2013

Evaluation and Hardware Realization for a Face Recognition System

Kaushik Ray
University of Windsor

Evaluation and Hardware Realization for a Face Recognition System

By

Kaushik Ray

A Thesis
Submitted to the Faculty of Graduate Studies
through Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science
at the University of Windsor

Windsor, Ontario, Canada

2013

Evaluation and Hardware Realization for a Face Recognition System

by

Kaushik Ray

APPROVED BY:

_____

Dr. Shaohong Cheng, Outside Dept. Reader
Department of Civil and Environmental Engineering

_____

Dr. Mohammed A. S. Khalid, Internal Dept. Reader
Dept. of Electrical & Computer Engineering

_____

Dr J. Wu, Advisor
Dept. of Electrical & Computer Engineering

18th June 2013

Co-Authorship Declaration

I hereby declare that this thesis incorporates material that is result of joint research, as follows:

This thesis also incorporates the outcome of a joint research undertaken in collaboration with Ashirbani Saha under the supervision of Professor Jonathan Wu. The collaboration is covered in Chapter 3 of the thesis. The experimental designs, data analysis and the hardware design were performed by the author. The contributions of co-authors were primarily through the face recognition algorithm idea and implementation, provision of proof reading and reviewing the research paper regarding the technical content.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-authors to include the above materials in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas,

techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Facial recognition from an image or a video sequence draws attention for many image processing researchers owing to its myriad applications in real world as well as in computer vision, human-computer interaction and intelligent systems. Facial structures have unique features which can be extracted using some mathematical tools. We have used Principal Component Analysis (PCA) and Local Binary Pattern (LBP) to extract them and stored them in a database. When the query image is given the facial features are extracted and compared to the previously obtained results using Sparse Face recognition. Detailed test methods have been defined and an extensive testing of the algorithm has been performed on various standard databases. The results have been tabulated with required graphs. The proposed algorithm has been compared to other different algorithms which show significant improvement in results with small number of training samples. Finally the algorithm was integrated in a hardware system so that it can be used as a self sufficient portable system.

to my

Mom and Dad

Acknowledgements

I express my sincere gratitude to my adviser, Dr. Q.M. Jonathan Wu for giving me the opportunity to work under his supervision as well as for his constant guidance and support. I would like to thank Mr. Frank Cicchello for his valuable suggestions and comments. I would like to thank Ashirbani Saha and Dibyendu Mukherjee for all their constant guidance, help and motivation during my research work.

I am thankful to my friends and colleagues, in particular Charrie, Carolyn, Conrad, Carina, Candace, and Gaurav for their help and frequent assistance and motivation throughout my work and stay. Last but not the least, I would like to thank my mom, dad and my sister for their love and support which motivated me throughout my research work.

Table of Contents

List of Tables

Table 4.4: Time taken for face recognition in seconds per frame.

List of Figures

List of Acronyms

AAM	Active Appearance Method
ADC	Analog to Digital Converter
ASCII	American Standard Code for Information Exchange
ASIC	Application-Specific Integrated Circuit
ASIP	Application Specific Instruction set Processor
ASM	Active Shape Model
ASSP	Application Specific Signal Processor
BJT	Bipolar Junction Transistors
CCD	Charged Coupled Device
CMOS	Complementary Metal-Oxide Semiconductor
CMRR	Common Mode Relation Ratio
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DAQ	Data Acquisition
DCT	Discrete Cosine Transform
DSP	Digital Signal Processors
DWT	Discrete Wavelet Transform

EBGM	Elastic Bunch Graph Matching
ED	Euclidean Distance
EEPROM	Electrically Erasable and Programmable Read-Only Memory
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GB	Gigabyte
GND	Ground
GPIO	General Purpose Input/Output
GPP	General Purpose Processors
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HD	Hamming Distance
HEX	Hexadecimal
HMM	Hidden Markov Model
HSD	Haar Spectral Diagram
IC	Integrated Circuits
IDE	Integrated Drive Electronics
IO	Input/Output
IR	Infra-red
JAFFE	Japanese Female Facial Expression
JTAG	Joint Test Action Group
KB	Kilobyte
KN	Known Neighbor
LBP	Local Binary Pattern

LCD	Liquid Cristal Display
LDA	Linear Discriminant Analysis
LED	Light Emitting Diode
LSB	Least Significant Bit
MB	Megabyte
MHz	Megahertz
MIPS	Million Instructions Per Second
OMP	Orthogonal Matching Pursuit
OS	Operating System
PC	Personal Computer
PCA	Principal Component Analysis
PDIP	Plastic Dual In-line Package
PWM	Pulse Width Modulation
RAM	Random Access Memory
RBF	Radical Based Function
RC	Resistance and Capacitance
RGB	Red Green Blue
RISC	Reduced Instruction Set Computing
ROM	Read-Only Memory
RX	Receive
SCL	Serial Clock
SD	Secure Digital
SDA	Serial Data
SFRLBP	Sparse Face Recognition with Local Binary Pattern

SoC	System on Chip
SRAM	Static Random Access Memory
STFT	Sort-Time Fourier Transform
SVM	Support Vector Machine
TTL	Transistor-Transistor Logic
TX	Transmit
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VI	Voltage Input
VO	Voltage Output
WP	Wright Protect
ZISC	Zero Instruction Set Computer

Chapter 1: Introduction

The primary objective for this thesis is to explain our proposed algorithm, design, application and tests for a robust face recognition algorithm and to explain how this system can be implemented onto hardware in order to realize its portability, surveillance potential to solve problems involving automotive theft and driver authentication. We believe this design conception will produce striking results as well as a critical, distinguished, state-of-the-art product.

1.1 Why use face recognition software?

Every day, thousands of terabytes of images around the world are captured in the form of videos from surveillance cameras. Of those captured images, 95 percent consist of facial data, so finding a correct algorithm to extract these facial structures and index them for facial recognition is vital. Presently, there are a myriad of applications for a robust face detection and recognition model. These applications are not only crucial within the theoretical world, but they have a gargantuan significance in the real world. For example, accurate facial recognition can be used for identification of criminals, for security system imaging, for image and film processing to retrieve character faces in

feature-length films, for use by investigators, scientists and engineers in the fields of human computer interaction and homeland security and for key barriers to automotive theft.

1.2 Challenges

Over the years, researchers have faced serious challenges with facial recognition systems. Although in the past few years we have seen significant improvement, most of the face recognition algorithms work well only with frontal faces. Occlusion of the face due to sunglasses, long hair, or some object that partially covers the face has made the production of a reliable face recognition system challenging. In addition, change in facial expression, low light, and low resolution pictures have proved detrimental for a trustworthy, consistent face recognition product. Difficulties with robustness verses processing speed have made the manufacture of this crucial recognition system even more arduous.

1.3 Motivation

The challenges mentioned in section 1.2 make face recognition one of the major research topics in computer vision. Face recognition being one of the diverse research topic with a lot of developments and new algorithms proposed every day but in retrospect that fact there are major gaps to be filled up for a perfect design. Everyday researchers are trying to fill these gaps and find a better solution. A trade-off between the speed and accuracy looms large as the primary research focus. Algorithms with higher accuracy suffer from greater time complexity whereas methods with lower

complexity have lower accuracy. One of the other major motivational factors is a Waterloo based company brought into focus for a requirement of a portable device capable of face recognition. The primary objective for the device is to be installed in a car for authentication of the eligible drivers.

1.4 Objective

The objective for the research detailed in this thesis is to improve the accuracy and speed for a face recognition system using *sparse face recognition* and *Local Binary Pattern* or LBP. Two different test procedures are presented to test the proposed algorithm on various standard datasets. Each of these datasets carries its own, unique features including variations in expression, lighting conditions, partial occultation, and, cluttered backgrounds. The final objective of this thesis is to demonstrate an available design and execution of a face recognition system in a portable hardware which can be used as a standalone device.

1.5 Scope

This research comprises a new approach towards face recognition with LBP and sparse face recognition along with rigorous testing and detailed observations on various standard datasets in order to reduce the whole system in small scale hardware and design a standalone system. Most embedded hardware systems use Hidden Markov Model (HMM) for embedded face recognition. This paper will present one of the first approaches with LBP and Sparse face recognition.

1.6 Organization

This thesis has been organized as follows:

- Chapter 2 details a review of the related work involved as well as a general workflow for a face recognition system.
- Chapter 3 specifies an overview of the present research along with required flowcharts and diagrams. A detailed tabulation of the results obtained is presented along with relevant graphs.
- Chapter 4 concentrates upon system miniaturization with a small sized board to make it a portable and standalone system along with other hardware support to assist the proposed algorithm.
- Chapter 5 discusses the future work that can be performed to make this particular face recognition device robust, efficient and cost effective.

Chapter 2: Literature Review

Faces are complex multi-dimensional structures and over the past three decades, image processing researchers have been working hard to execute a suitable procedure to extract, analyze and recognize facial features. This system is needed to find the identity of a test subject that will be significantly more reliable than the present systems.

Most of the early models were based on Construct Detailed models of retinal or striate activity. It was much later, with the increase in computational power, that face recognition turned to computational modeling. Most of these computational models of face recognition contain some basic blocks. Many different researchers have tried several different procedures to find a perfect algorithm to devise a facial recognition system that will be almost perfect.

2.1 General Workflow

Most facial recognition algorithms follow a generic frame work which is presented in Figure 2.1. The basic steps for facial recognition and some of the existing works in the literature have been discussed in the following chapters.

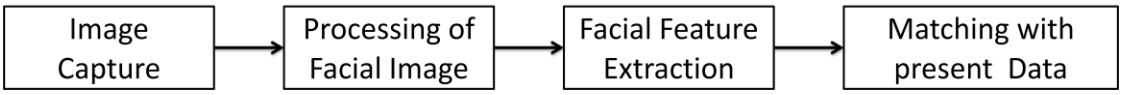


Figure 2.1: General workflow of a face recognition system.

2.1.1 Facial image acquisition

A face acquisition system identifies, locates and extracts features from a complex scene with cluttered backgrounds. For a face recognition system a good face detector is very important. There are many approaches for a face detection system - where some use an [91][92] exact location of faces, others operate [93][94] with a coarse location of faces. Modular eigenspace method was used by Essa and Pentland [91] for face detection. Principal component analysis or PCA coefficients of facial images were used to extract facial information from still images and image sequences. Hong et. al [92] used the Pearson Spotter system [95] for tracking the head so stereo disparity, skin color detection and convex region detector to figure out the exact face location. Methods like Active Appearance or AAM and local motion model [93] can work with a course location of faces. The most popular and effective face detection system used in recent research is Viola-Jones' face detector, which uses boosted haar features to detect faces.

2.1.2 Preprocessing of the facial image

Images captured in different resolution are computed in a uniform scale; RGB images can be converted to gray image; grey image can be converted to binary image. Noise reduction filter can be used or face alignment algorithm can be used for aligning the face properly.

2.1.3 Facial feature extraction

Feature extraction is one of the major tasks in an image processing algorithm. Feature extraction categorizes the image into correct abstract classes relevant to the context. Every image is a two dimensional, three dimensional or fore dimensional data set depending on the type of image or a video sequence. The basic idea is to use some mathematical tool on these data sets or matrices and extract meaningful information. In Figure 2.2 we can see some methods in which face detection algorithms are broadly classified. These methods use local information like portion of the eye or mouth [98] [99] for extracting the facial features. Holestic methods [92] [94] [97] use the entire face or points from different positions from the face. Methods which use both global and local features are also in existence and are commonly known as hybrid methods [91] [100].

Most commonly the image features are extracted in spatial domain or transform domain. In spatial domain it mostly calculates the density of the pixel or the distance between the lips and nose or the distance between the lips and the line joining the two eyes, mean standard deviation etc. In transformation domain methods like Fast Fourier Transform (FFT) [73], Discrete Cosine Transform (DCT) [74], Sort-Time Fourier Transform (STFT) [75], Discrete Wavelet Transform (DWT) [76] etc. are some of the common methods.

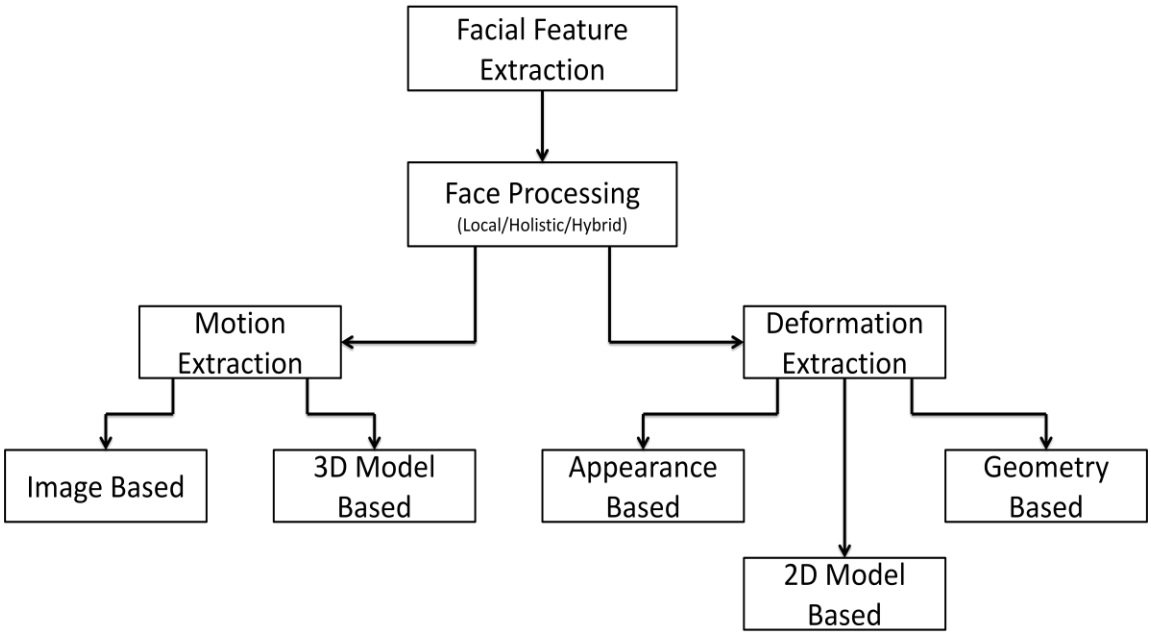


Figure 2.2: Different type of facial feature extraction approaches.

2.1.4 Matching the query image with the database image

While matching the query image, the features extracted from the training facial images are stored in a database. Next the query image is compared with the images from the database which have been processed and stored. Most common methods include Euclidean Distance (ED), Hamming Distance (HD), Support Vector Machine (SVM) [78], Known Neighbor (KN) [77] etc.

One of the most efficient computational models was proposed by Matthew Turk and Alex Pentland [43] in 1991; these are known as Eigenfaces were one of the best computational models at the time. Eigenfaces were used for a long time for face detection and recognition, all the while yielding good results. With further research, many drawbacks of the Eigenfaces were able to be overcome. Jeffery and Masatoshi [70] proposed Haar Spectral Diagram (HSD) in 1997. This new data structure was useful

to represent the Haar spectrum of Boolean function. In 2003 Kan Ma and Xiaoou Tang [71] proposed something similar to the Garber face graph known as Discrete Wavelet Face Graph. Duan and Zheng [72] proposed a method which uses grey level AdaBoost scheme for extracting Haar like features. In the next following years many new algorithms were developed which has an upper hand on top of the previous algorithm. Paul and Abbes [80] proposed a method to determine most discriminative coefficients in DWT/PCA based face recognition algorithm. Jun Ying Gan and Jun Feng Liu [81] proposed an innovative and novel approach based on wavelet features and their algorithm is known as Kernel Fisher Discriminant Analysis [82].

In more recent years, there have been more sophisticate algorithms proposed which are very robust and efficient. Some of the major approaches include PCA [43], Linear Discriminant Analysis (LDA) [83], Elastic Bunch Graph Matching (EBGM) [84], LBP [9], Sparse Representation [67] [69]. With the advent of all these new avenues, facial recognition techniques have been going uphill and reaching towards the goal to give an accuracy of close to 100% recognition rate. In the next following chapters we will explore a very new and unique way of designing Face recognition systems with sparse face recognition.

2.2 Embedded face recognition system

A Face recognition system is challenging by itself, so embedded face recognition systems must be even more challenging. Due to lower processing power and limitation in the availability of memory, the algorithms have to be modified accordingly. Some of

the most popular embedded face recognition systems available are neural network based face recognition systems [87]. Eigenfaces have been used for a long time; they are known to give good results using FFT based calculations for distance measurement as well as face detection [89], making hardware implementation easier and faster. In 2003 Fan Yang and Michel Paindavoine designed a Radical Based Function (RBF) Neural Network for real time tracing and face recognition on embedded systems. They implemented the same algorithm on different embedded hardware; the result and speed varied based on the hardware used: 92 percent accuracy on Fast Programmable Gate Array (FPGA); 85 percent accuracy on a ZISC processor [90]; and 98.2 percent accuracy on a Digital Signal Processor (DSP). With the increase in accuracy there is a decrease in the processing speed for the system. There are many other efforts for embedded face recognition systems which include designing a face recognition system by Liu *et al.* [84] using optical correlation technique.

In this thesis we have implemented the proposed algorithm on a portable hardware system. This is the first approach for designing a portable face recognition system with LBP, PCA and sparse face recognition.

Chapter 3: Software and Algorithm

3.1 Introduction

Development of an automatic face recognition algorithm is one of the major challenges for the image processing and pattern recognition researchers. Many algorithms have been proposed and all has some pros and cons. Some algorithms have a high efficiency rate but with a higher time complexity; others have a lower time complexity but with a lower efficiency rate (i.e. Eigenfaces). The prime goal for our thesis is to find an algorithm which can excel in two folds. In some recent research, it has been observed that LBP can extract useful correlated information out of a facial image [44]. We have used the same to design an algorithm which is robust as well as fast.

3.2 A short description on face recognition

Any face recognition system has a few steps in common like acquiring a set of training images for training the system. Find the face inside the image give the image a proper illumination and align the facial structure using some kind of algorithm and extract the region of interest for training and testing [43]. After these preliminary processing the training and the test algorithms are introduced.

In the following few section, we discuss an algorithm which has following features:

- Illumination invariant.
- Very robust even if trained with small number of training samples.
- And very fast recognition time on an average 0.3 sec*
- Gives a very good result even without facial alignment reducing the recognition time.

The proposed algorithm uses LBP [9] and PCA for feature extraction. Sparse Representation has been used to match the data from the test set to the test sample. In the Sparse Representation instead of l^1 minimization [69], Orthogonal Matching Pursuit (OMP) has been used to synthesize the matched image from the training dataset and the test sample. Let's say the sample image is of size $N_{PC} \times 1$ and the train data set is of size $N_{PC} \times N_{Train}$ now the synthesized image is derived from these two information which is of the order of $N_{Train} \times 1$. Orthogonal Matching Pursuit [68] is used instead of conventional l^1 minimization as it is faster and yields better performance [42]. Figure 3.1 shows the basic blocks and the process flow of the algorithm.

*All simulations in this chapter are tested on an AMD X2 1075 processor based computer with 16GB of RAM.

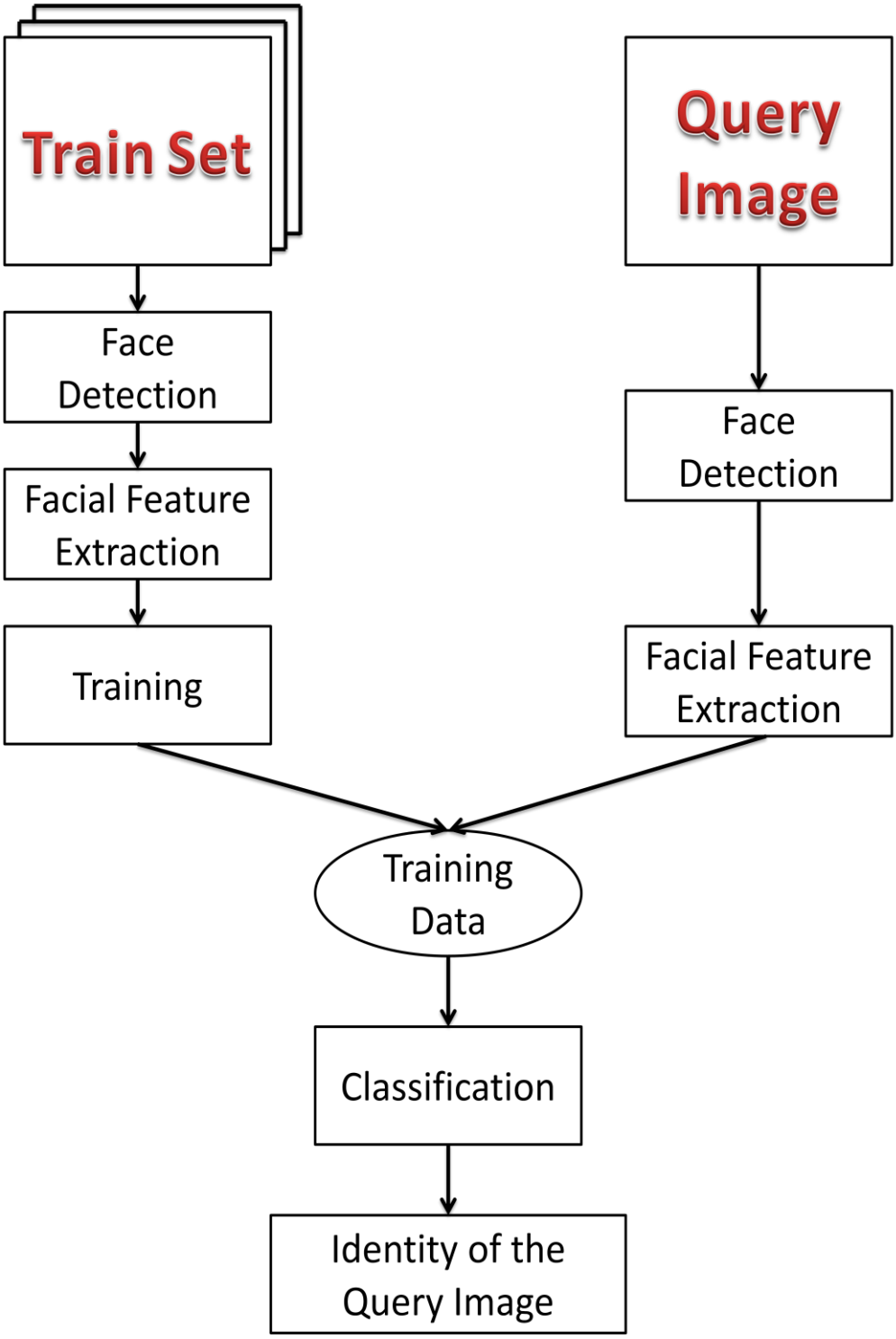


Figure 3.1: Shows the basic blocks and the process flow of the proposed facial recognition algorithm.

3.3 Face Detection

Viola Jones Face detector is one of the most impacting and robust face detector available till date. This real time face detector stands on three pillars namely:

- The integral image
- Classifier learning with AdaBoost
- Attentional cascade structure

3.3.1 The integral image:

Integral image is one of the major pillars for Viola Jones. This algorithm can be defined as a two dimensional lookup table in the form of a matrix with same dimensions as of the original image. Haar like features are very easy to compute as the algorithm follows the integral image technique and it provides a good performance for constructing a frontal face detector. This algorithm is used to quickly and efficiently calculate the sum of values in a rectangular subset in a grid. The integral image can be calculated by:

$$ii(x,y) = \sum_{x'\leq x, y'\leq y} i(x',y')$$

where $ii(x,y)$ is the integral image at pixel location (x,y) and $i(x',y')$ is the original image. Depending on the definition Haar like features may need more than four lookup tables. Viola and Jones's 2-rectangle features need six lookups, 3-rectangle features need eight lookups, and 4-rectangle features need nine lookups.

3.3.2 Classifier learning with AdaBoost

Adoptive Boosting or AdaBoost is the first step towards the practical boosting algorithm [2] [3]. In this algorithm we have used a modified version of AdaBoost which is commonly known as the *RealBoost* [1] [4] [5] [8]. RealBoost yields substantially better result than the AdaBoost.

Let us consider a training set say $S = \{(x_i, z_i), i = 1,2, \dots, N\}$ where x_i belongs to the sample space X and z_i belong to the finite label space Z. For a binary classification problem, $Z = \{1, -1\}$, where $z_i = 1$ for positive set and $z_i = -1$ for the negative set. As the name suggests AdaBoost is adaptive in nature and produces an adaptive model which can be represented by $F^T(x) = \sum_{t=1}^{T} f_t(x)$ for predicting the label of a test sample x. In the above equation $F^T(x)$ is a real function which is represented by $F^T: X \to R$ and the predicted label is $\hat{z}_i = sign\left(F^T(x_i)\right)$, where sign(.) is the sign function. From the statistical view point AdaBoost falls into a additive logistic regression model and by using adaptive Newton updates of minimization the whole system can be represented as

$$L^T = \sum_{i=1}^{N} \exp\{-z_i F^T(x_i)\}$$

Now AdaBoost can find the best additive base function $f_{t+1}(x)$ once $F^t(x)$ is known. Now let's consider for that $\{f(x)\}$ which is a base function and is in the form of confidence rated decision stump. This can be classified as certain form of real feature value $h(x)$ and these values are derived from $x, h : X \to R$. For example Viola Jones face

detector are Haar-like feature computed with integral images. A threshold H is obtained which will now divide the output obtained from $h(x)$ to two different parts say u_1, u_2 where $u_1 \cup u_2 = R$. Now the base function $f(x)$ can be rewritten as:

$$f(x) = c_j, \text{ if } h(x) \in u_j j = 1,2, \ldots$$

and this is often known as the stump classifier. c_j gives the confidence of the derived result and the optimal confidence value can be calculated from the following equations when $j = 1,2$ and $k = 1, -1$ Let

$$W_{kj} = \sum_{i:z_i=k, f(x_i)\in u_j} \exp\{-kF^T(x_i)\}$$

Now the expression for target criteria can be written as:

$$L^{t+1} = \sum_{j=1}^{2} [W_{+1j}e^{-cj} + W_{-1j}e^{cj}]$$

Using basic calculus we can minimize L^{t+1} when

$$c_j = \frac{1}{2}\ln\left(\frac{W_{+1j}}{W_{-1j}}\right)$$

Now if we combine both the equations we get

$$L^{t+1} = 2\sum_{j=1}^{2}\sqrt{W_{+1j}W_{-1j}}$$

This equation gives us the Z score [6]. For each and every iteration $t + 1$, the Haar-like feature $h(x)$ we can derive an optimum threshold H with a confidence score of c_1 and c_2 for minimizing the Z score L^{t+1}.

3.3.3 Attentional Cascade Structure

This is one of the major components in the Viola Jones face detector. The most important objective of this algorithm is to build a smaller and more efficient, boosted classifier which can reject most of the negative sub-windows while keeping almost all the positive sets. This will eliminate most of the sub-window in the early stages of the detection, making the process extremely efficient. Now the overall process of classifying a sub-window forms a decision tree which is also called a cascade [7]. Each of these sub-windows pass through a series of binary decision tree where it is decided whether the window will be kept for the next round or rejected [7]. As the sub-windows passes to the next level of binary decision tree the number of weak classifiers increases so that more details can be extracted. Having lower number of weak classifier in the early stages of the decision process increases the speed of the algorithm.

3.4 Facial Alignment

Facial image alignment has been one of the fundamental problems in computer vision since 1990. ASM [9] and AAM [2], [10] are the most popular model-based image alignment methods as they have very low computational overhead. For AAM, the basic idea is to use two eigenspaces for modeling the object and shape-free appearance. ASM and AAM have been used extensively in many computer vision algorithms such as facial

image processing [16] [17] [18], medical image analysis [19], industrial inspections [8], image coding [3], object appearance modeling [20] and many others. For many practical applications like pose estimation, expression analysis, and facial recognition, much research has been conducted on facial alignment. Although there are many available algorithms for facial alignment, the majority of them are based on ASM, AAM, or their variations [22] [23] [24] [27] [28] [29] [30] [31] [32] [33]. When AAM is used for a large dataset, the alignment has difficulty with generalization [65] [66], so a very simple, yet robust facial alignment algorithm has been proposed based on Viola Jones object detection algorithm [45].

As many facial recognition algorithms depend on careful positioning of an object into a canonical pose, so the position of features relative to a fixed coordinate system can be examined. Generally, this positioning is done either manually or by training a class-specialized learning algorithm as described before with samples of the class that have been hand-labeled with parts or poses. Proper positioning of the facial structures generally yields in better performance [64].

For the present work we have tried to implement a very fast and simple facial alignment algorithm and tested the efficiency of the algorithm on Caltech 101 facial database both Faces Easy and Hard and it gave an accuracy of 98.7%. Algorithm 1 below summarizes the complete alignment procedure.

Algorithm 1: Facial alignment

1. Find the face in the picture
2. After finding the face divide the obtained facial image in 3 parts out of which only two parts will be used.
3. Each of these two parts will contain the right and the left eye separately. Now the right and the left eyes are detected separately for each of those sub images using Viola Jones Eye detection algorithm. Refer to Figure 3.2 for details.
4. After the eyes has been detected separately we take the mid points for each eyes and align the picture using rotational matrix which is given by:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

5. After the rotation of the faces we can see some holes (refer Figure 3.3) in the pictures as there are overlapped points so Bilinear interpolation has been used to fill up the image holes which is given by:

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

where $R_1 = (x, y_1)$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

where $R_2 = (x, y_2)$

$$f(P) \approx \frac{y_2 - y}{y_2 - x_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2)$$

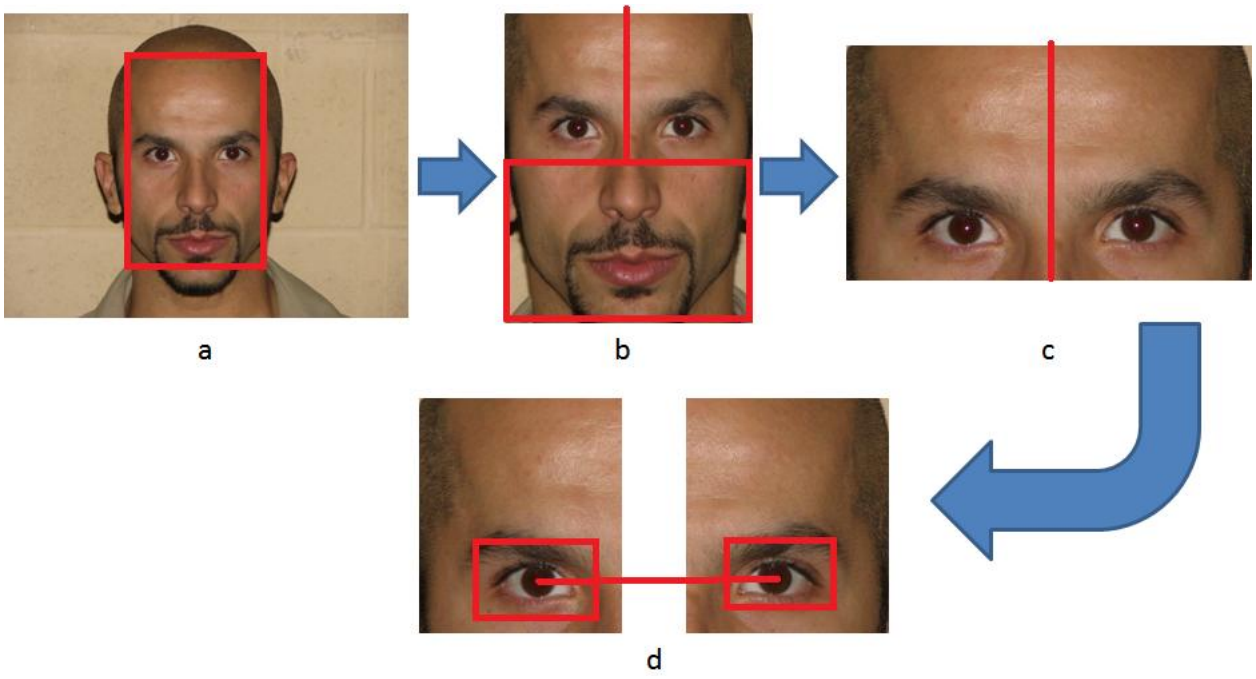


Figure 3.2: Alignment process flow.

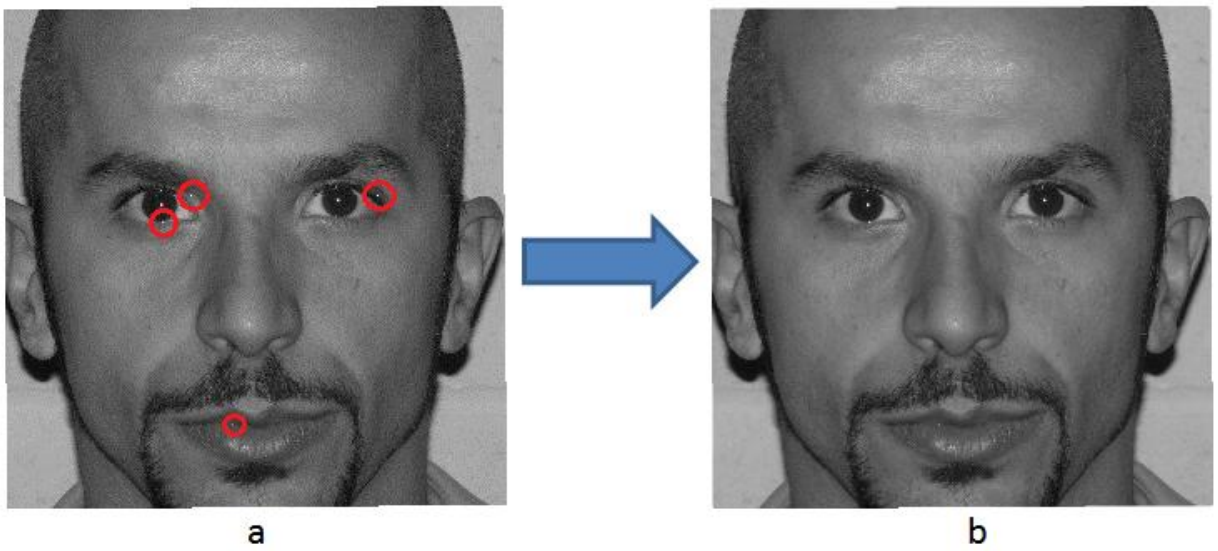


Figure 3.3: (a) Shows the holes present due to the rotation of the image marked by red circles. **(b)** Shows the image processed with a bilinear filter.

This algorithm is very simple and it uses Viola Jones object detection [ref] for both Face Detection and eye detection. For eye detection the Viola Jones algorithm was trained with left eye and right eye separately so that it can have a better result.

One of the major observations for this alignment algorithm is, if the alignment is not proper then the result of the facial recognition rate falls very rapidly which in turn can hinder the accuracy of the whole system rather than boosting it.

After aligning the faces with this above mentioned algorithm it has been observed that the facial recognition rate was falling by certain percentage in accuracy rate which has been tabulated in Table 3.1 so we finally decided not to include this algorithm for the system.

Table 3.1: Change in percentage accuracy with and without the alignment.

Name of Database	Accuracy with Face alignment	Accuracy without Face Alignment
Face94	98.07%	100%
Face95	97.0351%	99.0909%
Face96	92.0745%	95.0203%

3.5 Feature extraction using Local Binary Pattern (LBP)

One of the major challenges for face recognition algorithms are to find an efficient as well as discriminative facial appearance descriptor which can work well in real life conditions like variation in illumination, pose, change in facial expression, ageing, partial occlusions out of plane faces etc [14]. There are two ways to design facial appearance descriptors that can work well in the above mentioned real live conditions, namely

- geometric feature-based descriptors
- appearance-based descriptors

Geometric descriptors are not very reliable under variations in facial appearance, while appearance-based models blur out the micro-patterns from images [10] that is, the small details which increases the residual spatial registration errors. Eigenfaces is one of the major examples for appearance-based models. From the present research representations based on local pooling of local appearance descriptors has been gaining popularity. They can capture small appearance details in the descriptor while remaining robust to real life conditions and to registration errors owing to local pooling. This model is also persistent in extraction of smaller details in the highly correlated regions of the image and pooling local structural information or micro-patterns from images [10]. Some of the most popular methods include Gabor wavelets [12], local autocorrelation filters [11], and Local Binary Patterns [9].

In the following pages we focus on LBP and its generalizations. LBP is a computationally efficient nonparametric local image texture descriptor. LBP has successfully been used on numerous occasions including face recognition [9] [10] [13]. LBP features are invariant to monotonic gray-level changes so they don't require prior image preprocessing before use. In fact, LBP itself is sometimes used as a lighting normalization stage for other methods [16].

As we know that both Fisher faces and Eigen faces method take somewhat holistic approach to face recognition or feature extraction. We treat every image data as a

vector in a high-dimensional image space. As we all know high-dimensions make the algorithm complex, so a lower-dimensional subspace is identified, where more useful information is preserved. We've already seen that the Eigen faces approach is likely to find the wrong components on images with a lot of variation in illumination and are computationally more complex. They also fail to extract any micro-patterns, so small details and information loss is very common.

The LBP operator on the other hand was originally designed for texture description and it is one of the best of its kind [15]. It has a lot of key features which makes it very discriminative. LBP is invariant to monolithic grey level changes, it is illumination invariant and it is highly efficient as per the computational efficiency, which makes it ideal for image processing applications [67]. LBP can be used to describe facial structures very well as it is a composition of micro-patterns which is well described by this kind of operators. This operator operates on every pixel of the image to assign them a label by thresholding them on a 3 x 3 neighborhood of each pixel with the centre pixel value and considering the result as a binary number. Please refer to Figure 3.4 for the basic operation of LBP. If the intensity of the center pixel is greater-equal than its neighbor, then it is denoted with 1 and 0 if not. This will give a binary number for each pixel, just like 11001011. With 8 surrounding pixels we will get 2^8 possible combinations, which are called Local Binary Patterns.

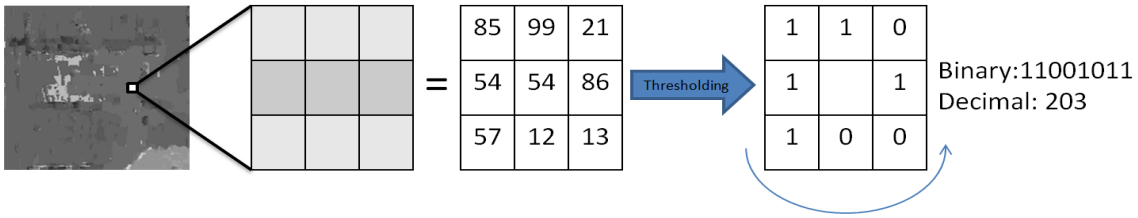


Figure 3.4: The Basic LBP operator.

LBP code for each pixel (x_c, y_c) is given by:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \qquad s(x) = \begin{cases} 1, if\ x \geq 0; \\ 0, \ otherwise \end{cases}$$

To deal with more complex structure and for even finer features, LBP is extended to use neighborhood of different sizes [16]. Defining the local neighborhood as a set of sampling points, evenly spaced on a circle, centered at the pixel to be labeled, allows any radius and number of sampling points. Bilinear interpolation is used when a sampling point does not fall in the center of a pixel. In the following, the notation (P;R) will be used for pixel neighborhoods which means P sampling points on a circle of radius of R. Another extension to the original operator is the definition of so-called uniform patterns [16]. A local binary pattern is called uniform if the binary pattern contains at most two bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular. For example, the patterns 00000000 (0 transitions), 01110000 (2 transitions) and 11001111 (2 transitions) are uniform whereas the patterns 11001001 (4 transitions) and 01010011 (6 transitions) are not. In the computation of the LBP histogram, uniform patterns are used so that the histogram has a separate bin for every uniform pattern and all nonuniform patterns are assigned to a single bin. In our proposed algorithm we

have used this Uniform LBP with radius equal two and with eight neighbors which is denoted by $LBP_{8,2}^{U2}$

We have divided the image into 10 x 10 pixel sub images for calculating the Uniform LBP with radius equal to two and with eight neighbors. Now with uniform LBP we can only get 59 different values from any given number between 0 and 255. Each of these vectors consists of the frequency of occurrence for the calculated histogram.

Example 1: if we have a image which is made of 100 x 100 pixels. Now we sub divide this image to 10 x 10 sub images and calculate $LBP_{8,2}^{U2}$. As we get the histogram for these sub images to get a vector of (10 x 10 sub images make 100 sub images) 1 x 5900 and this vector contains all the useful information in the image and also reduces the images dimension with some degree. Larger the image more will be the reduction of dimension as the value 59 is fixed [67].

3.6 Principal Component Analysis (PCA)

Principle component analysis (PCA) is a mathematical procedure that uses orthogonal transformations to convert a set of highly correlated variables to a set of linearly uncorrelated variables called principal components. In most cases, principal components are less than or equal to the components in the data set. The principal components are those that have the largest Eigen values and which account for the highest variants within the data set. The first principal component always has the largest possible variance. The second Principal Component is calculated under the constraint of being orthogonal to the first component and to have the largest inertia. All components

thereafter are calculated in the same way. These new variables are known as factor scores and these factor scores can be geometrically interpreted as projections of observations onto the principal components. The four main goals of PCA are as follows: The first goal is to extract the most significant data from a data table; the second goal is to compress the data keeping only the important information; the third goal is to simplify the data set; lastly, the fourth goal is to analyze the structure of the observations and variables.

3.7 Processing the Images with LBP and PCA

PCA and LBP [9] has been put together to process the data set before the matching algorithm has been introduced. We call this algorithm the ***Sparse face recognition with LBP***. Step by step working principal has been described in the following few lines here after.

Suppose we have a 100 x 100 pixel image TR_1 which we will use as a training sample and another image TS_1 of same dimensions which we will use as test sample. Now the first step we perform is $LBP_{8,2}^{U2}$ and these give us a vector of 1 x 5900 and let's call this TR_{LBP1}. From TR_{LBP1}we derive three useful information which will be crucial for designing the training data set:

- We find the mean Vector from $TR_{LBP1}$and lets name it TR_{MV}
- Principle components from theTR_{LBP1} which we will call TR_{PC}.
- Projected values of image features on a PCA space which is represented as:

$$A_i = \left[v_{i,1,}v_{i,2}, \dots, v_{in_{i,}}\right] \in IR^{m \times n_i}$$

With all these information in hand our training is complete. All the training sample data is kept in a training data set $N_{PC} \times N_{Train}$ where the vector A_i is stored in each row of $N_{PC} \times N_{Train}$. Now let's look, how the test sample is processed and matched with the train dataset [67].

As in training set, the test sample is first processed with $LBP_{8,2}^{U2}$ and we get a vector of the order of 1 x 5900 and let's call this TS_{LBP1}. Now we subtract TR_{MV} from TS_{LBP1}. So let's say we have $TS'_{LBP1} = TS_{LBP1} - TR_{MV}$. Now $TS'_{LBP1}$is projected on the PCA space using certain number of Principal Component from TR_{PC} which we have obtained while designing the training set. We do not use the entire results from TR_{PC} but we only take those components that have the largest Eigen values and which account for the most variance within the set [43]. Taking the entire PC set will make the testing algorithm slower.

Now we have both the training data and the test data in the PCA space which will help us to match the test sample with the training data set and we will use Sparse Face Recognition for the same.

3.8 Sparse Face Recognition

One of the basic challenges faced by researchers for object matching and recognition is to use labeled training samples from a set of k distinct objects and to match a query sample correctly with the train sample data. A given n_i training sample is arranged from the i^{th} class as column of a matrix $A_i = \left[v_{i,1}, v_{i,2}, \dots, v_{in_i,}\right] \in IR^{m \times n_i}$. For the purpose of face recognition we can consider a $w \times h$ grey-scale image with the vector $v \epsilon IR^m$ $(m =$

wh) given by stacking the column and now each column of A_i is the training face image of the i^{th} subject [67] [69].

There are many algorithms and statistical models to deal with A_i for face or object recognition. One of the very simple yet effective procedures is to sample from a single class as lying on a linear subspace. These subspace models are very effective as they can deal with most of the real data set, especially facial images. It has been observed that these facial images lie on a special lower-dimensional subspace [35] [36] often referred to as a face subspace. Now if we have enough training samples of the i^{th} object class, then $A_i = \left[v_{i,1,} v_{i,2}, \dots, v_{in_i,}\right] \in IR^{m \times n_i}$, so any given test sample $y \, \epsilon \, IR^m$ from the same class will lie in the span of training samples associated with i.

For a given new test sample y which belongs to the training set, we compute its sparse representation $\hat{x}_1$ either by l^1 minimization or by Orthogonal Matching Pursuit (OMP). For the proposed algorithm, we have used OMP over conventional l^1 minimization as OMP has a lower time complexity yielding a better performance [37] [38] [39].

3.8.1 Classification Based on Sparse Representation

Let's say that $\boldsymbol{x}$ is an arbitrary m-sparse signal, where $x \, \epsilon \, R^d$ and let $\{m_1, \dots \dots \dots, m_n\}$ be a family of N measurement vector. From an $N \times d$ matrix Φ whose rows are the measurement vector, and observe that the N measurements of the signal can be collected in an N-dimensional data vector $v = \Phi x$. We refer Φ as the *measurement matrix* and denote its columns by $\varphi_{1,\dots,}\varphi_d$ [68].

The above mentioned equation $v = \Phi x$ is exactly what we need to solve for $\hat{x}_1$ where v is the query image and Φ or the *measurement matrix* is the training dataset. In the training data set all the information are in the form of $A_i = [v_{i,1}, v_{i,2}, \dots, v_{in_i,}] \in IR^{m \times n_i}$. $\varphi_1 = v_{i,1}$, $\varphi_2 = v_{i,2}$ so forth where $\varphi_{1,\dots,}\varphi_d$ are the columns of the *measurement matrix*.

For each class i, let $\delta_i: IR^n \to IR^n$ be the characteristic function that selects the coefficients associated with the ith class. For $x \in IR^n, \delta_i(x) \in IR^n$ is a new vector whose only nonzero entries are the entries in x that are associated with class i. Using only the coefficients associated with the ith class, it can be approximated that the given test sample y as $\hat{y}_i = A\delta_i(\hat{x}_1)$. We then classify y based on these approximations by assigning it to the object class that minimizes the residual between y and $\hat{y}_i$:

$$\min_i r_i(y) \doteq \|y - A\delta_i(\hat{x}_1)\|_2$$

Algorithm 2 summarizes the complete recognition procedure on the next page.

Algorithm2: Sparse face recognition with LBP

1. **Input:** A matrix of training samples $A_i = [v_{i,1}, v_{i,2}, \dots, v_{in_i,}] \in IR^{m \times n_i}$ for k class and a test sample $y \in IR^m$
2. Each and every column of A is normalized to have unit l^2-norm.

   Let's say vector $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ so l^2-norm can be defined for a vector x as

$$|x| = \sqrt{\sum_{k=1}^{n} |x_k|^2}$$

3. Now we solve for $\hat{x}_1$ from the equation $v = \Phi x$ using Orthogonal Matching Pursuit (OMP). Algorithm defined in ***Algorithm 3.***
4. Compute the residual $r_i(y) = \|y - A\delta_i(\hat{x}_1)\|_2$ for $i = 1,2, \dots, k$.
5. **Output:** identity$(y) = \arg\min_i r_i(y)$

OMP tries to identify and regenerate $\hat{x}_1$ we need to find out which column of Φ participate in the measurement of vector v. The main objective of the algorithm is to pick columns in a greedy fashion. For each iteration a column of Φ is chosen which is strongly correlated with the remaining part of the vector v. After each iteration we subtract of its contribution to v and iterate on the residual hoping to find the correct set of columns after m iterations.

Algorithm 3: OMP for signal recovery

1. **Input:** An $N \times d$ dimensional matrix Φ (the training data set)

   An N-dimensional data vector v (the query image)

   The sparsity level m i.e. number of iterations to get back the ideal signal.

2. We have to initialize the residual $r_0 = x$ and the index count as $\Lambda_0 = \emptyset$. The iteration count t is set to $t = 1$.

3. Find the index λ_t which can give a solution for easy optimization problem $\lambda_t = \arg\max_{j=1,\ldots,d} \left|\langle r_{t-1}, \phi_j \rangle\right|$. If the maximum occurs for multiple indicates, break the tie deterministically.

4. Argument the index set and the matrix of the chosen atom: $\lambda_t = \lambda_{t-1} \cup \{\lambda_t\}$ and $\Phi_t = \left[\Phi_{t-1}\ \varphi_{\lambda_t}\right]$. We use the conversion that Φ_0 is an empty matrix.

5. Solve a least square problem to obtain a new single estimate:

   $x_t = \arg min_x \|v - \Phi_t x\|_2$

6. Calculate the new approximation of the data and the new residual

$$a_t = \Phi_t x_t$$

$$r_t = v - a_t$$

7. Increment t, and return to Step 3 if $t < m$.

8. The estimate $\hat{x}_1$ for the ideal signal has nonzero indices at the components listed in Λ_m. The value of the estimate $\hat{x}_1$ in component λ_j equals to the j th component of x_t.

9. **Output:** An estimate $\hat{x}_1$ in R^d for a ideal signal.

   A set $\Lambda_m$containing m elements from $\{1, \ldots, d\}$

An N-Dimensional approximation a_m of the data $\hat{x}_1$

An N-Dimensional residual $r_m = v - m$

For OMP it is important to reorganize the residual $r_t$which should always be orthogonal to the column of Φ_t. Provided that the residual r_{t-1} is nonzero the algorithm selects a new atom at iteration t and the matrix Φ_t has full column rank. In this case the solution for x_t to the list square problem is unique in step 5. The approximation and residual calculation in step 6 is always uniquely determined. For OMP step 3 has the most process overhead. To reduce the overhead we have used QR factorization for calculating Φ_t [40] [41].

3.9 Designing the training set

In the next few titles we have verified the clams of the proposed algorithm and its robustness. Standard facial database like Face 94, Face 95, Face 96, FEI Face database, JAFFE, PUT Vein Pattern Database, Yale Face database has been used for the purpose. Each database has some special features and each one of them has been used in different ways to get different results and many different observations were made. We have designed our training set and the test set in two different ways [34].

3.9.1 Procedure 1

In Procedure 1, we have divided the number of samples of each subject in the database into two different sets; the Train set and the Test Set. The Train set will be used to train the computer to recognize that specific subject's facial features. The Test Set will be

used to test the computer's facial recognition properties with the same subject. We will use so many of the images in each test for training and the rest will be used for testing.

The test results have been tabulated into two tables for this procedure. The first table indicates the samples of picture used for training and testing. The database has been divided into several subsets (i.e. Subset 1, Subset 2) with different numbers of test samples and training samples. The highest percentage of accuracy achieved is also indicated for each subset alongside. The higher the number of training samples, the higher the accuracy will be; however, the accuracy of the algorithm also depends on many other factors such as facial postures, in-plane and out-of-plane variations of the frontal face, facial alignment etc. We observed a major change in percent accuracy during one of the tests when the training set was trained with an in-plane, frontal face versus when it was trained with an out-of-plane, frontal face.

3.9.2 Procedure 2

In Procedure 2 we have divided the images into two subsets as in Procedure 1, the Train Set and the Test Set; however, in this case we have a fixed number and set of images we are using for the Test Set, though for each subset, we are using a different amount of images with ten being the most and one being the least for the Train Set.

In both procedures the second table shows the change in the increase of accuracy with the increasing change of principle component while running the recognition algorithm.

The accuracy and the time taken have been tabulated for each database in the following pages. Graphs have been included to visually explain how the percent accuracies of the

results increased and decreased with increasing numbers of images in the Train Set as well as with an increasing number of principal components. A short description of the database has been provided along with the detailed tabulation of the Test Set and the Train Set in the next following titles.

3.10 Results

In this section we have tabulated all the results that have been obtained by running the Sparse face recognition with LBP for different standard datasets.

3.10.1 Face 94

3.10.1.1 About the Database

Face 94 dataset consists of 153 subjects with 20 images of each. Of these subjects, 20 are female and 133 are male. This dataset has no varying illumination and change in facial expression. Each and every subject has a green background, and they are consistent with the posture and facial expression on each image with a 180 x 200 resolution. All images have been extracted from video sequences.

3.10.1.1.1 Database Description

- **Number of individuals:** 153
- **Image resolution:** 180 by 200 pixels (portrait format)
- **Subjects:** female (20), male (133)

3.10.1.1.2 Variation of individual's images

- **Backgrounds:** the background is plain green

- **Head Scale:** none
- **Head turn, tilt and slant:** very minor variation in these attributes
- **Position of face in image:** minor changes
- **Image lighting variation:** none
- **Expression variation:** considerable expression changes

Figure 3.5 shows a set of 20 images from Face 94 image database. As we can see there are a very little variation with change in lighting condition and each image has a green background.

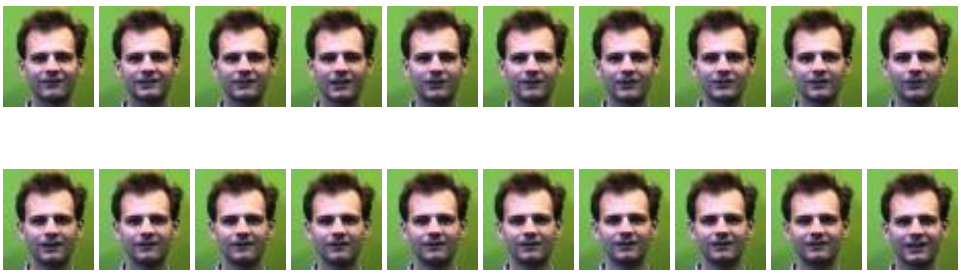


Figure 3.5: A sample image set from Face 94 database.

In order to assess the efficiency of the technique we have carried out a series of experiments using Face 94 dataset in two different procedures which have been discussed below.

3.10.1.2 Test Result from Procedure 1

In this procedure, the database has been divided into four different subsets. Subset 1 has 6 images which have been used for training and 14 images for testing. Subset 2 has 3 images for training and 17 images for testing, Subset 3 has used only 2 images for training the system, and the recognition algorithm was run on 18 images. Lastly, for

Subset 4, only one image was used to train the system and the algorithm was tested with 19 image samples which yielded an accuracy of 98.56%.

Table 3.2: Different images used for training and testing from Face 94 database.

	Images used for Training	**Images used for Testing**	**Max Accuracy achieved in %**
Subset 1	1,3,5,8,10,13	2,4,6,7,9,11,12,14,15,16,17,18,19,20	99.9524
Subset 2	5,9,14	1,2,3,4,6,7,8,10,11,12,13,15,16,17,18,19,20	100
Subset 3	1,3	2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20	99.8891
Subset 4	1	2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20	98.5639

Table 3.3 tabulates the result obtained by running the algorithm with a different number of principle component values and the percentage of accuracy achieved by the algorithm along with the average time taken by each facial recognition process.

Table 3.3: Results from Procedure 1 on Face 94 database.

Test Subset	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	14	6	50	99.9049	0.2209
			100	99.9524	0.2241
			150	99.9524	0.2258
			200	99.9049	0.2314
			250	99.8097	0.2412
Subset 2	17	3	50	99.9218	0.2007
			100	99.9218	0.2035
			150	100	0.2085
			200	100	0.2150
			250	100	0.2206
Subset 3	18	2	50	99.8152	0.1986
			100	99.8521	0.2011
			150	99.8891	0.2037
			200	99.8891	0.2054
			250	99.8891	0.2104
Subset 4	19	1	50	98.3537	0.1997
			100	98.5288	0.2012
			150	98.5639	0.2031
			200	98.5639	0.2034
			250	98.5639	0.2060

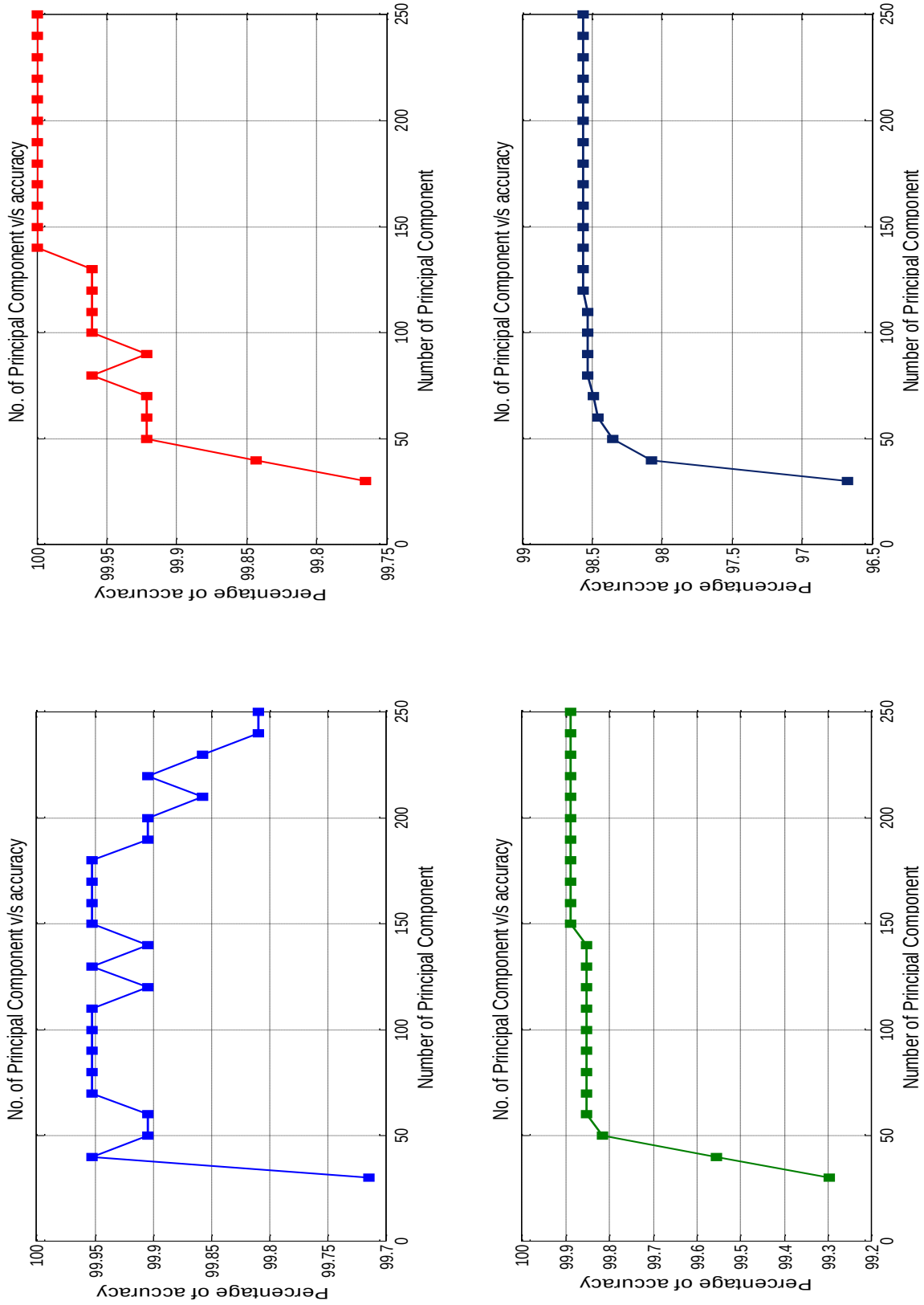


Figure 3.6: Graphs showing accuracy in percentage against the number of Principle Components.

3.10.1.3 Test Result from Procedure 2

In this procedure the database has been divided into 5 different subsets as in procedure 1. Subset 1 has 10 images which have been used for training and each Test Set has 10 images irrespective of the change in the number of training images. Subset 2 has 7 sample images for training; Subset 3 has 4 sample images for training; Subset 2 has 2 sample images for training; Subset 1 has only one sample image for training. The maximum accuracy achieved has been tabulated in Table 3.4 for each subset.

Table 3.4: Different images used for training and testing from Face 94 database.

	Images used for Training	**Images used for Testing**	**Max Accuracy achieved in %**
Subset 1	1,3,5,7,9,11,13,15,17,19	2,3,5,7,9,10,12,14,16,18	100
Subset 2	1,3,5,7,9,11,13		100
Subset 3	1,3,5,7		100
Subset 4	1,3		100
Subset 5	1		98.9355

Table 3.5 tabulates the result obtained by running the algorithm with different number of principle component values and the percentage of accuracy achieved by the algorithm along with the average time taken by each facial recognition process.

Table 3.5: Results from Procedure 2 on Face 94 database.

Test Set	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	10	10	50	99.9335	0.2668
			100	100	0.2498
			150	99.8669	0.2760
			200	100	0.2838
			250	99.9335	0.2735
Subset 2		7	50	100	0.2244
			100	100	0.2332
			150	100	0.2403
			200	100	0.2627
			250	99.9335	0.2090
Subset 3		4	50	99.9335	0.2150
			100	99.8669	0.2176
			150	100	0.2231
			200	100	0.2307
			250	99.9335	0.2431
Subset 4		2	50	99.9335	0.2230
			100	99.9335	0.2253
			150	100	0.2196
			200	100	0.2397
			250	100	0.2332
Subset 5		1	50	98.7359	0.2078
			100	98.8689	0.2195
			150	98.9355	0.2111
			200	98.9355	0.2195
			250	98.9355	0.2224

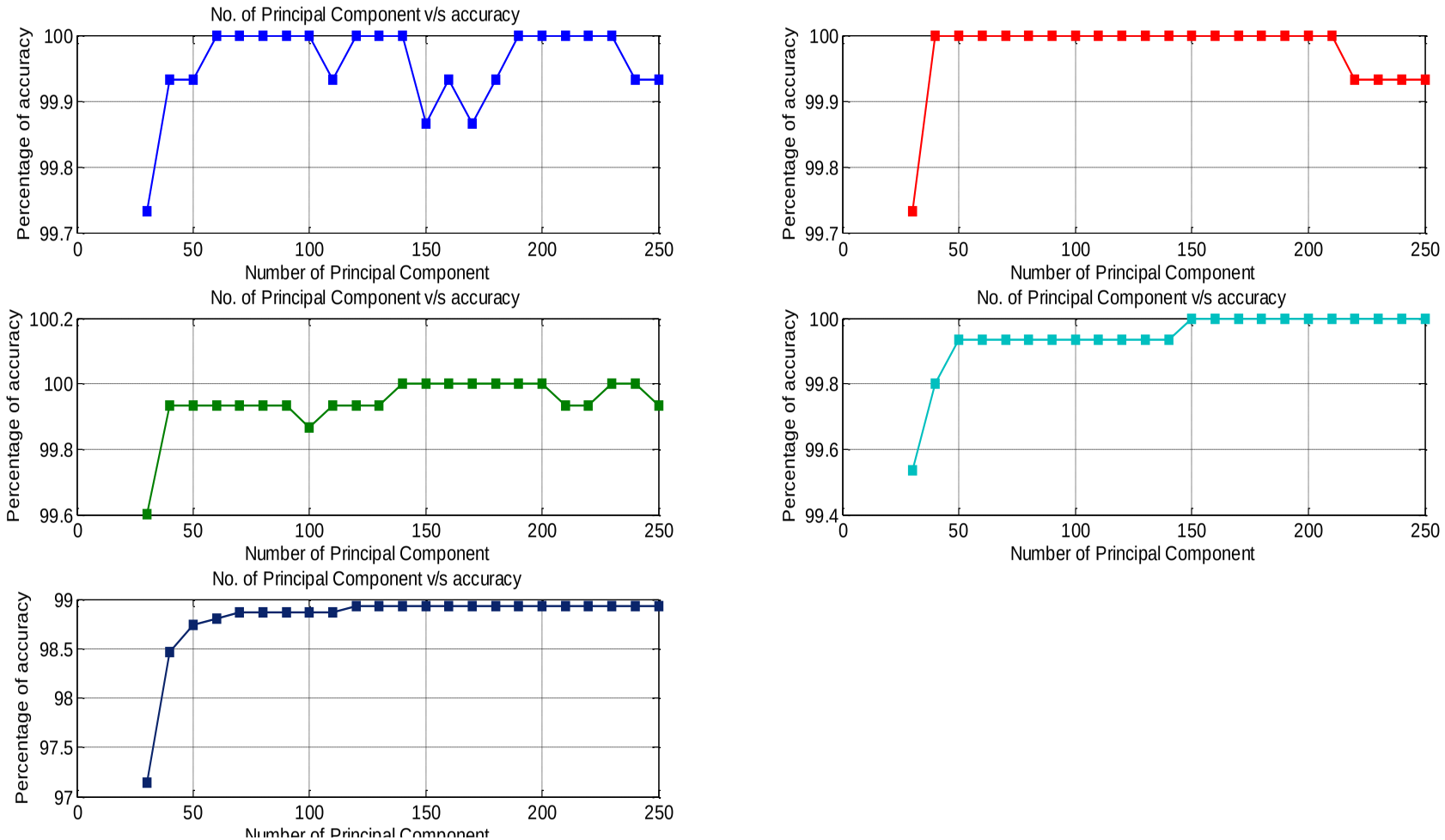


Figure 3.7: Graphs showing accuracy in percentage against the number of Principle Components.

3.10.2 Face95

3.10.2.1 About the Database

Using a fixed camera, a sequence of 20 images per individual was taken. During the sequence the subject takes one step forward towards the camera. This movement is used to introduce significant head scale variations between images of same individual. There is about 0.5 seconds between successive frames in the sequence.

3.10.2.1.1 Database Description

- **Number of individuals:** 72
- **Image resolution:** 180 by 200 pixels (portrait format).
- **Subjects:** Contains images of male and female subjects.

3.10.2.1.2 Variation of individual's images

- **Backgrounds:** the background consists of a red curtain. Background variation is caused by shadows as subject moves forward.
- **Head Scale:** large head scale variation.
- **Head turn, tilt and slant:** variation in these attributes.
- **Position of face in image:** some translation.
- **Image lighting variation:** as subject moves forward, significant lighting changes occur on faces due to the artificial lighting arrangement.
- **Expression variation:** some expression variation.

Figure 3.8 shows a set of 20 images of a subject from the Face95 dataset. As described the pictures consists of a red background and there are very small changes in the facial

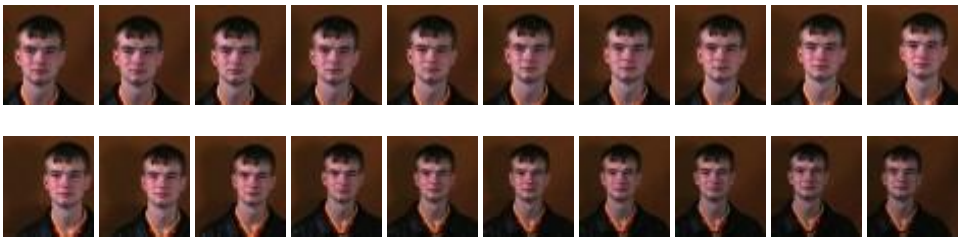


Figure 3.8: A sample image set from Face 95 database.

expressions or the variation of the background. There are significant changes in the head scale and size.

To check the efficiency of the proposed algorithm, the Face95 database has been used to check the robustness in many different subsets. The algorithm has been tested with different values of principle component and the change in the accuracy has been observed and tabulated in the following pages.

3.10.2.2 Test Result from Procedure 1

As described before in this procedure the database has been divided into 4 different subsets; Subset 1 has 6 images which have been used for training and 14 images for testing, the Subset 2 has 3 images for training and 17 images for testing, Subset 3 has 2 images for training and the recognition algorithm was run on 18 images and in the last Subset, only one image was used to train the system and the algorithm was tested on 19 images.

Table 3.6: Different images used for training and testing from Face 95 database.

	Images used for Training	Images used for Testing	Max Accuracy achieved in %
Subset 1	1,3,5,8,10,13	2,4,6,7,9,11,12,14,15,16,17,18,19,20	99.0909
Subset 2	5,9,14	1,2,3,4,6,7,8,10,11,12,13,15,16,17,18,19,20	98.9108
Subset 3	1,3	2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20	98.4599
Subset 4	1	2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20	91.4077

Table 3.7 tabulates the result obtained by running the algorithm with different principle component values and the percentage of accuracy achieved along with the average time taken for each facial recognition process.

Table 3.7: Results from Procedure 1 on Face 95 database.

Test Subset	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	14	6	50	97.9339	0.2125
			100	99.0083	0.2133
			150	99.0909	0.2191
			200	99.0909	0.2224
			250	99.9256	0.2262
Subset2	17	3	50	98.0259	0.2093
			100	98.8482	0.2098
			150	98.8428	0.2110
			200	98.7747	0.2129
			250	98.8428	0.2149
Subset3	18	2	50	97.4277	0.2087
			100	98.3280	0.2091
			150	98.4566	0.2097
			200	98.3923	0.2115
			250	98.3923	0.2132
Subset4	19	1	50	89.7623	0.2254
			100	91.4077	0.2222
			150	91.4077	0.2246
			200	91.4077	0.2247
			250	91.4077	0.2260

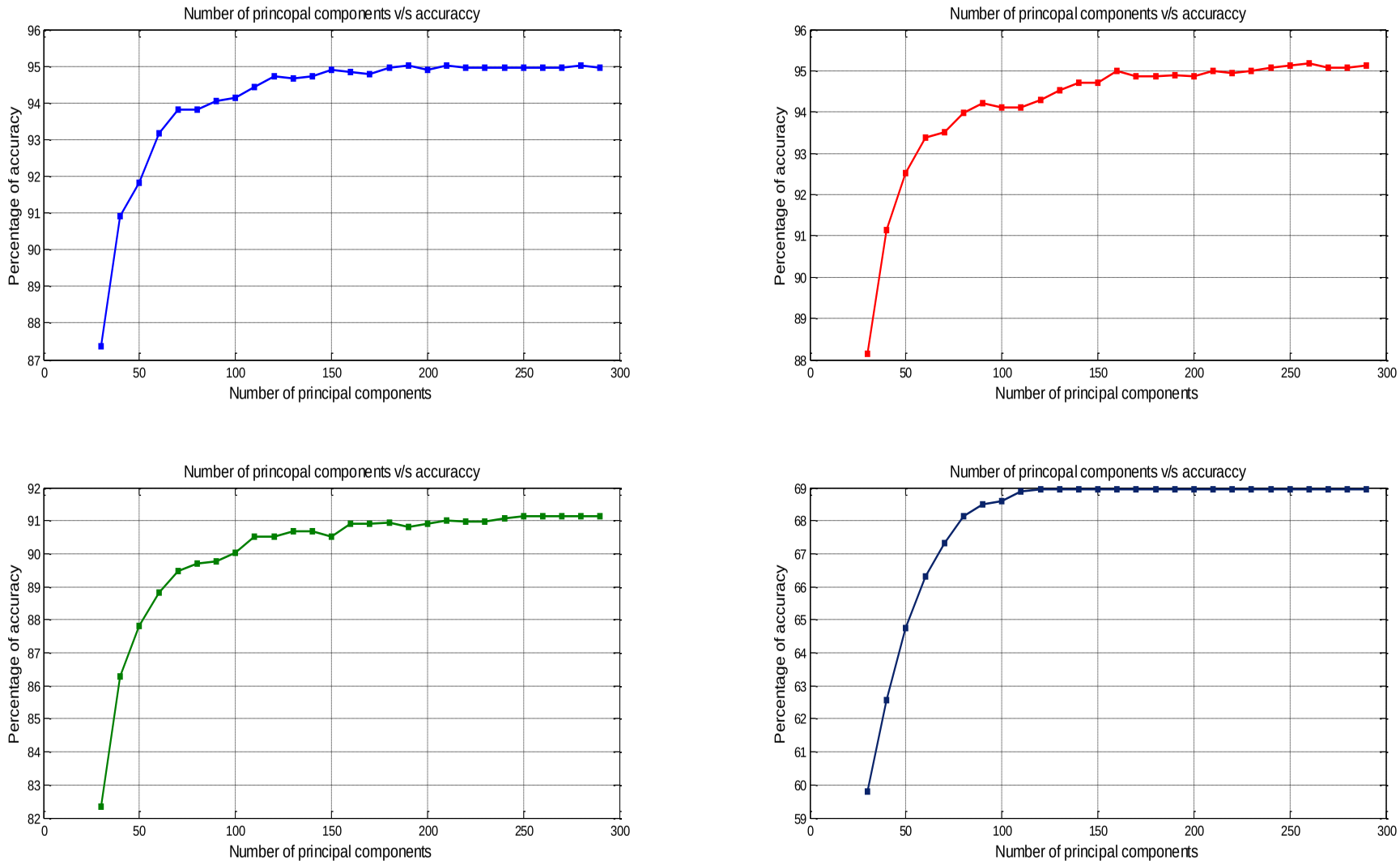


Figure 3.9: Graphs showing accuracy in percentage against the number of Principle Components.

3.10.2.3 Test Result from Procedure 2

In this test procedure the database has been divided into 4 Subsets as in procedure 1. Subset 1 has 10 images which have been used for the Train Set. The Test Set has 10 images irrespective of the change in the number of training images. Subset 2 has 4 images, Subset 3 has used 2 sample images for training the system and Subset 1 has used only 1 train image to train the algorithm which yielded a 92.03% of accuracy.

Table 3.8: Different images used for training and testing from Face 95 database.

	Images used for Training	**Images used for Testing**	**Max Accuracy achieved in %**
Subset 1	1,3,5,7,9,11,13,15,17,19	2,3,5,7,9,10,12,14,16,18	99.5381
Subset 3	1,3,5,7		98.8453
Subset 4	1,3		98.4988
Subset 5	1		92.0323

Table 3.9 tabulates the result obtained by running the algorithm with different number of principle component values and the percentage of accuracy achieved by the algorithm along with the average time taken by each facial recognition process while keeping the training set constant.

Table 3.9: Results from Procedure 2 on Face 95 database.

Test Subset	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	10	10	50	98.2679	0.2310
			100	99.3072	0.2364
			150	99.4226	0.2380
			200	99.4226	0.2418
			250	99.4226	0.2592
Subset 2		4	50	97.5751	0.2250
			100	98.6143	0.2256
			150	98.6143	0.2256
			200	98.6143	0.2314
			250	98.7298	0.2332
Subset 3		2	50	97.4596	0.2258
			100	98.1524	0.2277
			150	98.4988	0.2148
			200	98.3834	0.2305
			250	98.3834	0.2338
Subset 4		1	50	90.5312	0.2705
			100	92.0323	0.2493
			150	92.0323	0.2400
			200	92.0323	0.2530
			250	92.0323	0.2541

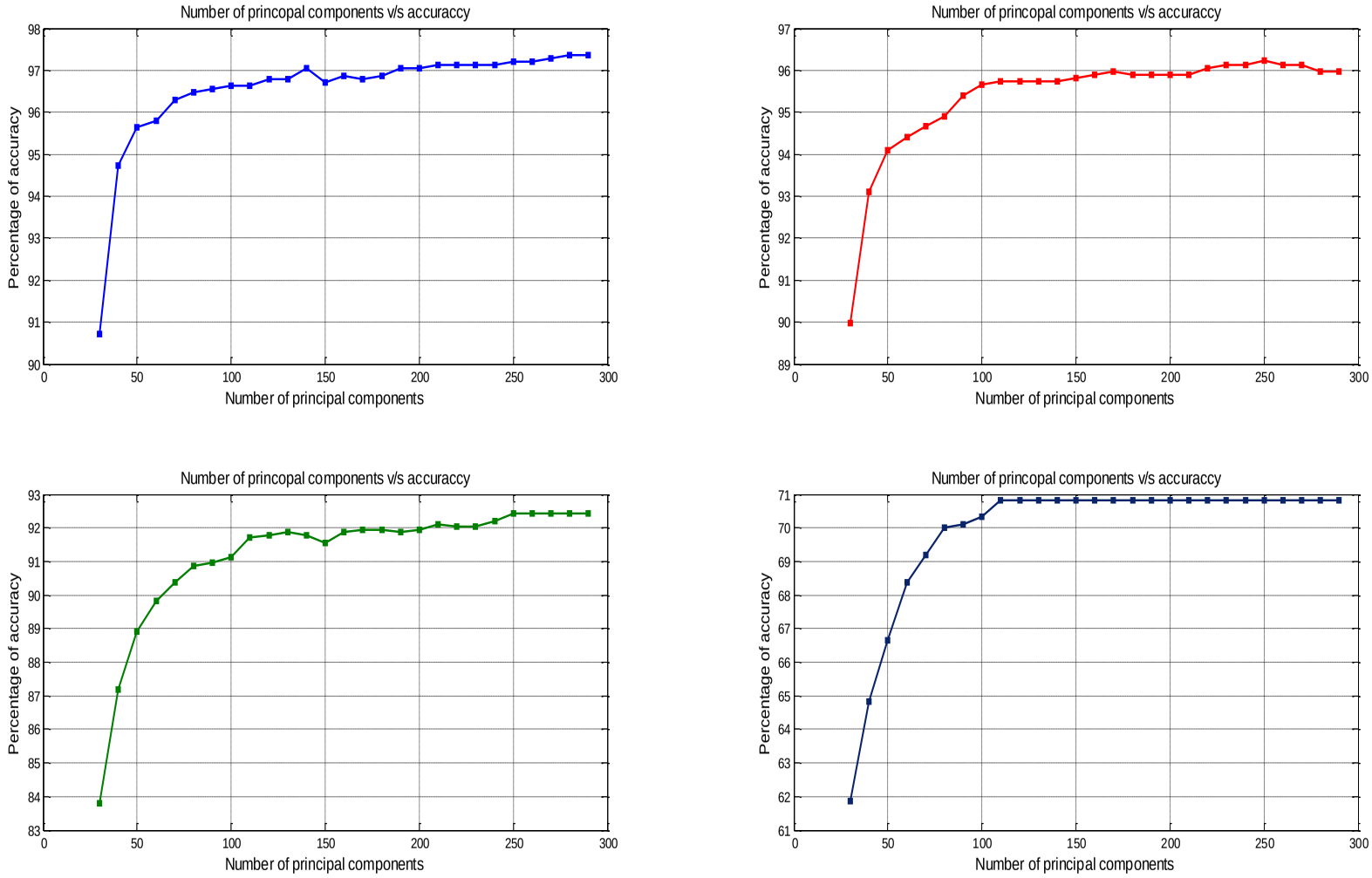


Figure 3.10: Graphs showing accuracy in percentage against the number of Principle Components.

3.10.3 Face96

3.10.3.1 About the Database

Using a fixed camera, a sequence of 20 images per individual was taken. During the sequence the subject takes one step forward towards the camera. This movement is used to introduce significant head variations between images of same individual. There is about 0.5 seconds between successive frames in the sequence.

3.10.3.1.1 Database Description

Number of individuals: 152

Image resolution: 196x196 pixels (square format)

Subjects: Contains images of male and female subjects

3.10.3.1.2 Variation of individual's images

- **Backgrounds:** the background is complex (glossy posters).
- **Head Scale:** large head scale variation.
- **Head turn, tilt and slant:** the images contain minor variation in these attributes.
- **Position of face in image:** some translation.
- **Image lighting variation:** as the subjects moves forward, significant lighting changes occur due to the artificial lighting arrangement.
- **Expression variation:** some expression variation.

Figure 3.11 shows a set of 20 images of a subject from the Face96 dataset. As described above the pictures consist of glossy background and there are some small changes in the facial expressions or the variation of the background, as well as many variations in the tilt and turn of the subject's head.

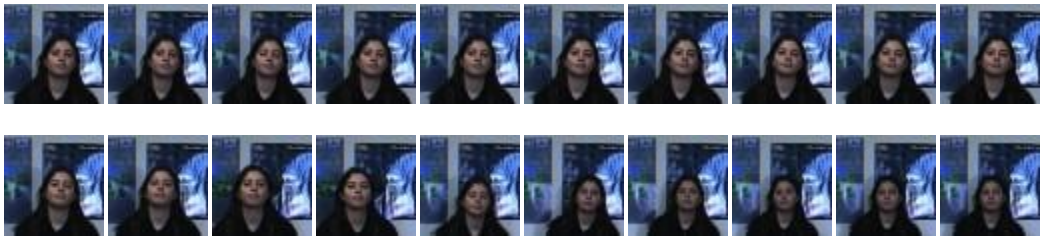


Figure 3.11: A sample image set from Face 96 database.

To check the efficiency of the proposed algorithm, this Face 96 database has been used as before. The algorithm has been tested with different values of principle component and the change in the accuracy has been observed and tabulated in the following pages with many different Subsets in two above mentioned procedures.

3.10.3.2 Test Result from Procedure 1

In this procedure the database has been divided into 4 different subsets, the Subset 1 has six images which have been used for training and fourteen images for testing. The Subset 2 has three images for training and seventeen image samples for testing, Subset 3 has used only two images for training and the recognition algorithm was run on eighteen images and in the last subset only one image was used to train the system and the algorithm was tested on nineteen sample images.

Table 3.10: Different images used for training and testing from Face 96 database.

	Images used for Training	Images used for Testing	Max Accuracy achieved in %
Subset 1	1,3,5,8,10,13	2,4,6,7,9,11,12,14,15,16,17,18,19,20	95.0203
Subset 2	5,9,14	1,2,3,4,6,7,8,10,11,12,13,15,16,17,18,19,20	95.1768
Subset 3	1,3	2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20	91.1512
Subset 4	1	2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20	68.9418

Table 3.11 tabulates the result obtained by running the algorithm with different principle component values and the percentage of accuracy achieved along with the average time taken for each facial recognition process.

Table 3.11: Results from Procedure 1 on Face 96 database.

Test Subset	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	14	6	50	91.8356	0.2575
			100	93.1517	0.2621
			150	94.9045	0.2679
			200	95.0203	0.2750
			250	94.9624	0.2862
Subset 2	17	3	50	92.5126	0.2435
			100	94.1203	0.2426
			150	94.7175	0.2441
			200	94.8553	0.2512
			250	95.1309	0.2559
Subset 3	18	2	50	87.8007	0.2524
			100	89.7766	0.2503
			150	90.5069	0.2494
			200	90.8935	0.2512
			250	91.1512	0.2524
Subset 4	19	1	50	64.7415	0.3360
			100	68.1341	0.3200
			150	68.9418	0.3192
			200	68.9418	0.3203
			250	68.9418	0.3236

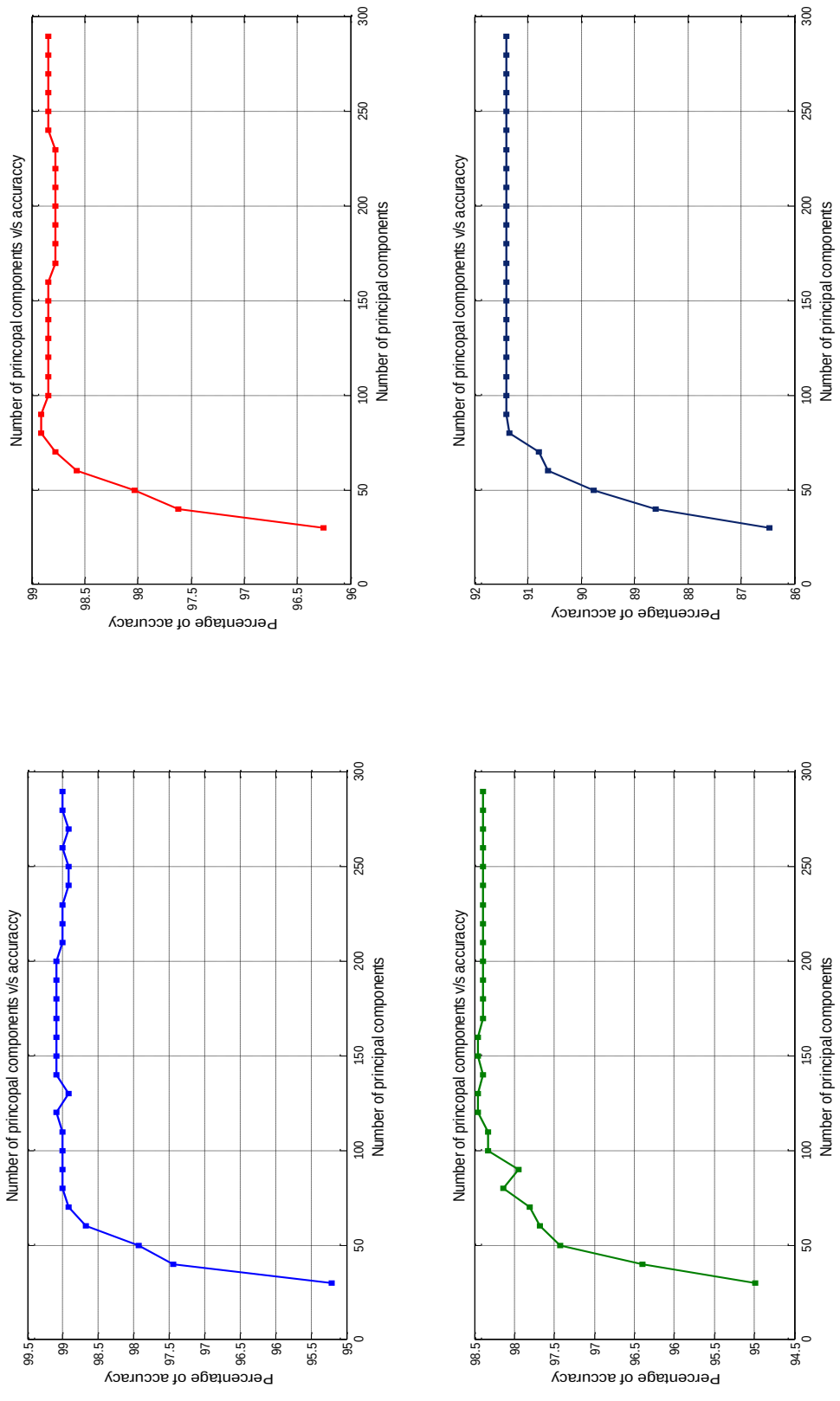


Figure 3.12: Graphs showing accuracy in percentage against the number of Principle Components.

3.10.3.3 Test Result from Procedure 2

In this test procedure the database has been divided into four different Subsets as in procedure 1. The subset one has ten images which have been used for training and all the test set has ten images irrespective of the change in the number of training images. Subset 2 has four images, Subset 3 has used two sample images for training the system and Subset 1 used only one training image to train the algorithm which yielded a 70.82% of accuracy.

Table 3.12: Different images used for training and testing from Face 96 database.

	Images used for Training	**Images used for Testing**	**Max Accuracy achieved in %**
Subset 1	1,3,5,7,9,11,13,15,17,19	2,3,5,7,9,10,12,14,16,18	97.3706
Subset 3	1,3,5,7		96.2202
Subset 4	1,3		92.4404
Subset 5	1		70.8299

Table 3.13 tabulates the result obtained by running the algorithm with a different number of principle component values and the percentage of accuracy achieved by the algorithm along with the average time taken by each facial recognition process while keeping the training set constant.

Table 3.13: Results from Procedure 2 on Face 96 database.

Test Subset	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	10	10	50	95.6450	0.2095
			100	96.5489	0.2184
			150	96.7132	0.2229
			200	97.0419	0.2750
			250	97.2062	0.2324
Subset 2		4	50	94.0838	0.2986
			100	95.6450	0.2981
			150	95.8094	0.2014
			200	95.8915	0.2026
			250	96.2202	0.2117
Subset 3		2	50	88.9071	0.2178
			100	90.9614	0.2056
			150	91.5366	0.2156
			200	91.8652	0.2174
			250	92.4404	0.2098
Subset 4		1	50	66.6393	0.2924
			100	70.3369	0.2540
			150	70.8299	0.2703
			200	70.8299	0.2486
			250	70.8299	0.2616

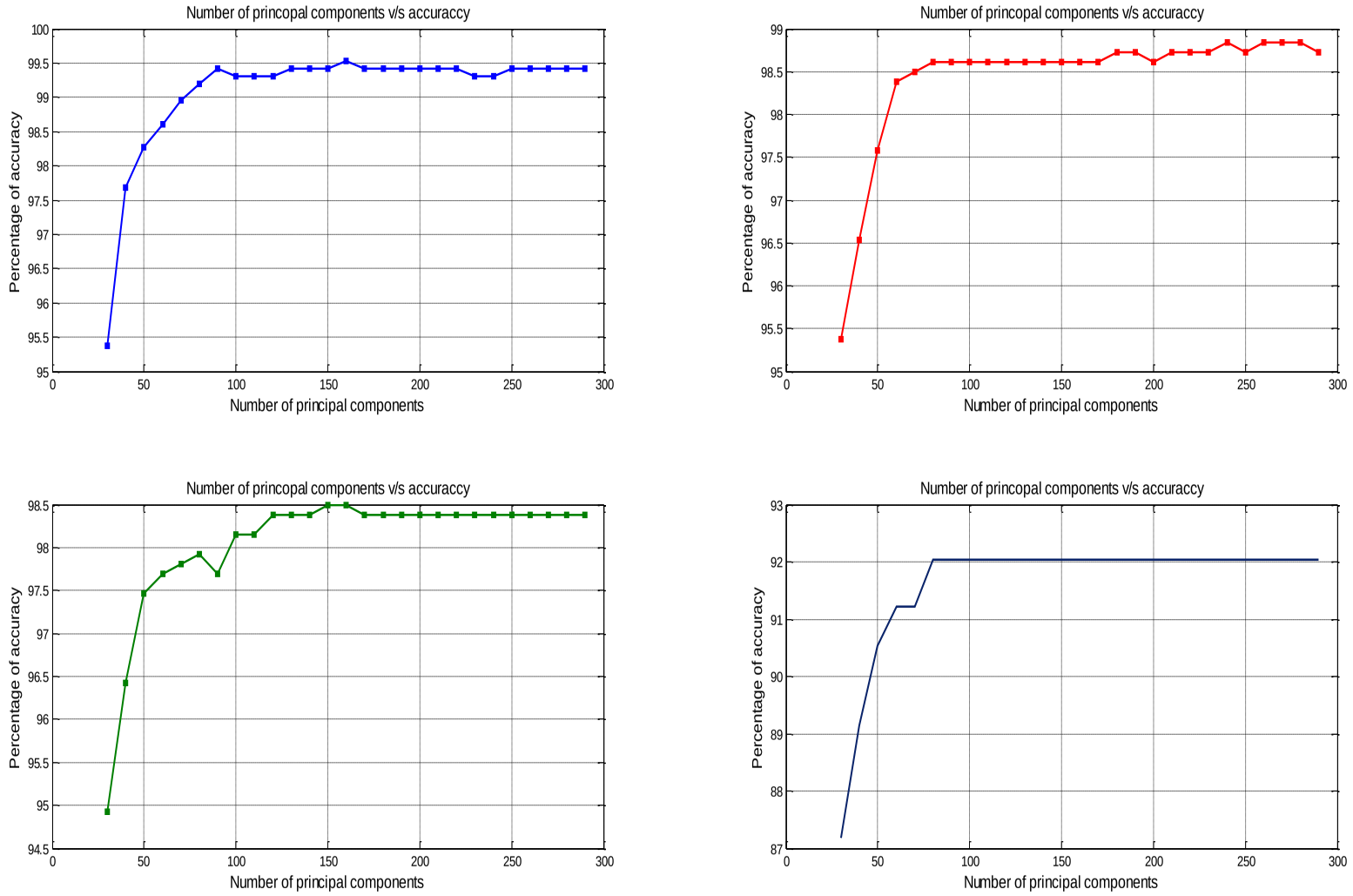


Figure 4.13: Graphs showing accuracy in percentage against the number of Principle Components.

3.10.4 FEI Face database

3.10.4.1 About the Database

The FEI Face Database [54] [55] [56] [57] is a Brazilian face database that contains a set of facial images taken between June 2005 and March 2006 at the Artificial Intelligence Laboratory of FEI in São Bernardo do Campo, São Paulo, Brazil. There are 14 images for each of 200 individuals, a total of 2800 images. All images are colorful and taken against a white homogenous background in an upright frontal position with profile rotation of up to about 180 degrees. Scale might vary about 10% and the original size of each image is 640x480 pixels. All faces are mainly represented by students and staff at FEI, between 19 and 40 years old with distinct appearance, hairstyle, and adorns. The numbers of male and female subjects are exactly the same and equal to 100. Figure 3.14 shows some examples of image variations from the FEI face database.

Figure 3.14: Some examples of image variations from the FEI face database.

3.10.4.1.1 Database Description

- **Number of individuals:** 200
- **Image resolution:** 640x480 pixels
- **Subjects:** Female 100, Male 100

3.10.4.1.2Variation of individual's images

- **Backgrounds:** White
- **Head Scale:** 10% change
- **Head turn, tilt and slant:** 180 degree turn
- **Position of face in image:** Minor changes
- **Image lighting variation:** Yes
- **Expression variation:** Yes

One of the primary requirements of the proposed algorithm is a frontal face image. Of the 14 images taken of each subject, only 5 were pertinent to this study. Figure 3.15 shows two sets of 5 images which have been selected for application with the proposed algorithm. As can be seen upon examination of the images, there are variations in the lighting, the subjects' facial expressions, the out-of-plane, frontal face in the 4^{th} sample and the white background. As there are 200 subjects in the database, we have divided the database into two parts, with each part containing 100 subjects to test the algorithm, but for both of them the Subset division will be same.

Figure 3.15: In this figure, 2 sets of five images have been shown which were used to test and train the algorithm from FEI Face Database.

Experiments have been carried out in the same way as in previous experiments on FEI Face database in two different procedures which has been discussed and the results are tabulated below.

3.10.4.2 Test Result from Procedure 1

As in previous experiments, this database has been divided into three different subsets. Subset 1 has three images used for training and two images used for testing the algorithm. Subset 2 has two images for training and three sample images for testing. Subset 3 has used only one image for training and four images are used for testing purpose. Table 3.14 shows the division of the database into different Subsets.

Table 3.14: Different images used for training and testing from FEI face database.

	Images used for Training	**Images used for Testing**	**Max Accuracy achieved in %**	**Max Accuracy achieved in %**
Subset 1	2,4,5	1,3	92.3664	86
Subset 2	2,4	1,3,5	86.3415	84.8101
Subset 3	2	1,3,4,5	81.2500	70.0461

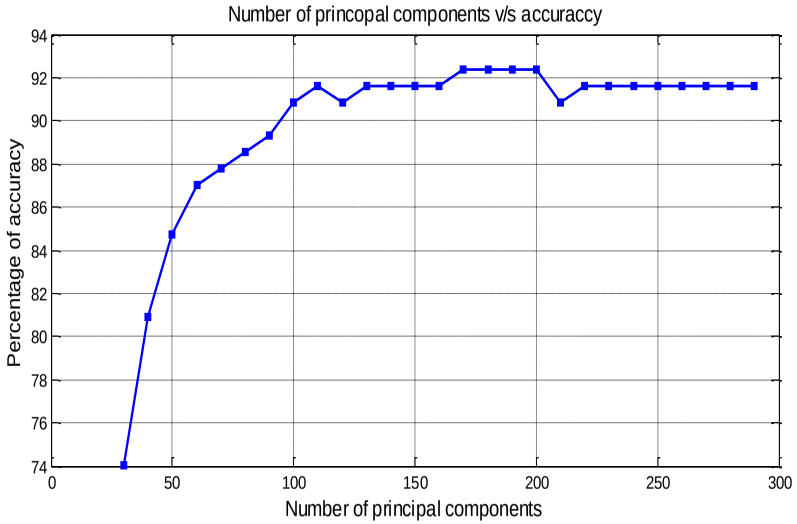
Table 3.15 tabulates the results obtained by running the algorithm with different principle component values and the percentage of accuracy achieved along with the average time taken for each facial recognition process. With the increase in the number of principle component the amount of time taken for the recognition process increases.

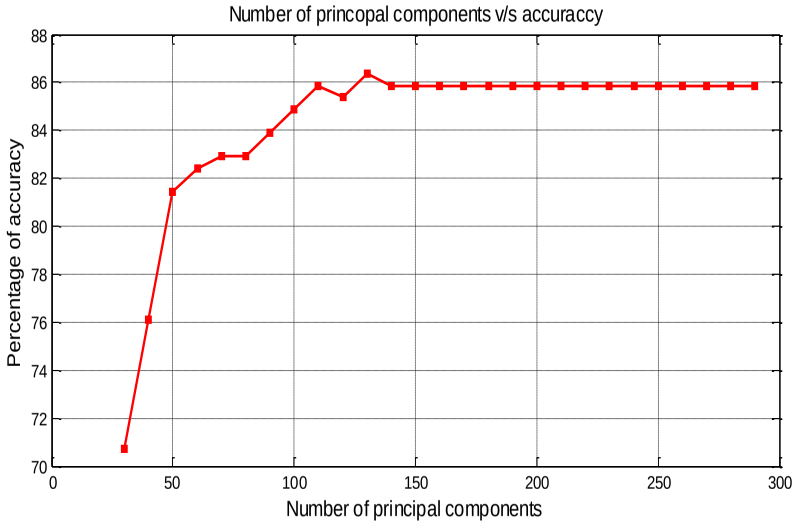
Table 3.15: Results from Procedure 1 on FEI face database for subject 1 to 100.

Test Subset	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	2	3	50	84.7328	0.3583
			100	90.8397	0.3337
			150	91.6031	0.3315
			200	92.3668	0.3304
			250	91.6031	0.3343
Subset 2	3	2	50	81.4634	0.3520
			100	84.8780	0.3478
			150	85.8537	0.3366
			200	85.8537	0.3343
			250	85.8537	0.3366
Subset 3	4	1	50	78.3008	0.3664
			100	81.2500	0.3502
			150	81.2500	0.3508
			200	81.2500	0.3520
			250	81.2500	0.3530

Table 3.16: Results from Procedure 1 on FEI face database for subject 100 to 200.

Test Subset	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	2	3	50	79	0.4860
			100	86	0.4837
			150	85	0.4918
			200	85	0.4850
			250	85	0.4883
Subset 2	3	2	50	80.3797	0.4817
			100	83.5443	0.4632
			150	84.1772	0.4595
			200	84.1772	0.4593
			250	84.1772	0. 4609
Subset 3	4	1	50	69.5853	0.5327
			100	70.0461	0.5306
			150	70.0461	0.5385
			200	70.0461	0.5351
			250	70.0461	0.5371





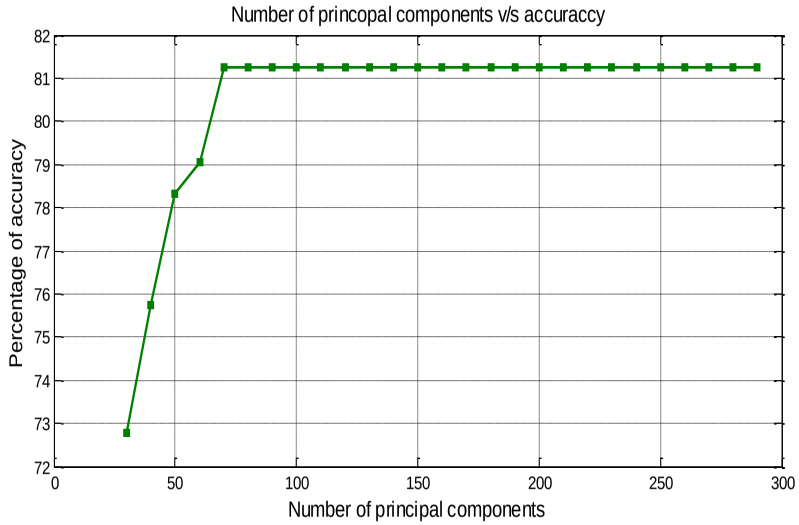
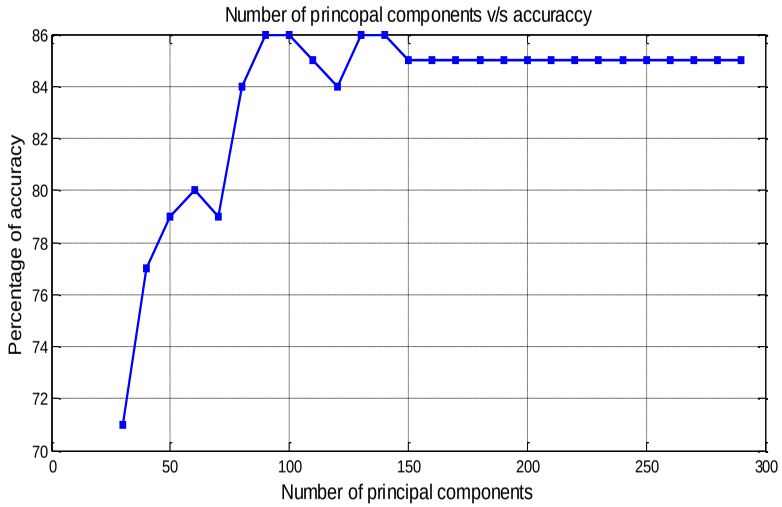
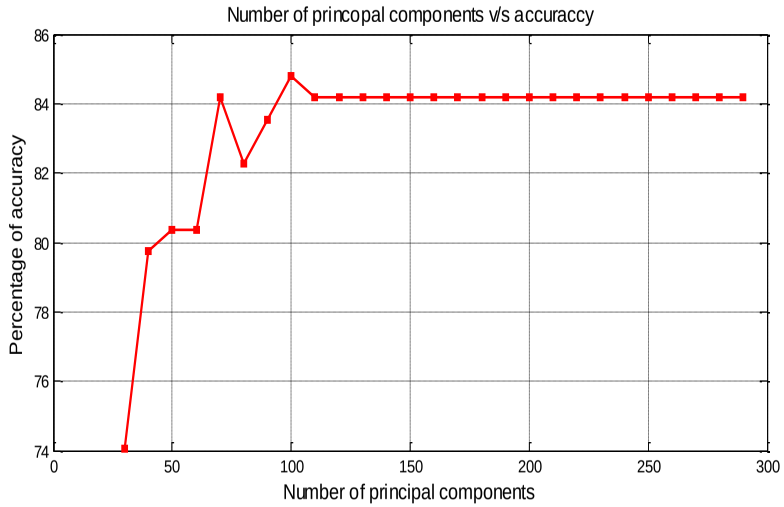


Figure 3.16: Graphs showing accuracy in percentage against the number of Principle Components for Set 1.





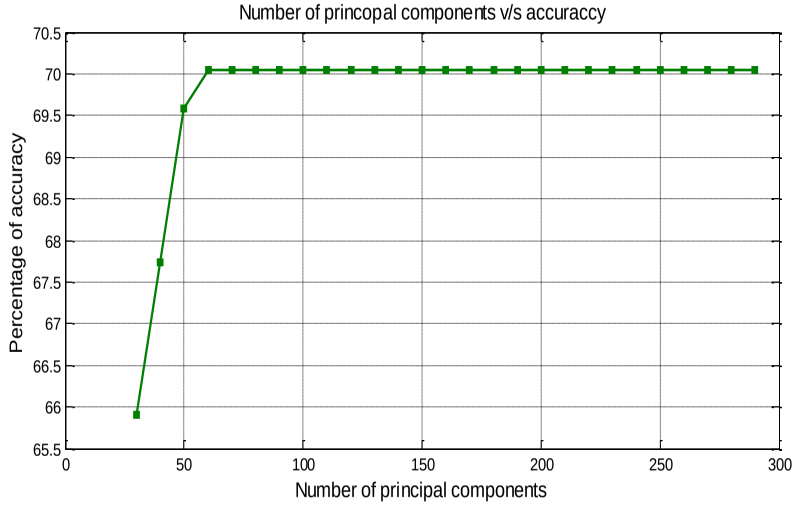


Figure 3.17: Graphs showing accuracy in percentage against the number of Principle Components for Set 2.

3.10.4.3 Test Result from Procedure 2

In this test procedure, the database has been divided into 3 Subsets as in Procedure 1 for the FEI face database. Subset 1 has three images used for training and the Test Set has two images irrespective of the change in the number of training images. Subset 2 has two images and Subset 1 has used only one sample image for training the system which yielded 77% accuracy for the algorithm.

Table 3.17: Different images used for training and testing from FEI face database.

	Images used for Training	**Images used for Testing**	**Max Accuracy achieved in %**	**Max Accuracy achieved in %**
Subset 1	2,4,5	1,3	92.3664	86
Subset 2	2,4		92.3664	87
Subset 3	2		84.7328	77

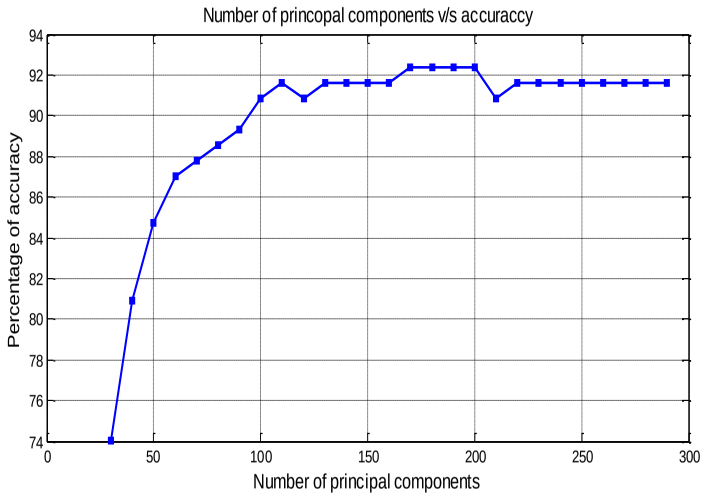
Table 3.18 and 3.19 tabulates the result obtained by running the algorithm on FEI face database with different number of principle component values and the percentage of accuracy achieved by the algorithm along with the average time taken by each facial recognition process while keeping the training set constant.

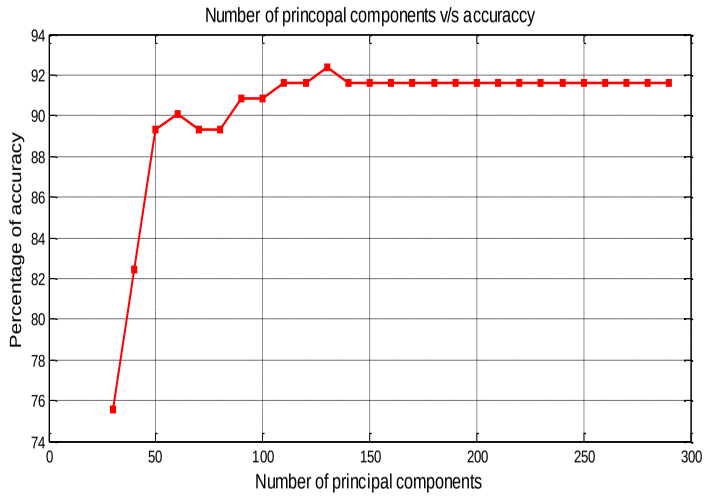
Table 3.18: Results from Procedure 2 on FEI face database for subject 1 to 100.

Test Set	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	2	3	50	84.7328	0.3757
			100	89.3130	0.3565
			150	91.6031	0.3797
			200	92.3664	0.3813
			250	91.6031	0.3817
Subset 2		2	50	89.3130	0.3833
			100	90.8397	0.3815
			150	91.6031	0.3874
			200	91.6031	0.3854
			250	91.6031	0.3771
Subset 3		1	50	83.2061	0.4039
			100	84.7328	0.3761
			150	84.7328	0.4034
			200	84.7328	0.3701
			250	84.7328	0.3746

Table 3.19: Results from Procedure 2 on FEI face database for subject 100 to 200.

Test Set	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (msec)
Subset 1	2	3	50	79	0.5140
			100	86	0.4716
			150	85	0.5221
			200	85	0.5282
			250	85	0.5187
Subset 2		2	50	81	0.5408
			100	87	0.5150
			150	85	0.5333
			200	85	0.5285
			250	85	0.5221
Subset 3		1	50	77	0.5618
			100	77	0.5350
			150	77	0.5782
			200	77	0.5161
			250	77	0.5236





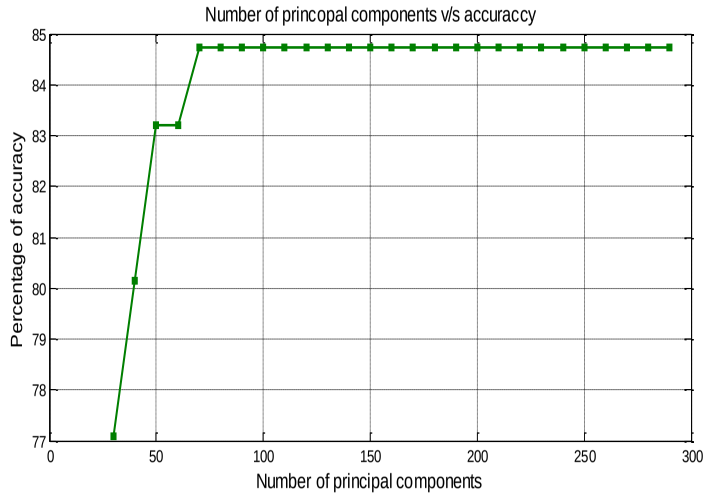
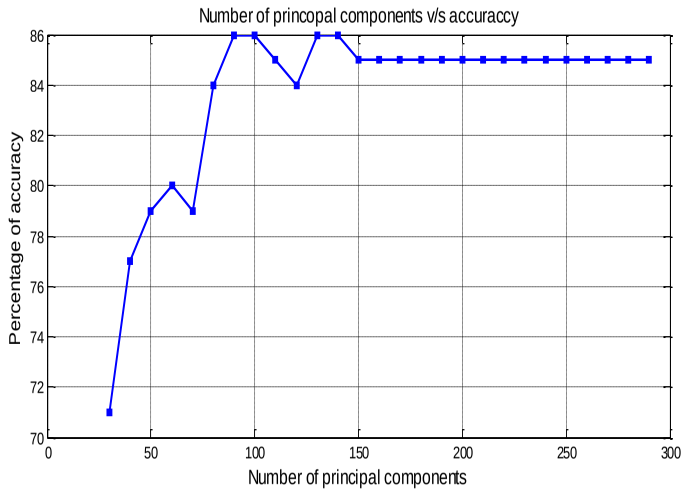
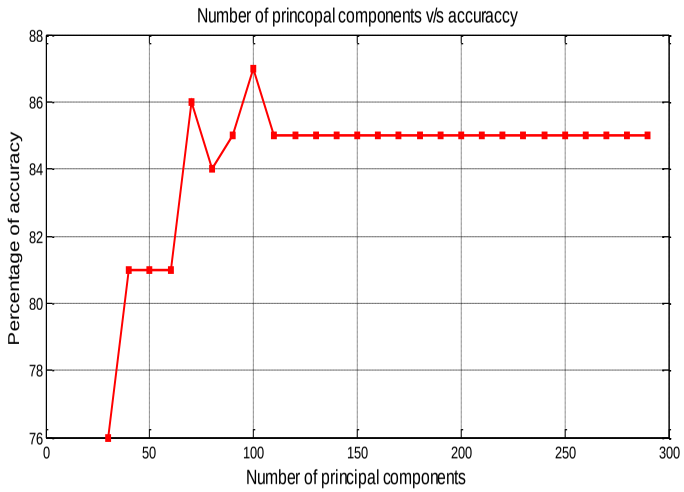


Figure 3.18: Graphs showing accuracy in percentage against the number of Principle Components for Set 1.





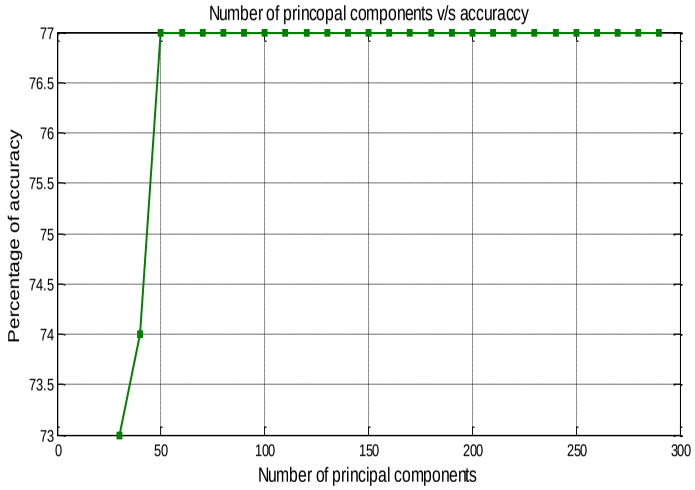


Figure 3.19: Graphs showing accuracy in percentage against the number of Principle Components for Set 2.

3.10.5 JAFFE

3.10.5.1 About the Database

The JAFFE Database [58] [59] [60] contains 213 images of 7 facial expressions (6 basic facial expressions + 1 neutral) posed by 10 Japanese female models. Each image has been rated on 6 emotion adjectives by 60 Japanese subjects.

3.10.5.1.1 Database Description

- **Number of individuals:** 10
- **Image resolution:** 256 x 256 pixels
- **Subjects:** Female 10

3.10.5.1.2 Variation of individual's images

- **Backgrounds:** White
- **Head Scale:** No significant change
- **Head turn, tilt and slant:** Lot of variations
- **Position of face in image:** Portrait
- **Image lighting variation:** None
- **Expression variation:** Extensive

Figure 3.20 shows a set of 20 images of a subject from the JAFFE dataset. As described, the sample pictures consist of white background and there are extensive changes in the facial expressions. The variation in the background is mainly due to the movements and the shadow of the lighting used which is very nominal. There are some variations in the

tilt and turn but not too extensive. All the sample images have extensive variations in the facial expression and emotion.

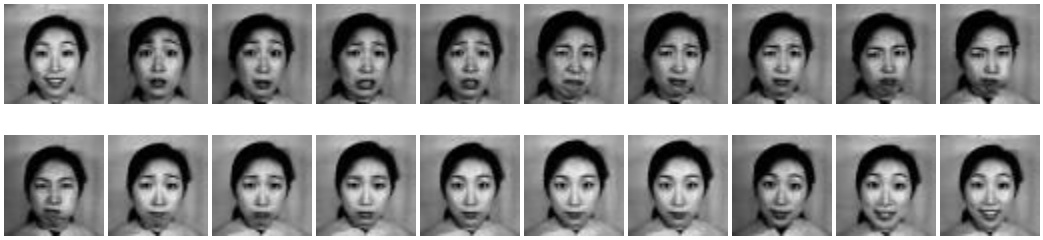


Figure 3.20: A sample image set from JAFFE database.

Experiments have been carried out in the same way as in previous experiments on JAFFE Face database and in two different procedures which have been discussed. The results have been tabulated below.

3.10.5.2 Test Result from Procedure 1

As in previous cases, this database has been divided into five different subsets. Each subset has a different set of images which have been used to train and test the facial recognition algorithm. As the total number of images for each subject is twenty, Subset 1 has used seven images for training the algorithm and thirteen images of the same subject to test the algorithm. In the same way, Subset 2 has used five images for training and fifteen image samples for testing. The Subset 3 has three training samples and seventeen test samples. Subset 4 has used two samples of a subject to train the algorithm and eighteen samples for testing and finally the last subset, Subset 5, has only one sample image to train the algorithm and the remaining nineteen images have been used for testing purposes. Table 3.20 shows the images used for training and testing purpose for every subset from the JAFFE database.

Table 3.20: Different images used for training and testing from JAFFE database in different Subsets.

	Images used for Training	**Images used for Testing**	**Max Accuracy achieved in %**
Subset 1	3,7,11,12,14,15,18	1,2,4,5,6,8,9,10,13,16,17,19,20	98.4615
Subset 2	11,12,14,15,18	1,2,3,4,5,6,7,8,9,10,13,16,17,19,20	98.0121
Subset 3	11,14,15	1,2,3,4,5,6,7,8,9,10,12,13,16,17,18,19,20	98.2353
Subset 4	11, 15	1,2,3,4,5,6,7,8,9,10,12,13,14,16,17,18,19,20	97.7778
Subset 5	15	1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,20	96.8412

Table 3.21 tabulates the result obtained by running the algorithm with different principle component values and the percentage of accuracy achieved along with the average time taken for each facial recognition process.

Table 3.21: Results from Procedure 1 on JAFFE database.

Test Subset	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	13	7	50	98.4615	0.2328
			100	98.4615	0.2406
			150	98.4615	0.2443
			200	98.4615	0.2654
			250	98.4615	0.2601
Subset 2	15	5	50	98	0.2547
			100	98	0.2554
			150	98	0.2576
			200	98	0.2607
			250	98	0.2618
Subset 3	17	3	50	98.2353	0.2529
			100	98.2353	0.2541
			150	98.2353	0.2537
			200	98.2353	0.2571
			250	98.2353	0.2552
Subset 4	18	2	50	97.7778	0.2521
			100	97.7778	0.2531
			150	97.7778	0.2553
			200	97.7778	0.2566
			250	97.7778	0.2562
Subset 5	19	1	50	96.8412	0.2541
			100	96.8412	0.2557
			150	96.8412	0.2487
			200	96.8412	0.2485
			250	96.8412	0.2510

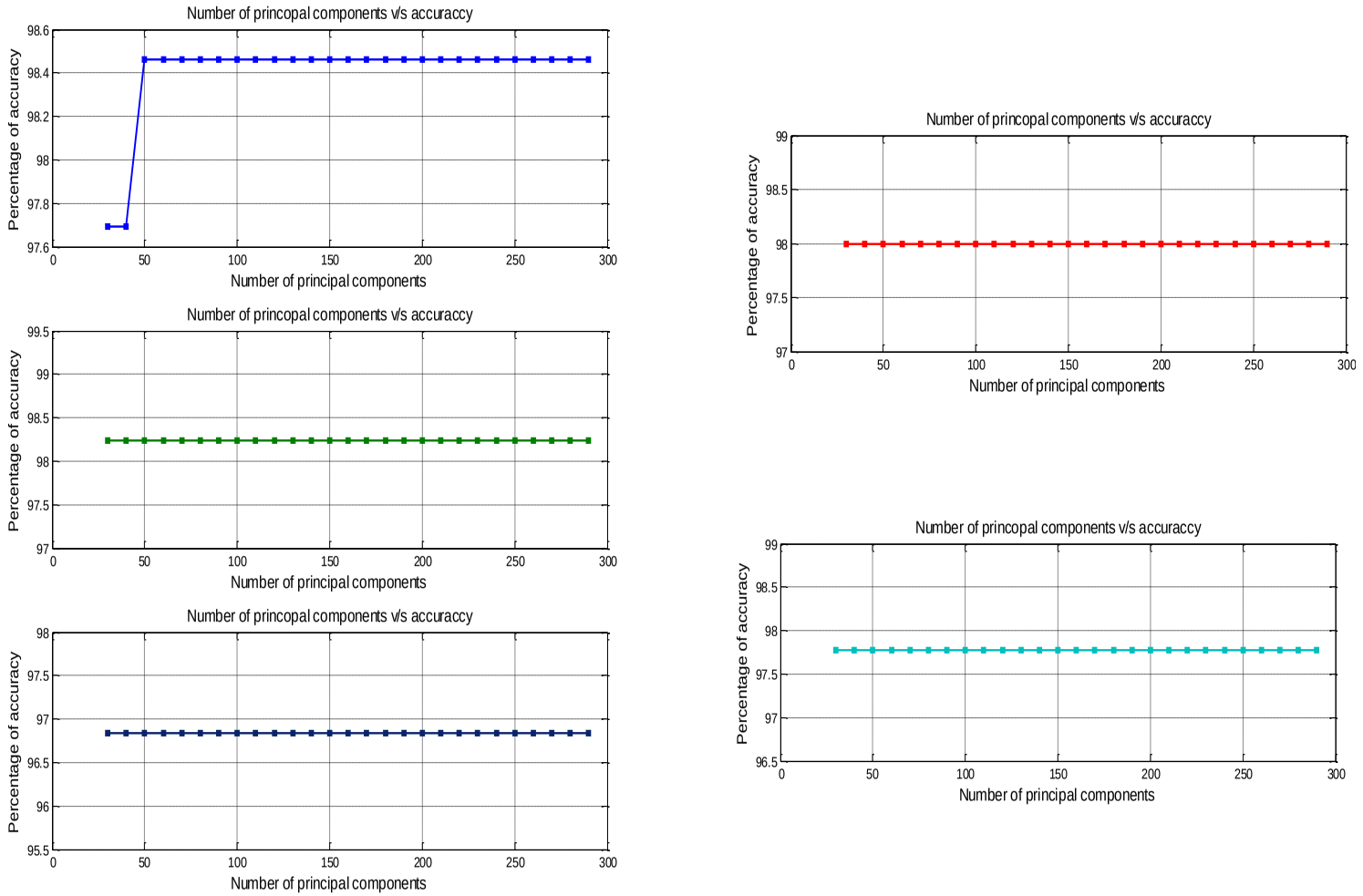


Figure 3.21: Graphs showing accuracy in percentage against the number of Principle Components.

3.10.5.3 Test Result from Procedure 2

In this test procedure, the database has been divided into five Subsets as in Procedure 1. Subset 1 has seven images used for training and the Test Set has thirteen images irrespective of the change in the number of training images. Subset 2 has five images, Subset 3 has used three samples images for training the system, Subset 4 has two training images and Subset 5 has used only one training image sample to train the algorithm which yielded a 96.15% of accuracy. Table 3.22 shows the details of which images from the JAFFE Face database have been used for training and which images have been used for testing.

Table 3.22: Different images used for training and testing from JAFFE database.

	Images used for Training	**Images used for Testing**	**Max Accuracy achieved in %**
Subset 1	3,7,11,12,14,15,18	1,2,4,5,6,8,9,10,13,16,17,19,20	98.4615
Subset 2	11,12,14,15,18		97.6923
Subset 3	11,14,15		97.6923
Subset 4	11, 15		96.9231
Subset 5	15		96.1538

Table 3.23 tabulates the result obtained by running the algorithm with varying number of principle component values and the percentage of accuracy achieved by the algorithm along with the average time taken by each facial recognition process while keeping the training set constant.

Table 3.23: Results from Procedure 2 on JAFFE database.

	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	13	7	50	98.4615	0.2395
			100	98.4615	0.2437
			150	98.4615	0.2615
			200	98.4615	0.2611
			250	98.4615	0.2609
Subset 2		5	50	97.6923	0.2577
			100	97.6923	0.2616
			150	97.6923	0.2616
			200	97.6923	0.2597
			250	97.6923	0.2629
Subset 3		3	50	97.6923	0.2567
			100	97.6923	0.2577
			150	97.6923	0.2608
			200	97.6923	0.2608
			250	97.6923	0.2592
Subset 4		2	50	96.9231	0.2592
			100	96.9231	0.2591
			150	96.9231	0.2606
			200	96.9231	0.2617
			250	96.9231	0.2641
Subset 5		1	50	96.1534	0.2600
			100	96.1534	0.2634
			150	96.1534	0.2622
			200	96.1534	0.2647
			250	96.1534	0.2651

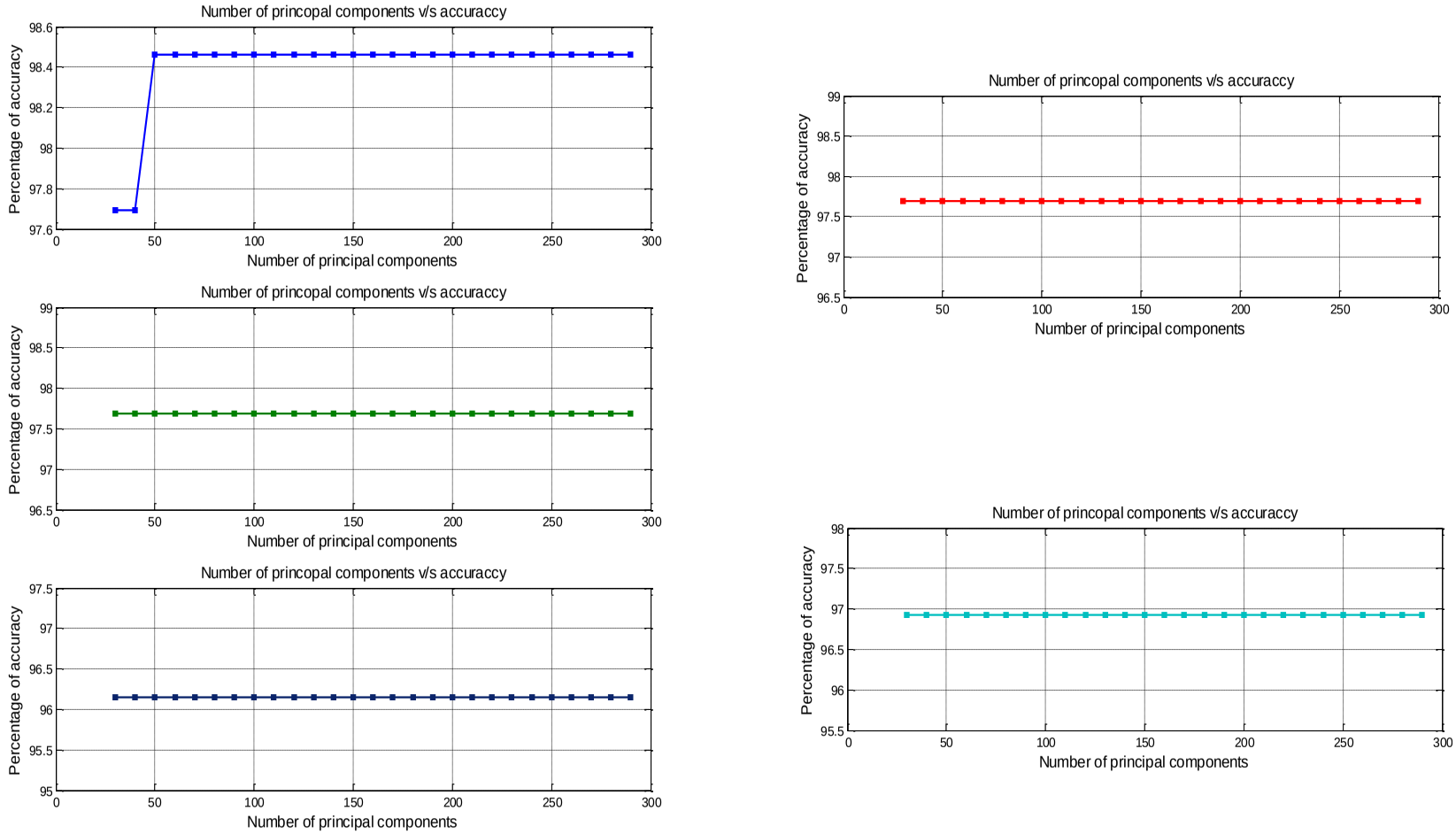


Figure 3.22: Graphs showing accuracy in percentage against the number of Principle Components.

3.10.6 PUT Vein Pattern Database

3.10.6.1 About the Database

PUT Vein Pattern database [61] was designed to test complex facial algorithms and has 22 sample images for each subjects. There are 100 subjects with 88 male and 12 female between the ages of 20 to 45. Each individual has been pictured in front of a pale yellow background and with proper lighting condition to remove occlusion.

3.10.6.1.1 Database Description

- **Number of individuals:** 100
- **Image resolution:** 2048x1536
- **Subjects:** Male: 88, Female 12

3.10.6.1.2 Variation of individual's images

- **Backgrounds:** Pale Yellow
- **Head Scale:** No change.
- **Head turn, tilt and slant:** Lot of variations
- **Position of face in image:** Middle
- **Image lighting variation:** None
- **Expression variation:** Few

Figure 3.23 shows a set of 22 images of a subject from the PUT Vein Pattern dataset. As described, the sample pictures consist of a pale yellow background and there are minor changes in the facial expressions. The variation in the background is very nominal and there are variations in the tilt and turn of the subject's head, as well as many image

samples with out-of-plane, facial structure. All the sample images have some variations in the facial expression and emotion.

Figure 3.23: In A sample image set from PUT Vein Pattern database.

Experiments have been carried out in the same way as in previous experiments from PUT Vein Pattern database in two different procedures which has been discussed and the results have been tabulated below. As the image size of this database is too big, before processing they have been reduced to one fourth of their original size.

3.10.6.2 Test Result from Procedure 1

In this test procedure, the database has been divided into 4 different subsets. Subset 1 has six images used for training and sixteen images used for testing. Subset 2 has four images for training and eighteen image samples for testing, Subset 3 has used only two images for training and recognition algorithm was run on twenty images and in the last subset, Subset 4, only one image was used to train the system and the algorithm was tested on twenty one image samples.

Table 3.24: Different images used for training and testing from PUT Vein Pattern database in different Subsets.

	Images used for Training	Images used for Testing	Max Accuracy achieved in %
Subset 1	5,6,7,16,20,21	1,2,3,4,8,9,10,11,12,13,14,15,17,18,19,22	97.2310
Subset 2	5,6,7,16	1,2,3,4,8,9,10,11,12,13,14,15,17,18,19,20,21,22	96.5884
Subset 3	5,6	1,2,3,4,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22	88.5312
Subset 4	6	1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22	87.4042

Table 3.25 tabulates the result obtained by running the algorithm with different principle component values and the percentage of accuracy achieved, along with the average time taken for each facial recognition process.

Table 3.25: Results from Procedure 1 on PUT Vein Pattern database.

Test Subset	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	16	6	50	94.0214	0.6006
			100	96.0352	0.5824
			150	96.8534	0.5890
			200	97.0422	0.6352
			250	97.0422	0.6271
Subset 2	18	4	50	93.7360	0.5941
			100	95.8613	0.5847
			150	96.3647	0.5838
			200	96.3647	0.5887
			250	96.5884	0.5870
Subset 3	20	2	50	84.6076	0.6594
			100	87.8270	0.6691
			150	88.3303	0.6267
			200	88.4306	0.6245
			250	88.5312	0.6637
Subset 4	21	1	50	83.8123	0.6632
			100	87.0690	0.6838
			150	87.4042	0.6721
			200	87.4042	0.6396
			250	87.4042	0.6074

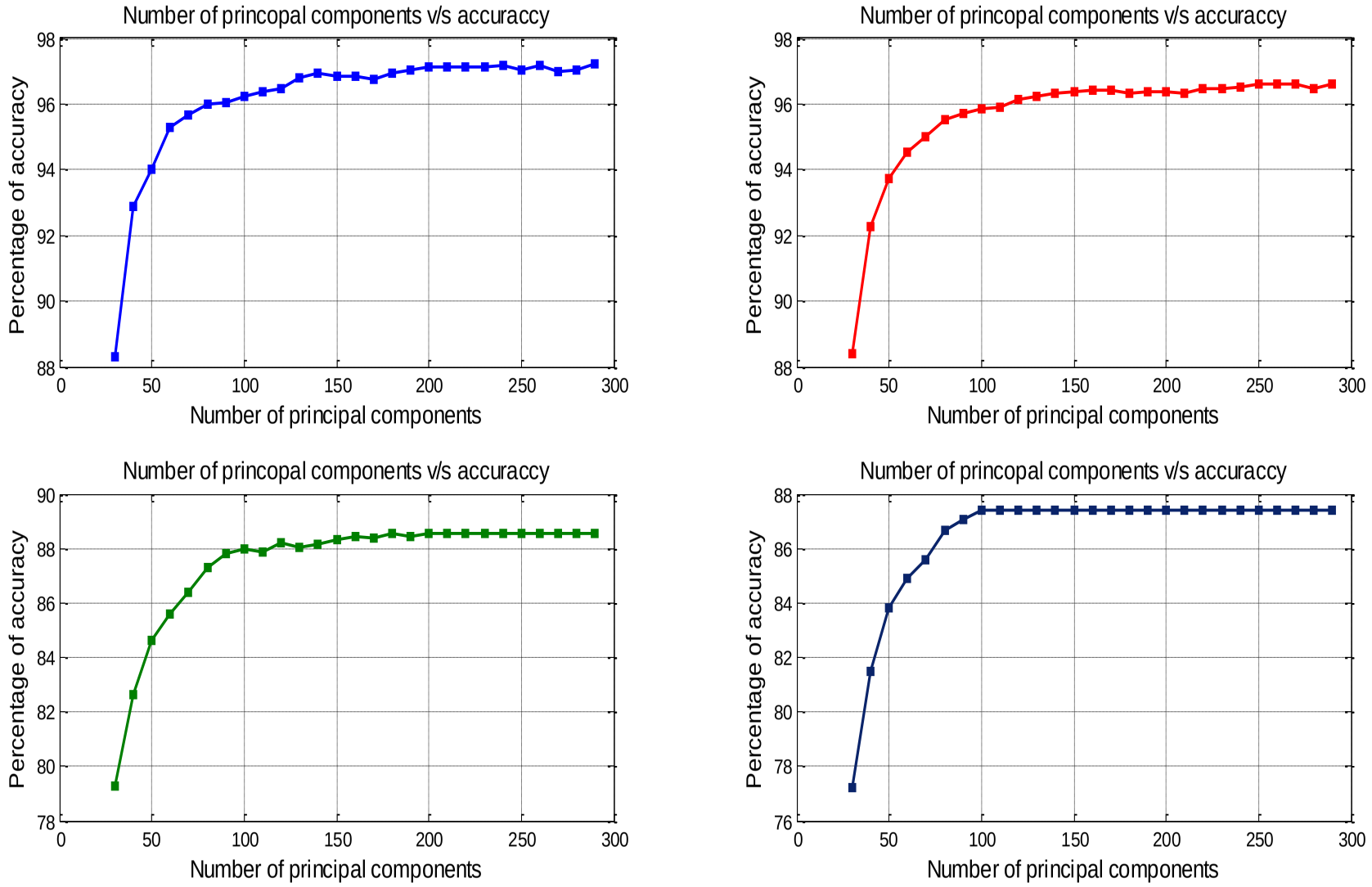


Figure 3.24: Graphs showing accuracy in percentage against the number of Principle Components.

3.10.6.3 Test Result from Procedure 2

In this test procedure, the database has been divided into four different Subsets as in procedure 1. Subset 1 has six images which have been used for training and the Test Set has sixteen images irrespective of the change in the number of training images. Subset 2 has four images, Subset 3 has used two sample images for training the system and Subset 4 used only one training image to train the algorithm which yielded an 85.14% of accuracy.

Table 3.26: Different images used for training and testing from PUT Vein Pattern database in different Subsets.

	Images used for Training	**Images used for Testing**	**Max Accuracy achieved in %**
Subset 1	5,6,7,16,20,21	1,2,3,4,8,9,10,11,12,13,14,15,17,18,19,22	97.2310
Subset 2	5,6,7,16		96.5387
Subset 3	5,6		87.1617
Subset 4	6		85.1479

Table 3.27 tabulates the result obtained by running the algorithm with varying number of principle component values and the percentage of accuracy achieved by the algorithm along with the average time taken by each facial recognition process while keeping the Train Set constant.

Table 3.27: Results from Procedure 2 on PUT Vein Pattern database.

Test Subset	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	16	6	50	94.0214	0.6006
			100	96.0352	0.5827
			150	96.8534	0.5890
			200	97.0422	0.6353
			250	97.0422	0.6264
Subset 2		4	50	93.5179	0.5953
			100	95.5318	0.5840
			150	96.2870	0.5856
			200	96.2870	0.5889
			250	96.5387	0.5903
Subset 3		2	50	82.6935	0.6693
			100	86.3436	0.6424
			150	86.9100	0.6785
			200	87.0359	0.6840
			250	87.1617	0.6392
Subset 4		1	50	80.9943	0.7124
			100	84.7703	0.6910
			150	85.1479	0.6640
			200	85.1479	0.6905
			250	85.1479	0.6959

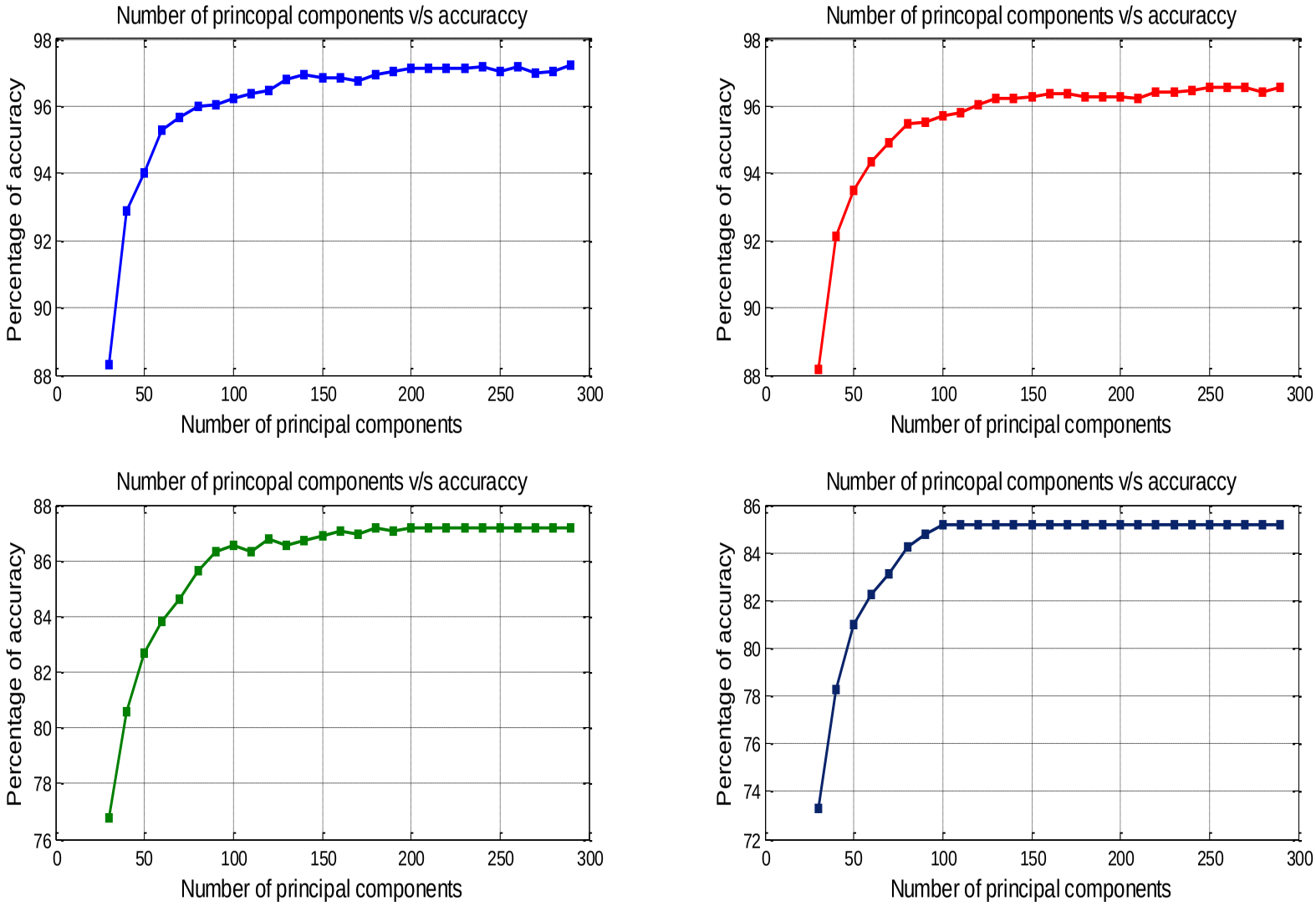


Figure 4.25: Graphs showing accuracy in percentage against the number of Principle Components.

3.10.7 Yale Face database

3.10.7.1 About the Database

The Yale Face Database contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, and sad, fatigued, surprised, and wink.

3.10.7.1.1 Database Description

- **Number of individuals:** 11
- **Image resolution:** 320 x 243 pixel
- **Subjects:** Male 10, Female 1.

3.10.7.1.2 Variation of individual's images

- **Backgrounds:** White and shadow
- **Head Scale:** None
- **Head turn, tilt and slant:** Small variation
- **Position of face in image:** Centered
- **Image lighting variation:** Moderate Variation
- **Expression variation:** Moderate Variation

Figure 2.26 shows two sets of eleven images which have been used for every subject to test the algorithm. As we can see there are variations in lighting and expression, and the

background is white as described but there are large variations in the lighting which greatly effects the different backgrounds.

Figure 2.26: 2 sets of five images have been showed which were used to test and train the algorithm.

Experiments have been carried out in the same way as in previous experiments on Yale Face database in two different procedures which have been discussed and the results have been tabulated below.

3.10.7.2 Test Result from Procedure 1

In this procedure the database has been divided into 4 different subsets. Subset 1 has six images used for training and five images used for testing. Subset 2 has four images for training and seven image samples for testing, Subset 3 has used only two images for training and recognition algorithm was run on nine images and in the last subset, Subset 4, only one image which was used to train the system and the algorithm was tested on ten image samples.

Table 3.28: Shows the different images used for training and testing the Yale Face database in different Subsets.

	Images used for Training	**Images used for Testing**	**Max Accuracy achieved in %**
Subset 1	1,3,5,8,9,11	2,4,6,7,10	100
Subset 2	5,8,9,11	1,2,3,4,6,7,10	100
Subset 3	5,8	1,2,3,4,6,7,9,10,11	86.5672
Subset 4	5	1,2,3,4,6,7,8,9,10,11	87.7551

Table 3.29 tabulates the result obtained by running the algorithm with different principle component values and the percentage of accuracy achieved, along with the average time taken for each facial recognition process for the Yale Face Database.

Table 4.29: Results from Procedure 1 on Yale Face database.

Test Set	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	5	6	50	100	0.2597
			100	100	0.2736
			150	100	0.2763
			200	100	0.2734
			250	100	0.2781
Subset 2	7	4	50	100	0.2753
			100	100	0.2689
			150	100	0.2701
			200	100	0.2716
			250	100	0.2739
Subset 3	9	2	50	86.5672	0.3154
			100	86.5672	0.3135
			150	86.5672	0.3248
			200	86.5672	0.3126
			250	86.5672	0.3254
Subset 4	10	1	50	87.7551	0.3194
			100	87.7551	0.2973
			150	87.7551	0.3012
			200	87.7551	0.2984
			250	87.7551	0.3041

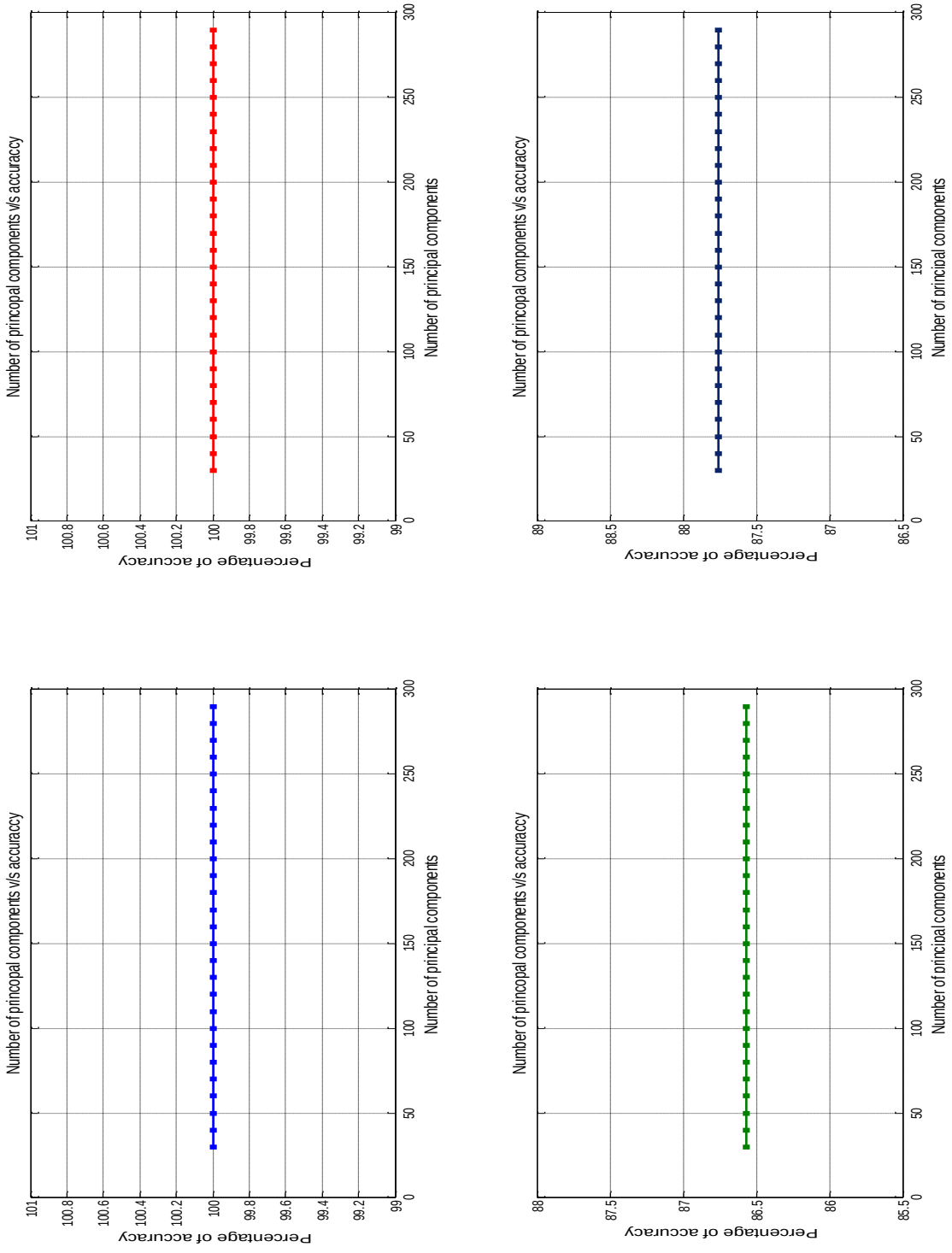


Figure 3.27: Graphs showing accuracy in percentage against the number of Principle Components for Set 2.

3.10.7.3 Test Result from Procedure 2

In this test procedure, the database has been divided into four different Subsets as in procedure 1. Subset 1 has six images used for training and the Test Set has five images irrespective of the change in the number of training images. Subset 2 has four images, Subset 3 has used two sample images for training the system and Subset 4 used only one training image to train the algorithm which yielded a 96% of accuracy.

Table 3.30: Shows the different images used for training and testing the Yale Face database in different Subsets.

	Images used for Training	**Images used for Testing**	**Max Accuracy achieved in %**
Subset 1	1,3,5,8,9,11	2,4,6,7,10	100
Subset 2	5,8,9,11		100
Subset 3	5,8		88
Subset 4	5		96

Table 3.31 tabulates the result obtained by running the algorithm with varying number of principle component values and the percentage of accuracy achieved by the algorithm along with the average time taken by each facial recognition process while keeping the training set constant.

Table 3.31: Results from Procedure 2 on Yale Face database.

Test Set	No of images used for Testing	No of images used for Training	Number of principal components	Recognition rate %	*Average time taken per recognition (sec)
Subset 1	5	6	50	100	0.2681
			100	100	0.2689
			150	100	0.2722
			200	100	0.2736
			250	100	0.2726
Subset 2		4	50	100	0.2736
			100	100	0.2710
			150	100	0.2716
			200	100	0.2754
			250	100	0.2754
Subset 3		2	50	88	0.3086
			100	88	0.3063
			150	88	0.3142
			200	88	0.3117
			250	88	0.3134
Subset 4		1	50	96	0.2800
			100	96	0.2813
			150	96	0.2818
			200	96	0.2853
			250	96	0.2892

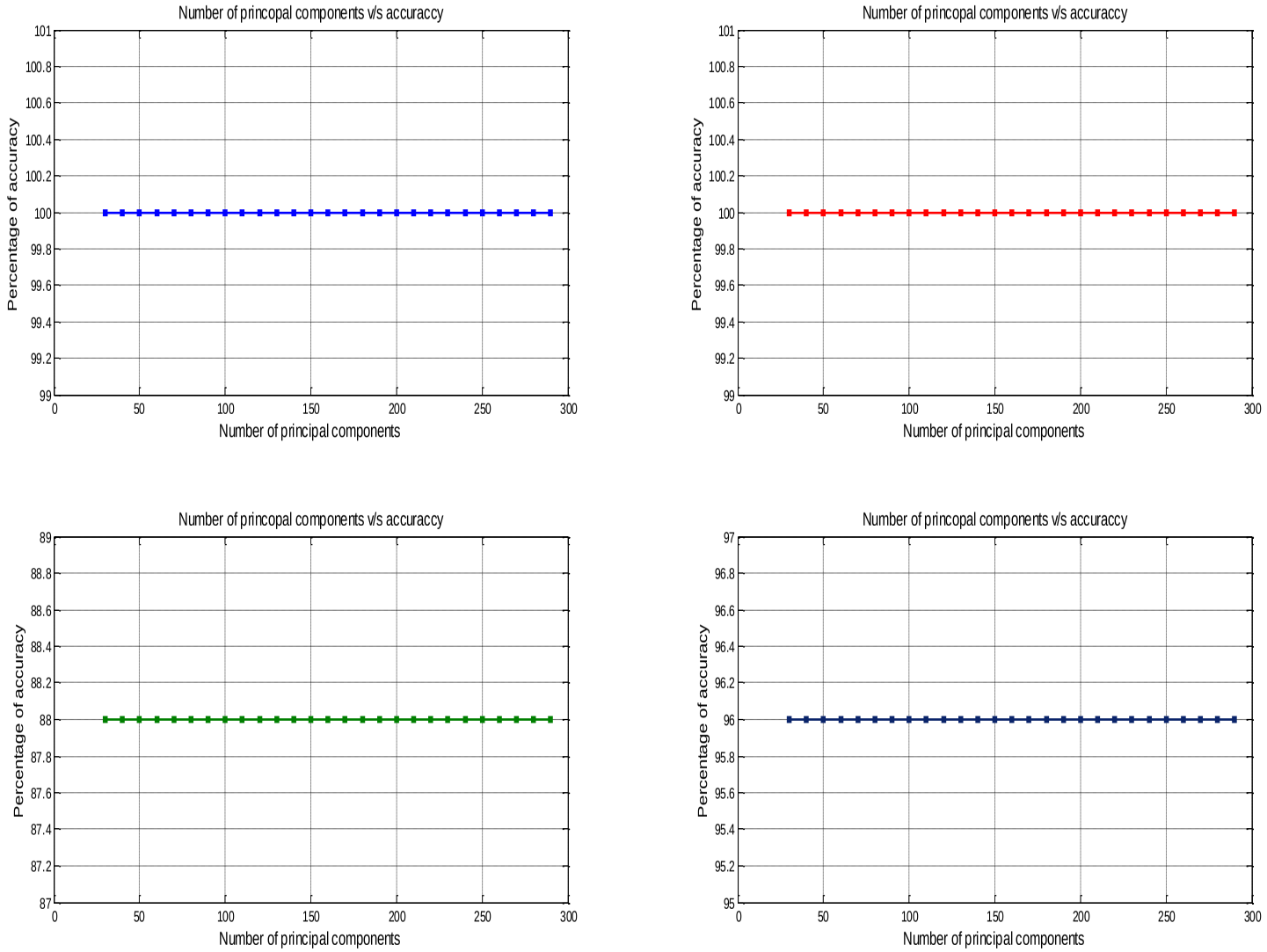


Figure 3.28: : Graphs showing accuracy in percentage against the number of Principle Components for Set 2.

3.11 Conclusion and Discussions

The performance of the proposed algorithm has outperformed on all the databases we have tested. Table 3.32 has summarized some of the top results archived with various different algorithms and the accuracy results obtained.

Table 3.32: Tablets the results obtained for facial recognition algorithms on different databases and compared it to SFRLBP

Name of database	Highest Accuracy achieved with the proposed technique (SFRLBP)	Highest recorded accuracy	Method used to get the result in column 3
Faces 94	100%	99.2325	*NCFS [46]
Faces 95	99.5381%	94%	**LDA [47]
Faces 96	97.3706%	81%	***PCA [48]
JAFFE	100%	100%	NCFS [46]
YALE	100%	97.7%	LDA [47]

*Nonlinear Curvelet feature Subspace [46]
**Linear Discriminate Analysis [50]
***Principle Component Analysis

The accuracy for the proposed algorithm *Sparse face recognition with LBP* (SFRLBP) depends on many factors. First of all larger the training set images higher will be the accuracy. SFRLBP has a higher tolerance when trained with smaller number of images too. For example the result obtained for Yale database is 100% with SFRLBP but with Discriminative common Vector its 97.33% [49]. Some other methods were used to test the recognition accuracy on Yale Face Database and they are as tabulated in Table 3.33 [47] [49].

Table 3.33: Tabulates results for face recognition with different listed algorithms and the accuracy obtained for Yale face database.

Test Method	**Accuracy in %**
Eigenfaces[43] [51]	76
Fisherface [51]	96
Direct-LDA [50]	92
Discriminative Common Vector [49]	97.33
PCA [49]	86.6
Independent component analysis [49] [52]	73.3
Gabor SVM [49] [53]	88.8
LBP SVM [49]	97.8
LDA [49]	97.7

All these methods used 70% of the image for training the system and 30% of images were used for testing yet 100% accuracy were not achieved. 100% accuracy is archived only when 37% images are used for training and the rest used for testing purpose with SFRLBP.

Another very important factor that comes into play for increasing the accuracy is to designing and selecting the training set images. During one of the experiments with PUT Vein Pattern Database the SFRLBP algorithm was trained with only one image sample and the image used was a out of plane facial image, refer Figure 3.29 (a)

a

b

Figure 3.29: Shows two sample images from PUT Vein Pattern Database

and the accuracy archived was 57.65% now the SFRLBP algorithm was trained again with a in plane facial image refer Figure 3.29 (b). The accuracy achieved this time is 87.4%. In both cases the test set included rest of the 21 images in the dataset. We can see that designing the training database can make a huge difference in the result.

Chapter 4: Hardware System

4.1 Objective

The main objective for this chapter is to design a Hardware Graphical User Interface (GUI) and a light optimization program for the camera to assure proper exposure. In addition, our reasons for choosing the main processing board will be explained which will be running the algorithms mentioned in the previous chapter.

4.2 System Overview

Our image processing system is a real-life example of image processing algorithms in hardware systems. We are endeavoring to implement a ready-to-install face detection and recognition system for a car. The objective of this system is to authenticate the driver before allowing him/her to drive. Moreover, this device will allow the user to update the system's database to enable more than one authentic driver. As a result, only authenticated drivers, those drivers already in the system database, will be able to drive the car. Installing this system in a car will help consumers in two ways:

1. It will dramatically decrease the chance that their car will be stolen.
2. It will enable only an authorized driver to automatically start their car.

The face recognition system will also help to determine if the driver is holding a G2 or a G driver's license, to assess the blood alcohol level of the operator and to grant permission for only an authorized driver to start and drive the car.

As mentioned in the previous chapter, the image processing system was first designed on a desktop computer and all the algorithms were implemented using Matlab. However, because this was not the proper size or procedure to fit the system in an actual car, the whole system needed to be miniaturized. The miniaturization not only reduced the dimension of the system but also reduced the processor speed and amount of memory available. Thus, with the reduction of the processor's speed, we found many constraints that made a significant difference in the programming. All the algorithms had to be re-written in C++, and to make the system real-time, the program had to be optimized accordingly. Due to the absence of a monitor, mouse or keyboard, a GUI (Graphical User Interface) was designed using electrical components to reduce complexity. To design this GUI, a simple AtMega 32 microcontroller was used.

This hardware GUI receives input through multiple control switches and displays the options and outputs on a 20 x 4 alphanumeric LCD (Liquid Cristal Display) mounted with the GUI. The input instructions are sent to the main board where the facial recognition programs are executed. The computer module is a small board which runs on customized Windows Compact 7. The module has an 800 MHz single core x86 based processor and 512 MB of RAM. The module is equipped with a 16 GB Class 10 SD card for the installation of the OS. With the customization of Windows Compact 7, the

system ran as expected; however, the execution speed of the algorithm was hindered because of the lower processing speed. Figure 4.1 shows the major 3 blocks for the whole system which include:

- The camera module
- The main processing board
- GUI and IR intensity level controller

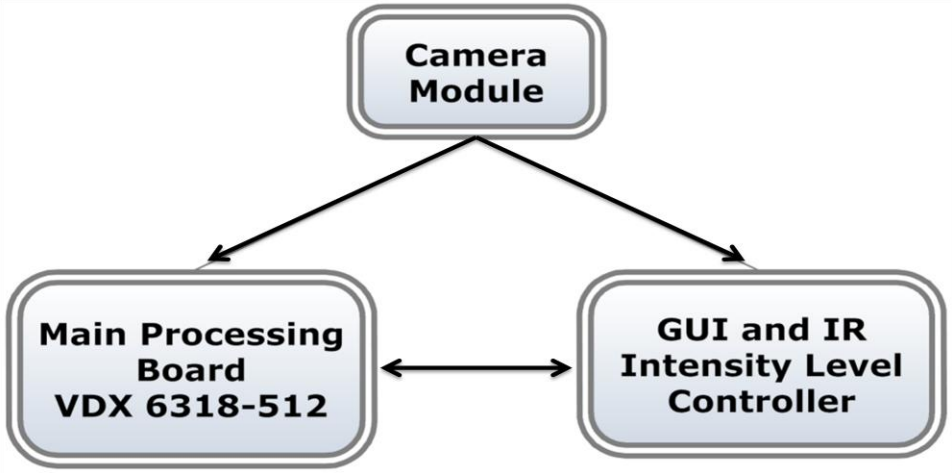


Figure 4.1: The above figure shows the major three blocks of the proposed system.

4.3 Camera Module

One of the major challenges for our design was that our device needed to perform in all kinds of real-life lighting conditions. It needed to operate properly during day as well as during the night. Therefore, we required a camera capable of taking pictures in both day and night situations. Consequently, we decided to use an infrared camera capable of taking pictures both during the day and the night. We used an array of Infra-Red LED on the side of the camera which acted as a light source when there was absence of visible

light. Presently, there are various types of cameras available to meet our needs. Following, are some of the options we are exploring before deciding upon one:

4.3.1 CMOS vs. CCD sensors

Digital cameras use mostly solid state sensors which contain millions of photosensitive diodes called “photosites”. When a sensor is exposed to light, each of these photosites is uncovered to collect and store photons or light. As soon as the exposure is complete, the camera begins calculating how many photons were captured by each of these photosites. In other words, these sites will integrate the intensity of the light falling on the diodes by accumulating the charge during the brief moment of the exposure. A CCD (Charged Coupled Device) utilizes ADC or Analog to Digital Convertor to convert the charge into a digital number. The charges are coupled in such a way that only one row can be read at a time which makes the system complex for enhancements. CMOS (Complementary Metal-Oxide Semiconductor) sensors use photo transistors instead of diodes which in turn, makes them more costly but efficient.

Each of these photo transistors is utilized to read each pixel value individually. Each of these pixels has its own charge to voltage convertors and other enchantment circuitry like amplification, noise-correction, digitalization circuits, etc., so the chip outputs in a digital bus making it much more complex, reducing effective area to capture light and making the system non-uniform. However, additional chip features can be added at very little extra cost, which may include image stabilization, auto gain correction, white balance etc. All these integrated features within the chip make the camera smaller and

lighter and cheaper to produce. Indeed, CMOS sensors consume very low power and are faster (they can capture 10,000 frames pre sec) which makes them suitable for embedded system designs and low power devices. Unfortunately, integrating all these features in a CCD camera will make the manufacturing process very complex and costly.

The percentage of a pixel devoted to collect light is called the pixel fill factor which has a direct correlation with the sensitivity of the sensor and is inversely correlated with the exposure time. CCD's have a 100% fill factor, but CMOS cameras have a much less fill factor. To increase the fill factor for CMOS sensors, micro-lenses are integrated into the sensor packages. Both the CMOS and CCD sensors have almost the same response to IR (Infra Red) regions, so either of the two can be used for infrared imaging.

With these advantages, CMOS image sensors are emerging as a complementary solution to CCDs. Their present applications include internet camera, digital still camera, machine vision, automotive, children's toy, medicine and dentistry, fingerprint ID, surveillance, aerospace, motion analysis, industrial inspection, quality control, process control, target tracking, and spectroscopy. We have used a standard USB CMOS camera but it has been hacked and customized as per our own need.

4.3.2 The Camera Setup

As mentioned before we have used a normal USB camera, but the IR filter in the camera has been removed so that it can work in the IR spectrum. As shown in Figure 4.2, we have arranged an array of IR LED around the camera so that when there is not enough

light for a proper exposure, these LED's can act as a light source. Figure 4.3 shows some of the pictures taken by the camera in different lighting conditions.

Going clockwise from the top left, the first picture was taken when the camera was directly exposed to sunlight, and the second picture was taken in a normal light exposure. In both cases, IR LED's did not light up because there was sufficient light. The third image was taken in total darkness, and the only available light source was the IR LED. For the fourth and final picture, there was some ambient light but not enough to have a proper exposure, so the IR LRD's lit up along with the available surrounding light.

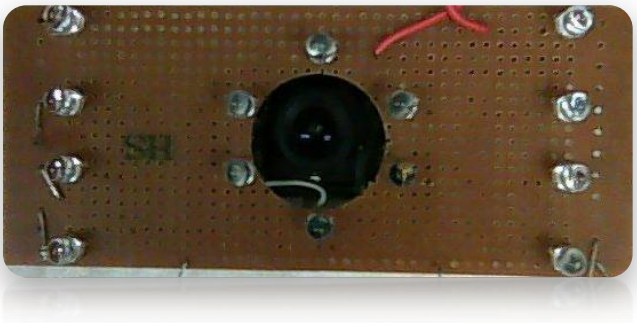


Figure 4.2: Shows the camera module with the IR LED's mounted to it.

Figure 4.3: Shows images captured in different lighting conditions.

4.4 GUI and IR Intensity level Controller

This unit is the control panel for the main face recognition program; it includes the IR intensity level controller.

The user interface or the GUI is a menu driven interface. The system takes input from the user and sends command back to the main board so that the software on the main board can run certain algorithms to perform the assigned task.

The LCD and the key board are connected to a controller. When the user generates an event on the keyboard, the controller sends an appropriate command to the main board and displays a certain set of output on the LCD so that the user can understand what is happening with the system.

4.4.1 System Configuration

The system consists of an AtMega32L microcontroller, a 20 x 4 alphanumerical LCD to display all the options, configuration and control, 12 keys in the key board with one

reset, one enter and numerical keys from 0 to 9, an 8KB ROM device (24C64) for storing the configuration data. A MAX232 has been used as a buffer device to convert the TTL signals into a RS232 voltage level signal. A RS232 signal is fed to the main processing board instructing it as to what operation needs to perform. A PWM controller is used to control the intensity of the IR LED array so that the camera takes the image with proper exposure depending on the ambient lighting condition.

To monitor the ambient light, a light sensor has been connected to the Analog to Digital Converter (ADC) of the controller. Since the light sensor gives out a current output, an OP07 OPAMP has been employed to design a current to voltage convertor; the output is fed into the ADC of the microcontroller. A block diagram showing the main blocks is displayed in Figure 4.4 and a picture of the actual board has been displayed in Figure 4.5. Table 4.1 lists all the components on the board marked in Figure 4.5. Each and every block has been described in the following chapters and shown in Figure 4.4.

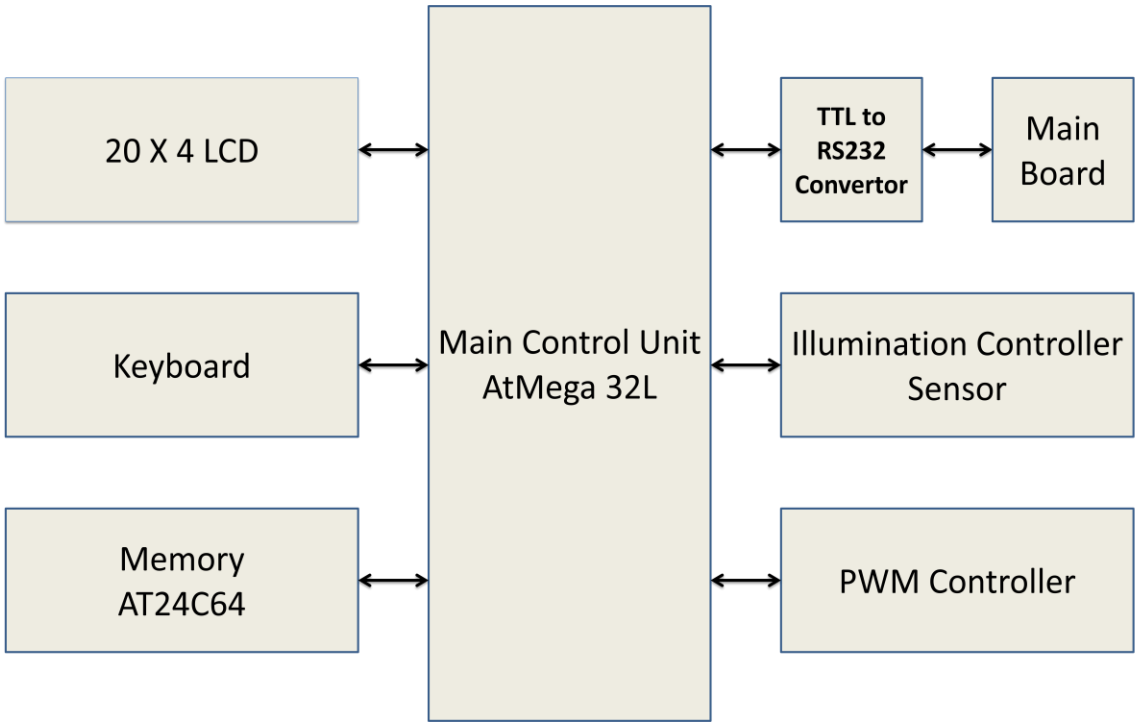


Figure 4.4: This diagram shows the basic blocks of the GUI and the IR intensity controller.

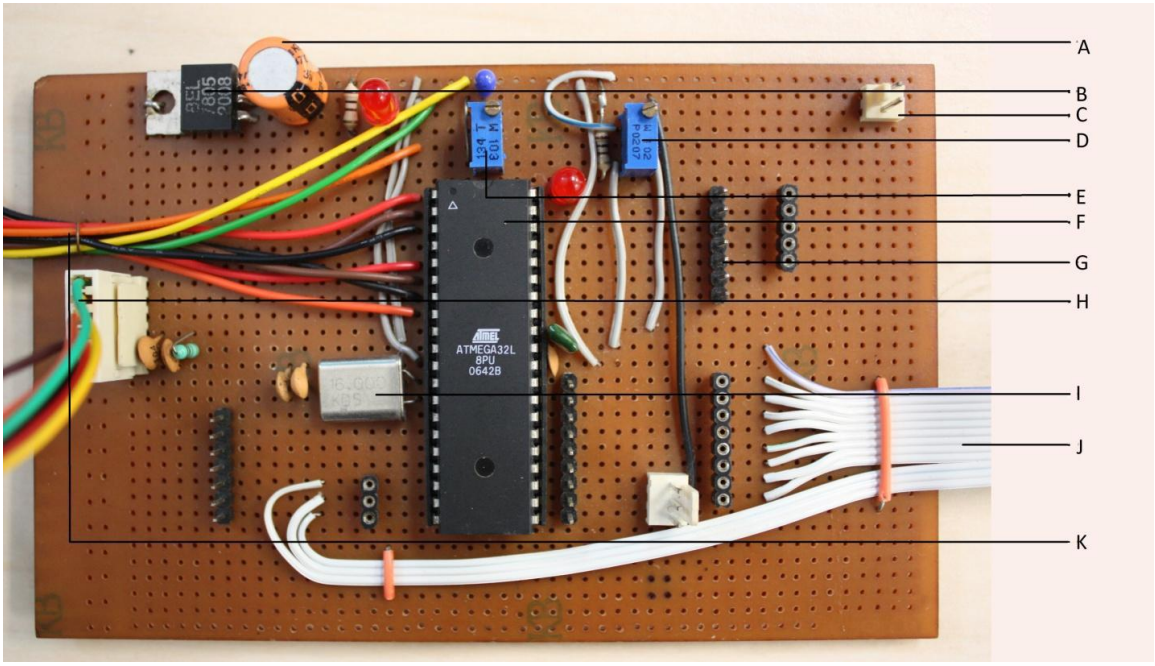


Figure 4.5: This picture shows the main board.

Table 4.1: lists all the components on the board marked in Figure 4.5.

A	Noise reduction and surge protection capacitor
B	Voltage regulator IC
C	Power connector
D	ADC reference voltage adjuster
E	LCD contrast adjust
F	ATMEGA 32 controller
G	Sensor connector
H	Programmer Connector
I	Crystal
J	Keyboard Cable
K	LCD Cables

Noise reduction and surge protection capacitor

Many DC circuits encounter various types of power surges due to external or internal forces acting on the electrical system. Lightning, short circuits and component failures are the most common causes of power surges. Capacitors can be installed on DC circuits to help reduce the impact of a power surge. This helps to protect the equipment and reduce repair and replacement costs. Installing a capacitor on a DC circuit requires knowledge of the size of power surges that the system might encounter.

Voltage regulator IC

A voltage regulator IC has been used to bring down the higher supplied voltage to the required voltage level. A 7805 Voltage level IC has been used for this purpose. These ICs do not require any external component, so they are very economical for board space and price. These 70XX series ICs have built-in protection against a circuit drawing too much power. They have protection against overheating and against short-circuits, which makes them very robust for low power electronic circuit design.

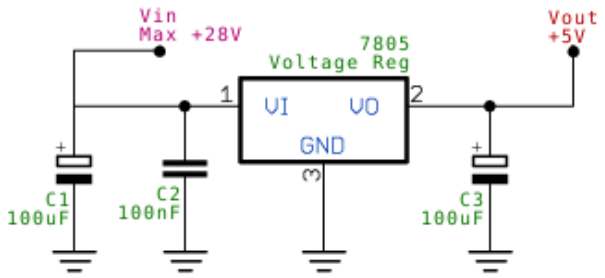


Figure 4.6: A simple circuit diagram for 7805 has been displayed in the above figure.

Power connector

A power connector has been used so that the board can be easily powered from an external power source.

ADC reference voltage adjuster

An analog voltage is given as input to the ADC which must be greater than $0V$, and smaller than the ADC's reference voltage $Aref$. The reference voltage is an external voltage that is supplied at the $Aref$ pin of the AtMega 32 chip. The voltage at the input is converted to a digital value and can be calculated with the following formula:

$$ADC\ conversion\ value\ =\ round\left(\left(\frac{v_{in}}{v_{ref}}\right)\times 1023\right)$$

Since AtMega 32 has a 10-bit ADC, we can have 1024 possible output values (from 0 to 1023). So, if v_{ADC} is equal to 0V, the result of the conversion will be 0, if v_{ADC} is equal to v_{ref}, it will be 1023, and if v_{ADC} is equal to $v_{ref}/2$ it will be 511. As we are converting an analog value, it can have infinite possible values at the output; we have only a few

finite numbers of possible values called *discrete variable*, and this ADC conversion process produces an error, known as *quantization error*.

LCD contrast adjust

An LCD contrast adjuster is used to adjust the brightness vs. the contrast of the 20 x 4 LCD display used for the design.

ATMEGA 32 controller

This part has been discussed in one of the later chapter in details..

Sensor connector

Sensor connectors are used to easily plug the light sensor (BPW34) with the board.

Programmer Connector

This connecter is used to connect the programmer to the board. A programmer is a device which is used to download the program from the computer to the microcontroller.

Crystal

An AtMega 32 controller needs a clock source. It can use the internal RC (Resistance and Capacitance) oscillator to generate the clock. With prolonged usage and heating up of the controller, there is a change in the value of resistance and capacitance which are used to design the oscillator. With the change in these values, the clock frequency changes too, which makes it difficult for the controller to communicate through UART

because it needs very precise timing and causes erroneous result. Consequently, an external clock source has been utilized so that it can keep a precise timing.

Keyboard Cable

The keyboard cable connects the keyboard with the controller's input ports.

LCD Cables

An LCD cable connects the LCD with the input/output port of the controller.

4.4.2 20 x 4 LCD

The LCD used for this project uses a Hitachi HD44780 LCD controller. It is one of the most common and widely used dot matrix liquid crystal display (LCD) controller available. The controller in this LCD was designed to interface directly to a microprocessor or a controller. The device can display ASCII (American Standard Code for Information Exchange) characters, Japanese characters and other special characters.

These LCD screens are mostly limited to single color text and are often used in embedded system designs such as those in copiers, fax machines, laser printers, industrial test equipment and networking equipment such as routers and storage devices etc. The screens come in a small number of standard configurations. Common sizes are 8x1 (one row of eight characters), 16×2, 20×2 and 20×4. Larger custom sizes are made with 32, 40 and 80 characters and with 1, 2, 4 or 8 lines. The most commonly manufactured larger configurations are 40×4 characters, which require two individually addressable HD44780 controllers with expansion chips because a single HD44780 chip can only address up to 80 characters. A common smaller size is 16×2; this size is readily

available as surplus stock for hobbyist and prototyping work. Character LCDs can come with or without backlights which may be LED, fluorescent, or electroluminescent. Character LCDs use a standard 16 contact interface, commonly using pins or card edge connections on 0.1 inch / 2.54mm centers. Those without backlights may have only 14 pins, omitting the final two pins powering the light.

The one used for this project is a 20×4 character LCD and is connected to the Controller to show the options in a menu driven format to control the face recognition program. A picture and the circuit diagram for this LCD is shown in Figure 4.7and 4.8; Table 4.2 lists which pin is used for what purpose for a 20 x 4 LCD.

Figure 4.7: A 20×4 LCD display module.

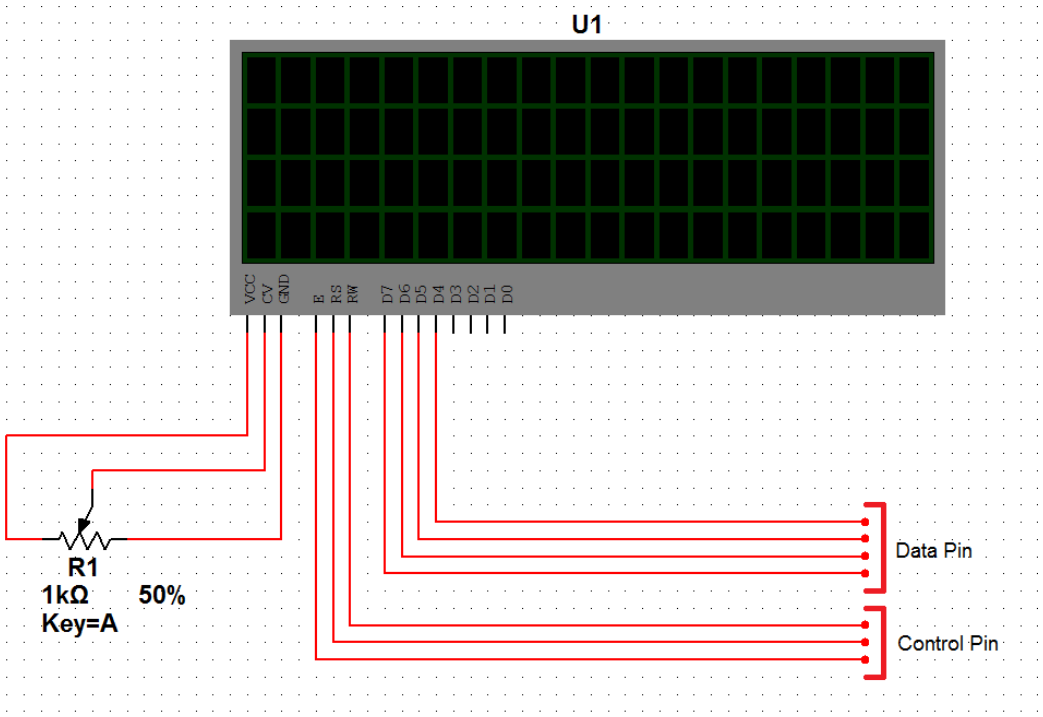


Figure 4.8: Circuit diagram showing the basic wiring for a LCD module.

Table 4.2: Lists the function of each pin in a 20 x 4 LCD.

Pin 1	*Ground*
Pin 2	*V_{CC} (+3.3 to +5V)*
Pin 3	*Contrast adjustment (VO)*
Pin 4	*Register Select (RS). RS=0: Command, RS=1: Data*
Pin 5	*Read/Write (R/W). R/W=0: Write, R/W=1: Read*
Pin 6	*Clock (Enable). Falling edge triggered*
Pin 7	*Bit 0 (Not used in 4-bit operation)*
Pin 8	*Bit 1 (Not used in 4-bit operation)*
Pin 9	*Bit 2 (Not used in 4-bit operation)*
Pin 10	*Bit 3 (Not used in 4-bit operation)*
Pin 11	*Bit 4*
Pin 12	*Bit 5*
Pin 13	*Bit 6*
Pin 14	*Bit 7*
Pin 15	*Backlight Anode (+)*
Pin 16	*Backlight Cathode (-)*

The HD44780 interface allows for two modes of operation, 8-bit and 4-bit. Using the 4-bit mode is more complex, but reduces the number of active connections needed. The chip starts in an 8-bit mode, with the instruction set designed to allow switching without requiring the lower four data pins. Once in 4-bit mode, character and control data is transferred as pairs of 4-bit "nibbles" on the upper data pins, D4-D7. We have used the 4-bit mode operation for this design as it reduces the number of pins used for the LCD operation and makes the hardware design more flexible.

4.4.3 Keyboard

The numeric keyboard consists of twelve keys. It has an Enter key and a Reset key. The rest of the keys are mainly the numeric keys consisting of numbers from 0 to 9. Each of these keys has two terminals, one of them is connected to ground and the other one is connected to the input/output port of the microcontroller. A picture of the keyboard is shown below in Figure 4.9.

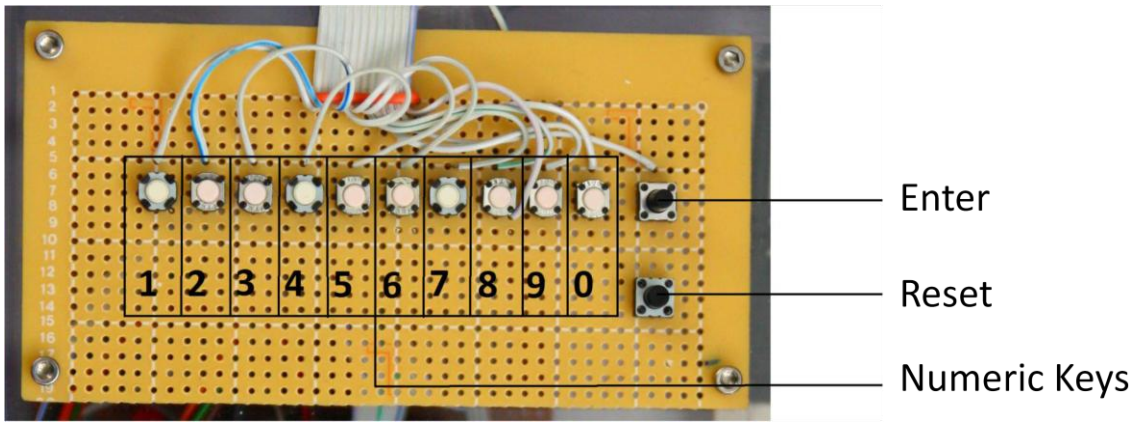


Figure 4.9: A picture of the keyboard used for the GUI Design.

4.4.4 AT24C64

This device uses a serial (I2C) ROM device from Atmel Corporation to store configuration data and other information such as the user database. The user database consists of the name of the driver matched with the label provided (the numeric value). The configuration settings consist of the number of users in the data base, a log for who was using the car and the time the car was used.

The AT24C64 provides 65,536 bits of serial Electrically Erasable and Programmable Read-Only Memory (EEPROM) organized as 8192 words of 8 bits each. The device's cascadable feature allows up to 8 devices to share a common 2-wire bus. The device is optimized for use in many industrial and commercial applications where low power and low voltage operation are essential. The AT24C64 is available in space saving 8-pin PDIP, packages and is accessed via a 2-wire serial interface. In addition, the entire family is available in 2.7V (2.7V to 5.5V) and 1.8V (1.8V to 5.5V) versions. Figure 4.10 shows the pin out of a 24C64.

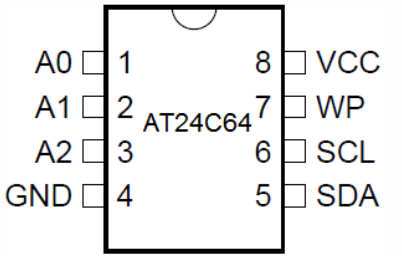


Figure 4.10: Displays the pin out of a AT24C64 EEPROM.

4.4.5 TTL to RS232 Convertor

The microcontroller AtMega 32 has a built-in UART, but the output voltage level is in TTL level. The main board uses a conventional RS232 voltage level for UART, so we needed a TTL to RS232 and RS232 to TTL convertor. For this purpose a MAX232 IC has been used. A brief circuit diagram has been shown in Figure 4.11

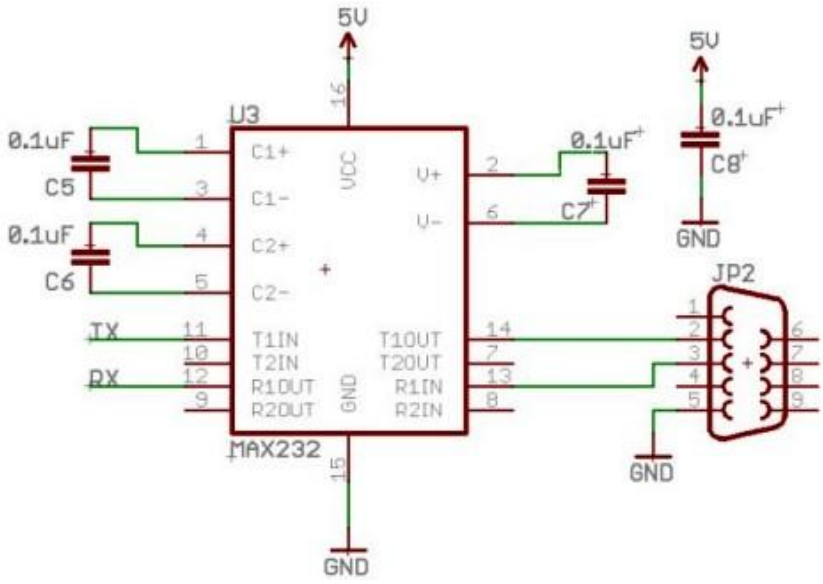


Figure 4.11: Shows the pin out and the basic circuit diagram for a MAX 232.

TX and the RX pin on the left of the diagram are connected to the microcontroller and the JP2 is a DB9 port which is connected to the main controller board.

4.4.5.1 TTL and RS232

A Transistor Logic or TTL is a digital voltage level mainly designed for Bipolar Junction Transistors (BJT). For TTL logic level, $+5V$ is considered as logic 1 and $0V$ is considered as logic 0. TTL logic is used for logic communication and data transfer.

RS232 is more of a communication protocol and is used for long distant communication (25 meters). The logic level is 0 when the voltage lies anywhere between $3V$ to $25V$, and the logic level is 1 when the voltage level is anywhere between $-3V$ to $-25V$.

4.4.6 Illumination Control Sensor

As a part of a real-life project, this device should work during day, night and other challenging lighting conditions. Visible light during the night might cause disturbance to the passengers in the car, so with that in mind, IR (Infra Red) LED's (Light Emitting Diodes) have been used for illumination.

To understand the ambient lighting condition, a photo sensor has been used to get the intensity of the ambient light near the camera. The main objective is to have an understanding of the surrounding lighting condition and determine if light balancing algorithm should be used so that we can have a proper exposure of the image. A good exposure of the image helps in better performance of the image processing algorithms. This sensor has been used to derive a threshold value so that the intensity of the IR LED's can be controlled using PWM (Pulse Width Modulation) to get a proper exposure.

BPW34 is a PIN photodiode with high speed and high radiant sensitivity in miniature, flat, top view, clear plastic package. It is sensitive to visible and near infrared radiation. It has a very fast response time and angle of half sensitivity is φ = ± 65°

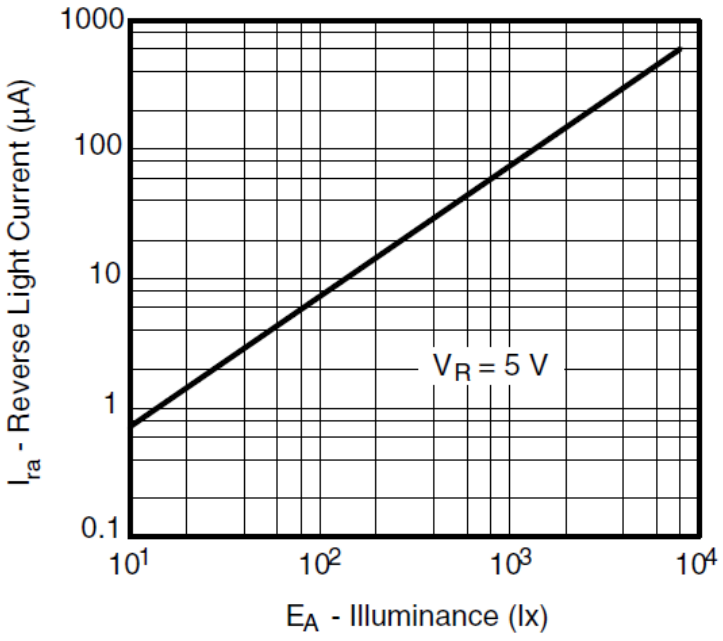


Figure 4.12: This graph shows the illumination verses current output in μA for BPW34 sensor.

Figure 4.13: The above picture shows the BPW34 sensor.

BPW34 can be used in either zero bias or reverse bias mode. Diodes have extremely high resistance when reverse biased. This resistance is reduced when light of an appropriate frequency shines on the junction. Hence, a reverse biased diode can be used as a light detector by monitoring the current running through it. Coupled to a

10Kohm resistor, and given the specification of the BPW34, a simple relationship between lux (light intensity) and voltage is given by

$$lux = 1333 \times Vo$$

With the Data Acquisition Module (DAQ module), a simple microcontroller can be used to measure this voltage. The DAQ module contains a 10-bit ADC (analog to digital converter). During analog to digital conversion, a digital value can correspond to a range of analog values. Any analog signal within the zone of one Least Significant Bit (LSB) will have the same digital value. This error is known as quantization error. The relationship between this error and the bit resolution is given by

$Error = \left(\frac{1}{2}\right)^n$ where n is the resolution in bits of the ADC

For a 10-bit ADC operating over a 5V range, the accuracy we obtain cannot be better than 5/1024 V or 4.8 mV.

Because the sensor gives out a linear current output with the change in intensity of light falling on it, we designed a current to voltage convertor to interface it with a 10 bit ADC.

While designing the current to voltage convertor, a OP07 opamp was used to get better output results. This opamp has a wide input voltage range of ±13 V minimum combined with high CMRR of 106 dB, and high input impedance provides high accuracy in the non-inverting circuit configuration. Excellent linearity and gain accuracy can be maintained even at high closed-loop gains. The accuracy and stability of the OP07, even at high gain,

combined with the freedom from external nulling, have made the OP07 an excellent choice to use in this project.

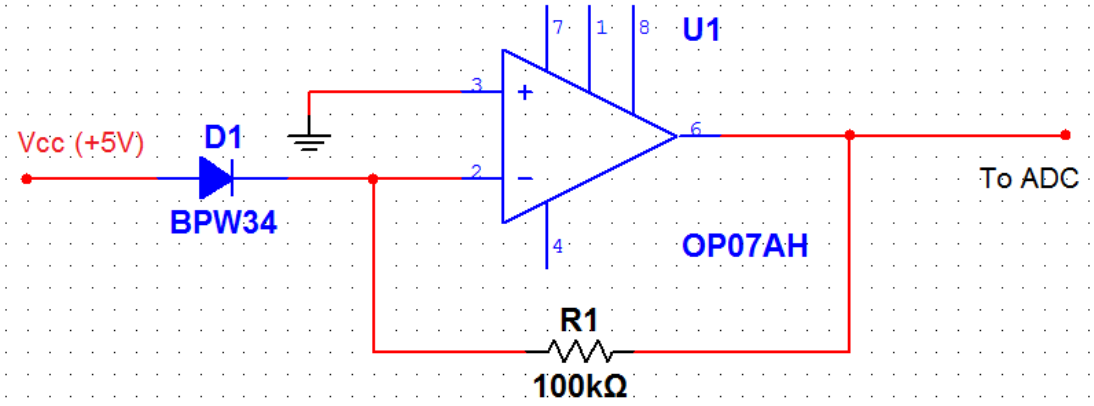


Figure 4.14: In this diagram we have a simple design showing a current to voltage convertor with D1 as the BPW34.

The output from this opamp was fed to a 10-bit ADC. We used AtMega 32 microcontroller for this purpose and used the in-built ADC of the microcontroller.

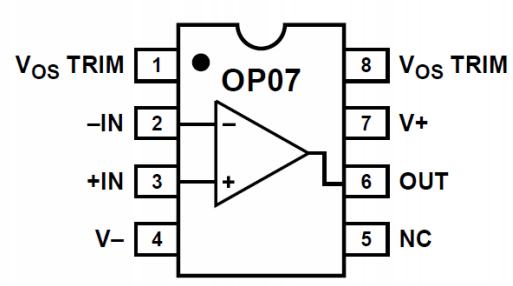


Figure 4.15: Shows the pin out for the opamp.

Figure 4.14 shows a simple circuit diagram used for conversion of the current to voltage, and Figure 4.15 shoes the pin out of the opamp OP07.

4.4.7 Pulse Width Modulation (PWM) Controller

Pulse width modulation (PWM) is generally a rectangular pulse wave, and the average power of the waveform is changed by modulating the pulse width which is also commonly known as duty cycle. If $f(t)$ is the pulse waveform with the lowest value as y_{min} and the highest value as y_{max} and the duty cycle as D, then the average value of the PWM waveform is given by:

$$\bar{y} = \frac{1}{T}\left(\int_0^{DT} y_{max}dt + \int_{DT}^{T} y_{min}dt\right)$$

4.4.8 AtMega 32L

The high-performance, low-power Atmel 8-bit AVR RISC-based microcontroller combines 32KB of programmable flash memory, 2KB SRAM, 1KB EEPROM, an 8-channel 10-bit A/D converter, and a JTAG interface for on-chip debugging. The device supports throughput of 16 MIPS at 16 MHz and operates between 4.5-5.5 volts.

By executing instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed. Figure 4.16 shows the pin out of the controller, and Figure 4.17 shows the basic circuit diagram for AtMega 32.

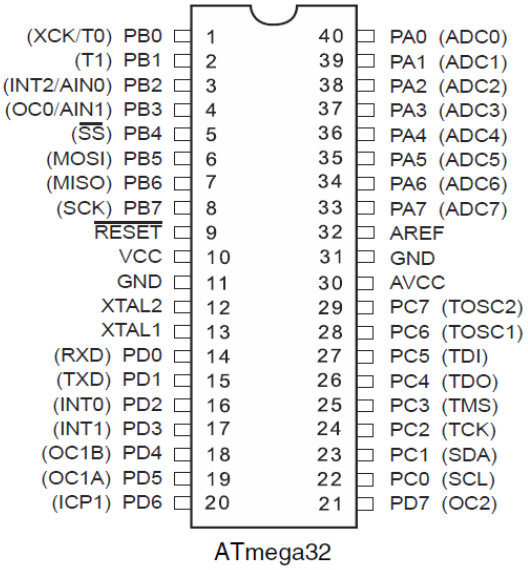


Figure 4.16: Pin diagram of an AtMega 32 controller.

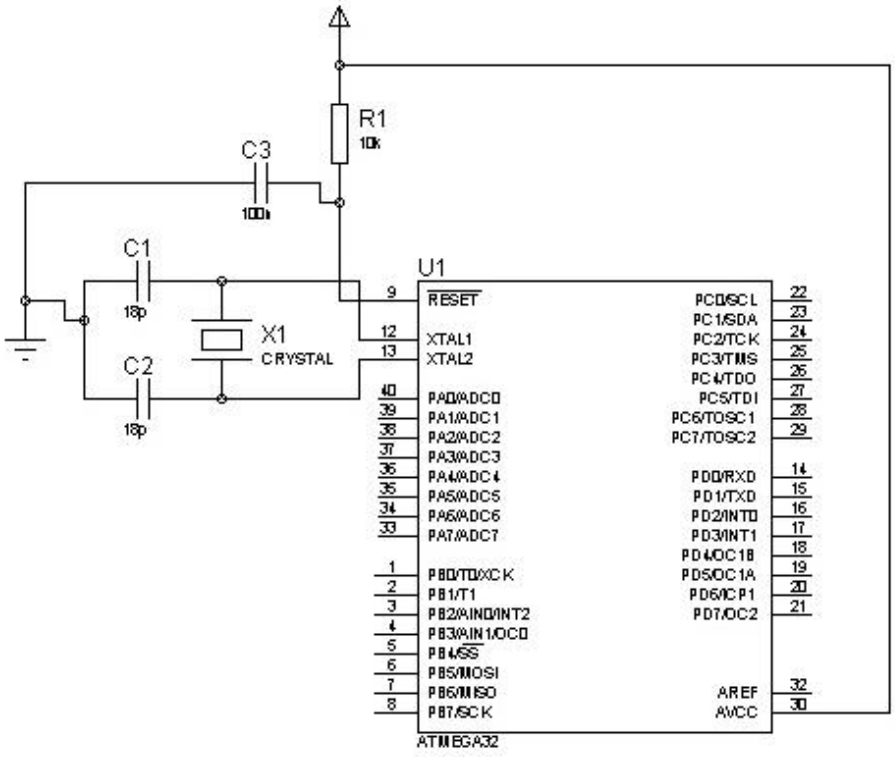


Figure 4.17: The basic circuit diagram of an AtMega 32 controller.

4.4.9 Programming the Hardware

The program for this device has been written in *Embedded C* using Code Vision AVR compiler. This compiler converts the C code to assembly and then converts the assembly to machine level HEX code which is downloaded through a programmer to the microcontroller chip.

The interface for the program running in the system is simple and menu driven. All the menu options can be directly accessed from the numerical switches and are self explanatory.

When the device is turned on, we can see a welcome note; then, a menu with two options pops up. The flow chart on the next page shows all the options as well as the process flow of the system. The flowchart is designed in a hierarchical manner; each and every menu appears as shown in the flowchart in figure 4.18.

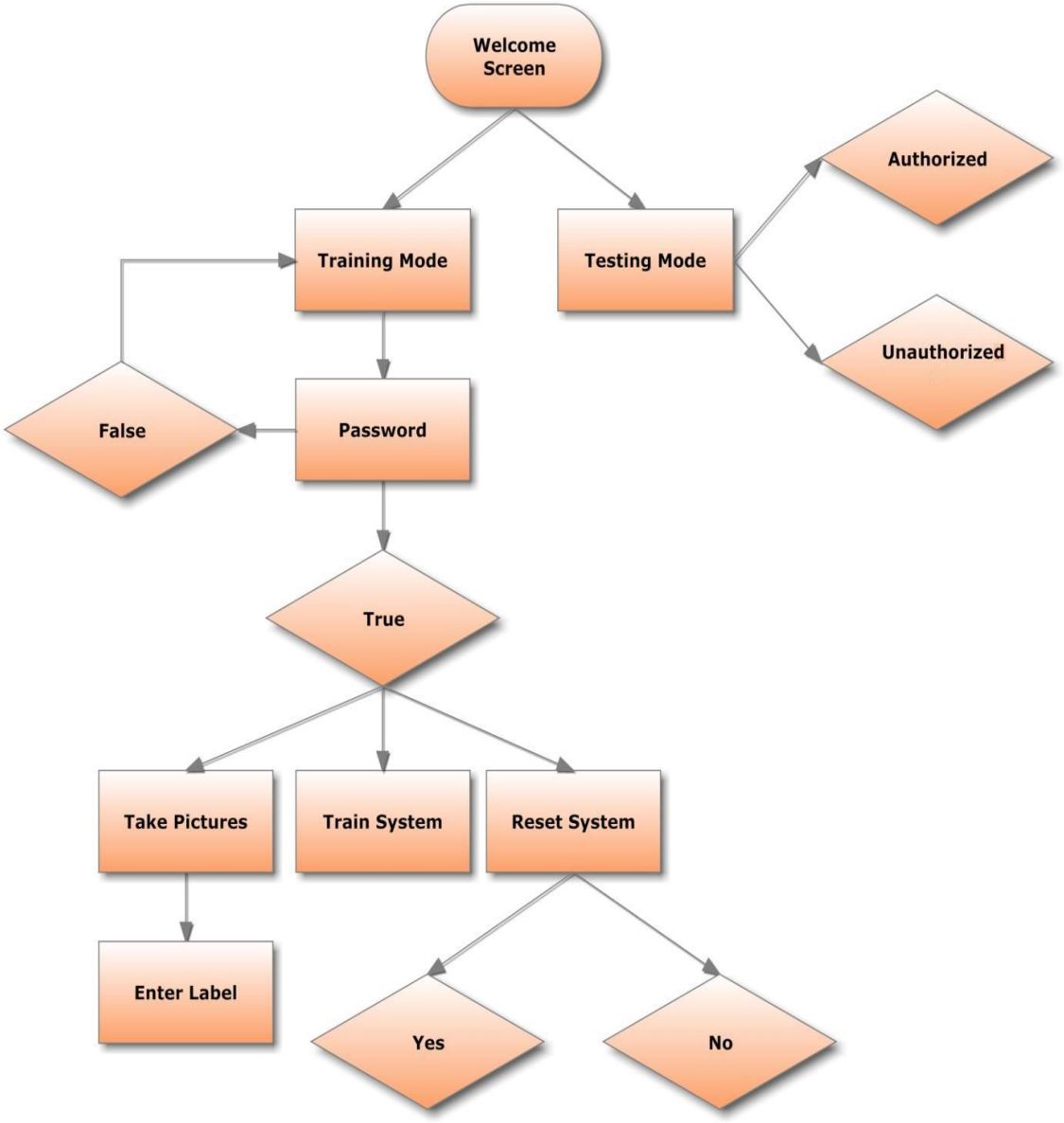


Figure 4.18: Flow diagram of the program in the Hardware GUI.

For controlling the IR LEDs for the Camera, the light sensor data (BPW34) is read from the ADC channel; it has been mapped to the PWM channel to control the intensity of the LED. The algorithm used a very simple inverse proportional mapping function. The higher the intensity of the ambient light, the lower will be the output of the IR LED; the lower the ambient light, the higher will be the intensity of the IR LED.

4.5 Main Processing Board

4.5.1 Processor selection

There are many different kinds of processors available which are specially optimized and designed for specific purposes. Selecting the correct processor for a computationally intense face detection and recognition application is of critical importance. The choice will directly influence the product cost, performance and power consumption of the system. There are many types of processors available in the market such as: PC CPU, ASIC, DSP processors, multimedia processors, FPGA and others. The selection criteria for choosing the proper type of processor for this application includes:

- Performance considerations (like speed, memory handling, bus width, data bus, energy consumption, benchmarking results, etc.
- Cost of integration.
- Availability and roadmap.
- Development consideration (like single v/s multi-core, number of IO/GPIO ports, instruction set, compatibility tools, compiler etc.)
- Packaging requirement.
- Operating temperature and robustness to shocks, among others.

Media processors like ASIPs provide better performance than most DSPs and GPPs (General Purpose Processors) and have better support for video processing; however, they have complex programming models, higher development cost, and higher associated risk because their roadmaps are unclear. FPGAs can be reconfigured

dynamically, offer architectural flexibility, high throughput and performance, all resulting in higher efficiency. However, FPGA have a very complex program structure which will require recoding the entire OpenCV library. ASSPs are often inflexible, have a sharp learning curve and require extensive tuning. Their roadmap is unclear, and the benefits of low cost could only be realized if they were produced in mass quantities. Application processors offer adequate performance, portability, energy efficiency, integration, and support for video-based applications; however they are less powerful than the other type of processors mentioned above.

Recently the advent of a new family of x86 based processors manufactured by DM&P known as Vortex86DX seems to be ideal for these algorithms to be implemented. As the processor supports x86 instruction, set OpenCV libraries can be directly used for programming the device. It is ideal as it has high clock speed, low power dissipation per unit of processing, small form factor and flexible programming.

The convergent process allows the developers to create applications in C/C++; the processor is optimized not only for computation on real-time video data but also for control task. The benefits include best utilization of existing skill sets within a team, reduction time to market and lifecycle costs, higher processing speed and easy maintenance. Thus, we have chosen the Vortex86DX as our processor platform.

4.5.2 A brief description/specification of the processor / board

The system hardware consists of a state-of-the-art processor Votex86DX on an embedded computer board by ICOP Technologies. It consists of an 800MHz Votex86DX

processor with 512MB of RAM @ 333MHz, 4MB of SPI flash Disk, an enhanced IDE port for hard drive interfacing, a four RS232 (com port), a four USB v2, 16 GPIO port and a 10/100MBPs Ethernet. Table 4.3 shows some of the features of the board. Figure 4.2 shows the picture of the board used, and Figure 4.3 shows a simple block diagram of the board.

Table 4.3: Lists some of the features of the main board.

Clock Speed	800MHz
RAM	512MB @ 333MHz
Core Voltage	0.9V ~ 1.1V
Power Requirement	Single Voltage +5V @ 740mA
Operating Temperature	-40C to+85C
Com Port	4
USB Port	4
GPIO	16
Ethernet 10/100 MB	1

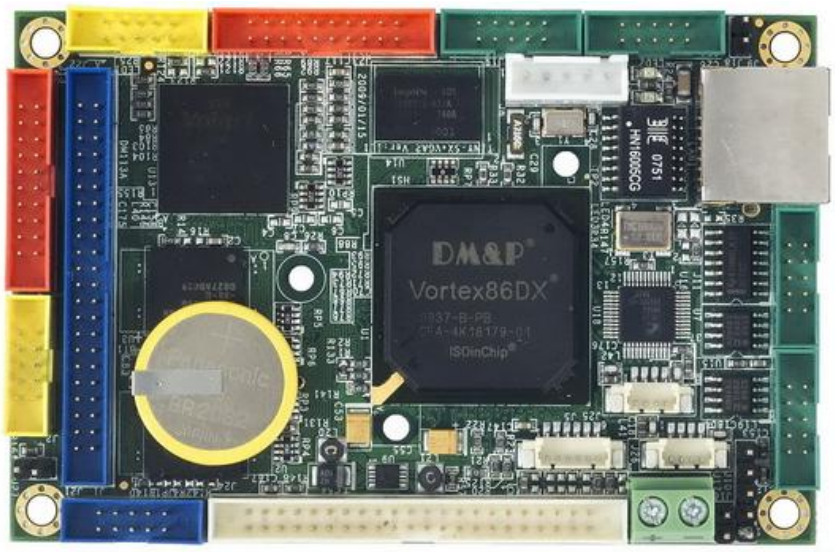


Figure 4.19: Displays a picture of the main processing board.

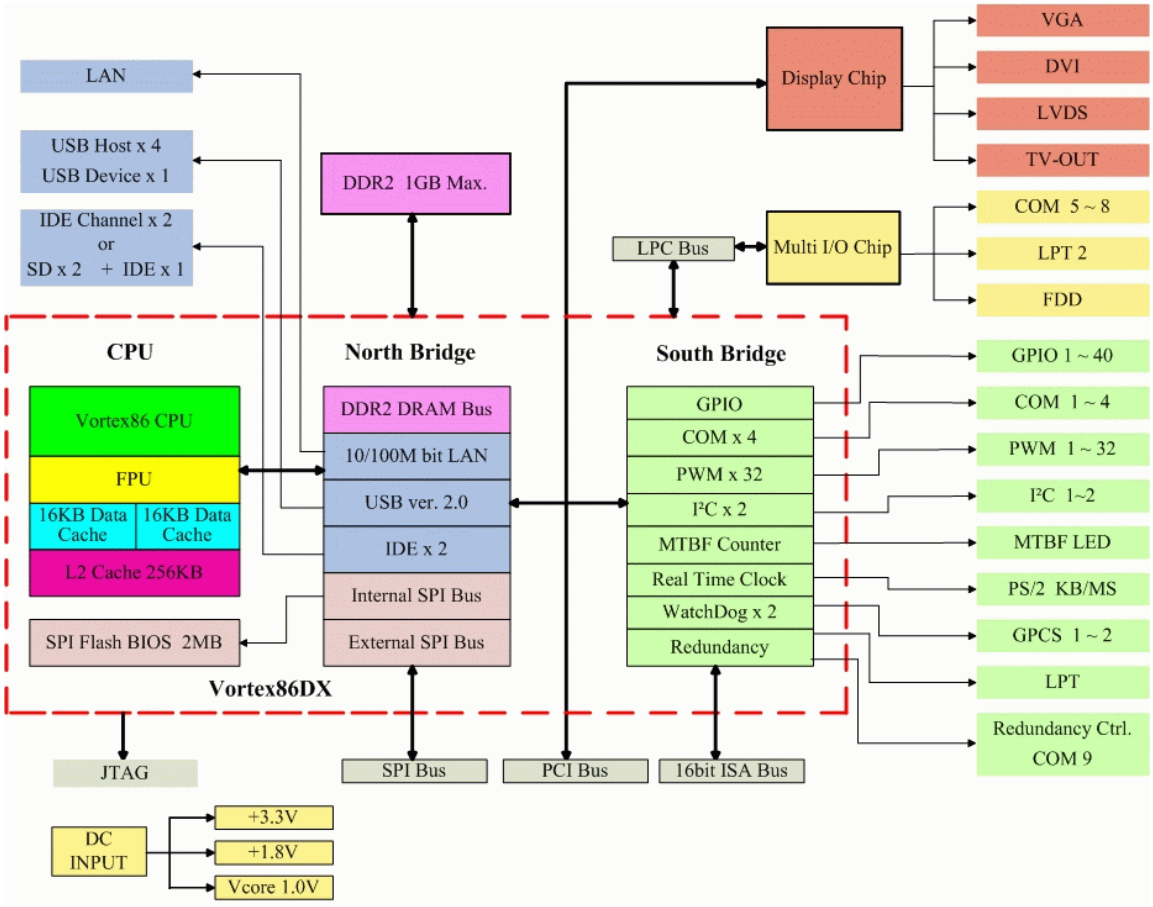


Figure 4.20: Shows the basic blocks of the main processing board.

Because of the many bundled features and single power supply, this board is ideal for this design. In addition, a small form factor with industrial-grade temperature range gives this board an upper hand over other available boards within the same price range.

4.6 Results Obtained

After implementing the whole algorithm on the above mentioned board we checked it for accuracy and speed. As observed there was no change in accuracy was observed but the speed has decreased dramatically. Average speed per recognition has been tabulated in the table below when 50 principle components has been used for facoal recognition.

Table 4.4: Time taken for face recognition in seconds per frame.

Name of the Database	Recognition time per face (Sec)
Face 94	1.6
Face 95	1.8
Face 96	2.1

Chapter 5:
Conclusion and Future Work

This thesis is conducted with a discussion, Research scope for further work and how it can be improved in future studies.

5.1 Contribution of the Research Work

The research work conducted on this thesis focuses on the use of LBP and PCA with sparse face recognition; for sparse face recognition, OMP has been used instead of l^1 minimization. This is one of the first approaches for using LBP, PCA and sparse face recognition for a face recognition algorithm. Experiments conducted on various databases show that this method gives a better result than some of the existing methods even when trained with only a small number of training samples.

Extensive and rigorous testing has been conducted on seven different standard databases in two different test methods. Method two is a new way to test the face recognition algorithm which has not been observed in the current literature.

This is also one of the first approaches to design a portable hardware system with the above mentioned algorithm. The Algorithm works well both in day and in night lighting conditions using Infra-red LED's as the light source.

As per the current literature, the use of sparse face recognition with LBP and PCA is a novel idea; its implementation on a portable hardware system is an original approach. As we have observed, there is a vast area of research to be done for the betterment of this work; It can be made more robust and cost effective. In the next chapter, further discussion about the future of this research will be conducted in more detail.

5.2 Scope for future work

The method has assumed that all the facial images are in-plane and aligned; however, without any alignment the algorithm has put forth expected results. Facial alignment can yield a higher accuracy.

In the present work, we have implemented the algorithm on an x86 based Vortex processor board. This board uses a windows based operating system. The processor is not too advanced and has lower processing power, but recently, with the advancement of technology, there are new classes of processors available which are known as System on Chip or SoC. Many of the recent embedded development boards use these kinds of processors to design the system. Using SoC makes the system cheaper and faster as all the components are integrated in one chip, such as RAM, data bus, address bus, flash memory, EPROM and other features depending on the features provided. Most of these development boards use Android or Linux based operating systems; both are free of

cost and therefore make the system cheaper. Two of the most potential hardware that can be used for this design is Raspberry Pi and Beaglebone Black. Both of these hardware systems cost 10 percent of the price as compared to the board used in the research above. This hardware was made available in the market pretty recently hence they were not used to design the hardware for this research. Another very powerful processor was introduced by NVIDIA fairly recently and would be ideal for such a high performance system. It packs a Quad Core processor and each and every core is 1.6 Ghz. These types of processors have multiple GPU core and it supports CUDA. Using CUDA can be beneficial as it can use the GPU cores for intensive floating point operations making the system faster and more accurate.

An outline for future research can be seen in the following list:

- Finding the right portable hardware.
- Rewriting codes for Linux or Android based systems either using Python or C++.
- Executing algorithm tests on different types of hardware to determine the best one for both accuracy and speed.
- Analyze and optimize the algorithm for embedded hardware.
- Find the right camera module depending on the processing board used to run the algorithm.
- Design a robust facial alignment algorithm to increase the accuracy of the system.

- Increase program functionality and speed by designing a multiple processor based system.
- Design an optimal system, making the hardware self sufficient, cost effective and check its efficiency on real life conditions.
- Program the algorithm using CUDA and check for the increase in speed and efficiency (only available in CUDA supported devices)
- Most recent embedded face recognition systems are designed using HMM (Hidden Markov Model) or neural network [97][98]. Our algorithm is based on LBP, PCA and Sparse face recognition. This is the first approach with Sparse Face Recognition LBP and PCA on embedded system for this kind of system. A detailed study and comparison is necessary with the present systems to prove its robustness over the present system.
- Finally, optimize the system as per the industrial standards.

References

[1] C. Bishop and P. Viola. "Learning and Vision: Discriminative Methods." ICCV *Course on Learning and Vision 2*, no. 7, pp. 11, 2003.

[2] Y. Freund and R. E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." In *European Conf. on Computational Learning Theory*, pp. 2, 7, 1994.

[3] Y. Freund and R.E. Schapire. " A decision-theoretic generalization of on-line learning and an application to boosting." *Journal of computer and system sciences* 55, no. 1, pp. 119-139, 1997.

[4] S. Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum. " Statistical learning of multi-view face detection." In *Proceedings of the 7th European Conference on Computer Vision-Part IV*, pp. 67-81. Springer-Verlag, 2002.

[5] T. Mita, T. Kaneko, and O. Hori. "Joint haar-like features for face detection." In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, pp. 1619-1626. IEEE, 2005.

[6] R. E. Schapire, E. Robert, and Y. Singer. "Improved boosting algorithms using confidence-rated predictions." *Machine learning* 37, no. 3, pp 297-336, 1999.

[7] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features." In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I-511. IEEE, 2001.

[8] B. Wu, H. Ai, C. Huang, and S. Lao. "Fast rotation invariant multi-view face detection based on real adaboost." In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pp. 79-84. IEEE, 2004.

[9] T. Ahonen, A. Hadid, and M. Pietikäinen. "Face recognition with local binary patterns." In *Computer Vision-ECCV 2004*, pp. 469-481. Springer Berlin Heidelberg, 2004.

[10] T. Ahonen, A. Hadid, and M. Pietikainen. "Face description with local binary patterns: Application to face recognition." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28, no. 12, pp 2037-2041, 2006.

[11] F. Goudail, E. Lange, T. Iwamoto, K. Kyuma, and N. Otsu. "Face recognition system using local autocorrelations and multiscale integration." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 18, no. 10, pp 1024-1028, 1996.

[12] C. Liu. "Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28, no. 5, pp 725-737, 2006.

[13] Y. Rodriguez, and S. Marcel. "Face authentication using adapted local binary pattern histograms." In *Computer Vision–ECCV 2006*, pp. 321-332. Springer Berlin Heidelberg, 2006.

[14] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. "Face recognition: A literature survey." *Acm Computing Surveys (CSUR)* 35, no. 4, pp 399-458, 2003.

[15] T. Ojala, M. Pietikäinen, and D. Harwood. "A comparative study of texture measures with classification based on featured distributions." *Pattern recognition* 29, no. 1, pp 51-59, 1996.

[16] T. Ojala, M. Pietikainen, and T. Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24, no. 7, pp 971-987, 2002.

[17] I. Matthews, and S. Baker. "Active appearance models revisited." *International Journal of Computer Vision* 60, no. 2, pp 135-164, 2004.

[18] S. Baker, I. Matthews, and J. Schneider. "Automatic construction of active appearance models as an image coding problem." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 26, no. 10, pp 1380-1384, 2004.

[19] B. F. Rolfe, M. J. Cardew-Hall, S. M. Abdallah, and GA W. West. "Geometric shape errors in forging: developing a metric and an inverse model." *Proceedings of the institution of mechanical engineers, part B: journal of engineering manufacture* 215, no. 9, pp 1229-1240, 2001.

[20] T. F. Cootes, D. H. Cooper, C. J. Taylor, and J. Graham. "Trainable method of parametric shape description." *Image and Vision Computing* 10, no. 5, pp 289-294, 2004.

[21] T. F. Cootes, G. J. Edwards, and C. J. Taylor. "Active appearance models." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23, no. 6, pp 681-685, 2001.

[22] S. Yan, C. Liu, S. Z. Li, H. Zhang, H-Y. Shum, and Q. Cheng. "Face alignment using texture-constrained active shape models." *Image and Vision Computing* 21, no. 1, pp 69-75, 2003.

[23] D. Cristinacce, and T. F. Cootes. "Facial feature detection and tracking with automatic template selection." In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pp. 429-434. IEEE, 2006.

[24] G. Dedeoglu, T. Kanade, and S. Baker. "The asymmetry of image registration and its application to face tracking." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29, no. 5, pp 807-823, 2007.

[25] R. Beichel, H. Bischof, F. Leberl, and M. Sonka. "Robust active appearance models and their application to medical image analysis." *Medical Imaging, IEEE Transactions on* 24, no. 9, pp 1151-1169, 2005.

[26] E. Jones, and S. Soatto. "Layered active appearance models." In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, pp. 1097-1102. IEEE, 2005.

[27] D. Cristinacce, and T. Cootes. "Boosted regression active shape models." In *Proc. British Machine Vision Conference*, vol. 2, pp. 880-889. 2007.

[28] A. U. Batur, and M. H. Hayes. "Adaptive active appearance models." *Image Processing, IEEE Transactions on* 14, no. 11, pp. 1707-1721, 2005.

[29] C. Butakoff, and A. F. Frangi. "A framework for weighted fusion of multiple statistical models of shape and appearance." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28, no. 11, pp. 1847-1857, 2006.

[30] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof. "Fast active appearance model search using canonical correlation analysis." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28, no. 10, pp. 1690-1694, 2006.

[31] A. Kanaujia, and D. N. Metaxas. "Large scale learning of active shape models." In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 1, pp. I-265. IEEE, 2007.

[32] J. Tu, Z. Zhang, Z. Zeng, and T. Huang. "Face localization via hierarchical condensation with fisher boosting feature selection." In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II-719. IEEE, 2004.

[33] F. De la Torre, A. Collet, M. Quero, J. F. Cohn, and T. Kanade. "Filtered component analysis to increase robustness to local minima in appearance models." In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1-8. IEEE, 2007.

[34] K. Lee, J. Ho, and D. J. Kriegman. "Acquiring linear subspaces for face recognition under variable lighting." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27, no. 5, pp. 684-698, 2005.

[35] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19, no. 7, pp. 711-720, 1997.

[36] R. Basri, and D. W. Jacobs. "Lambertian reflectance and linear subspaces." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25, no. 2, pp. 218-233, 2003.

[37] D. L. Donoho, and Y. Tsaig. *Fast solution of l1-norm minimization problems when the solution may be sparse*. Department of Statistics, Stanford University, 2006.

[38] D. L. Donoho, M. Elad, and V. N. Temlyakov. "Stable recovery of sparse overcomplete representations in the presence of noise." *Information Theory, IEEE Transactions on* 52, no. 1, pp. 6-18, 2006.

[39] J. Chen, and X. Huo. "Sparse representations for multiple measurement vectors (MMV) in an over-complete dictionary." In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, vol. 4, pp. iv-257. IEEE, 2005.

[40] A. Bjorck,. *Numerical methods for least squares problems*. No. 51. Society for Industrial and Applied Mathematics, 1996.

[41] S. Kunis, and H. Rauhut. "Random sampling of sparse trigonometric polynomials, II. Orthogonal matching pursuit versus basis pursuit." *Foundations of Computational Mathematics* 8, no. 6, pp. 737-763, 2008.

[42] S. Kunis, and H. Rauhut. "Random sampling of sparse trigonometric polynomials, II. Orthogonal matching pursuit versus basis pursuit." *Foundations of Computational Mathematics* 8, no. 6, pp. 737-763, 2008.

[43] M. Turk, and A. Pentland. "Eigenfaces for recognition." *Journal of cognitive neuroscience* 3, no. 1, pp 71-86, 1991.

[44] T. Ahonen, A. Hadid, and M. Pietikäinen. "Face recognition with local binary patterns." In *Computer Vision-ECCV 2004*, pp. 469-481. Springer Berlin Heidelberg, 2004.

[45] P. Viola, and M. Jones. "Rapid object detection using a boosted cascade of simple features." In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I-511. IEEE, 2001.

[46] A. A. Mohammed, R. Minhas, Q. M. J. Wu, and M. A. Sid-Ahmed. "A novel technique for human face recognition using nonlinear curvelet feature subspace." In *Image Analysis and Recognition*, pp. 512-521. Springer Berlin Heidelberg, 2009.

[47] J.E. Al Daoud. "Enhancement of the Face Recognition Using a Modified Fourier-Gabor Filter." *International Journal Advance. Software Computer. Applications* 1, no. 2, 2009.

[48] P. Forczmański. "On the Dimensionality of PCA Method and Color Space in Face Recognition." In *Image Processing and Communications Challenges 4*, pp. 55-63. Springer Berlin Heidelberg, 2013.

[49] H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana. "Discriminative common vectors for face recognition." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27, no. 1, pp. 4-13, 2005.

[50] H. Yu, and J. Yang. "A direct LDA algorithm for high-dimensional data-with application to face recognition." *Pattern recognition* 34, no. 10, pp 2067, 2001.

[51] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19, no. 7, pp 711-720, 1997.

[52] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski. "Face recognition by independent component analysis." *Neural Networks, IEEE Transactions on* 13, no. 6, pp. 1450-1464, 2002.

[53] H. Zhang, A. C. Berg, M. Maire, and J. Malik. "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition." In *Computer Vision and Pattern*

Recognition, 2006 IEEE Computer Society Conference on, vol. 2, pp. 2126-2136. IEEE, 2006.

[54] C. E. Thomaz, and G. A. Giraldi. "A new ranking method for principal components analysis and its application to face image analysis." *Image and Vision Computing* 28, no. 6, pp. 902-913, 2010.

[55] E. Z. Tenorio, and C. E. Thomaz. "Analise Multinlinear Discrimanante de Formas Frontais de Imagens 2D de Face." In *Simpósio Brasileiro de Automação Inteligente,* 2011.

[56] V.D. Amaral, C. Fígaro-Garcia, G. J. F. Gattas, and C. E. Thomaz. "Normalização espacial de imagens frontais de face em ambientes controlados e não-controlados." *FaSCi-Tech* 1, no. 1, 2012.

[57] L. L. OLIVEIRA JR, and C. E. Thomaz. "Captura e alinhamento de imagens: Um banco de faces brasileiro." *Relatório de iniciação científica, Depto. Eng. Elétrica da FEI, São Bernardo do Campo, SP* 10, 2006.

[58] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba. "Coding facial expressions with gabor wavelets." In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 200-205. IEEE, 1998.

[59] M. J. Lyons, J. Budynek, and S. Akamatsu. "Automatic classification of single facial images." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 21, no. 12. pp. 1357-1362, 1999.

[60] M. N. Dailey, C. Joyce, M. J. Lyons, M. Kamachi, H. Ishi, J. Gyoba, and G. W. Cottrell. "Evidence and a computational explanation of cultural differences in facial expression recognition." *Emotion* 10, no. 6, pp 874, 2010.

[61] A. Kasinski, A. Florek, and A. Schmidt. "The PUT face database." *Image Processing and Communications* 13, no. 3-4, pp. 59-64, 2008.

[62] M. Abdullah, M. Wazzan, and S. Bo-saeed. "Optimizing Face Recognition Using PCA." *arXiv preprint arXiv:1206.1515,* 2012.

[63] X. Tan, S. Chen, Z. H. Zhou, and F. Zhang. "Face recognition from a single image per person: A survey." *Pattern Recognition* 39, no. 9, pp. 1725-1745, 2006.

[64] X. Liu, "Discriminative face alignment." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31, no. 11, pp. 1941-1954, 2009.

[65] R. Gross, I. Matthews, and S. Baker. "Generic vs. person specific active appearance models." *Image and Vision Computing* 23, no. 12, pp. 1080-1093, 2005.

[66] X. Liu, P. Tu, and F. Wheeler. "Face model fitting on low resolution images." In *Proc. 17th British Machine Vision Conference, Edinburgh, UK*, vol. 3, pp. 1079-1088, 2006.

[67] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. "Robust face recognition via sparse representation." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31, no. 2, pp. 210-227, 2009.

[68] J. A. Tropp, and A. C. Gilbert. "Signal recovery from random measurements via orthogonal matching pursuit." *Information Theory, IEEE Transactions on* 53, no. 12, pp. 4655-4666, 2007.

[69] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan. "Sparse representation for computer vision and pattern recognition." *Proceedings of the IEEE* 98, no. 6, pp. 1031-1044, 2010.

[70] J. P. Hansen, and M. Sekine. "Decision diagram based techniques for the haar wavelet transform." In *Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on*, vol. 1, pp. 59-63. IEEE, 1997.

[71] K. Ma, and X. Tang. "Face recognition using discrete wavelet graph." In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 6, pp. VI-117. IEEE, 2003.

[72] D. S. Chen, and L. Zheng-Kai. "Generalized Haar-Like Features for Fast Face Detection." In *Machine Learning and Cybernetics, 2007 International Conference on*, vol. 4, pp. 2131-2135. IEEE, 2007.

[73] C. J. Ju, "Equivalent relationship and unified indexing of FFT algorithm." In *Circuits and Systems, 1993., ISCAS'93, 1993 IEEE International Symposium on*, pp. 742-745. IEEE, 1993.

[74] D. Omaia, and L. V. Batista. "2D-DCT Distance Based Face Recognition Using a Reduced Number of Coefficients." In *Computer Graphics and Image Processing (SIBGRAPI), 2009 XXII Brazilian Symposium on*, pp. 291-298. IEEE, 2009.

[75] S. Tomazic, and S. Znidar. "A fast recursive STFT algorithm." In *Electrotechnical Conference, 1996. MELECON'96., 8th Mediterranean*, vol. 2, pp. 1025-1028. IEEE, 1996.

[76] Y. Li, S. Wang, and X. Ding. "Person-independent head pose estimation based on random forest regression." In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pp. 1521-1524. IEEE, 2010.

[77] J. S. Beis, and D. G. Lowe. "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces." In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 1000-1006. IEEE, 1997.

[78] D. R. Kisku, H. Mehrotra, J. K. Sing, and P. Gupta. "SVM-based Multiview Face Recognition by Generalization of Discriminant Analysis." 2010.

[79] K. Ray, Q. M. J. Wu, G. Basu, and P. K. Panigrahi. "Random matrix route to image denoising." In *Systems and Informatics (ICSAI), 2012 International Conference on*, pp. 1975-1980. IEEE, 2012.

[80] P. Nicholl, and A. Amira. "DWT/PCA face recognition using automatic coefficient selection." In *Electronic Design, Test and Applications, 2008. DELTA 2008. 4th IEEE International Symposium on*, pp. 390-393. IEEE, 2008.

[81] J. Y. Gan, and J. F. Liu. "Fusion and recognition of face and iris feature based on wavelet feature and KFDA." In *Wavelet Analysis and Pattern Recognition, 2009. ICWAPR 2009. International Conference on*, pp. 47-50. IEEE, 2009.

[82] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers. "Fisher discriminant analysis with kernels." In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, pp. 41-48. IEEE, 1999.

[83] K. Etemad, and R. Chellappa. "Discriminant analysis for recognition of human face images." *JOSA A* 14, no. 8, pp. 1724-1733, 1997.

[84] L. Wiskott, J-M. Fellous, N. Kuiger, and Christopher von der Malsburg. "Face recognition by elastic bunch graph matching." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19, no. 7, pp. 775-779, 1997.

[85] W. Wolf, B. Ozer, and T. Lv. "Smart cameras as embedded systems." *Computer* 35, no. 9, pp. 48-53, 2002.

[86] K. Ramani, and A. Davis. "Application driven embedded system design: A face recognition case study." In *Proceedings of the 2007 international conference on Compilers, architecture, and synthesis for embedded systems*, pp. 103-114. ACM, 2007.

[87] F. Yang, and M. Paindavoine. "Implementation of an RBF neural network on embedded systems: real-time face tracking and identity verification." *Neural Networks, IEEE Transactions on* 14, no. 5, pp. 1162-1175, 2003.

[88] D. Ginhac, F. Yang, and M. Paindavoine. "Design, Implementation and Evaluation of Hardware Vision Systems dedicated to Real-Time Face Recognition." *Face Recognition*, pp. 123-148. 2007.

[89] J. P. S. George, S. Tevaramani, and K. B. Raja. "Performance Comparison of Face Recognition using Transform Domain Techniques." *World of Computer Science and Information Technology Journal* 2, no. 3 pp. 82-86, 2012.

[90] A. Eide, T. Lindblad, C. S. Lindsey, M. Minerskjöld, G. Sekhnialdze, and G. Székely. "Implementation of the Zero Instruction Set Computer (ZISC036) on a PC/ISA-Bus Card." In *PROCEEDINGS-SPIE THE INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING*, pp. 40-53. 1996.

[91] I. Essa and A. Pentland. “Coding, analysis, interpretation and recognition of facial expressions.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), pp. 757-763, 1997.

[92] H. Hong, H. Neven, and C. V. Malsburg. “Online facial expression recognition based on personalized galleries.” In *IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 354-359, 1998.

[93] M. Black and Y. Yacoob. “Recognizing facial expressions in image sequences using local parameterized models of image motion.” *International Journal of Computer Vision*, 25(1) pp. 23-48, 1997.

[94] A. Lanitis, C. Taylor, and T. Cootes. “Automatic interpretation and coding of face images using flexible models.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7) pp. 743-756, 1997.

[95] J. Steffens, E. Elagin, and H. Neven. “Personspotter-fast and robust system for human detection, tracking and recognition.” In *International Conference Automatic Face and Gesture Recognition*, pages 516-521, 1998.

[96] S. Bashyal and G. K. Venayagamoorthy. “Recognition of facial expressions using gabor wavelets and learning vector quantization.” *Engineering Applications of Artifcial Intelligence*, 21(7), pp. 1056-1064, 2008.

[97] C. Shan, S. Gong, and P.W. McOwan. “Facial expression recognition based on local binary patterns: A comprehensive study.” *Image and Vision Computing*, 27(4) pp. 803-816, 2008.

[98] G. Edwards, T. Cootes, and C. Taylor. “Face recognition using active appearance models.” In *European Conference on Computer Vision*, volume 2, pp 581-695, 1998.

[99] M. Pantic and L.J.M. Rothkrantz. “Expert system for automatic analysis of facial expression.” *Image and Vision Computing*, 18(11), pp. 881-905, 2000.

[100] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu. “Comparison between geometry-based and gabor wavelets-based facial expression recognition using multi-layer perceptron.” In *International Conference on Automatic Face and Gesture Recognition*, pp. 454-459, 1998.

Vita Auctoris

NAME: Kaushik Ray

PLACE OF BIRTH: INDIA

YEAR OF BIRTH: 1986

EDUCATION:

Master of Applied Science
Electrical and Computer Engineering
University of Windsor, Windsor, ON, Canada, 2013

Bachelor of Technology
Electronics and Communication Enginnering
West Bengal University of technology, Kolkata, India, 2009