##### A NOVEL FACE RECOGNITION METHOD USING PCA, LDA AND BAYESIAN CLASSIFIER

##### A PROJECT REPORT

###### ***Submitted by***

##### KARTHIKRAJAA A T (12007107043)

**JERWIN SAMUEL J (12007107039)**

**ESAKKI RAJ M (12007107025)**

**VISHAL I V (12007107110)**

***In partial fulfillment for the award of the degree***

***Of***

##### BACHELOR OF ENGINEERING

*In*

ELECTRONICS AND INSTRUMENTATION ENGINEERING



****

**VELAMMAL ENGINEERING COLLEGE**

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2011

ANNAUNIVERSITY: CHENNAI 600 025

**BONAFIDE CERTIFICATE**

Certified that this project report **“A NOVEL FACE RECOGNITON METHOD**

**USING PCA, LDA AND BAYESIAN CLASSIFIER”** is the bonafide work of

**KARTHIKRAJAA A T**

**JERWIN SAMUEL J**

**ESAKKI RAJ M**

**VISHAL I V**

**SIGNATURE SIGNATURE**

**Dr. N.BHARATHI M.Tech, PhD., Mr.C.MURUKESH, M.E.,**

**PROFESSOR & SUPERVISOR**

**Head of the Department** Velammal Engineering College,

Department of EIE, Ambattur- Red hills Road,

Velammal Engineering College, Surapet, Chennai – 66.

Ambattur-Red hills Road,

Surapet, Chennai – 66.

**CERTIFICATE OF EVALUATION**

**College Name: VELAMMAL ENGINEERING COLLEGE**

**Department : Electronics & Instrumentation Engineering**

**Semester : VIII**

|  |  |  |
| --- | --- | --- |
| **NAME OF THE STUDENTS** | **TITLE OF THE PROJECT** | **NAME OF THE SUPERVISOR WITH DESIGNATION** |
| KARTHIKRAJAA A T  JERWIN SAMUEL J  ESAKKI RAJ M  VISHAL I V | A NOVEL FACE RECOGNITION METHOD USING PCA, LDA AND BAYESIAN CLASSIFIER | Mr.C.MURUKESH,M.E.,  ASSISTANT PROFESSOR / EIE |

The report of the project work submitted by the above students in the partial fulfillment for the award of Bachelor of Engineering degree in **ELECTRONICS AND INSTRUMENTATION ENGINEERING** of AnnaUniversity was evaluated and confirmed to be the report of the work done by the above students and then evaluated.

Submitted on: \_\_\_\_\_\_\_\_\_\_\_

**(INTERNAL EXAMINER) (EXTERNAL EXAMINER)**

**ACKNOWLEDGEMENT**

At the outset we would like to express our sincere gratitude to our beloved **Chairman, Thiru.M.V.Muthuramalingam** for his constant guidance and support.

We would like to express our heartfelt thanks to our respected **Chief Executive Officer, Thiru.M.V.M.Velmurugan** for his kind encouragement and blessings.

We express our deepest gratitude and thanks to our beloved **Principal Dr.K**. **Thirunavukkarasu** for his inspirational ideas during the course of the project.

We wish to express our sincere thanks and gratitude to **Dr. N.Bharathi, M.Tech, PhD., Head of the Department of Electronics and Instrumentation Engineering** for her patient guidance and assistance in solving the various intricacies involved in the project.

Our special thanks to our project guide **Mr.C.Murukesh, M.E., Assistant Professor, Department of Electronics and Instrumentation Engineering**. For the deluge of ideas and assistance they have provided to us all through this project.

Finally we thank our parents and friends who helped us in the successful completion of this project.

**ABSTRACT**

Humans often use faces to recognize individuals and advancements in computing capability over the past few decades now enable similar recognitions automatically. Early face recognition algorithms used simple geometric models, but the recognition process has now matured into a science of sophisticated mathematical representations and matching processes. Face recognition can be used for both verification and identification. In this we have proposed a new face recognition method using PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), and Bayesian Classifier. PCA is used for reducing the Dimensionality of the image and LDA is used for feature extraction. LDA is generally superior to PCA, when enough training samples persubject are available but it fails to perform well when the images per subject are very small. Combination of PCA and LDA is used for improving the capability of LDA when a few samples of images are available.LDA algorithm selects features that are most effective for class separability while PCA selects features important for class representation. Bayesian Classifier uses Probabilistic Measure of Similarity primarily based on Maximum Likelihood Estimation (MLE) of image differences. The proposed method was tested on FERET and YALE face database.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  |  |  |
|  | **ABSTRACT** | **iv** |
|  | **LIST OF FIGURES** | **x** |
|  | **LIST OF SYMBOLS AND ABBREVIATIONS** | **xi** |
|  |  |  |
| **1** | **INTRODUCTION** |  |
|  | 1.1INTRODUCTION | 1 |
|  | 1.2 OBJECTIVES | 2 |
|  | 1.3WHY FACE RECOGNITION? | 2 |
|  | 1.4NOVEL APPLICATIONS OF FACE  RECOGNITION SYSTEMS | 2 |
|  | 1.5COMMERCIAL SYSTEMS AND APPLICATIONS | 3 |
|  | 1.6 LITERATURE REVIEW | 3 |
|  | 1.7 REQUIREMENT SPECIFICATIONS | 7 |
|  | 1.7.1INTRODUCTION | 7 |
|  | 1.7.2SOFTWARE USED | 7 |
|  | 1.7.2.1MATLAB 7.6.0 | 7 |
|  | 1.7.2.2IMAGE PROCESSING TOOLBOX | 7 |
|  | 1.8 ORGANISATION OF THE PROJECT | 8 |
| **2** | **FUNDAMENTALS OF IMAGE PROCESSING AND FACE RECOGNITION SYSTEM** | **9** |
|  | 2.1 INTRODUCTION | 9 |
|  | 2.1.1 IMAGE | 9 |
|  | 2.1.2 ANALOG IMAGE | 9 |
|  | 2.1.3 DIGITAL IMAGE | 10 |
|  | 2.2 DIGITAL IMAGE PROCESSING | 10 |
|  | 2.3 DIGITAL IMAGE REPRESENTATION | 10 |
|  | 2.3.1 RESOLUTION | 11 |
|  | 2.4 IMAGE TYPES | 11 |
|  | 2.4.1 BINARY IMAGES | 11 |
|  | 2.4.2 GRAYSCALE IMAGES | 12 |
|  | 2.4.3 COLOR IMAGES | 13 |
|  | 2.5 ELEMENTS OF IMAGE PROCESSING SYSTEM | 13 |
|  | 2.6 INTRODUCTION | 14 |
|  | 2.7 RELATIONSHIP BETWEEN IMAGE PROCESSING AND  FACE RECOGNITION | 15 |
|  | 2.8 RECOGNIZING FACES IN A SINGLE IMAGE | 15 |
|  | 2.9 APPEARANCE BASED METHODS | 15 |
|  | 2.10 CHALLENGES IN FACE RECOGNITION | 15 |
|  | 2.11 APPROACHES TO FACE RECOGNITION | 16 |
|  | 2.12 AUTOMATED FACE RECOGNITION SYSTEM | 16 |
|  | 2.13 REPRESENTATION OF PATTERN CLASSES | 17 |
|  | 2.14 VECTOR REPRESENTATION OF PATTERNS | 17 |
| **3** | **PREPROCESSING** | **19** |
|  | 3.1 POINT OPERATION | 20 |
|  | 3.1.1 HISTOGRAM MANIPULATION | 20 |
|  | 3.1.2 HISTOGRAM EQUALIZATION | 21 |
|  | 3.2 NORMALIZATION | 26 |
|  | 3.3 GEOMETRIC OPERATION | 29 |
|  | 3.3.1 GEOMETRIC SCALING | 30 |
|  | 3.3.2 IMAGE ROTATION | 33 |
| **4** | **DIMENSIONALITY REDUCTION USING PRINCIPAL COMPONENT ANALYSIS** | **34** |
|  | 4.1 PRINCIPAL COMPONENT ANALYSIS | 35 |
|  | 4.2 MATHEMATICS | 36 |
|  | 4.3 ASSUMPTIONS | 37 |
|  | 4.4 ALGORITHM FOR FINDING EIGENFACES | 37 |
|  | 4.5 FINDING WEIGHTS | 39 |
|  | 4.6 RECOGNITION TASK | 40 |
|  | 4.7 DISTANCE MEASURES | 41 |
|  | 4.7.1 EUCLIDEAN DISTANCE | 41 |
|  | 4.7.2 THE MAHALANOBIS DISTANCE | 41 |
| **5** | **FEATURE EXTRACTION USING LINEAR DISCRIMINANT ANALYSIS** | **42** |
|  | 5.1 INTRODUCTION | 42 |
|  | 5.2 DIFFERENT APPROACHES TO LDA | 43 |
|  | 5.3 MATHEMATICAL OPERATIONS | 44 |
| **6** | **CLASSIFICATION USING BAYESIAN CLASSIFIER** | **47** |
|  | 6.1 CLASSIFICATION | 47 |
|  | 6.2 BAYES THEOREM | 49 |
|  | 6.3 MAXIMUM LIKELIJHOOD ESTIMATION (MLE) | 50 |
|  | 6.4 CONDITIONAL PROBABILITY | 50 |
|  | 6.5 BAYESIAN MAXIMUM LIKELIHOOD ESTIMATION | 50 |
|  | 6.5.1 PROBABILISTIC SIMILARITY MEASURES | 51 |
|  | 6.5.2 SUBSPACE DENSITY ESTIMATION | 52 |
|  | 6.5.3 EFFICIENT SIMILARITY COMPUTATION | 53 |
|  | 6.5.4 EIGENFACES MATCHING | 55 |
|  | 6.5.5 BAYESIAN MATCHING | 55 |
|  | 6.5.6 DUAL EIGENFACES | 56 |
| **7** | **RESULTS AND DISCUSSIONS** | **59** |
| **8** | **CONCLUSION AND FUTURE WORK** | **71** |
|  | **APPENDIX** | **73** |
|  | **REFERENCES** | **85** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| FIGURE.NO | TITLE | PAGE NO |
| 2.1 | Analog image to Digital image conversion | 8 |
| 2.2 | Digital Image Representation | 9 |
| 2.3 | Binary Image Representation | 11 |
| 2.4 | Grayscale Image Representation | 12 |
| 2.5 | Color Image Representation | 12 |
| 2.6 | Pattern Recognition Algorithm | 17 |
| 3.1 | Input image for histogram equalization | 23 |
| 3.2 | Histogram of Input Image | 24 |
| 3.3 | Histogram equalized Image | 25 |
| 3.4 | Histogram of equalized image | 26 |
| 3.5 | Input Image for Normalization | 29 |
| 3.6 | Normalized Image | 30 |
| 3.7 | Input Image for Scaling | 33 |
| 3.8 | Scaled Image | 34 |
| 6.1 | Face Alignment System | 58 |
| 6.2 | Dual Eigen faces | 59 |
| 6.3 | Eigen face similarity | 60 |
| 6.4 | Bayesian Similarity | 61 |

**LIST OF SYMBOLS AND ABBREVIATIONS**

|  |  |
| --- | --- |
| PCA | PRINCIPLE COMPONENT ANALYSIS |
| LDA | LINEAR DISCRIMINANT ANALYSIS |
| MLE | MAXIMUM LIKELIHOOD ESTIMATION |
| M | NUMBER OF TRAINING IMAGES |
| C | COVARIANCE MATRIX |
| U | EIGENVALUE |
| V | EIGENVECTOR |
|  | A POSTERIOR PROBABILITY |
|  | LIKELIHOODS |
|  | PRIORS |

# ****CHAPTER 1****

# 

# INTRODUCTION

## 1.1 Introduction

Humans are very good at recognizing faces and complex patterns. Even a passage of time doesn't affect this capability and therefore it would help if computers become as robust as humans in face recognition. Face recognition system can help in many ways:

1. Checking for criminal records.
2. Enhancement of security by using surveillance cameras in conjunction with face recognition system.
3. Finding lost children's by using the images received  from the cameras fitted at some public places .
4. Knowing in advance if some VIP is entering the hotel.
5. Detection of a criminal at public place.
6. Can be used in different areas of science for comparing a entity with a set of entities.
7. Pattern Recognition.

This project is a step towards developing a face recognition system which can recognize static images. It can be modified to work with dynamic images. In that case the dynamic images received from the camera can first be converted in to the static one's and then the same procedure can be applied on them. But then there are lots of other things that should be considered . Like distance between the camera and the person , magnification factor, view [top ,side, front] etc.

### 1.2 Objectives:

1) To recognize a sample face from a set of given faces .

2) Use of Principal Component Analysis [Using Eigen face approach].

3) To Use Bayesian approach for recognition and compare it with Eigen face

approach.

## 1.3 Why Face Recognition?

Given the requirement for determining people's identity, the obvious question is what technology is best suited to supply this information? There are many different identification technologies available, many of which have been in wide-spread commercial use for years. The most common person verification and identification methods today are Password/PIN (Personal Identification Number) systems, and Token systems (such as your driver's license). Because such systems have trouble with forgery, theft, and lapses in users' memory, there has developed considerable interest in biometric identification systems, which use pattern recognition techniques to identify people using their physiological characteristics. Fingerprints are a classic example of a biometric; newer technologies include retina and iris recognition.

**1.4 Novel Applications of Face Recognition Systems**

Face recognition systems are no longer limited to identity verification and surveillance tasks. Growing numbers of applications are starting to use face-recognition as the initial step towards interpreting human actions, intention, and behavior, as a central part of next-generation smart environments. Many of the actions and behaviors humans display can only be interpreted if you also know the person's identity, and the identity of the people around them. Examples are a valued repeat customer entering a store, or behavior monitoring in an eldercare or childcare facility, and command-and-control interfaces in a military or industrial setting. In each of these applications identity information is crucial in order to provide machines with the background knowledge needed to interpret measurements and observations of human actions.

## 1.5 Commercial Systems and Applications

Currently, several face-recognition products are commercially available. Algorithms developed by the top contenders of the FERET competition are the basis of some of the available systems; others were developed outside of the FERET testing framework. While it is extremely difficult to judge, three systems Visionics, Viisage, and Miros are the current market leaders in face recognition.

**1.6 Literature Review**

[1] **Baback Moghaddam**, Mitsubishi Electric Research Laboratory, Cambridge, USA, **Tony Jebara**, and **Alex Pentland**, Massachusetts Institute of Technology, Cambridge, USA worked on continuous development of face recognition and proposed a new method. They proposed a new technique for direct visual matching of images for the purpose of face recognition and image retrieval, using probabilistic measure of similarity, based on Bayesian analysis of image differences. (Pattern Recognition, Vol. 33, No. 11, pps. 1771-1782, November, 2000).

**[2]** Matthew Turk and Alex Pentland, Vision and Modeling Group, The Media Laboratory, Massachusetts Institute of Technology, USA worked on Eigen Faces based Face Recognition using principal component analysis. M. Turk, A. Pentland, Eigen faces for Recognition, Journal of Cognitive Neuroscience, Vol. 3, No. 1, 1991, pp. 71-86.

2D face recognition using eigenfaces is one of the oldest types of face recognition. Turk and Pentland published the groundbreaking “Face Recognition Using Eigenfaces” in 1991. The method works by analyzing face images and computing eigenfaces which are faces composed of eigenvectors. The comparison of eigenfaces is used to identify the presence of a face and its identity.

There is a five step process involved with the system developed by Turk and Pentland. First, the system needs to be initialized by feeding it a set of training images of faces. This is used these to define the face space which is set of images that are face like. Next, when a face is encountered it calculates an eigenface for it. By comparing it with known faces and using some statistical analysis it can be determined whether the image presented is a face at all. Then, if an image is determined to be a face the system will determine whether it knows the identity of it or not. The optional final step is that if an unknown face is seen repeatedly, the system can learn to recognize it.

The eigenface technique is simple, efficient, and yields generally good results in controlled circumstances [1]. The system was even tested to track faces on film. There are also some limitations of eigenfaces. There is limited robustness to changes in lighting, angle, and distance [6]. 2D recognition systems do not capture the actual size of the face, which is a fundamental problem [4]. These limits affect the technique’s application with security cameras because frontal shots and consistent lighting cannot be relied upon.

[3] **T. De Bie, N. Cristianini, R. Rosipal**, worked on Eigenproblems in Pattern Recognition. Their work have brought Handbook of Computational Geometry for Pattern Recognition, Computer Vision, Neurocomputing and Robotics, E. Bayro-Corrochano (editor), Springer-Verlag, Heidelberg, April 2004

[4] **Srinivas Nagamalla**, Indian Statistical Institute, Kolkata, India and **Bibhas Chandra Dhara**, Department of Information Technology, Jadavpur University, Kolkata, India worked on continuous development of face recognition and proposed a new method. In this approach, first the probable positions of the facial landmarks are located from the gradient image. Secondly template matching is employed over a region around the probable positions to detect exact location of the facial landmarks. Then statistical and geometrical features are extracted from them. To reduce the dimension of the feature vector PCA is employed. The system has been proved to have advantage over the traditional face recognition system (2009 Seventh international conference on advances in pattern recognition, IEEE).

**[5] Wei-Min Lui and Chein-I Chang,** Dept. of CSE, Univ. of Maryland, Baltimore County, Baltimore, MD, USA, worked on different variants of principal component analysis. This paper presents four different variants of principal component analysis such as simultaneous PCA , Progressive PCA, Successive PCA, Prioritized PCA and except the simultaneous PCA all the other are new variants and each of it has own merits and it has not been explored in the literature(IEEE International Geoscience and Remote Sensing Symposium 2007).

# 1.7 Requirement Specifications

# 1.7.1 Introduction

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

## 1.7.2 Software Used

* MATLAB 7.8.0.347 (R2009a)
* Image Processing Toolbox 6.3(R2009a)

### 1.7.2.1 MATLAB 7.8.0.347 (R2009a)

### The MATLAB® high-performance language for technical computing integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

* Math and computation
* Algorithm development
* Data acquisition
* Modeling, simulation, and prototyping
* Data analysis, exploration, and visualization Scientific and engineering graphics
* Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. It allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar no interactive language such as C or FORTRAN.

Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. You can add on toolboxes for signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many other areas.

**1.7.2.2 Image Processing Toolbox**

Image Processing Toolbox™ provides a comprehensive set of reference-standard algorithms and graphical tools for image processing, analysis, visualization, and algorithm development. You can perform image enhancement, image deblurring, feature detection, noise reduction, image segmentation, spatial transformations, and image registration. Many functions in the toolbox are multithreaded to take advantage of multicore and multiprocessor computers.

1.8 Organization of the Project

In Chapter 2 we briefly review the basics of image processing and face recognition methods.

Chapter 3 deals with the preprocessing method for enhancing the input image.

Chapter 4 describes the Eigenfaces method for face recognition using Principal Component Analysis.

Chapter 5 deals with the Linear Discriminant Analysis for Feature Extraction.

Chapter 6 describes the Bayesian Classifier for face recognition.

Our experimental results are shown in Chapter 7.

Finally we draw a Conclusion at Chapter 8.

**CHAPTER 2**

**FUNDAMENTALS OF IMAGE PROCESSING AND FACE RECOGNITION SYSTEM**

# IMAGE PROCESSING SYSTEM

## 2.1 INTRODUCTION

# Digital images play an important role, both in daily-life applications such as satellite television, magnetic resonance imaging, and computed tomography as well as in areas of research and technology such as geographical information systems and astronomy.

### 2.1.1 Image

# An image is a two-dimensional function that represents a measure of some characteristic such as brightness or color of a viewed scene. An image is a projection of a 3D scene into a 2D projection plane. It can be defined as a two-variable function f(x, y) where for each position (x, y) in the projection plane, f(x, y) defines the light intensity at this point. A digital image is basically a numerical representation of an object.

### 2.1.2 Analog Image

# Am analog image can be mathematically represented as a continuous range of values representing position and intensity. An analog image is characterized by a physical magnitude varying continuously in space. For example, the image produced on the screen of a CRT monitor is analog in nature.

### 2.1.3 Digital Image

# A digital image is composed of picture elements called *pixels*. Pixels are the smallest sample of an image. A pixel represents the brightness at one point. Conversion of an analog image into a digital image involves two important operations, namely, sampling and quantization.

# Figure 2.1 Digital image from analog image

## 2.2 Digital Image Processing:

# Processing of an image by means of computer algorithms is termed as Digital Image Processing. The advantages of using computers for the processing of images are,

# Flexibility and Adaptability

# Data Storage and Transmission

## 2.3 Digital Image Representation:

# A digital image is a two dimensional discrete signal. A digital image is an *N* x *N* array of elements. Each element in the array is a number which represents the sampled intensity. For example, the representation of 4 x 4 images in matrix format and its three-dimensional view is shown:

# Figure 2.2 Digital image representation

### 2.3.1 Resolution

# Resolution is the ability to distinguish fine spatial detail. The spatial frequency at which a digital image is sampled (the sampling frequency) is often a good indicator of resolution. This is why dots-per-inch (dpi) or pixels-per-inch (ppi) are common and synonymous terms used to express resolution for digital images.

## 2.4 IMAGE TYPES

# Images can be classified under four categories:

# Binary image

# Grayscale Image

# Color Image

# Multispectral Image

### 2.4.1 Binary Images

# Binary images take only two values, i.e., either ‘0’ or ‘1’. The brightness graduation cannot be differentiated in the binary image. The binary image representation is illustrated in Figure.

# 

# Figure 2.3 Binary Image Representation

### 2.4.2 Grayscale Images:

# Grayscale images contain only brightness information. Each pixel value in a grayscale image corresponds to an amount or quantity of light. The brightness graduation can be differentiated in grayscale image. In a gray scale image, each pixel is represented by a byte or word, the value of which represents the light intensity at that point in the image. An 8-bit image will have a brightness variation form 0 to 255 where ‘0 represents black and ‘255’ represents white. A grayscale image measures only the light intensity. Each pixels scalar proportional to brightness.

# 

# Figure 2.4 Gray Scale Image Representation

### 2.4.2 Color Image:

# A color image has three values per pixel and they measure the intensity and chrominance of light. Each pixel is a vector of color components. Color images can be modeled as three-band monochrome image data, where each band of data corresponds to a different color.

# 

# Figure 2.5 Color Image

## 2.5 Elements of Image Processing System

# The different elements in an image processing system are

# Image acquisition element

# Image storage devices

# Image processing elements

# Image display devices

# 

# 

# 

# FACE RECOGNITION SYSTEM

## 2.6 Introduction

Biometrics has been widely used in forensic applications such as criminal identification and prison security. Biometric technology is rapidly evolving and has a very strong potential to be widely adopted in civilian applications such as electronic banking, e-commerce and access control. With the progress in biometric technology, these applications will increasingly use biometrics for authentication. The biometrics normally used for personal identification is,

1. Face Recognition
2. Fingerprint Recognition
3. Iris Recognition
4. Vein-pattern Recognition

# Face recognition is basically an attempt to develop a machine to mimic the human capability to distinguish different human faces. The term ‘recognition’ has been used to refer to many different visual abilities, including identification, categorization, and discrimination.

Faces are diverse, semi-rigid, semi-flexible, culturally significant, and part of our individual entity. There are many approaches to the face-recognition problem. Some techniques rely on a single face template or a model for recognition; others rely on facial sub-features. A variety of face recognition techniques are employed form correlation, neural networks, eigenfaces, Bayesian models. Given an arbitrary image, the goal of face recognition is to determine whether or not there are many faces in the image and, if present, returns the image location and extent of each face.

## 2.7 Relationship between Image Processing and Face Recognition:

## Image processing deals with different techniques which can improve the visual quality of the input image, whereas face recognition is concerned with the description and classification of objects or entities in the image. Image processing is often necessary as a preprocessing stage preceding face recognition. Image processing is required to improve the qualities and measurement of an image.

## 2.8 Recognizing Faces in a Single Image:

The single image recognition can be broadly classified into four categories:

1. Knowledge based methods
2. Feature invariant approaches
3. Template matching methods
4. Appearance based methods

## 2.9 Appearance based methods:

Appearance based methods rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face and non-face images. The learned characteristics are in the form of distribution models or discriminant functions that are consequently used for face recognition. An image or feature vector derived from an image is viewed as a random variable *x*, and this random variable is characterized for faces and non-faces by the class=conditional density functions *p*(*x/*face) and *p*(*x*/non-face). Bayesian classification or maximum likelihood can be used to classify a candidate image location as a face or non-face.

## 2.10 Challenges in Face Recognition:

The challenges associated with face recognition are

1. Facial Expression
2. Image Orientation:
3. Imaging Conditions:
4. Pose

**2.11 Approaches to Face Recognition:**

# Basically, there are two approaches to face recognition. They are (i) decision – theoretic approach, which is also called statistical approach, and (ii) Structural approach, or syntactic approach. In the decision theoretic approach, the pattern is represented as a vector in a vector space (feature space). Then a decision algorithm, which is mainly based on the statistical concept, is used to decide which class the pattern belongs to. In the structural method, the pattern is represented by its structure, e.g., a string of symbols, a graph connecting the primary elements, etc. The decision-theoretic method can be broadly classified into the classical and neural network approaches. The classical approach depends on the statistics of the input data to be classified; hence the classical method is also known as statistical approach to pattern recognition.

## 2.12 Automated Face Recognition System:

Automated face recognition systems use computers that execute programs of instructions to implement mathematical algorithms. An automated pattern recognition system use algorithms to process data collected either electronically through sensors or transcribed by humans. Figure 2.6 Face Recognition algorithm

The face recognition system is divided into two parts, Feature extraction and classifier. The feature extractor reduces the dimensionality of the input vectors to the classifier. The classification task uses the classifier to map a feature vector to a group. Once formulated, the mapping can be used to assign identification to each unlabeled feature vector subsequently presented to the classifier. A special case of feature extraction is feature selection that selects a subset of given measurements as features. Feature selection can be considered as a mapping from the primitive n-dimensional space to a lower-dimensional space. The feature selected is discriminate an object belonging to a different class.

## 2.13 Representations of Pattern Classes:

Three common representations of Patterns are:

1. Vector representation
2. String representation
3. Tree representation

## 2.14 Vector Representation of Patterns:

# Vectors are basically quantitative description of patterns. Suppose that *N* features are to be measured from each input pattern. Each set of *N* features can be considered as a vector *X* called a feature vector.

A face recognition system generally consists of 4 modules

i) Pre processing

ii) Dimensionality reduction using PCA

iii) Feature extraction for class separability using LDA

iv)Classification using Bayesian network

Figure 2.7 Basic Block diagram of Face Recognition

**CHAPTER 3**

# PREPROCESSING

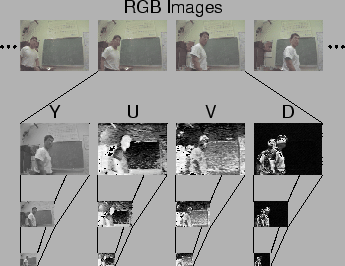


Figure 3 Preprocessing

Image pre-processing is the name for operations on images at the lowest level of abstraction whose aim is an improvement of the image data that suppress undesired distortions or enhances some image features important for further processing. It does not increase image information content. Its methods use the considerable redundancy in images. Neighboring pixels corresponding to one object in real images have the same or similar brightness value and if a distorted pixel can be picked out from the image, it can be restored as an average value of neighboring pixels. Various Preprocessing Operations include:

1. Point Operation
2. Geometric Operation

## 

## 3.1 Point operation

## Single-point processing is a simple method of http://homepages.inf.ed.ac.uk/rbf/HIPR2/mote.gifimage enhancement. This technique determines a pixel value in the enhanced image dependent only on the value of the corresponding pixel in the input image. The process can be described with the http://homepages.inf.ed.ac.uk/rbf/HIPR2/mote.gifmapping function. Point Operation includes

1 .**Histogram Equalization** - general method of modifying intensity distribution.

  2. **Normalization** - spreading out gray level distribution

#### 3.1.1 Histogram Manipulation

Histogram manipulation basically modifies the histogram of an input image so as to improve the visual quality of the image. In order to understand histogram manipulation, it is necessary that one should have some basic knowledge about the histogram of the image. The following section gives basic idea about histogram of an image and the histogram equalization technique used to improve the visual quality of an image.

##### 3.1.3 Histogram:

The histogram of an image is a plot of the number of occurrences of gray levels in the image against the gray levels values. The histogram provides a convenient summary of the intensities in an image, but it is unable to convey any information regarding spatial relationships between pixels. The histogram provides more insight about image contrast and brightness.

1. The histogram of a dark image will be clustered towards the lower gray level.
2. The histogram of a bright image will be clustered towards higher gray level.
3. For a low contrast image, the histogram will not be spread equally, that is, the histogram will be narrow.
4. For a high contrast image, the histogram will have an equal spread in the gray level.

Image brightness may be improved by modifying the histogram of the image.

##### 3.1.4 Histogram equalization

Equalization is process that attempt to spread out the gay levels in an image so that they are evenly distributed across their range. Histogram equalization reassigns the brightness values of pixels based on the image Histogram. Histogram equalization is a technique where the histogram of the resultant image is as flat as possible. Histogram equalization provides more visually pleasing results across a wider range of images.

**INPUT IMAGE**

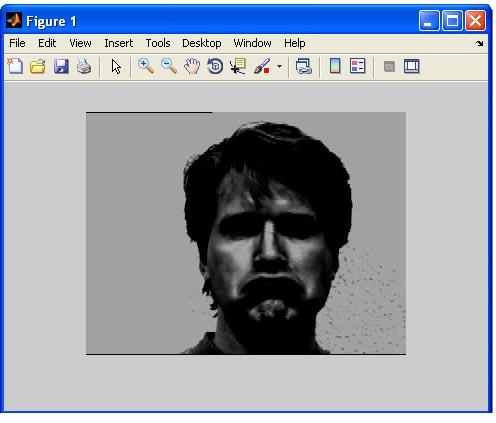


Figure 3.1 Input Image for Histogram Equalization

**HISTOGRAM of INPUT IMAGE**

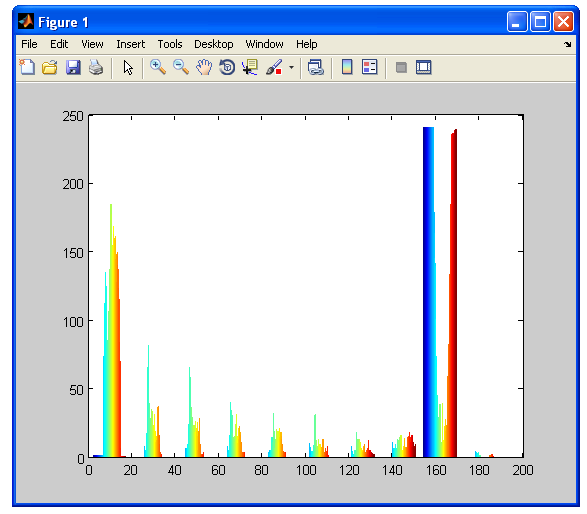


Figure 3.2 Histogram of Input Image

**HISTOGRAM EQUALIZED IMAGE**

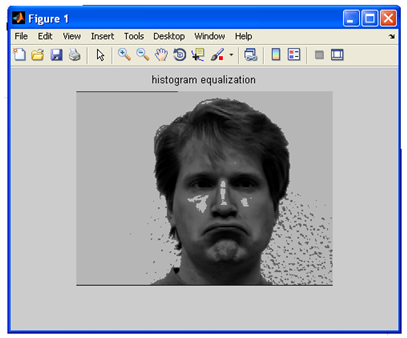
****

Figure 3.3 Histogram Equalized Image

**HISTOGRAM OF INPUT IMAGE AFTER EQUALIZATION**

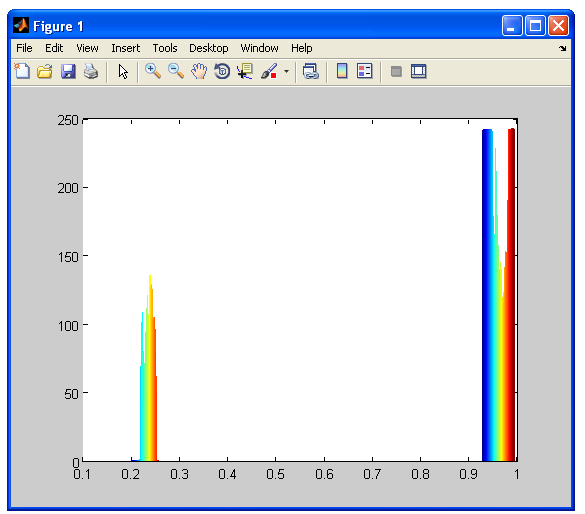
****

Figure 3.4 Histogram of Equalized Image

## 

## NORMALIZATIO

## 3.2. Normalization:

Contrast stretching or normalization is a simple image enhancement technique used to improve the contrast in an image by `stretching' the range of intensity values it contains to span a desired range of values, e.g. the full range of pixel values that the image type concerned allows. http://homepages.inf.ed.ac.uk/rbf/HIPR2/mote.gif

Contrast adjustment is done by scaling all the pixels of the image by a constant *k*. It is given by

g [*m, n*] = *f* [*m, n*]*\*k*

### Changing the contrast of an image, changes the range of luminance values present in the image. Specifying a value above 1 will increase the contrast by making bright samples brighter and dark samples darker, thus expanding on the range used. A value below 1 will do the opposite and reduce a smaller range of sample values.

### 

Figure 3.5 Input Image for Normalization

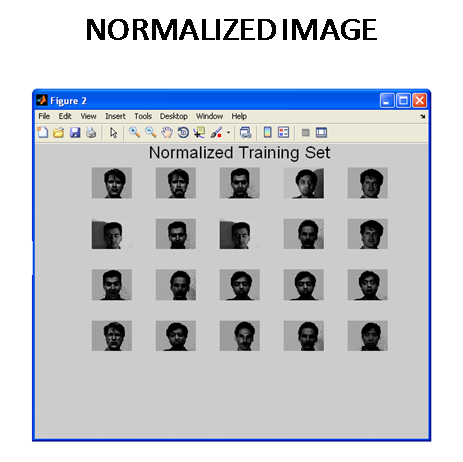
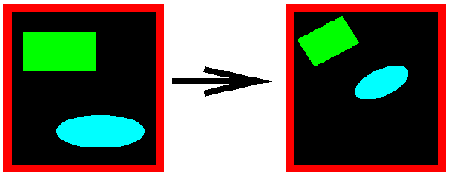


Figure 3.6 Normalized Image

# GEOMETRIC OPERATIONS



CONTENTS

[http://homepages.inf.ed.ac.uk/rbf/HIPR2/scale.gif](http://homepages.inf.ed.ac.uk/rbf/HIPR2/scale.htm)Scale - change image content size

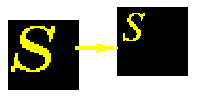
[http://homepages.inf.ed.ac.uk/rbf/HIPR2/rotate.gif](http://homepages.inf.ed.ac.uk/rbf/HIPR2/rotate.htm)Rotate - change image content orientation

## 3.3 Geometric operation:

A geometric operation maps pixel information (*i.e.* the intensity values at each pixel location Eqn:eqnxy1) in an input image to another location Eqn:eqnxy2 in an output image. For basic operators described in this package. Geometric operations includes

1. Geometric Scaling
2. Geometric Rotation

### 3.3.1 Geometric Scaling



The scale operator performs a geometric transformation which can be used to shrink or zoom the size of an image (or part of an image). Image reduction, commonly known as http://homepages.inf.ed.ac.uk/rbf/HIPR2/mote.gifsub sampling, is performed by replacement (of a group of pixel values by one arbitrarily chosen pixel value from within this group) or by interpolating between pixel values in a local neighborhoods. Image zooming is achieved by pixel replication or by interpolation. Scaling is used to change the visual appearance of an image, to alter the quantity of information stored in a scene representation, or as a low-level preprocessor in multi-stage image processing chain which operates on features of a particular scale.

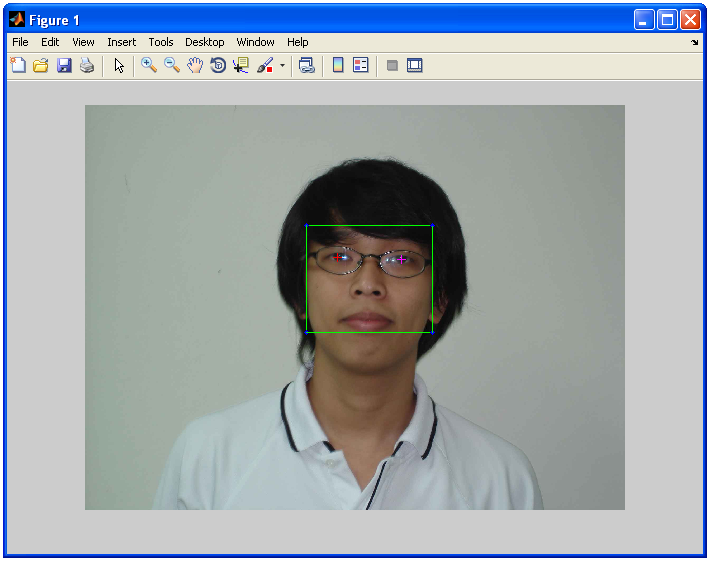
Scaling compresses or expands an image along the coordinate directions. As different techniques can be used to subsample and zoom, each is discussed in turn.

**INPUT IMAGE**



Figure 3.7 Input Image for Scaling

**SCALED IMAGE**



# Figure 3.8 Scaled Image

### 3.3.2 Image Rotation (change in image content orientation)

# http://homepages.inf.ed.ac.uk/rbf/HIPR2/rotateb.gif

# Image rotation is performed by computing the inverse transformation for every destination pixel. Output pixels are computed using bilinear interpolation. RGB images are computed by evaluating one color plane at a time. There are no gamma corrections so purists might want to correct for image gamma before and after rotation.

# Small degree counter-clockwise rotation with bilinear interpolation produces an image that is slightly larger than the original surrounded by black regions that do not map to any pixels in the original image.

**CHAPTER 4**

# DIMENSIONALITY REDUCTION USING PRINCIPAL COMPONENT ANALYSIS

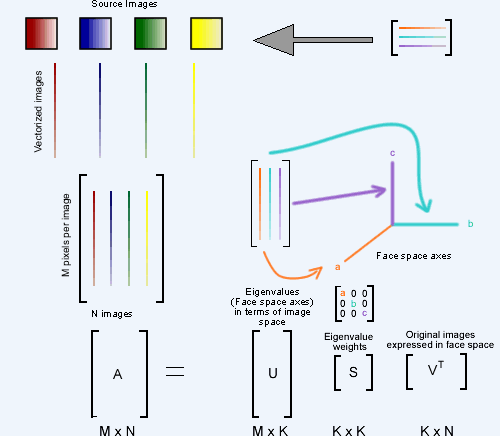
****

Figure 4 Principal Component Analysis

## 4.1Principal Component Analysis

Principal component analysis (PCA) is a classical statistical method. This linear transform has been widely used in data analysis and compression. Principal Components Analysis is a method that reduces data dimensionality by performing a covariance analysis between factors.

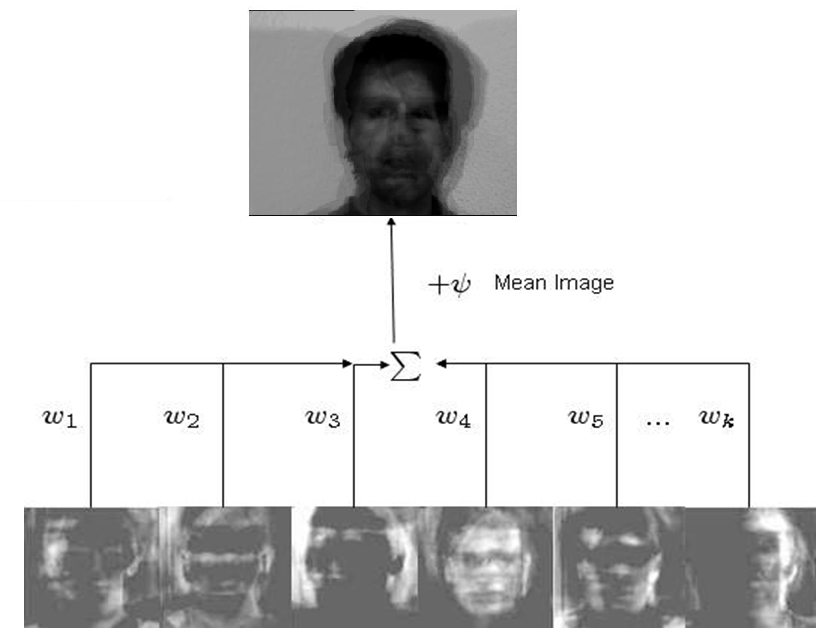
# Face description can be obtained by computing the Eigen vectors of the image’s autocorrelation matrix. These Eigen vectors are known as eigenfaces. The task of facial recognition is discrimination the image data into several classes. The input signals are highly noisy, yet the input images are not completely random and in spite of their differences there are patterns such as eyes, nose, mouth, etc., present in the facial image. These patterns are separated by specific distances. These characteristics features are called eigenfaces. The original face can be reconstructed by combining all the eigenfaces in the right proportion. The eigenvector of the covariance matrix forms the basis of the KL transform which is also known as the Principal Component Analysis (PCA) or the Hottelling transform. PCA generates a set of orthogonal axes of projections known as the principal components, or the eigenvectors, of the input data distribution in the order of decreasing variance. By means of PCA, one can transform each original image into a corresponding eigenfaces. In order to determine whether the object in the images is the desired face or not, extract the weights from the eigenfaces and the face to be recognized. Similar images possess similar eigenfaces, and hence similar weights. Thus all images having similar weights are likely to be similar faces.

### ****4.2 Mathematics****

The aim is to represent a face as a linear combination of a set of basis images. That is:

(4.1)

Where  represents the face with the mean subtracted from it, represent weights and  the eigenvectors.



The Principal Components basically seek directions in which it is more efficient to represent the data. This is particularly useful for reducing the computational effort. To understand this,  suppose we get 60 such directions, out of these about 40 might be insignificant and only 20 might represent the variation in data significantly, so for calculations it would work quite well to only use the 20 and leave out the rest.

The number of Eigenfaces that we would obtain therefore would be equal to the number of images in the training set. Let us take this number to be *M*. Some of the Eigenfaces are more important in encoding the variation in face images, thus we could approximate faces using only the *K* most significant Eigenfaces.

### ****4.3 Assumptions:****

1. There are *M* images in the training set.

2. There are *K* most significant Eigenfaces using which we can satisfactorily approximate a face. K < M.

3. All images are *N* x *N* matrices, which can be represented as *N2* x 1dimensional vectors. The same logic would apply to images that are not of equal length and breadths. To take an example: An image of size 112 x 112 can be represented as a vector of dimension 12544 or simply as a point in a 12544 dimensional space.

**4.4 Algorithm for Finding Eigenfaces:**

**1.** Obtain *M* training images,  I1 , I2 ,….. IM ,it is very important that the images are centered.

**2.** Represent each image*I* i as a vector Γi as discussed above.

**= (4.2)**

**3.** Find the average face vector.

(4.3)

**4.** Subtract the mean face from each face vector  to get a set of vectors. The purpose of subtracting the mean image from each image vector is to be left with only the distinguishing features from each face and “removing” in way information that is common.

(4.4)

**5.** Find the Covariance matrix *C*:

*C* = *AAT* , where  (4.5)

Note that the Covariance matrix has simply been made by putting one modified image vector obtained in one column each.

Also note that *C* is a *N*2 x *N*2 matrix and *A* is an *N* 2 x Mmatrix.

**6.** We now need to calculate the Eigenvectors *ui* of *C*, However note that *C* is a *N*2  x *N*2  matrix and it would return  *N*2 Eigenvectors each being *N*2 dimensional. For an image this number is HUGE.  The computations required would easily make your system run out of memory. How do we get around this problem?

**7.**Instead of the *AAT* Matrix consider the matrix *ATA.*. Remember*A* is a *N* 2 x M matrix, thus is *ATA* a *M* x *M* matrix. If we find the Eigenvectors of this matrix, it would return *M* Eigenvectors, each of Dimensions *M* x 1, let’s call these Eigenvectors*vi.*

Now from some properties of matrices, it follows that: *ui=Avi* we have found out *vi* earlier. This implies that using *vi* we can calculate the *M* largest Eigenvectors of *AAT*. Remember that *M<<N*2 as *M* is simply the number of training images.

**8.** Find the best M Eigenvectors of *C* = *AAT* by using the relation discussed above. That is: *ui=Avi* . Also keep in mind that ||*ui* =1||

**9.** Select the best *K* Eigenvectors

### ****4.5 Finding Weights:****

The Eigenvectors found at the end of the previous section, *ui* when converted to a matrix in a process that is reverse to that in STEP 2, have a face like appearance. **Since** these are Eigenvectors and have a face like appearance, they are called Eigen faces. Sometimes, they are also called as **Ghost Images** because of their weird appearance.

Now each face in the training set (minus the mean),  can be represented as a linear combination of these Eigenvectors *ui* :

, (4.6)

where *uj* ‘s are Eigenfaces.

These weights can be calculated as:

(4.7)

Each normalized training image is represented in this basis as a vector.

(4.8)

Where i = 1, 2… M. This means we have to calculate such a vector corresponding to every image in the training set and store them as templates.

### ****4.6 Recognition Task:****

Now consider we have found out the Eigenfaces for the training images, their associated weights after selecting a set of most relevant Eigenfaces and have stored these vectors corresponding to each training image.

If an unknown probe face \Gamma is to be recognized then:

**1.** We normalize the incoming probe  as. (4.9)

**2.** We then project this normalized probe onto the Eigen space (the collection of Eigenvectors/faces) and find out the weights.

(4.10)

**3.**The normalized probe \Phi can then simply be represented as:

After the feature vector (weight vector) for the probe has been found out, we simply need to classify it. In case we use distance measures, classification is done as:

Find. This means we take the weight vector of the probe we have just found out and find its distance with the weight vectors associated with each of the training image.

And if, where \Theta is a threshold chosen heuristically, then we can say that the probe image is recognized as the image with which it gives the lowest score.

If however  then the probe does not belong to the database. I will come to the point on how the threshold should be chosen.

For distance measures the most commonly used measure is the Euclidean Distance. The other being the Mahalanobis Distance. The Mahalanobis distance generally gives superior performance. Let’s take a brief digression and look at these two simple distance measures and then return to the task of choosing a threshold.

### ****4.7 Distance Measures****

**4.7.1Euclidean Distance**

The Euclidean Distance is probably the most widely used distance metric. It is a special case of a general class of norms and is given as:

(4.11)

**4.7.2 The Mahalanobis Distance**

The Mahalanobis Distance is a better distance measure when it comes to pattern recognition problems. It takes into account the covariance between the variables and hence removes the problems related to scale and correlation that are inherent with the Euclidean Distance. It is given as:

(4.12)

Where *C* is the covariance between the variables involved.

**CHAPTER 5**

# FEATURE EXTRACTION USING

# LINEAR DISCRIMINANT ANALYSIS

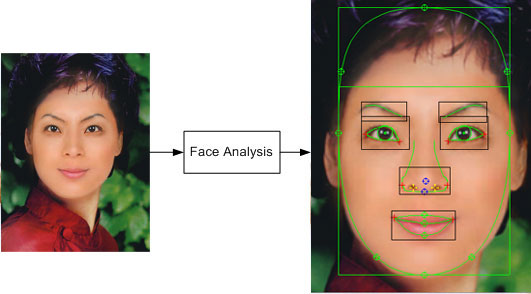


Figure 5 Feature Extraction

## 

## 5.1 Introduction

Linear Discriminant Analysis (LDA) is a well-known scheme for feature extraction and dimension reduction. It has been used widely in many applications involving high-dimensional data, such as face recognition and image retrieval. An intrinsic limitation of classical LDA is the so-called *singularity problem*, that is, it fails when all scatter matrices are singular. A well-known approach to deal with the singularity problem is to apply an intermediate dimension reduction stage using Principal Component Analysis (PCA) before LDA. The algorithm, called PCA+LDA, is used widely in face recognition. However, PCA+LDA has high costs in time and space, due to the need for an eigen-decomposition involving the scatter matrices.

There are many possible techniques for classification of data. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two commonly used techniques for data classification and dimensionality reduction. Linear Discriminant Analysis easily handles the case where the within-class frequencies are unequal and their performance has been examined on randomly generated test data. This method maximizes the ratio of between-class variance to the within-class variance in any particular data set thereby guaranteeing maximal separability. The use of Linear Discriminant Analysis for data classification is applied to classification problem in speech recognition. We decided to implement an algorithm for LDA in hopes of providing better classification compared to Principal Components Analysis. The prime difference between LDA and PCA is that PCA does more of feature classification and LDA does data classification. In PCA, the shape and location of the original data sets changes when transformed to a different space whereas LDA doesn’t change the location but only tries to provide more class separability and draw a decision region between the given classes. This method also helps to better understand the distribution of the feature data.

## 5.2 Different Approaches to LDA

Data sets can be transformed and test vectors can be classified in the transformed space by two different approaches.

*Class-dependent transformation*: This type of approach involves maximizing the ratio of betweenclass variance to within class variance. The main objective is to maximize this ratio so that adequate class separability is obtained. The class-specific type approach involves using two optimizing criteria for transforming the data sets independently.

*Class-independent transformation*: This approach involves maximizing the ratio of overall varianceto within class variance. This approach uses only one optimizing criterion to transform the data sets and hence all data points irrespective of their class identity are transformed using this transform. In this type of LDA, each class is considered as a separate class against all other classes.

## 5.3 Mathematical operations

In this section, the mathematical operations involved in using LDA will be analyzed the aid of sample set. For ease of understanding, this concept is applied to a two-class problem. Each data set has 100 2-D data points.

Formulate the data sets and the test sets, which are to be classified in the original space. For ease of understanding let us represent the data sets as a matrix consisting of features in the form given below:

|  |
| --- |
| s (5.1) |
|  |
| Compute the mean of each data set and mean of entire data set. Let and be the mean of set 1 and set 2 respectively and be mean of entire data,which is obtained by merging set 1 and set 2, is given by Equation5.2.  (5.2) | |
|

where *p*1 and *p*2

(5.3)

Therefore, for the two-class problem,  
 (5.4)

*Sw* = 0.5x *cov*1+ 0.5x *cov*2  (5.5)

the probability factor is assumed to be 0.5.

In LDA, within-class and between-class scatter are used to formulate criteria for class separability. Within-class scatter is the expected covariance of each of the classes. The scatter measures are computed using Equations  [3](#page4) and  [4](#page4).

All the covariance matrices are symmetric. Let *cov*1 and *cov*2 be the covariance of set 1 and set 2 respectively. Covariance matrix is computed using the following equation.

(5.6)

The between-class scatter is computed using the following equation.

(5.7)

(5.8)

For the class independent transform, the optimizing criterion is computed as

(5.9)

By definition, an eigen vector of a transformation represents a 1-D invariant subspace of the vector space in which the transformation is applied. A set of these eigen vectors whose corresponding eigen values are non-zero are all linearly independent and are invariant under the transformation. Thus any vector space can be represented in terms of linear combinations of the eigen vectors. A linear dependency between features is indicated by a

zero eigen value. To obtain a non-redundant set of features all eigen vectors corresponding to non-zero eigen values only are considered and the ones corresponding to zero eigen values are neglected

**CHAPTER 6**

# CLASSIFICATION USING BAYESIAN CLASSIFIER

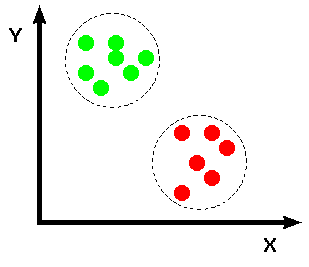


Figure 6 Classification

## 6.1 Classification:

Classification includes a broad range of decision-theoretic approaches to the identification of images (or parts thereof). All classification algorithms are based on the assumption that the image in question depicts one or more features (e.g., geometric parts in the case of a manufacturing classification system, or spectral regions in the case of remote sensing, as shown in the examples below) and that each of these features belongs to one of several distinct and exclusive classes. The classes may be specified a priori by an analyst (as in supervised classification) or automatically clustered (i.e. as in unsupervised classification) into sets of prototype classes, where the analyst merely specifies the number of desired categories.

Image classification analyzes the numerical properties of various image features and organizes data into categories. Classification algorithms typically employ two phases of processing: training and testing. In the initial training phase, characteristic properties of typical image features are isolated and, based on these, a unique description of each classification category, i.e. training class, is created. In the subsequent testing phase, these feature-space partitions are used to classify image features. The description of training classes is an extremely important component of the classification process. In supervised classification, statistical processes (i.e. based on an a priori knowledge of probability distribution functions) or distribution-free processes can be used to extract class descriptors. Unsupervised classification relies on clustering algorithms to automatically segment the training data into prototype classes. In either case, the motivating criteria for constructing training classes is that they are:

* independent, i.e. a change in the description of one training class should not change the value of another,
* discriminatory, i.e. different image features should have significantly different descriptions, and
* reliable, all image features within a training group should share the common definitive descriptions of that group.

A convenient way of building a parametric description of this sort is via a http://homepages.inf.ed.ac.uk/rbf/HIPR2/mote.giffeature vector , where n is the number of attributes which describe each image feature and training class. This representation allows us to consider each image feature as occupying a point, and each training class as occupying a sub-space (i.e. a representative point surrounded by some spread, or deviation), within the n-dimensional classification space.

Prior probability:

A prior probability is an initial probability value originally obtained before anyadditional information is obtained.

Posterior probability:

A posterior probability is a probability value that has been revised by using additional information that is later obtained

**6.2 Bayes Theorem:**

Bayes' theorem expresses the conditional probability, or "posterior probability", of a hypothesis H (i.e. its probability after evidence E is observed) in terms of the "prior probability" of H, the prior probability of E, and the conditional probability of E given H. It implies that evidence has a confirming effect if it is more likely given H than given not-H.

Mathematically, Bayes' rule states

or, in symbols,

**6.3 Maximum likelihood estimation (MLE):**

Likelihoodi**s** a function of the parameters of a statistical model. The *likelihood* of a set of parameter values given some observed outcomes is equal to the *probability* of those observed outcomes given those parameter values

Maximum likelihood estimation (MLE) is a statistical method used for fitting a statistical model to data, and providing estimates for the model's parameters.

6.4 Conditiional Probability:

Conditional probability is the probability of some event *A*, given the occurrence of some other event *B*. Conditional probability is written *P*(*A*|*B*), and is read "the (conditional) probability of *A*, given *B*" or "the probability of *A* under the condition *B*".

## 

## 6.5 Bayesian Maximum Likelihood Estimation:

## We have used a probabilistic similarity measure based on the Bayesian belief that the image intensity differences, denoted by are characteristic of typical variations in appearance of a individual. In particular, we define two classes of facial image variations: intrapersonal variations (corresponding, for example to different facial expressions of the same individual) and extra personal variations (corresponding to variations between different individuals). Our similarity measure is then expressed in terms of the probability.

(5.1)

Where,

is the *a posteriori probability*

Given by Bayes rule, using estimates of the *likelihoods*  and.

These likelihoods are derived from training data using an efficient subspace method for density estimation of high-dimensional data. Bayesian approach to face recognition is possible the first instance of a non-Euclidean similarity measure used in face recognition. The mechanics of Bayesian matching has computational and storage advantages over most linear methods for large databases.

## 6.5.1 Probabilistic Similarity Measures:

In Probabilistic Similarity Measures We define two distinct and mutually exclusive classes: representing intrapersonal variations between multiple images of the same individual (e.g., with different expressions and lighting conditions), and representing variations in matching two different individuals. We will assume that both classes are Gaussian-distributed and seek to obtain estimates of the likelihood functions and for a given intensity difference.

Given these likelihoods we can evaluate a similarity score between a pair of images directly in terms of the intrapersonal a posteriori probability as given by Bayes rule:

(5.2)

Where the priors can be set to reflect specific operating conditions (e.g., number of test images vs. the size of the database) or other sources of a priori knowledge regarding the two images being matched. An alternative probabilistic similarity measure can be defined in simpler form using the Intrapersonal likelihood alone,

(5.3)

Thus leading to maximum likelihood (ML) recognition as opposed to MAP recognition.

## 6.5.2 Subspace Density Estimation:

One difficulty with this approach is that the intensity difference vector is high dimensional, with with *N* typically of. Therefore few almost always lack sufficient independent training samples to compute reliable 2nd order statistics for the likelihood densities (i.e., singular covariance matrices will result). Even if we were able to estimate these statistics, the computational cost of evaluating the likelihood is formidable. Furthermore, this computation would be highly inefficient since the intrinsic dimensionality or major degrees of freedom of is likely to be significantly smaller than *N*.

To deal with the high dimensionality of , we make use of the efficient density estimation method proposed by moghaddam and pentland which divides the vector space into two complementary subspaces using an eigenfaces decomposition. This method relies on a Principle Component Analysis (PCA) to form a low dimensional estimate of the complete likelihood which can be evaluated using only the first *M* principle components, where *M<<N.*

The complete likelihood estimate can be written as the product of two independent marginal Gaussian densities

. (5.4)

= (5.5)

Where is the true marginal density in *F* is the estimated marginal density jin the orthogonal complement , are the principal components and is the residual (DFFS). The optimal value for the weighting paramenter found by minimizing cross entropy is simply the average of the Eigen values.

(5.6)

6.5.3 Efficient Similarity Computation:

Consider a feature space ofvectors, the differences between two images ( and.). The two classes of interest in this space corrospond to intrapersonal and extrapeersonal variations and eachi is modeled as a high dimensional Gaussian density.

(5.7)

(5.8)

However, these computations can be greatly simplified by offline transformations. To compute the likelihoodsand we pre process theimages with whitening transformations and consequently every image is stored as two vectors of whitened subspace coefficients; **i** for **e** intrapersonal and e for extra personal.

(5.9)

Where, and are matrices of the largest eigenvalues and eigenvectors of or, with subspace dimensionalities of and, respectively. After this preprocessing and with the normalizing denominators precomputed, evaluating the likelihoods is reduced to computing simple Euclidean distances for the exponents

(5.10)

(5.11)

These likelihoods are then used to compute the MAP similarity *S* in eq2; since the Euclidean distances in the exponents of are of dimensions andfor the **i** and **e** vectors, respectively, only arithmetic operations are required for each similarity computation. In this manner, one avoids unnecessary and repeated image differencing and online projections.

(5.12)

## 6.5.4 Eigenfaces Matching:

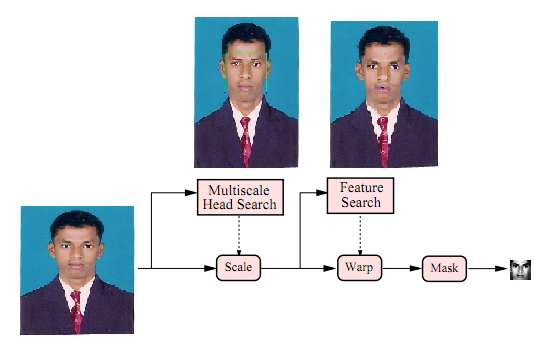
Eigen faces method uses the normalized images from the training set and the test set were projected onto eigenspace. A nearest neighbor rule based on a Euclidean distance was then used to match each test image to a training image. 

Figure 6.1 Face alignment system

## 6.5.5 Bayesian Matching:

visualizing this is to plot their mutual principal components-i.e., perform PCA on the combined dataset and project each vector onto the principal eigenvectors. Such a visualization is shown in fig5a. Which is a 3D scatter plot of the first 3 principal components. This plot shows what appears to be two completely enmeshed distributions, both having near-zero means and differing primarily in the amount of scatter, with displaying smaller intensity differences as expected.

It therefore appears that one cannot reliably distinguish low-amplitude extra personal differences (of which there are many) from intrapersonal ones.

## 6.5.6 DUAL EIGENFACES:

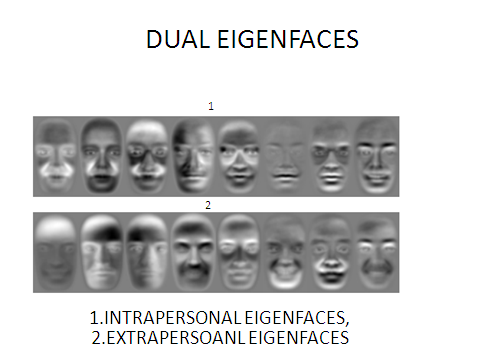


Figure 6.2 Dual Eigenfaces

We note that the two mutually exclusive classes and corrospond to a dual set of eigenfaces. Note that the intrapersonal variations represent subtle variations due mostly to expression changes (and lighting) whereas the extrapersonal variations are more representative of standard variations such as hair color, facial hair and glasses. This supports the basic intuition that intensity differences of the extrapersonal type span a larger vector space similar to the volume of facespace spanned by standard eigenfaces, whereas the intrapersonal eigenspace corrosponds to a more tightly constrained subspace.

**OPERATIONAL FLOW DIAGRAM**

EIGENFACE SIMILARITY

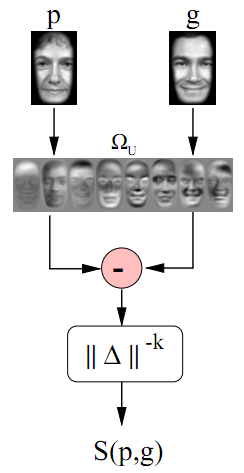


Figure 6.3 Eigen face similarity

**PROBABILISTIC SIMILARITY**

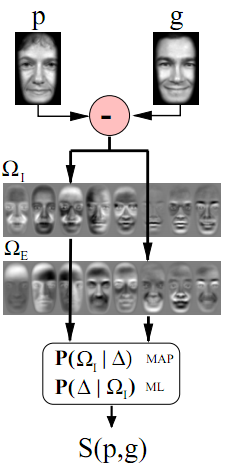


Figure 6.4 Bayesian Similarity

# CHAPTER 7

# RESULTS AND DISCUSSIONS

# EIGENFACES BASED FACE RECOGNITION

# 

# Figure 7.1 Input Image

# 

# Figure 7.2 Normalized Image

# MEAN IMAGE

# 

# Figure 7.3 Mean Image

# EIGENFACES

# 

# Figure 7.4 Eigen faces

# RECONSTRUCTED IMAGE

# 

# Figure 7.5 Reconstructed Image

# EUCLIDEAN DISTANCE

# 

# Figure 7.6 Euclidean Distance

# UNIDENTIFIED IMAGE

# 

# Figure 7.7 Unidentified Image

# BAYESIAN APPROACH BASED FACE RECOGNITION

# INPUT IMAGE

# 

# Figure 7.8 Test image

# SELECTING FEATURES

# 

# Figure 7.9 Selecting Coordinates

# 24 X 24 SCALED IMAGE

# 

# Figure 7.10 Scaled Image

# GRAYSCALE IMAGE

# 

# Figure 7.11 Grayscale Image

# RECOGNIZED IMAGE

# 

# Figure 7.12 Recognized Image

# 

# Figure 7.13 Output

# INPUT IMAGE

# 

# Figure 7.14 Test Image

# SELECTING FEATURES

# 

# Figure 7.15 Selecting Coordinates

# 24 X 24 SCALED IMAGE

# 

# Figure 7.16 Scaled Image

# OUTPUT

# 

# Figure 7.17 Output

# CONCLUSION

Face recognition technology has come a long way in the last twenty years. Today, machines are able to automatically verify identity information for secure transactions, for surveillance and security tasks, and for access control to buildings etc. These applications usually work in controlled environments and recognition algorithms can take advantage of the environmental constraints to obtain high recognition accuracy. However, next generation face recognition systems are going to have widespread application in smart environments where computers and machines are more like helpful assistants.

To achieve this goal computer must be able to reliably identify nearby people in a manner that fits naturally within the pattern of normal human interactions. They must not require special interactions and must conform to human intuitions about when recognition is likely. This implies that future smart environments should use the same modalities as humans, and have approximately the same limitations.

Face recognition technologies have been associated generally with very costly top secure applications. Today the core technologies have evolved and the cost of equipments is going down dramatically due to the integration and the increasing processing power. Certain applications of face recognition technology are now cost effective, reliable and highly accurate.

**FUTURE WORK**

Face recognition system proposed in this work very well under constrained conditions, although all systems work much better with frontal mug-shot images and constant lighting. All current face recognition algorithms fail under the vastly varying conditions under which humans need to and are able to identify other people. Next generation person recognition systems will need to recognize people in real-time and in much less constrained situations.

We believe that identification systems that are robust in natural environments, in the presence of noise and illumination changes, cannot rely on a single modality, so that fusion with other modalities is essential.

# APPENDIX

Source Code

function [aimg alex aley arex arey amx amy] =imalign (bigimg, imheight, imwidth, filesavepath, leyex, reyex, eyey, mouthy)

if (nargin<3)

Imheight=292;

Imwidth=240;

Leyex=56;

Reyex=184;

Eyey=88;

Mouthy=224;

End

If nargin>2 && nargin<7

Leyex=imwidth/4;

Reyex=imwidth\*3/4;

Eyey=imheight/3;

Mouthy=imheight\*3/4;

End

Img=imread (bigimg);

Mouthx= (leyex+reyex)/2;

Os=size(img);

Figure; imshow (img);

[x,y]=getpts;

% align image

%compute rot scale deltX deltY

lex=x (1);

ley=y (1);

rex=x(2);

rey=y(2);

mx=x(3);

my=y(3);

hold on;

plot([lex rex mx],[ley rey my],'bh');

X=[lex rex mx];

Y=[ley rey my];

Xc=[leyex reyex mouthx];

Yc=[eyey eyey mouthy];

A1=X\*X'+Y\*Y';

A2=lex+rex+mx;

A3=ley+rey+my;

A4=X\*Xc'+Y\*Yc';

A5=X\*Yc'-Y\*Xc';

A6=leyex+reyex+mouthx;

A7=eyey+eyey+mouthy;

%At=b, t=A\b

A=[A1 0 A2 A3;0 A1 -A3 A2;A2 -A3 3 0;A3 A2 0 3];

b=[A4 A5 A6 A7]';

t=A\b;

k1=t(1,1);

k2=t(2,1);

t1=t(3,1);

t2=t(4,1);

rot=atan2(k2,k1);

scale=(k1\*k1+k2\*k2)^0.5;

ox=-t1;

oy=-t2;

angle=-rot\*180/pi;

aimg=imrotate(img,angle,'bilinear');

% after rotation, the (0,0) is changed;

% calculate the shifts to adjust the crop

sx=0;sy=0;

if 0<angle && angle<=90 sy=os(2)\*sin(-rot); end

if -90<=angle && angle<0 sx=os(1)\*sin(rot); end

if 90<=angle && angle<=180

sx=os(2)\*sin(-rot-pi/2);

sy=size(aimg,1);

end

if -90>=angle && angle>=-180

sx=size(aimg,2);

sy=os(1)\*sin(rot-pi/2);

end

sx=sx\*scale;

sy=sy\*scale;

aimg=imresize(aimg,scale,'bilinear');

cx=round(ox+sx+1);

cy=round(oy+sy+1);

aimg=imcrop(aimg,[cx,cy,imwidth-1,imheight-1]);

%if img not big enough, pad zeros

if size(aimg,1)<imheight || size(aimg,2)<imwidth

timg =cast (zeros (imheight, imwidth, size (aimg, 3)),class(aimg));

if cx<1 px=-cx+1; else px=1; end

if cy<1 py=-cy+1; else py=1; end

timg(py:py+size(aimg,1)-1,px:px+size(aimg,2)-1,:)=aimg;

aimg=timg;

end

alex=(lex\*cos(rot)-ley\*sin(rot))\*scale-ox;

aley=(lex\*sin(rot)+ley\*cos(rot))\*scale-oy;

arex=(rex\*cos(rot)-rey\*sin(rot))\*scale-ox;

arey=(rex\*sin(rot)+rey\*cos(rot))\*scale-oy;

amx=(mx\*cos(rot)-my\*sin(rot))\*scale-ox;

amy=(mx\*sin(rot)+my\*cos(rot))\*scale-oy;

figure;

imshow(aimg)

aimg=rgb2gray(aimg);

colormap(gray(256));

imwrite(aimg,filesavepath);

hold on;

%plot ([alex arex amx],[aley arey amy],'r+');

IntensityDif(filesavepath);

function ans= IntensityDif (inputimage)

M=60;

filepath='Training set\';

indexa='a';

indexb='b';

img\_matrix=[];

ImgDifVecs=[];

Vectors=[];

Values=[]; % To hold the eigenvalues (sorted) from pc\_evectors

Psi=[];% To hold the mean image difference vector from pc\_evectors

IntraCovMat=[]; % To hold the covariance matrix of the intrapersonal difference vectors

DimDesired =25; % The number of top eigenvectors I want

DiagEigValMat=[]; % To store the diagonal matrix of the top DimDesired Eigenvalues

imgMatrixA=[]; %To store 'a' filename images

imgMatrixB=[];%To store 'b' filename images

ExtraDifImgMatrix=[];%To hold the extrapersonal image difference vectors

ExtraDesiredSubDim=25; % The number of top extrapersonal eigenvalues

for i=1:M

strimgnameA=strcat(int2str(i), indexa);

strfilenamea=strcat(strimgnameA, '.png');

strfilepath=strcat(filepath, strfilenamea);

img=imread(strfilepath);

sizeA=numel(img);

img=reshape(img', sizeA,1);

img\_matrix =[img\_matrix img];

imgMatrixA =[imgMatrixA img];

strimgnameB=strcat(int2str(i), indexb);

strfilenamea=strcat(strimgnameB, '.png');

strfilepath=strcat(filepath, strfilenamea);

img=imread(strfilepath);

sizeA=numel(img);

img=reshape(img', sizeA,1);

img\_matrix =[img\_matrix img];

imgMatrixB =[imgMatrixB img];

end

for i =1:M

ImgDifVecs(:,i) = imgMatrixA(:,i) - imgMatrixB(:,i);

ImgDifVecs(:,i+1) = imgMatrixB(:,i) - imgMatrixA(:,i);

end

[Vectors,Values,Psi,IntraCovMat] = pc\_evectors(ImgDifVecs,DimDesired);

i=30;

[normalizing\_denom] = cal\_normalizingDenom(Values,DimDesired);

[transVecMat]= perform\_whitening(Vectors, Values,img\_matrix,DimDesired);

imgtest= imread(inputimage);

sizeA=numel(imgtest);

imgtest=reshape(imgtest', sizeA,1);

[whitenedImgVec]= perform\_whitening(Vectors, Values,imgtest,DimDesired);

[MLresult,index]= cal\_Likelihoods (transVecMat,whitenedImgVec,normalizing\_denom);

if MLresult>1.3119e-063

fprintf('recognized')

msgbox('RECOGNIZED', 'Output')

else

fprintf('not recognized');

msgbox('Not recognized', 'Output')

return

end

if(mod(index,2)==0)

strimgname=strcat(int2str(index/2),'b');

else

strimgname=strcat(int2str(index/2),'a');

end

strimgname

%Show the input image for feedback purpose

figure(3)

strfilepath=inputimage;

img=imread(strfilepath);

img=histeq(img,255);

subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)

title('grayscale image')

imshow(img)

drawnow;

%Show the image found for feedback purpose

figure(4);

strfilenamea=strcat(strimgname, '.png');

strfilepath=strcat(filepath, strfilenamea);

img=imread(strfilepath);

img=histeq(img,255);

subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)

title('recognized image')

imshow(img)

drawnow;

function [transVecMat]= perform\_whitening(Vectors, Values, imgM,subDim)

DiagVal=[];

TempVal=[];

w=[];

for i =1:subDim

TempVal(i) = (Values(i))^(-1/2);

end

DiagVal= diag(TempVal);

transVecMat=[];%The matrix to hold the whitened image vectors

for i=1:size(imgM,2)

transVecMat(:,i) = DiagVal \* Vectors'\* double(imgM(:,i)); %Perform whitening

end

function [normalizing\_denom] = cal\_normalizingDenom(Values,DimDesired) %L is the covariance matrix returned from pc\_evectors

normalizing\_const = 2\*pi;

normalizing\_const= normalizing\_const ^(DimDesired/2); % This is the constant as specified in the first part of the normalizing formula

subVal=[];

subVal=Values(1:DimDesired);

temp=1;

for i =1:DimDesired

temp =temp \* subVal(i);

end

normalizing\_denom = normalizing\_const \* (temp)^(1/2)

function [normalizedWhitVecs] = cal\_NormalizeWhit(normalizing\_denom,transVecMat)

normalizedWhitVecs=[];

for i=1:size(transVecMat,2)

normalizedWhitVecs(:,i) = transVecMat(:,i)/normalizing\_denom;

end

function [MLresult,index]= cal\_Likelihoods (WhitenVecs,ImgColVec,normalizing\_denom)

likelihoods=[]

DBsize=size(WhitenVecs,2);

for i= 1:DBsize

temp = ImgColVec - WhitenVecs(:,i);

tempnorm = (-1/2)\*((norm(temp))^2);

likelihoods(i) = exp(tempnorm)/normalizing\_denom;

end

[MLresult,index]=max(likelihoods)

function [Vectors,Values,Psi,L] = pc\_evectors(A,numvecs)

if nargin ~= 2

error('usage: pc\_evectors(A,numvecs)');

end;

nexamp = size(A,2);

fprintf(1,'Computing average vector and vector differences from avg...\n');

Psi = mean(A')';

for i = 1:nexamp

A(:,i) =A(:,i) - Psi;

end;

ele=size(A,2);

fprintf(1,'Calculating L=A''A\n');

L=A'\*A;

fprintf(1,'Calculating eigenvectors of L...\n');

[Vectors,Values] = eig(L);

fprintf(1,'Sorting evectors/values...\n');

[Vectors,Values] = sortem2(Vectors,Values);

fprintf(1,'Computing eigenvectors of the real covariance matrix..\n');

Vectors = A\*Vectors;

Values = diag(Values);

Values = Values / (nexamp-1);

num\_good = 0;

for i = 1:nexamp

Vectors(:,i) = Vectors(:,i)/norm(Vectors(:,i));

if Values(i) < 0.0000001

Values(i) = 0;

Vectors(:,i) = zeros(size(Vectors,1),1);

else

num\_good = num\_good + 1;

end;

end;

if (numvecs > num\_good)

fprintf(1,'Warning: numvecs is %d; only %d exist.\n',numvecs,num\_good);

numvecs = num\_good;

end;

Vectors = Vectors(:,1:numvecs);

function [vectors values] = sortem2(vectors, values)

if nargin ~= 2

error('Must specify vector matrix and diag value matrix')

end;

vals = max(values); %create a row vector containing only the eigenvalues

[svals inds] = sort(vals,'descend'); %sort the row vector and get the indicies

vectors = vectors(:,inds); %sort the vectors according to the indicies from sort

values = max(values(:,inds)); %sort the eigenvalues according to the indicies from sort

values = diag(values); %place the values into a diagonal matrix

**REFERENCES**

# 1. A New Face Recognition using PCA, LDA and Neural Classifier [www.waset.org/journals/waset/v41/v41-2.pdf](http://www.waset.org/journals/waset/v41/v41-2.pdf)

# 2. Atick.J, and Penev P. ``Local feature analysis: A general statistical theory for object Representation,'' *Network: Computation in Neural Systems*, vol. 7, pp. 477-500, 1996. 3 *Chai*. D.*,* S. *L. Phung and* A. BouzerdoumA Bayesian Skin/Non- Skin Color Classifier Using Nonparametric Density Estimation in the proceedings of IEEE.

# 4 Chellappa.R, K. Etemad and ` Discriminant analysis for recognition of human face images,'' *Journal of the Optical Society of America*, vol. 14, pp. 1724-1733,

# 5 Choudhury T, B. Clarkson, T. Jebara, and A. Pentland, ``Multimodal person recognition using unconstrained audio and video,'' in *Proceedings of the Second Conference on Audio- and Video-based Person Authentication*, Washington, D.C., March 1999.

6 Evaluation of Face recognition algorithm. The CSU Face Identification Evaluation System, Version 5.0. [http://www.cs.colostate.edu/evalfacerec/algorithms5 .html], 2005.

7 Face Recognition Home Page., www.face-rec.org .

8 FERET Database <http://www.itl.nist.gov/iad/humanid/feret/>.

9 Gonzalez, Woods, and Eddins., Digital Image Processing Using MATLAB 2nd Ed by © 2009. [Gatesmark Publishing](http://www.imageprocessingplace.com/root_files_V3/top_navigation_bar/contact_us.htm).

# 10 Jayaraman .S, 2009. ‘Digital Image Processing, McGraw-Hil’.

# 11 Moghaddam B and A. Pentland, “Probabilistic visual recognition for object recognition,'' *IEEE Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696-710.

# 12 Smith L. I. A tutorial on principal components analysis

# 13 M. Turk and A. Pentland. Eigen faces for recognition. *Journal of Cognitive Neuroscience*, 3 (1), 1991a URL <http://www.cs.ucsb.edu/> mturk/Papers/jcn.pdf .

14 Turk D.G,R. 0. Duda, and P. E. Hart, Pattern

Classification. New York John Wiley & sons, Inc., 2001

15 W. Zhao, R. Chellappa, P. J. Phillips and A. Rosenfeld. Face Recognition: A Literature Survey. ACM Computing Surveys, vol. 35(4):399-458, Dec 2003.

16 Yale Database Group <http://db.cs.yale.edu/>

# 