUE
28	22.881670	TRUE
29	22.696386	TRUE
30	22.556495	TRUE
31	23.730826	TRUE
32	22.545399	TRUE
33	22.665064	TRUE
34	22.362857	TRUE
35	22.493440	TRUE
36	22.754727	TRUE
37	22.438454	TRUE
38	22.546615	TRUE
39	22.801075	TRUE
40	22.602522	TRUE
41	23.009534	TRUE
42	22.752176	TRUE
43	22.753178	TRUE
44	22.493677	TRUE
45	22.614323	TRUE
46	22.687592	TRUE
47	22.551719	TRUE
48	22.788042	TRUE
49	22.682521	TRUE
50	22.560405	TRUE

CHAPTER FOUR

ANALYSIS OF RESULTS

The results from all the experiments indicated 100% accurate recognitionrate as illustrated in Tables 4-1 and 4-2regardless of increase in number of images per class (person) and its corresponding increase in the number of images in all the training datasets. Conversely, observing the result from the highest image per person (class) with its corresponding highest number of images in the training database considered and reducing them to the lowest in all the projection-metric combination experimented the same accurate recognition rate was obtained.

Thus, the generalisation abilities of all the projection-metric combinations (PCA-M1, PCA-M2, PCA-M3, LDA-M1, LDA-M2, and LDA-M3) were excellent.

TABLE 4-1: ACCURATE RECOGNITION RATE FOR EXECUTION OF PROJECTIONMETRIC ALGORITHM WITH AT& T DATABASE

AT & T Database									
	Dataset 1			Dataset 2			Dataset 3		
PROJECTION-/METRIC	A	B	C	A	B	C	A	B	C
PCA-M1	40	40	100%	40	40	100%	40	40	100%
PCA-M2	40	40	100%	40	40	100%	40	40	100%
PCA-M3	40	40	100%	40	40	100%	40	40	100%
LDA-M1	40	40	100%	40	40	100%	40	40	100%
LDA-M2	40	40	100%	40	40	100%	40	40	100%
LDA-M3	40	40	100%	40	40	100%	40	40	100%

M1-Euclidean Distance Metric M2-City Block Metric M3-Cosine Metric

A=Numberof Correctly Matched Images,

B= NumberofProbe/Test Images, C=Recognition Rate =A/B*100%

TABLE 4-2: ACCURATE RECOGNITION RATE FOR EXECUTION OF PROJECTIONMETRIC ALGORITHM WITH INDIAN FACE DATABASE

INDIAN FACE DATABASE									
	Dataset 1			Dataset 2			Dataset 3		
PROJECTION-/METRIC	A	B	C	A	B	C	A	B	C
PCA-M1	50	50	100%	50	50	100%	50	50	100%

PCA-M2	50	50	100%	50	50	100%	50	50	100%
PCA-M3	50	50	100%	50	50	100%	50	50	100%
LDA-M1	50	50	100%	50	50	100%	50	50	100%
LDA-M2	50	50	100%	50	50	100%	50	50	100%
LDA-M3	50	50	100%	50	50	100%	50	50	100%

M1-Euclidean Distance Metric   M2-City Block Metric   M3-Cosine Metric

A=Number of Correctly Matched Images

B= Number of Probe/Test Images, C=Recognition Rate =A/B*100%

Though IFD had images where each subject or class was portrayed with highly varying orientation angles than AT & T images, it did not have any effect on the recognition rate of any of the techniques used .Obviously, the results of all the projection-metric combination considered on the two Databases had 100% accurate recognition.

In all the algorithms executed with AT & T Database, the Table 4-3 showed that average execution time increased as the image size per person in the training database increased. Take an instance. PCA-M1 (PCA combined with Euclidean distance Metric) had average execution time of 3.2

seconds with dataset1 (3 images per person in the training database), 4.44 seconds with dataset2 (5 images per person in the training database) and 10.62 seconds with dataset3 (10 images per person in the training database). LDA-M2 (LDA combined with City Block Metric) had average execution time of 2.59 seconds with dataset 1, 3.52 seconds with dataset 2 and 8.69 seconds with dataset3.This behaviour can easily be noticed in the figure 4-1.

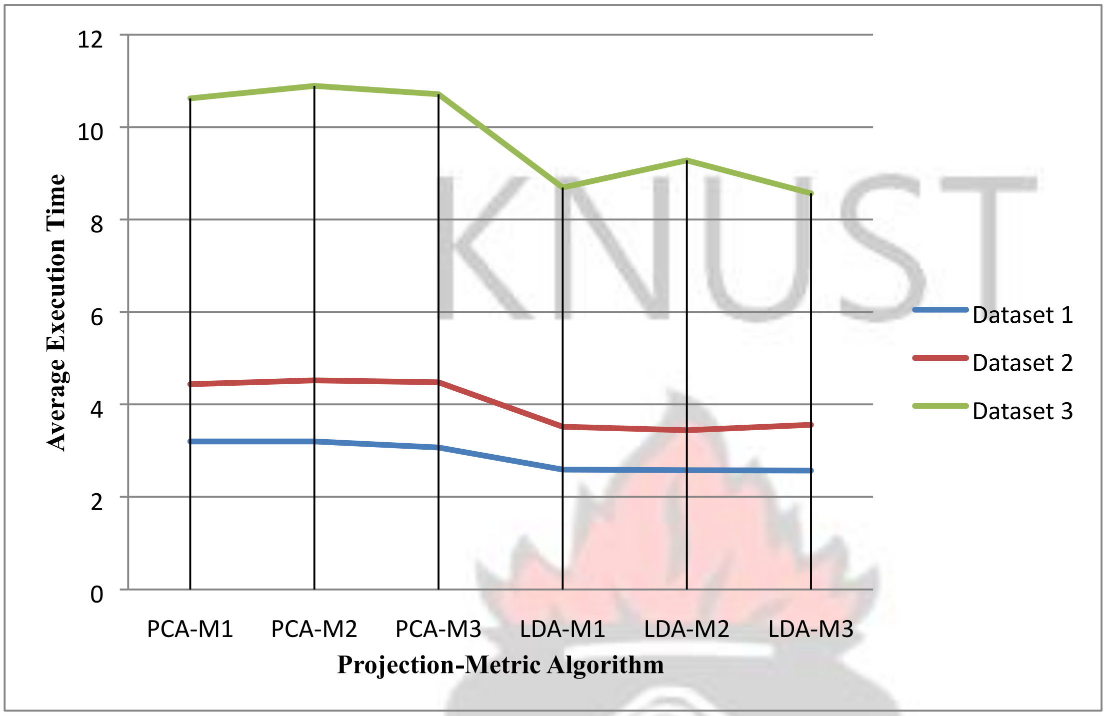
However, from the same table and chart, LDA appeared to have average execution time increased but lower than PCA in all the projection-metric considered as the image size per person increased.

TABLE 4-3: Average Execution Time (in seconds) For Projection-Metric Algorithms with AT& T Database

ALGORITHMS	AVERAGE EXECUTION TIME(SECONDS)		
Projection-Metric	Dataset 1	Dataset 2	Dataset 3
PCA-M1	3.20	4.44	10.62
PCA-M2	3.20	4.52	10.89
PCA-M3	3.07	4.48	10.71
LDA-M1	2.59	3.52	8.69
LDA-M2	2.58	3.44	9.28
LDA-M3	2.57	3.56	8.57

M1-Euclidean Distance Metric M2-City Block Metric M3-Cosine Metric

Figure 4-1:Average Execution Time (in seconds) For Projection-Metric Algorithms with AT & T Database



Similarly, the table 4-4 showed increase in average execution time for all the projection-metric algorithms as the image size per person increased. For example, PCA with Euclidean distance metric (PCA-M1) had 5.54 seconds with dataset1, 8.42 seconds with dataset 2 and 23.73 seconds with dataset 3. The figure 4-2 depicted the behaviour in the table 4-4.

However, the difference between the result in Table 4-3 and 4-4 was that for all the algorithms, the average execution times for Table 4-4 were more than that of Table 4-3 due to the fact that the total number of images in the training database of each of the datasets were more in the case of Indian Face Database (Dataset1: 180 images, Dataset2: 300, Dataset3: 600 images) than AT &

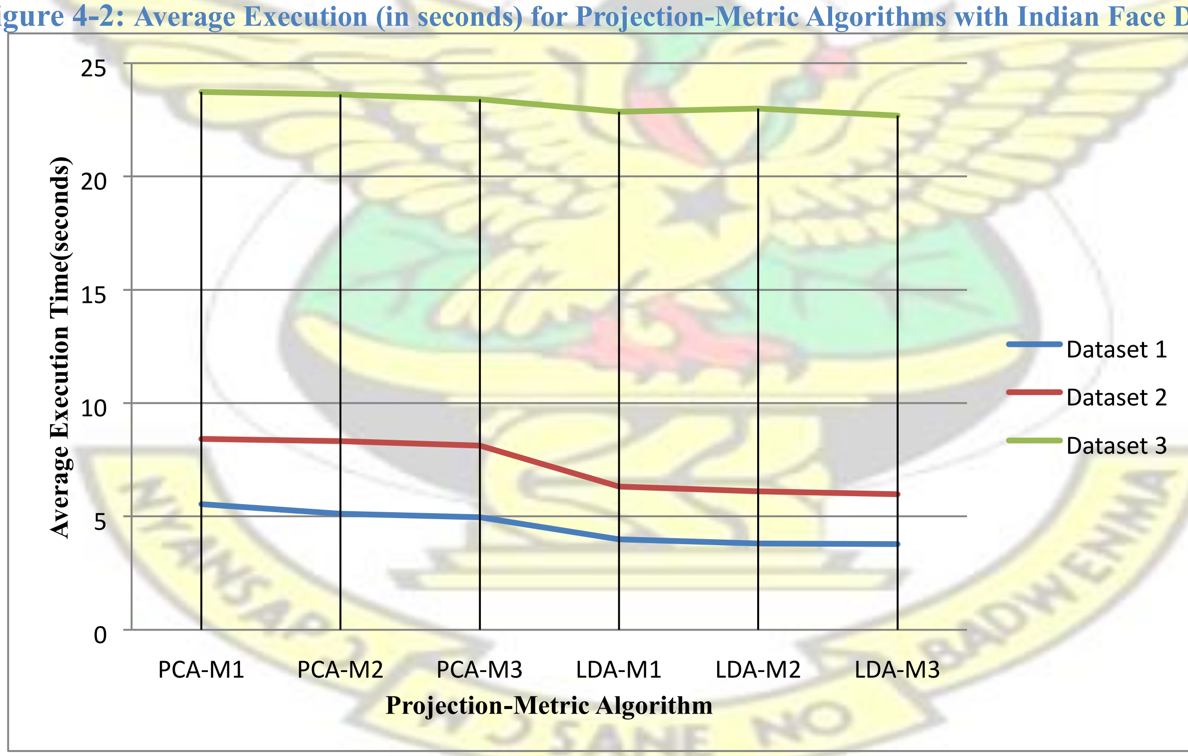
T Database (Dataset 1: 120 images, Dataset2: 200 images, Dataset3 : 300 images) employed. The analysis in the two tables 4-3 and 4-4 had shown the effect of varying the image size per person and consequently the size of images in the training database. That is, both PCA and LDA proved that time of execution of each algorithm averagely increases as the size of the training database increases.

TABLE 4-4: Average Execution Time (in seconds) For Projection-Metric Algorithms with Indian Face Database

ALGORITHM	AVERAGE EXECUTION TIME (SECONDS)		
Projection-Metric	Dataset 1	Dataset 2	Dataset 3
PCA-M1	5.54	8.42	23.73
PCA-M2	5.12	8.32	23.62
PCA-M3	4.96	8.12	23.4
LDA-M1	3.99	6.31	22.85
LDA-M2	3.81	6.11	23
LDA-M3	3.77	5.98	22.69

M1-Euclidean Distance Metric   M2-City Block Metric   M3-Cosine Metric

Figure 4-2: Average Execution (in seconds) for Projection-Metric Algorithms with Indian Face Database



Again, it is important to note that the average time taken for LDAs to recognise a face was better than that of the PCAs in all the training image size of both database used as illustrated in Table 4-3 and 4-4 as well as the figures 4-1,4-2. Therefore it can be concluded that LDA outperformed the PCA.

In an attempt to specifically point out which projection-metric combination is more efficient and appropriate for what situation, rankings for projection-metrics (in terms of their average time taken to recognize a face when applied on a specific dataset of a particular database)were done (as illustrated in Table4-5 and 4-6).The Table 4-5 showed that with AT& T database, PCA-M1 and LDA-M3 ranked best among the PCA and LDA methods respectively. However, comparing the average execution time of both PCA-M1 and LDA-M3 for dataset3(with the largest image data size of 400 images) it turned out that LDA-M3 (**8.57 seconds**) was better than PCAM1(**10.62 seconds**).

Similarly, with the IFD PCA-M3 and LDA-M3 also ranked the best among the PCA and LDA methods respectively. Again, comparing the average execution time of both PCA-M3 and LDAM3 for dataset3 (with the largest image data size of 600 images) LDA-M3 (**22.69 seconds**) was preferred to PCA-M3 for dataset3 (**23.40 seconds**).Obviously, the LDA-M3 was the preferred choice in all cases. This became clear when the overall ranking was considered as shown in Table 4-6 that LDA-M3 is the most efficient among all the projection-metric algorithms executed with AT& T Database and IFD. It also meant that LDA-M3 was capable of withstanding large training database of face images than the other projection-metric algorithms. On the other hand, PCA-M1 could not stand large database hence it would be improper to recommend it for Voter registration and Verification which usually has large database.

The above results also showed that the increase in size of image per person and corresponding increase in size of the training database had severe impact on the PCA compare to LDA as far as performances (in terms of time taken) of the algorithms were concerned. That is to say, in all cases LDA algorithms were executed much faster than that of PCA. Hence LDA outperformed PCA in respect of efficiency (in terms time taken to execute algorithm).

TABLE 4-5: RANKING PCA AND LDA SEPARATELY

	AT&T DATABASE					INDIAN DATABASE			
	Ranking Projection-Metrics					Ranking Projection-Metrics			
	Dataset 1	Dataset 2	Dataset 3	Average Ranking		Dataset 1	Dataset 2	Dataset 3	Average Ranking
PCA-M1	2	1	1	1.33		3	3	3	3.00
PCA-M2	2	3	3	2.67		2	2	2	2.00
PCA-M3	1	2	2	1.67		1	1	1	1.00
LDA-M1	3	2	2	2.33		3	3	2	2.67
LDA-M2	2	1	3	2.00		2	2	3	2.33
LDA-M3	1	3	1	1.67		1	1	1	1.00

Table 4-6: OVERALL RANKING OF THE PROJECTION-METRICS

	AT&T DATABASE				INDIAN DATABASE			
	Ranking Projection-Metrics				Ranking Projection-Metrics			
	Dataset 1	Dataset 2	Dataset 3		Dataset 1	Dataset 2	Dataset 3	Average Ranking
PCA-M1	5	5	4		6	6	6	5.33
PCA-M2	5	6	6		5	5	5	5.33
PCA-M3	4	4	5		4	4	4	4.17
LDA-M1	3	2	2		3	3	2	2.50
LDA-M2	2	1	3		2	2	3	2.17
LDA-M3	1	3	1		1	1	1	1.33

CHAPTER FIVE

CONCLUSION,RECOMMENDATION AND FUTURE WORK

Empirical assessment was made of the two popular appearance based techniques of face recognition (PCA and LDA). The publicly available databases AT & T and Indian Face image database were used to implement algorithms of the two techniques and consequently evaluated . The research considered the effect on accurate recognition and the computational cost (in terms of time of execution of algorithm) of implementing the two techniques indicated above combined with three similarity measure as image database size changes. The study showed that regardless of the size of image per person and the training database size the accuracy of the techniques was not compromised. The generalization abilities of all the projection-metric combination considered were excellent. The study demonstrated that there is lower computational cost in all projection-metric algorithms of the LDA techniques than that of PCA. The LDA-Cosine Metric was preferred to other projection-metrics studied since with the larger data size it turned out to be the most efficient. PCA-Euclidean distance metric performed worse in terms of average time taken and appear not to have the capability to withstand larger training database.

As far as recognition rate is concerned the study agrees with Delac et al (2006) which stated that "the performance of the appearance based methods is heavily dependent on the employed distance measure and that with the right combination of appearance based method and distance no claim regarding the superiority of any of the techniques i.e., PCA, LDA can be made".

RECOMMENDATION

Obviously, LDA especially LDA-Cosine metric will be much more preferred in situation where large number of people are involved in the recognition process and little amount of time is required for an individual to be recognised by the system. Voter Registration and Verification Exercise, Customs and Immigration activities are few examples

In organisations where there is not much traffic on their security access control systems any of the projection-metric methods can be adopted since in terms of accuracy (recognition rate) none of the methods implemented is superior to the other. However, where the access control systems will have to record time of entry of staff (especially reporting time) then LDA-cosine metric method is most appropriate compare to the other five methods implemented.

FUTURE WORK

Future research should focus on employing more than three similarity measures to combine with the LDA and PCA and expanding the image data size beyond 600 images to determine which projection-metric methods is most efficient and can withstand larger training database.

REFERENCES

Behnam K. "Comparative Analysis of Face Recognition Algorithms and Investigation on Significance of Color", Master Thesis, Concordia University, Montreal Quebec, Canada. August 2006.

Belhumeur P., Hespanha J., and Kriegman D. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

`Beveridge, J. R., She, K., Draper, B. A., and Givens, G. H. 2001. A nonparametric statistical comparison of principal component and linear discriminant subspaces for face recognition. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, pages 535-542*.

Cox I. J., Ghosn J., and Yianilos P. N. Feature-based face recognition using mixture-distance. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 209–216, June 1996.

Darwin C. The Expression of the Emotions in Man and Animals. London: John Murray, 1872, pages 120-126.

Delac K., Mislav G and Sonja G, "Independent comparative study of PCA, ICA, and LDA on the FERET data set," *International Journal of Imaging Systems and Technology*, vol. 15, no. 5, pp. 252-260, 2006.

Delac K, Mislav G. and Sonja G, "Generalization Abilities of Appearance-Based SubspaceFace Recognition Algorithms" *12th Int. Workshop on Systems, Signals & Image Processing, 22-24 September 2005 page 273-276*

Delac K, Mislav G., Panos L. "Appearance-based Statistical Methods for Face Recognition"47th International Symposium ELMAR-2005, 08-10 June 2005, Zadar, Croatia. Page 151-156.

Fischler M. A. and Elschlager R. A. The representation and matching of pictorial structures. IEEE Transactions on Computers, 22(1):67–92, 1973.

Galton F. Personal identification and description. Nature, pages 173–188, 1888.

Goldstein A. J. Lesk A. B. and Harmon L. D. "Identification of Human Faces," Proceedings of the IEEE, Vol. 59, No. 5, pp.748-760, 1971.

Jain A.K., "Small sample size effects in statistical pattern Recognition: Recommendations for practitioners" *IEEE Transactions on Pattern Analysis and Machine Intelligence 13,* I3 (1991), 252-264.

Jens F, and Kongens L 2005, Master Thesis IMM- Thesis-2005-74

Kanade T. Picture Processing System by Computer Complex and Recognition of Haman Faces. Doctoral Dissertation, Kyoto University, 1973.

Keyur B , Premal J. P., Samip A P., Udesang K. J. "A Comparative Study of PCA & LDA Human Face Recognition Methods", National Conference on Recent Trends in Engineering & Technology, 13-14 May 2011 B.V.M. Engineering College, V.V.Nagar,Gujarat,India. Page 1-6

Li S. Z. and Jain A. K. Handbook of face recognition. Springer, 2005, page 13-14

Mukesh G, "Comparative Analysis of Face Recognition Algorithms", IJREAS Volume 2, Issue 2 (February 2012) ISSN: 2249-3905, page 292-299.

Navarrete P., Ruiz-del-Solar J. "Analysis and Comparison of Eigenspace-Based Face Recognition Approaches", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 16, No. 7, November 2002, pp. 817-830

`Raj K S, Abhijit K, Yash P S "A Comparative Study of Face Recognition System Using PCA and LDA", *International Journal of IT, Engineering and Applied Sciences Research (IJIEASR) ISSN: 2319-4413 Volume 2, No. 10, October 2013 page 12-20*

Sanjeev K. and Harpreet K. "Face Recognition Techniques: Classification and Comparisons" **,** *International Journal of Information Technology and Knowledge Management July December 2012, Volume 5, No. 2, pp. 361-363*

Sasan K, Alireza H. , Azizah A. M, Mazdak Z., Shahidan M. A, "An Overview of Principal Component Analysis", *Journal of Signal and Information Processing*, 2013, 4, page 173 175 doi:10.4236/jsip.2013.43B031 Published Online August 2013 (http://www.scirp.org/journal/jsip)

Satonkar S. S , Kurhe A B., Prakash K. " Face Recognition Using Principal Component Analysis and Linear Discriminant Analysis on Holistic Approach in Facial Images Database" , IOSR Journal of Engineering **e**-ISSN: 2250-3021, **p**-ISSN: 2278-8719, www.iosrphr.org Vol. 2, Issue 12 (Dec. 2012), ||V4|| PP 15-23

Shah Z.M.N., Radi H.R., Muniroh A. S. , Rosman A. R, Muzafar I.M., Idzdihar I M., Sulaiman H.A., Jaafar A. "Face Recognition Using Principal Component", International Journal of

Electrical & Computer Sciences IJECS-IJENS Vol:12 No:05. Page 50-54, October 2012
Sirovich L. and Kirby M., "Low-dimensional Procedure for the Characterization of Human Faces," Vol. 4, No. 3, 1987, pp. 519-524.

Sujata G. B. and Mankar V. H., "A Review Paper on Face Recognition Techniques",*International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 1, Issue 8, October 2012 page 339-343*

Sukhvinder S, Meenakshi S and Rao N S " Accurate Face Recognition Using PCA and LDA " International Conference on Emerging Trends in Computer and Image Processing (ICETCIP'2011) Bangkok page 62-66, Dec., 2011
Turk M. and Pentland A. "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, 1991, pp. 71-86.

Zhao W., Rosenfeld A., Phillips J., Chellappa R., "Face Recognition in Still and Video Images: A Literature Surrey", *ACM Computing Surveys*, Vol. 35, Dec. 2003, pp. 399-458.

Zunxiong L, Linfeng H , Zheng X, "A Comparative Study of Distance Metrics used in face recognition" Journal of Theoretical and Applied Information Technology© 2005 - 2009 JATIT page 206- 210