Face Detection and Recognition Methodology using Viola-Jones Face Detection Algorithm and presenting its utility in an Organization

Dissertation submitted in
partial fulfilment of the requirement
for the award of the degree of

Master of Technology

in

Computer Science and Engineering

by

ABHIMANYU SINGH

University Roll No. 2K12/CSE/02

Under the Esteemed Guidance of

Mr. VINOD KUMAR
Associate Professor, Computer Engineering Department, DTU

2012-2014

COMPUTER ENGINEERING DEPARTMENT

DELHI TECHNOLOGICAL UNIVERSITY

DELHI – 110042, INDIA

Computer Engineering Department
Delhi Technological University
Delhi-110042
www.dce.edu

CERTIFICATE

This is to certify that the dissertation titled "**Face Detection and Recognition Methodology using Viola-Jones Face Detection Algorithm and presenting its utility in an Organization**" is a bonafide record of work done by **Abhimanyu Singh, Roll No. 2K12/CSE/02** at **Delhi Technological University** for partial fulfilment of the requirements for the degree of Master of Technology in Computer Science & Engineering. This project was carried out under my supervision and has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma to the best of my knowledge and belief.

Mr. Vinod Kumar
Associate Professor
Department of Computer Engineering
Delhi Technological University

Date: _____________

ACKNOWLEDGEMENT

First of all, I would like to express my deep sense of respect and gratitude to my project supervisor Mr. Vinod Kumar for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to him for the support, advice and encouragement he provided without which the project could not have been a success.

Secondly, I am grateful to Dr. O.P.Verma, HOD, Computer Engineering Department, DTU for his immense support. I would also like to acknowledge Delhi Technological University for providing the right academic resources and environment for this work to be carried out.

Last but not the least I would like to express sincere gratitude to my parents for constantly encouraging me during the completion of work.

Abhimanyu Singh
University Roll no: 2K12/CSE/02
M.Tech (Computer Science & Engineering)
Department of Computer Engineering
Delhi Technological University
Delhi – 110042

Date………………

ABSTRACT

The idea of this project is to make a computer application for automatically identifying or verifying a student from a digital image. One of the ways to do this is by comparing selected facial features from the image and a facial database. The system will record the images of the students over the span of lecture and send a message to the mobile phone of the faculty if someone is absent. The message will contain the roll number of the person that is absent. The main idea is to reduce the overhead of attendance. Professor will receive the SMS updates regarding attendance on daily basis. Thus providing a complete solution to the problem of attendance in a secure and reliable manner. The system also aims to be fast and free from errors and human effort.

Table of Contents

Table of Figures

Chapter 1 Introduction

Organizations of all sizes use time and attendance systems to record when employees start and stop work, and the department where the work is performed. However, it's also common to track meals and breaks, the type of work performed, and the number of items produced. In addition to tracking when employees work, organizations also need to keep tabs on when employees are not working. Things like vacation time, compensation time, FMLA time, and jury duty must be recorded. Some organizations also keep detailed records of attendance issues such as who calls in sick and who comes in late.

A time and attendance system provides many benefits to organizations. It enables an employer to have full control of all employees working hours. It helps control labor costs by reducing over-payments, which are often caused by transcription error, interpretation error and intentional error. Manual processes are also eliminated as well as the staff needed to maintain them. It is often difficult to comply with labor regulation, but a time and attendance system is invaluable for ensuring compliance with labor regulations regarding proof of attendance. Companies with large employee numbers might need to install several time clock stations in order to speed up the process of getting all employees to clock in or out quickly or to record activity in dispersed locations.

Attendance Management System is the easiest way to keep track of attendance for community organizations such as school clubs, scouting units, church groups, athletics, or volunteer groups. Attendance Management System covers the requirements of the Personnel Department in terms of Manpower Analysis, day-to-day monitoring of the Attendance, Maintaining Statutory Registers, Monitoring of Leave Records, Calculation of Overtime and transfer of relevant information to the Payroll System.Depending on the supplier, identification method and number of clocking points required, prices vary widely. A time and attendance system protects a company from payroll fraud and provides both employer and employees with confidence in the accuracy of their wage payments all while improving productivity.

1.1. Objective

The objective is to create a software system which will replace the current attendance mechanism system of our college with an entirely automated attendance system. This reduces the manual drudgery of Data Entry, Register maintenance. It will save both time and paper.

1.2. Methodology

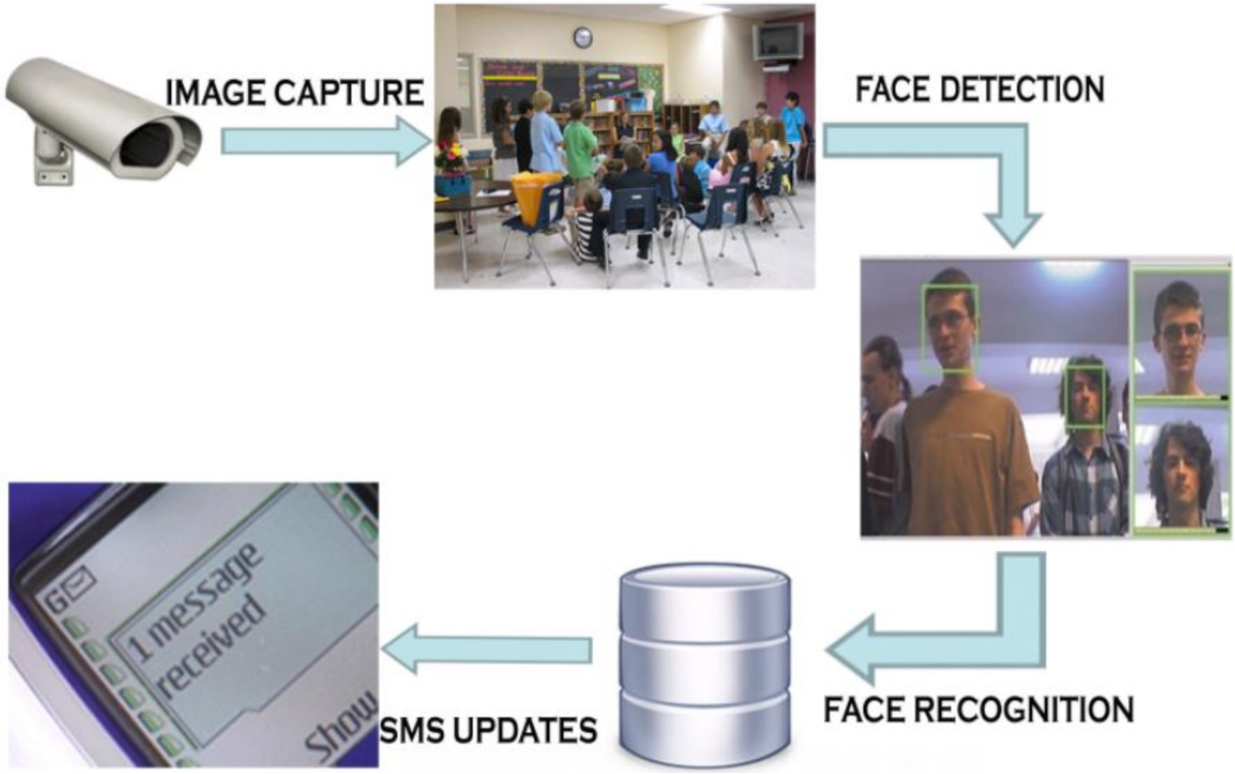


Figure 1.1 Methodology

1.3. Motivation for the Project

I cannot deny the fact that attendances do take a lot of time and there is always a chance of buddy punching or proxy. This is where our Smart Attendance System steps in and could be an important means of e-attendance. This reduces the manual drudgery of Data Entry, Register maintenance and monthly requirement of Punch Cards for conventional Time Clocks to a minimum. Thus designing an efficient system for the same would certainly help in making our college attendance system much more efficient. This serves as the major motivation for us to take up this project.

Perfect Attendance

awarded to

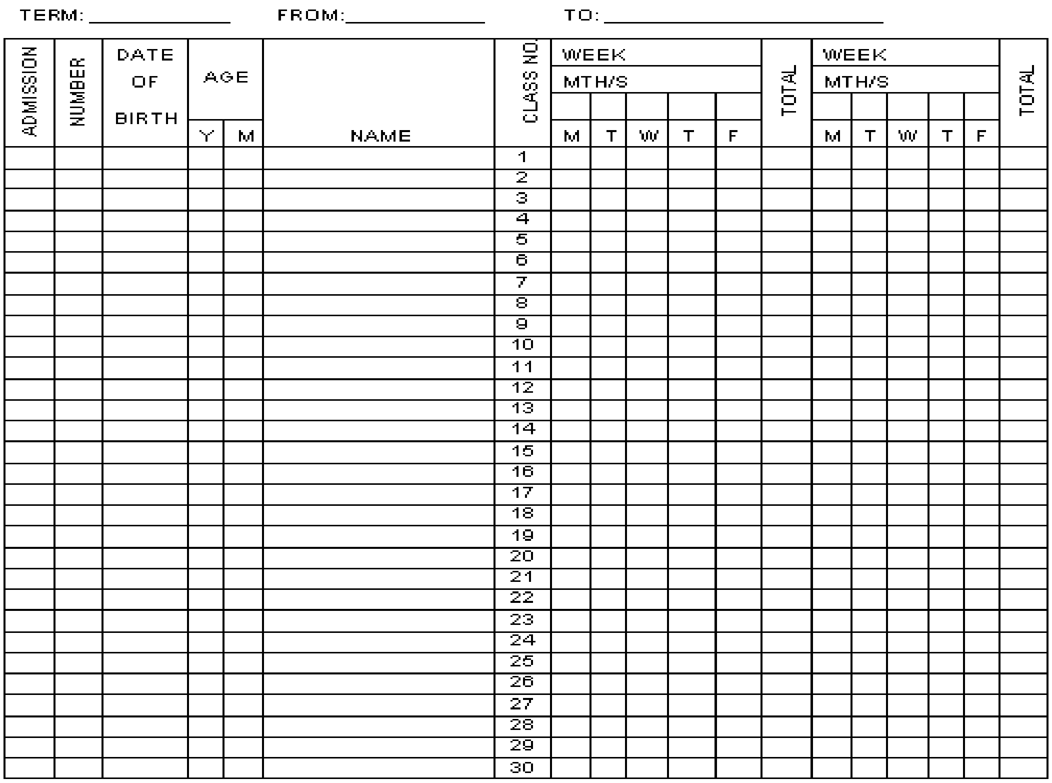
© 2000 Teachnet.com

Figure 1.2 Perfect Attendance

1.4. Commonly Used Modes of Attendance

1.4.1. Manual Data Entry

Most commonly used mode where a person makes a roll call to check presence or absence of individuals. It is the cheapest mode of attendance widely used in schools but it very time consuming and involves a lot of human effort.



TERM: ____________ FROM: ____________ TO: ________________________

ADMISSION	NUMBER	DATE OF BIRTH	AGE		NAME	CLASS NO.	WEEK					TOTAL	WEEK					TOTAL
							MTH/S						MTH/S					
			Y	M			M	T	W	T	F		M	T	W	T	F	
						1												
						2												
						3												
						4												
						5												
						6												
						7												
						8												
						9												
						10												
						11												
						12												
						13												
						14												
						15												
						16												
						17												
						18												
						19												
						20												
						21												
						22												
						23												
						24												
						25												
						26												
						27												
						28												
						29												
						30												

Figure 1.3 Manual Data Entry

1.4.2. Smart Cards

Smart card system is a system used in big companies and big institutes for the attendance of their employees and workers. This system is very important part of different organizations. This system replaced the old tradition of register attendance system in which a person takes the attendance of employees, those old systems were not too much accurate and could be changed and corrupted but now a day this modern system, smart card attendance system is very much accurate and does not require any person for care. This system totally based on microcontrollers which provide this a proper functioning. Mostly all the companies that may be national or international use this attendance system. Companies provide a special card called as smart card, to their Employees to enter their presence daily.

Figure 1.4 Smat Cards

1.4.3. Biometric Systems

Biometric attendance systems are fool proof systems. Fingerprint is one such system where a person's finger print is used as his biometric identification. This technology is being used in ofices and colleges to take attendance. The teacher will bring the small handheld with a finger scanner. Students press their fingers on it for attendance. This system has got many advantages. Manual Data entry is avoided No Proxy attendances can be made. Guardians can be informed about absent.

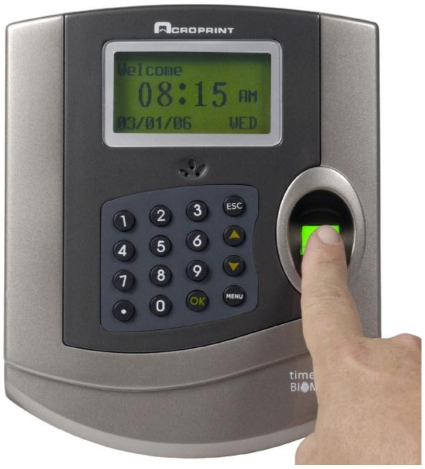


Figure 1.5 Biometric System

Disadvantages with the approaches discussed above:

- Manual drudgery of Data Entry
- Register Maintenance
- Expensive
- Unreliable and Unsecure
- Manual or Semi-Automated only

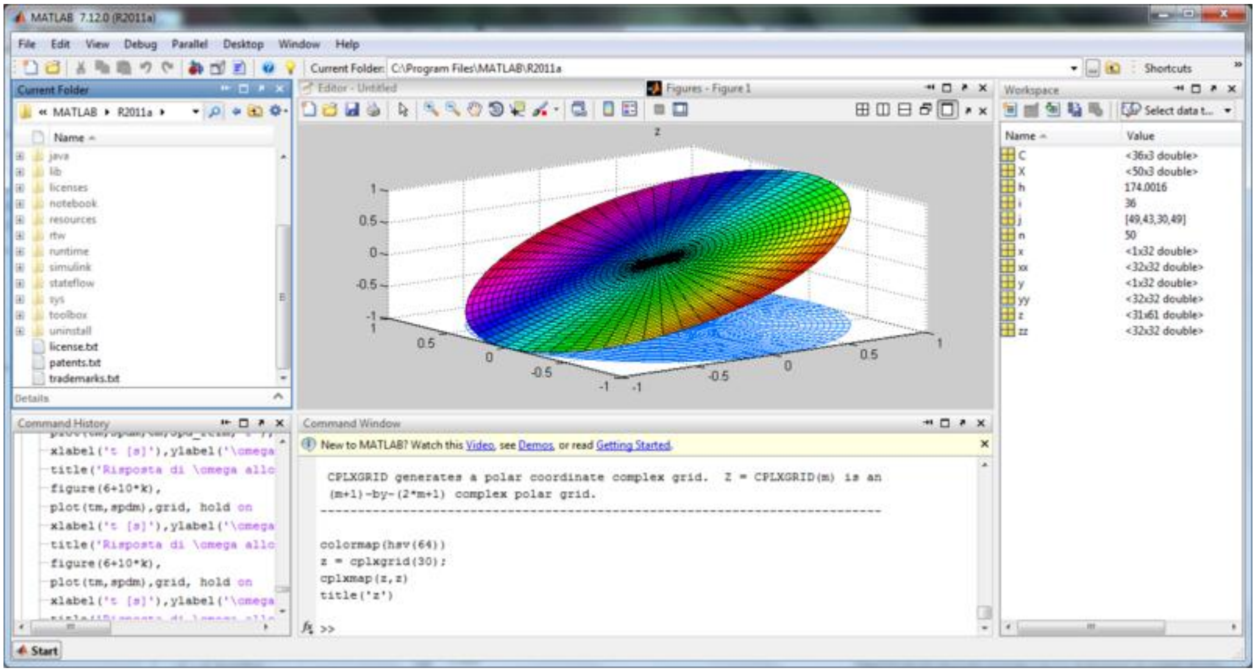
1.5. Proposed Solution

Attendance Management System is the easiest way to keep track of attendance for community organizations such as school clubs, scouting units, church groups, athletics, or volunteer groups. But attendance does take a lot of time and there is always a chance of buddy punching or proxy. In recent decades the various techniques like smart cards and biometric identification. But they still involve some human effort and these systems still lack intelligence and there can be issues of buddy punching or proxy.

The idea of this project is to make a computer application for automatically identifying or verifying a student from a digital image or a video frame from a video source. One of the ways to do this is by comparing selected facial features from the image and a facial database. The system will record the images of the students over the span of lecture and send a message to the mobile phone of the faculty if someone is absent. The message will contain the name of the person that is absent. The main idea is to reduce the overhead of attendance. Professor will receive the SMS updates regarding attendance on daily basic. Thus providing a complete solution to the problem of attendance in a secure and reliable manner. The system also aims to be fast and free from errors and human effort.

1.6. Tools Used

1.6.1. Matlab



```
xlabel('t [s]'),ylabel('\omega
title('Risposta di \omega allo
figure(6+10*k),
plot(tm,spdm),grid, hold on
xlabel('t [s]'),ylabel('\omega
title('Risposta di \omega allo
figure(6+10*k),
plot(tm,spdm),grid, hold on
xlabel('t [s]'),ylabel('\omega
```

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
 CPLXGRID generates a polar coordinate complex grid.  Z = CPLXGRID(m) is an
 (m+1)-by-(2*m+1) complex polar grid.
-------------------------------------------------------------------------------

colormap(hsv(64))
z = cplxgrid(30);
cplxmap(z,z)
title('z')

>>
```

Name	Value
C	<36x3 double>
X	<50x3 double>
h	174.0016
i	36
j	[49,43,30,49]
n	50
x	<1x32 double>
xx	<32x32 double>
y	<1x32 double>
yy	<32x32 double>
z	<31x61 double>
zz	<32x32 double>

Figure 1.6 Matlab

MATLAB (matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

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

Image Processing Toolbox supports a diverse set of image types, including high dynamic range, gigapixel resolution, embedded ICC profile, and tomographic. Graphical tools let you explore an image, examine a region of pixels, adjust the contrast, create contours or histograms, and manipulate regions of interest (ROIs). With toolbox algorithms you can restore degraded images, detect and measure features, analyze shapes and textures, and adjust color balance.

1.7. AT Commands

Computers use AT commands to communicate with modems, i.e. you can use these commands to use data connections via your PremiCell terminal. Most communications applications, however, have a user-friendly interface that hides these AT commands from the user. The AT commands supported by the data adapter are listed below according to their uses. A short description, the syntax, the possible setting values and responses of the AT commands are presented. Note that the "AT" or "at" prefix must be included at the beginning of each command line. You may type several AT commands on the same line, and in such cases you need to type the "AT" or "at" prefix only at the beginning of the command line. Some AT commands are not supported by all Nokia products. Giving a command which is not supported causes an error response, and if the extended error report feature has been enabled with the AT+CMEE command, the error response also reports that the command is not supported with the product used. Also, some parameters may not be supported by all products, and in this case those parameters are just ignored in commands or left blank in responses.

1.7.1. Command Syntax

The letter <n> used in command syntax is the setting value typed in as a part of the command. If the value is optional it is enclosed in square brackets. Setting values for the commands are presented below under the description of the command. When you select a setting value with an AT command, the setting is valid until you change it. The functionality of a particular command form is described on the right side of the syntax, and after that follows the command response.

1.7.1.1. Syntax of AT+ Commands

When you want to set or display a value for a setting, or you want to know the valid values, type in AT+ the command string followed by “=n”, “?”, or “=?”, and press <ENTER>.

1. AT+__=n To write in a new setting with the command.
2. AT+__? Displays the current setting for the command.
3. AT+__=? Displays all setting values that can be used with the command.

1.8. Outline of the Report

Chapter 2: In this chapter, an abstract about all the papers that I have studied for this project

Chapter3: This chapter discusses about the face detection algorithm (Viola Jones) that has been implemented in this project.

Chapter 4: This part comprises of the details of face detection algorithm (Eigenface).

Chapter 5: This chapter explains the AT commands used for sending SMS through a GSM modem.

Chapter 6: In this chapter, the results and discussion regarding the project has been done.

Chapter 7: This chapter comprises of conclusion and future scope of work.

Chapter 2 Literature Survey

2.1. Robust Real-Time Face Detection by Paul Viola and Michael J. Jones [1]

This paper describes a face detection framework that is capable of processing images extremely rapidly while achieving high detection rates. There are three key contributions. The first is the introduction of a new image representation called the "Integral Image" which allows the features used by our detector to be computed very quickly. The second is a simple and efficient classifier which is built using the AdaBoost learning algorithm (Freund and Schapire, 1995) to select a small number of critical visual features from a very large set of potential features. The third contribution is a method for combining classifiers in a "cascade" which allows background egions of the image to be quickly discarded while spending more computation on promising face-like regions. A set of experiments in the domain of face detection is presented. The system yields face detection performance comparable to the best previous systems (Sung and Poggio, 1998; Rowley et al., 1998; Schneiderman and Kanade, 2000; Roth et al., 2000). Implemented on a conventional desktop, face detection proceeds at 15 frames per second.

2.2. Simple Face-detection Algorithm Based on Minimum Facial Features by Yao-Jiunn Chen and Yen-Chun Lin [2]

This article presents an algorithm for rapid and accurate face-detection. The algorithm detects human faces by the geometric correlations between locations of faces and hairs. Ranges of skin color are used to figure out possible face regions so as to initially localize the faces; furthermore, probable hair blocks in an image are determined by means of hair color spectrums. Grouped skin and hair blocks decide candidate face areas in light of the geometric relation. The accuracy of the single-face detection is higher than 92% with the frame-rate of 10 fps, if tested with a simple background and sufficient light source. The algorithm can be transferred from a PC to embedded devices, such as a DSP platform. This system-level implementing ability brings great potential to customized and reusable applications as well as miniature systems.

2.3. Face Recognition Using Principal Component Analysis by S. Thakur [3]

In this paper, an efficient method for face recognition using principal component analysis (PCA) is presented. Recently, the PCA has been extensively employed for face recognition algorithms. It is one of the most popular representation methods for a face image. It not only reduces the dimensionality of the image, but also retains some of the variations in the image data. After performing the PCA, the hidden layer neurons of the RBF neural networks have been modeled by considering intra-class discriminating characteristics of the training images. This helps the RBF neural networks to acquire wide variations in the lower-dimensional input space and improves its generalization capabilities. The proposed method has been evaluated using the AT and T (formerly ORL) and UMIST face databases. Experimental results show that the proposed method has encouraging recognition performance.

2.4. Detecting Faces in Images: A Survey by Ming-Hsuan Yang, David J. Kriegman and Narendra Ahuja [4]

Images containing faces are essential to intelligent vision-based human computer interaction, and research efforts in face processing include face recognition, face tracking, pose estimation, and expression recognition. However, many reported methods assume that the faces in an image or an image sequence have been identified and localized. To build fully automated systems that analyze the information contained in face images, robust and efficient face detection algorithms are required. Given a single image, the goal of face detection is to identify all image regions which contain a face regardless of its three-dimensional position, orientation, and lighting conditions. Such a problem is challenging because faces are non-rigid and have a high degree of variability in size, shape, color, and texture. Numerous techniques have been developed to detect faces in a single image, and the purpose of this paper is to categorize and evaluate these algorithms. I also discuss relevant issues such as data collection, evaluation metrics, and benchmarking. After analyzing these algorithms and identifying their limitations, I conclude with several promising directions for future research.

2.5. Face Recognition Using Eigenfaces by Matthew A. Turk and Alex P. Pentland [5]

This paper presents an approach to the detection and identification of human faces and describes a working, near real-time face recognition system which tracks a subject's head and then recognizes the person by comparing characteristics of the face to those of known individuals. The approach treats face recognition as a two-dimensional recognition problem, taking advantage of the fact that faces are normally upright and thus may be described by a small set of 2-D characteristic views. Face images are projected into feature space ("face space") that best encodes the variation among known face images. The face space is defined by the "Eigen faces", which are eigenvectors of the set of faces; they do not necessarily correspond to isolated features such as eyes, ears, and noses. The framework provides the ability to learn to recognize new faces in an unsupervised manner.

Chapter 3 Face Detection (Phase I)

Face detection is defined as the process of extracting faces from any given image. It differentiates between faces and non-faces in an image. The locations of the faces are found and are then extracted.

3.1. Challenges in Face Detection

Face detection must deal with several well-known challenges. They are usually present in images captured in uncontrolled environments. These challenges can be attributed to some factors:

- Pose variation: The ideal scenario for face detection would be one in which only frontal images are involved. But, this is very unlikely in general uncontrolled conditions. Moreover, the performance of face detection algorithms drops severely when there are large pose variations. It's a major research issue. Pose variation can happen due to subject's movements or camera's angle.

- Feature occlusion: The presence of elements like beards, glasses or hats introduces high variability. Faces can also be partially covered by objects or other faces.

- Facial expression: Facial features also vary greatly because of different facial gestures.

- Imaging conditions: Different cameras and ambient conditions can affect the quality of an image, affecting the appearance of a face.

3.2. Face Detection Methods

There are various methods for face detection. These methods are divided into four categories. These categories may overlap, so an algorithm could belong to two or more categories. This classification can be made as follows:

- Knowledge-based methods: Ruled-based methods that encode our knowledge of human faces.

- Feature-invariant methods: Algorithms that try to find invariant features of a face despite its angle or position.

- Template matching methods: These algorithms compare input images with stored patterns of faces or features.

- Appearance-based methods: A template matching method whose pattern database is learnt from a set of training images.

3.2.1. Knowledge-based methods

These are rule-based methods. They try to capture our knowledge of faces, and translate them into a set of rules. It's easy to guess some simple rules. For example, a face usually has two symmetric eyes, and the eye area is darker than the cheeks. Facial features could be the distance between eyes or the color intensity difference between the eye area and the lower zone. The big problem with these methods is the difficulty in building an appropriate set of rules. There could be many false positives if the rules were too general. On the other hand, there could be many false negatives if the rules were too detailed. A solution is to build hierarchical knowledge-based methods to overcome these problems. However, this approach alone is very limited. It's unable to find many faces in a complex image.

3.2.2. Template matching method

Template matching methods try to define a face as a function. I tried to find a standard template of all the faces. Different features can be defined independently. For example, a face can be divided into eyes, face contour, nose and mouth. Also a face model can be built by edges. But these methods are limited to faces that are frontal and un-occluded. A face can also be represented as a silhouette. Other templates use the relation between face regions in terms of brightness and darkness. These standard patterns are compared to the input images to detect faces. This approach is simple to implement, but it's inadequate for face detection. It cannot achieve good results with variations in pose, scale and shape. However, deformable templates have been proposed to deal with these problems.

3.2.3. Appearance-based methods

The templates in appearance-based methods are learned from the examples in the images. In general, appearance-based methods rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face images. Some appearance-based methods work in a probabilistic network. An image or feature vector is a random variable with some probability of belonging to a face or not.

3.3. Face Detection Algorithm Used

There are various algorithms for face detection based on the methods discussed above. I have used a well-known algorithm by Viola & Jones. The breakthrough in face detection took place with this algorithm. Using a cascade of weak-classifiers, simple Haar features and training, Viola & Jones were able to yield impressive results. This approach is now the most commonly used algorithm for face detection

3.4. Face Detection Using Viola Jones Face Detection Algorithm

The basic principle of the Viola-Jones algorithm is to scan a sub-window capable of detecting faces across a given input image. The standard image processing approach would be to rescale the input image to different sizes and then run the fixed size detector through these images. This approach turns out to be rather time consuming due to the calculation of the different size images. Contrary to the standard approach Viola-Jones rescale the detector instead of the input image and run the detector many times through the image – each time with a different size. At first one might suspect both approaches to be equally time consuming, but Viola-Jones have devised a scale invariant detector that requires the same number of calculations whatever the size. This detector is constructed using a so-called integral image and some simple rectangular features reminiscent of Haar wavelets. The next section elaborates on this detector.

3.4.1. The scale invariant detector

The first step of the Viola-Jones face detection algorithm is to turn the input image into an integral image. This is done by making each pixel equal to the entire sum of all pixels above and to the left of the concerned pixel. This is demonstrated in Figure 4.1.

1	1	1
1	1	1
1	1	1

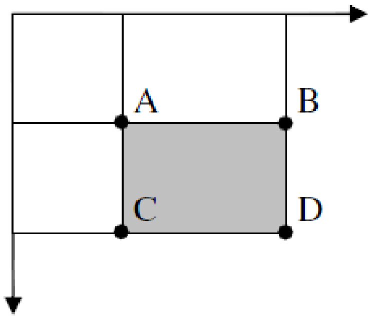
Input image

1	2	3
2	4	6
3	6	9

Integral image

Figure 3.1 The integral image

This allows for the calculation of the sum of all pixels inside any given rectangle using only four values. These values are the pixels in the integral image that coincide with the corners of the rectangle in the input image. This is demonstrated in Figure 4.2.



Sum of grey rectangle = D - (B + C) + A

Figure 3.2 Sum Calculation

It has now been demonstrated how the sum of pixels within rectangles of arbitrary size can be calculated in constant time. The Viola-Jones face detector analyzes a given sub-window using features consisting of two or more rectangles. The different types of features are shown in Figure 4.3.

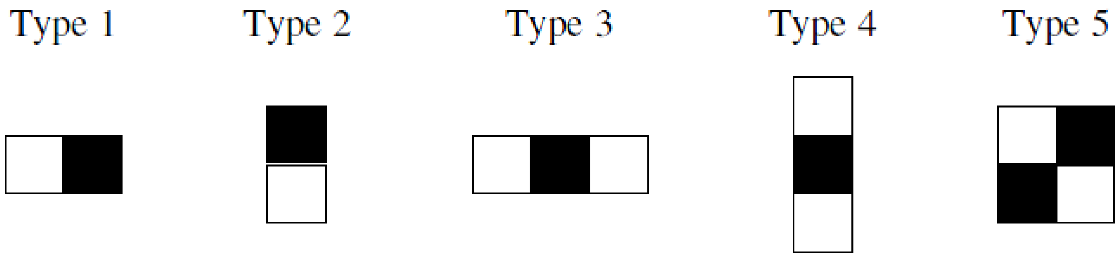


Figure 3.3 The different types of features

Each feature results in a single value which is calculated by subtracting the sum of the white rectangle(s) from the sum of the black rectangle(s).

Viola & Jones have empirically found that a detector with a base resolution of 24*24 pixels gives satisfactory results. When allowing for all possible sizes and positions of the features in Figure 4 a total of approximately 160.000 different features can then be constructed. Thus, the amount of possible features vastly outnumbers the 576 pixels contained in the detector at base resolution. These features may seem overly simple to perform such an advanced task as face detection, but what the features lack in complexity they most certainly have in computational efficiency.

One could understand the features as the computer's way of perceiving an input image. The hope being that some features will yield large values when on top of a face. Of course operations could also be carried out directly on the raw pixels, but the variation due to different pose and individual characteristics would be expected to hamper this approach. The goal is now to smartly construct a mesh of features capable of detecting faces and this is the topic of the next section.

3.4.2. The modified AdaBoost algorithm

AdaBoost is a machine learning boosting algorithm capable of constructing a strong classifier through a weighted combination of weak classifiers. (A weak classifier classifies correctly in only a little bit more than half the cases.) To match this terminology to the presented theory each feature is considered to be a potential weak classifier. A weak classifier is mathematically described as:

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{if } pf(x) > p\theta \\ 0 & \text{otherwise} \end{cases}$$

Where *x* is a 24*24 pixel sub-window, *f* is the applied feature, *p* the polarity and _ the threshold that decides whether *x* should be classified as a positive (a face) or a negative (a non-face). Since only a small amount of the possible 160.000 feature values are expected to be potential weak classifiers the AdaBoost algorithm is modified to select only the best features.

3.4.3. The cascaded classifier

The basic principle of the Viola-Jones face detection algorithm is to scan the detector many times through the same image – each time with a new size. Even if an image should contain one or more faces it is obvious that an excessive large amount of the evaluated sub-windows would still be negatives (non-faces). This realization leads to a different formulation of the problem:

Instead of finding faces, the algorithm should discard non-faces.

The thought behind this statement is that it is faster to discard a non-face than to find a face. With this in mind a detector consisting of only one (strong) classifier suddenly seems inefficient since the evaluation time is constant no matter the input. Hence the need for a cascaded classifier arises. The cascaded classifier is composed of stages each containing a strong classifier. The job of each stage is to determine whether a given sub-window is definitely not a face or maybe a face. When a sub-window is classified to be a non-face by a given stage it is immediately discarded. Conversely a sub-window classified as a maybe-face is passed on to the next stage in the cascade. It follows that the more stages a given sub-window passes, the higher the chance the sub-window actually contains a face. The concept is illustrated with two stages in Figure 4.4.

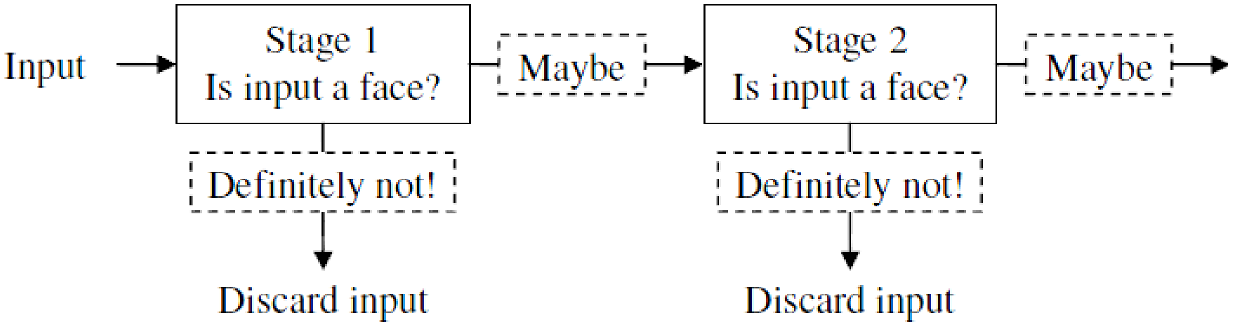


Figure 3.4 The cascaded classifier

In a single stage classifier one would normally accept false negatives in order to reduce the false positive rate. However, for the first stages in the staged classifier false positives are not considered to be a problem since the succeeding stages are expected to sort them out. Therefore Viola-Jones prescribes the acceptance of many false positives in the initial stages. Consequently the amount of false negatives in the final staged classifier is expected to be very small.

3.5. Refining Viola-Jones' Result

The output of the Viola-Jones face detection algorithm is a series of bounding boxes around a face (See Figure 4.5). As I have to provide this output to the face recognition phase, the series of bounding boxes representing a single face causes a duplicate match in the face recognition phase. To address this issue, a refinement in the Viola-Jones' output was inevitable.

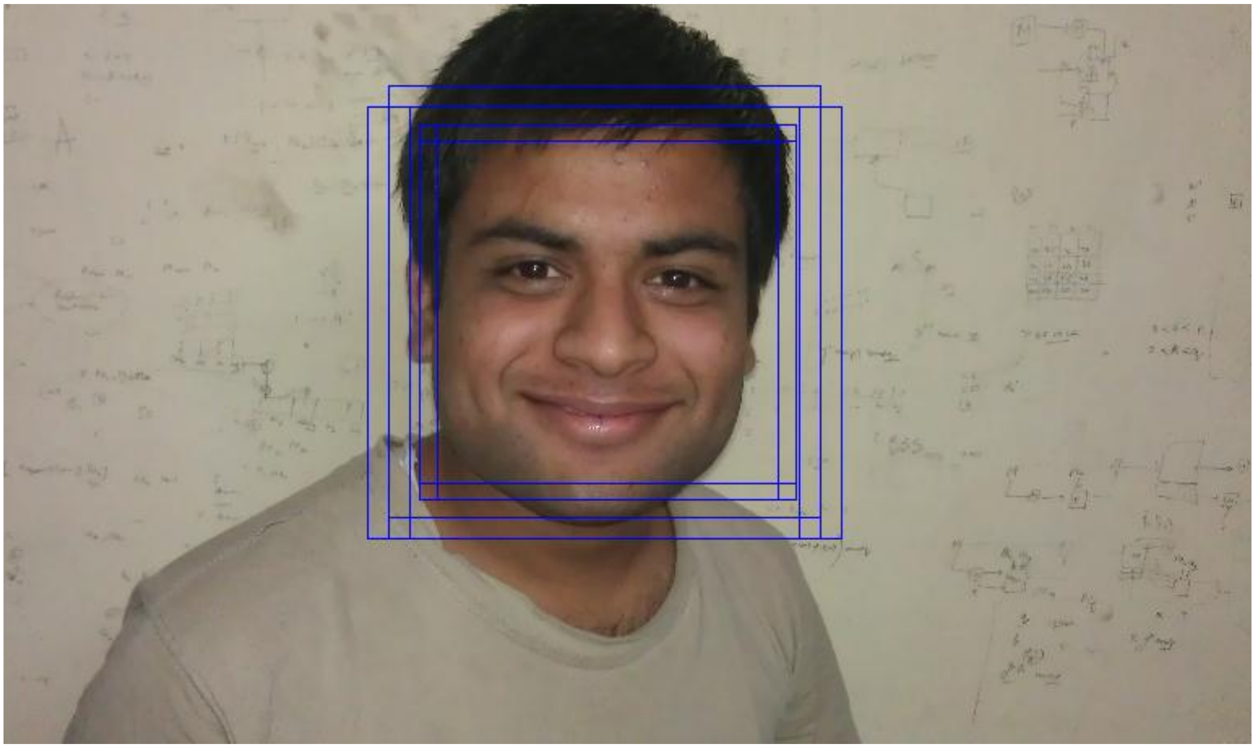


Figure 3.5 Series of bounding boxes around a face

3.5.1. Removing Redundancies

In order to reduce the series of bounding boxes around a single face to a single bounding box around a single image, I compared the top-left corner co-ordinates of the bounding boxes. Then I conducted a series of experiment on different images to calculate an optimum threshold. I found that a threshold value of 230 pixels of x and y co-ordinates each was optimum for our images. If the difference between the corresponding x and y co-ordinates is not more than 230 pixels, the bounding boxes are considered to be around the same face. The average of these bounding boxes is then taken as the final single bounding box that represents a single face. This removed the redundancies quiet efficiently. Figure 4.6 shows the result after removing redundancies.

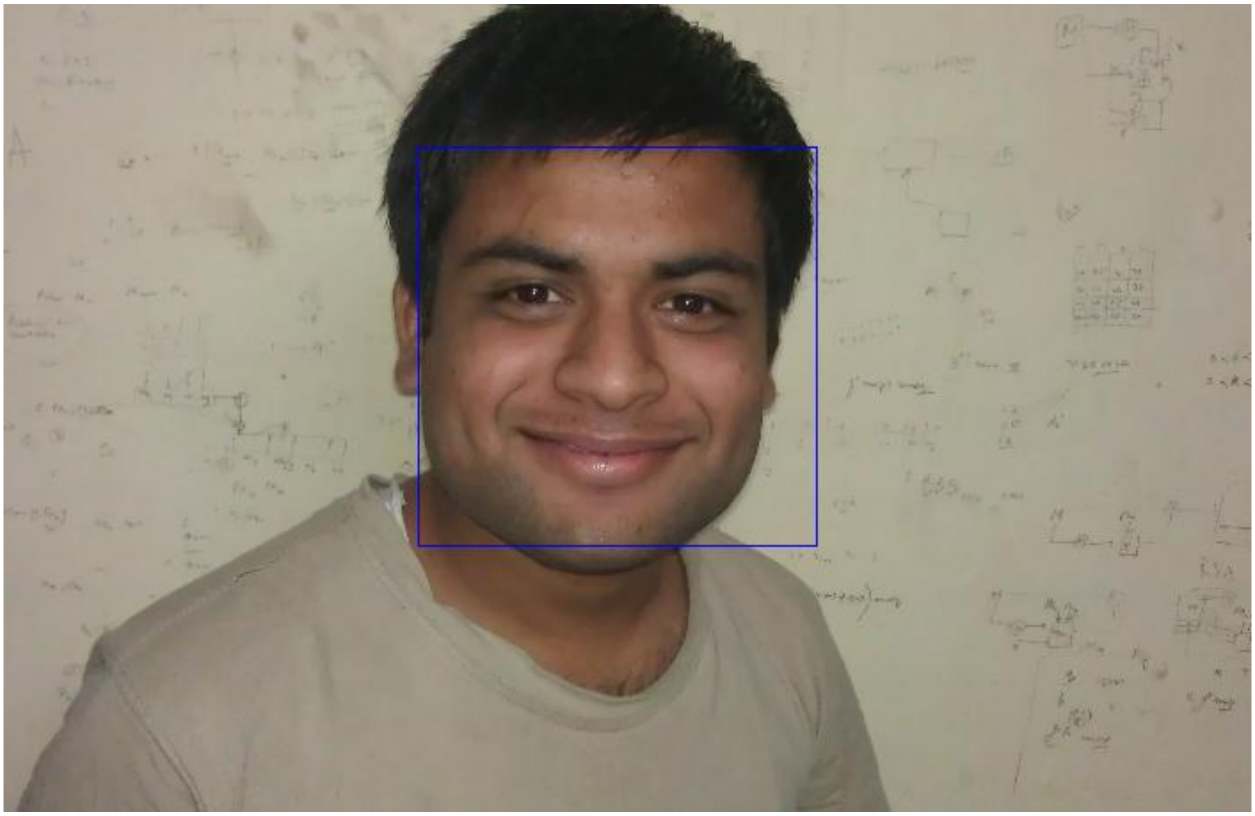


Figure 3.6 After removing redundancies

3.6. Cropping the Face

After the face has been detected in the image, the final step of this phase is to crop the face and save it. The face is cropped and saved as an image in a folder named “Cropped” with the name “1.jpg”, “2.jpg”, “3.jpg”, etc.

3.6.1. Converting JPG to PGM

The Face Recognition phase reads grey scale PGM images. As a result, the cropped JPG images are converted to PGM images.

Chapter 4 Face Recognition (Phase II)

Face recognition is defined as the process of comparing the given face with the faces in the database and giving a match.

4.1. Challenges in Face Recognition

Face recognition very strongly depends on the face detection phase. As a result, the challenges of face detection are common to face recognition as well.

- Pose variation: The ideal scenario for face recognition would be one in which only frontal images are involved. But, this is very unlikely in general uncontrolled conditions. Moreover, the performance of face recognition algorithms drops when there are large pose variations. It's a major research issue. Pose variation can happen due to subject's movements or camera's angle.

- Occlusion: We understand occlusion as the state of being obstructed. In the face recognition context, it involves that some parts of the face can't be obtained. For example, a face photograph taken from a surveillance camera could be partially hidden behind a column. The recognition process can rely heavily on the availability of a full input face. The presence of elements like beards, glasses or hats introduces high variability. Faces can also be partially covered by objects or other faces.

- Facial expression: Facial features also vary greatly because of different facial gestures.

- Imaging conditions: Different cameras and ambient conditions can affect the quality of an image, affecting the appearance of a face.

- Illumination: There can be relevant illumination variations on images taken under uncontrolled environment. The chromaticity is an essential factor in face recognition. The intensity of the color in a pixel can vary greatly depending on the lighting conditions. The big problem is that two faces of the same subject but with illumination variations may show more differences between them than compared to another subject. Summing up, illumination is one of the big challenges of automated face recognition systems.

- Optical technology: A face recognition system should be aware of the format in which the input images are provided. There are different cameras with different features, different resolutions, different weaknesses and problems.

- Expression: Facial expression is another variability provider. However, it isn't as strong as illumination or pose. The addition of expression variability to pose and illumination problems can become a real impediment for accurate face recognition.

4.2. Face Recognition Algorithm Used

There are various algorithms for face recognition. I have used an algorithm which is known as Eigenfaces. Eigenfaces are a set of eigenvectors used in the computer vision problem of human face recognition. The approach of using eigenfaces for recognition was developed by Sirovich & Kirby. It is considered the first successful example of facial recognition technology.

A set of eigenfaces can be generated by performing a mathematical process called principal component analysis (PCA) on a large set of images depicting different human faces. Informally, eigenfaces can be considered as a set of "standardized face ingredients", derived from statistical analysis of many pictures of faces. Any human face can be considered to be a combination of these standard faces

4.2.1. Principal Component Analysis (PCA)

Principal component analysis (PCA) was invented in 1901 by Karl Pearson. PCA involves a mathematical procedure that transforms a number of possibly correlated variables into a number of uncorrelated variables called principal components, related to the original variables by an orthogonal transformation. This transformation is defined in such a way that the first principal component has as high a variance as possible (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to the preceding components. PCA is sensitive to the relative scaling of the original variables. Depending on the field of application, it is also named the discrete Karhunen–Loève transform (KLT), the Hotelling transform or proper orthogonal decomposition (POD).

PCA tries to identify the directions of maximum variation in the training data. Using the SVD decomposition, I can find the first k-vectors whose singular values are the largest. We basically take the best projection of the training images onto a subspace. The images when represented in this subspace have maximum variance (in the coordinates).

The major advantage of PCA is that the eigenface approach helps reducing the size of the database required for recognition of a test image. The trained images are not stored as raw images rather they are stored as their weights which are found out projecting each and every trained image to the set of eigenfaces obtained.

4.2.1.1. Procedure

To identify an input image we proceed as follows

1. Evaluate the components of input image along the selected k eigen vectors.

2. Reconstruct the image from the components.

3. If the distance between the reconstructed image and the original image is above a threshold e the input image is not a face image. We tried this threshold but it did not turn out to be a reliable estimate.

4. Now compute the distance of the input image from the training images in the space spanned by the k Eigen vectors. If the minimum distance is above threshold e then the input image is not a face from the training database else report the training image with the minimum distance as the recognized image.

The threshold e is defined as

$$\epsilon = \frac{1}{2} max\{||d_{ij}||\}$$

where dij denotes the distance between training images I and j.

Instead of working with individual images, we can cluster the images of an individual in a single class. Distance to a class may be defined as average of the distances to each of the individual images within the class.

4.2.1.2. Mathematics approach

A 2-D facial image can be represented as 1-D vector by concatenating each row (or column) into a long thin vector. Let's suppose we have M vectors of size N (= rows of image x columns of image) representing a set of sampled images. pj's represent the pixel values.

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

The images are mean centered by subtracting the mean image from each image vector. Let m represent the mean image.

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

And let wi be defined as mean centered image

$$w_i = x_i - m$$

Our goal is to find a set of ei’s which have the largest possible projection onto each of the wi’s. We wish to find a set of M orthonormal vectors ei for which the quantity

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

is maximized with the orthonormality constraint

$$e_l^T e_k = \delta_{lk}$$

It has been shown that the ei’s and ¸i’s are given by the eigenvectors and eigenvalues of the covariance matrix

$$C = WW^T$$

where W is a matrix composed of the column vectors wi placed side by side. The size of C is N x N which could be enormous. For example, images of size 64 x 64 create the covariance matrix of size 4096x4096. It is not practical to solve for the eigenvectors of C directly. A common theorem in linear algebra states that the vectors ei and scalars ¸i can be obtained by solving for the eigenvectors and eigenvalues of the MxM matrix WTW. Let di and μi be the eigenvectors and eigenvalues of WTW, respectively.

$$W^T W d_i = \mu_i d_i$$

By multiplying left to both sides by W

$$W W^T (W d_i) = \mu_i (W d_i)$$

which means that the first M ¡ 1 eigenvectors ei and eigenvalues ¸i of WWT are given by Wdi and ^{1}i, respectively. Wdi needs to be normalized in order to be equal to ei. Since we only sum up a finite number of image vectors, M, the rank of the covariance matrix cannot exceed M ¡ 1(The ¡1 come from the subtraction of the mean vector m).

The eigenvectors corresponding to nonzero eigenvalues of the covariance matrix produce an orthonormal basis for the subspace within which most image data can be represented with a small amount of error. The eigenvectors are sorted from high to low according to their corresponding eigenvalues. The eigenvector associated with the largest eigenvalue is one that reflects the greatest variance in the image. That is, the smallest eigenvalue is associated with the eigenvector that finds the least variance. They decrease in exponential fashion, meaning that the roughly 90% of the total variance is contained in the first 5% to 10% of the dimensions.

A facial image can be projected onto M' (<<M) dimensions by computing

$$\Omega = [v_1 v_2 \ldots v_{M'}]^T$$

where vi = eTi wi. vi is the ith coordinate of the facial image in the new space, which came to be the principal component. The vectors ei are also images, so called, eigenimages, or eigenfaces. They can be viewed as images and indeed look like faces. So, Ω describes the contribution of each eigenface in representing the facial image by treating the eigenfaces as a basis set for facial images. The simplest method for determining which face class provides the best description of an input facial image is to find the face class k that minimizes the Euclidean distance

$$\epsilon_k = \|(\Omega - \Omega_k)\|$$

where Ω_k is a vector describing the kth face class. If ϵ_k is less than some predefined threshold θ_{ϵ}, a face is classified as belonging to the class k.

4.3. Face Recognition Using Eigenfaces

In the language of information theory, the relevant information in a face needs to be extracted, encoded efficiently and one face encoding is compared with the similarly encoded database. The trick behind extracting such kind of information is to capture as many variations as possible from the set of training images.

Mathematically, the principal components of the distribution of faces are found out using the eigenface approach. First the eigenvectors of the covariance matrix of the set of face images is found out and then they are sorted according to their corresponding eigenvalues. Then a threshold eigenvalue is taken into account and eigenvectors with eigenvalues less than that threshold values are discarded. So ultimately the eigenvectors having the most significant eigenvalues are selected. Then the set of face images are projected into the significant eigenvectors to obtain a set called eigenfaces. Every face has a contribution to the eigenfaces obtained. The best M eigenfaces from a M dimensional subspace is called "face space".

Each individual face can be represented exactly as the linear combination of "eigenfaces" or each face can also be approximated using those significant eigenfaces obtained using the most significant eigen values.

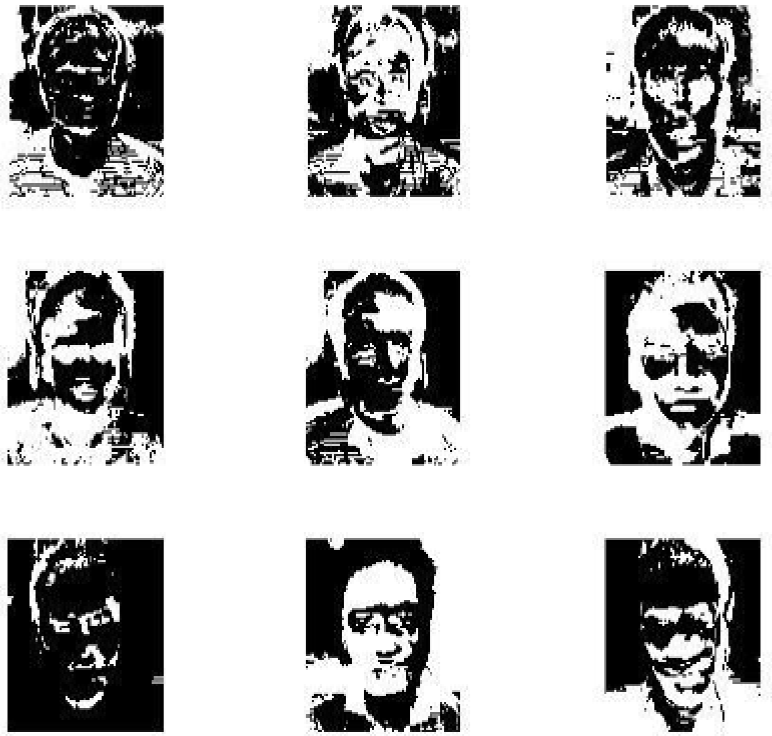


Figure 4.1 Eigenfaces

Now the test image subjected to recognition is also projected to the face space and then the weights corresponding to each eigenface are found out. Also the weights of all the training images are found out and stored. Now the weights of the test image are compared to the set of weights of the training images and the best possible match is found out. The comparison is done using the "Euclidean distance" measurement. Minimum the distance, maximum is the match.

The approach to face recognition involves the following initialization operations:

- Acquire an initial set of N face images (training images).
- Calculate the eigenface from the training set keeping only the M images that correspond to the highest eigenvalues. These M images define the "facespace". As new faces are encountered, the "eigenfaces" can be updated or recalculated accordingly.
- Calculate the corresponding distribution in M dimensional weight space for each known individual by projecting their face images onto the "face space".
- Calculate a set of weights projecting the input image to the M "eigenfaces".

- Determine whether the image is a face or not by checking the closeness of the image to the “face space”.

- If it is close enough, classify, the weight pattern as either a known person or as an unknown based on the Euclidean distance measured.

- If it is close enough then cite the recognition successful and provide relevant information about the recognized face form the database which contains information about the faces.

4.3.1. Mathematical approach

Let Γ1, Γ2, …, Γm be the set of train images.

Average face of set can be defined as $\Psi = (1/M)\sum_{n=1}^{M}(\Gamma_n)$

Each face differs from the average by the vector Φi = Γi – ψ

When subjected to PCA, this large set of vectors seeks a set of M orthogonal vectors Un, which best describes the distribution of data.

The kth vector Uk is chosen such that

$$\lambda_k=(1/M)\ \sum_{n=1}^{M}[(\mathbf{U}_k)^{\mathrm{T}} * \Phi_n]^2$$

is maximum, subject to

$$(\mathbf{U}_l)^{\mathrm{T}}\ \mathbf{U}_K = \delta_{lk} = \begin{cases} 1, & \text{if l=k} \\ 0, & \text{otherwise} \end{cases}$$

The vector Uk and scalar λk are the eigenvectors and eigenvalues respectively of the covariance matrix

$$\mathbf{C} = (\mathbf{1}/\boldsymbol{M})\ \sum_{n=1}^{M}(\Phi_n)(\Phi_n)^{\mathrm{T}}$$

$$\mathbf{C = AA^T}$$

Where the matrix A= [Φ1 Φ2 ….. ΦM].

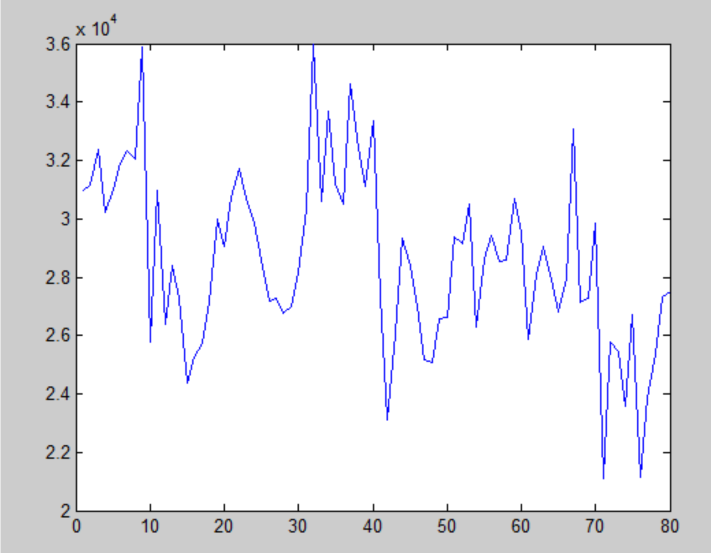


Figure 4.2 Graph of image differences

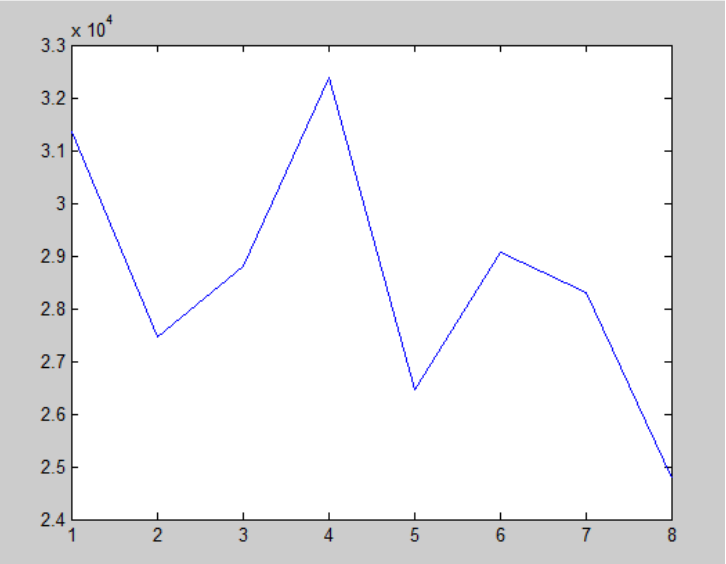


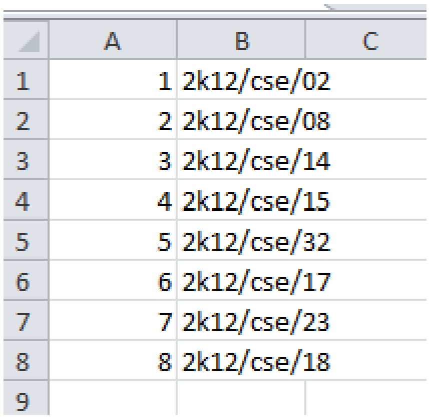
Figure 4.3 Subject wise graph of image difference

4.4. Creation of Excel Sheets

An excel sheet named “Students” is already created that contains the folder number and the roll number of the student whose images are in that folder. When a match is found, the code refers to this excel sheet and finds out the roll number of the student with whom match is found.

Based on the recognized faces, separate excel sheets are created for the students that are present and the students that are absent containing the students' roll numbers in its first column.

A new folder is created in the current directory with the current date in the format DD-Month-YYYY. The two excel sheets are created inside this folder. Name of the excel sheet is kept based on the starting time of the lecture. For example, if a lecture is from 2PM-3PM, the excel sheet will be created with the name "2 PM Present" for students that are present and "2 PM Absent" for students that are absent.



	A	B	C
1	1	2k12/cse/02	
2	2	2k12/cse/08	
3	3	2k12/cse/14	
4	4	2k12/cse/15	
5	5	2k12/cse/32	
6	6	2k12/cse/17	
7	7	2k12/cse/23	
8	8	2k12/cse/18	
9			

Figure 4.4 Students.xls

Chapter 5 SMS Updates (Phase III)

The last phase of our project was to send SMS updates to the teacher containing the list of students that were absent in his/her lecture. Also, SMS are sent to the students that are absent for double-checking.

First of all we configured and tested our GSM modem using HyperTerminal. Then we configured our modem to send messages through Matlab.

5.1. Setting up GSM Modem

Most GSM modems come with simple manual and necessary drivers. We installed our modem's driver and its software. The GSM modem mapped itself as a COM serial port on our computer. The COM serial port was found out through the device manager.

5.2. Using HyperTerminal

Some initial setup is required after HyperTerminal is launched. It involves setting the following fields:

- COM Port: As indicated in the device manager
- Bits per second: 9600
- Data Bits: 8
- Parity: None
- Stop Bits: 1
- Flow Control: Hardware

After this, we are ready to work with AT commands. The first step was to make sure our GSM modem was connected properly and that it understands AT commands. For this, we send an "AT" command and we get a response "OK". This meant our GSM modem is configured properly and it understands AT commands and it is ready to be used for sending SMS.

5.2.1. Initial Setup AT Commands

We check the status of our GSM modem. Some commands that we used are as follows:

- AT : Returns “OK” to confirm that modem is working.
- AT+CPIN=”xxxx” : To enter the PIN for the SIM (if enabled).
- AT+CREG? : A “0,1″ reply confirms that the modem is connected to GSM network.
- AT+CSQ : Indicates the signal strength, 31.99 is maximum.

5.2.2. Sending SMS using AT Commands

A series of AT commands are used to send text messages.

- AT+CMGF=1: To format SMS as a TEXT message.
- AT+CSCA=”+xxxxx”: This sets the SMS center number.
- To send a SMS, the AT command to use is AT+CMGS.
  - AT+CMGS=”+yyyyy”: The “+yyyyy” is recipient’s mobile number.
  - Press Enter.
  - Write SMS text message here
  - Press Ctrl+Z

The SMS is sent if the above steps are followed correctly.

5.3. Using Matlab

After successfully testing the GSM modem in HyperTerminal, it was time to configure it through Matlab. We developed a code that would run directly after the face recognition phase. It access the modem and the excel sheets created in the previous phase and sends messages accordingly.

A SMS update is sent to the teacher containing the current date and time and the list of students who are absent in the lecture. Also, SMS are sent to the students that are absent for cross-checking.

5.3.1. Procedure

1. Excel sheet of the students that are absent is loaded and read.

2. Roll numbers and mobile numbers of the students is extracted from the excel sheet.

3. A SMS is sent to the concerned teacher with the list of the absent students using a GSM Modem.

4. Another set of SMS are sent to the students that are absent.

Chapter 6 Results

Here I present my results.

6.1. Results of Phase 1

Here I discuss different cases for the face detection phase.

6.1.1. Case 1: Frontal face

In this case the person is facing the camera directly. In other words, all the facial features are available to the algorithm.

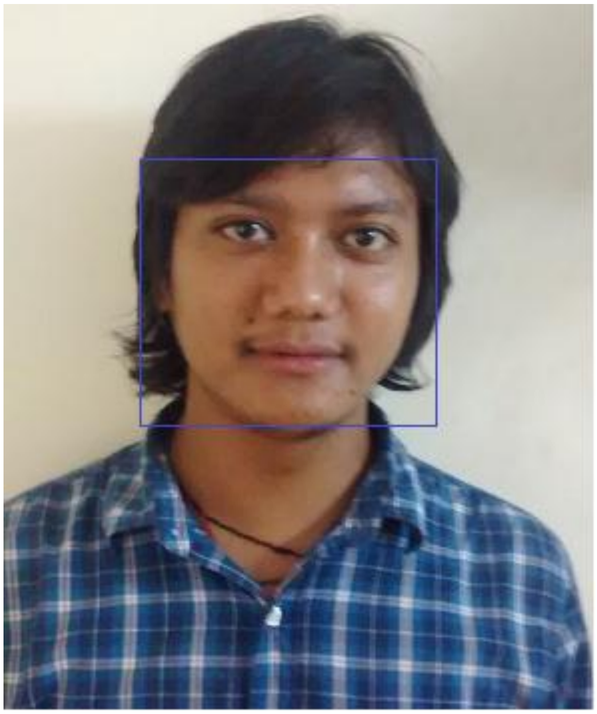


Figure 6.1 Frontal face

6.1.2. Case 2: Frontal face with spectacles

In this case the person is wearing spectacles and is facing the camera directly. The face is not detected in this case, so we made changes in the haar cascade library and increased the threshold value to counter the effect of spectacles.

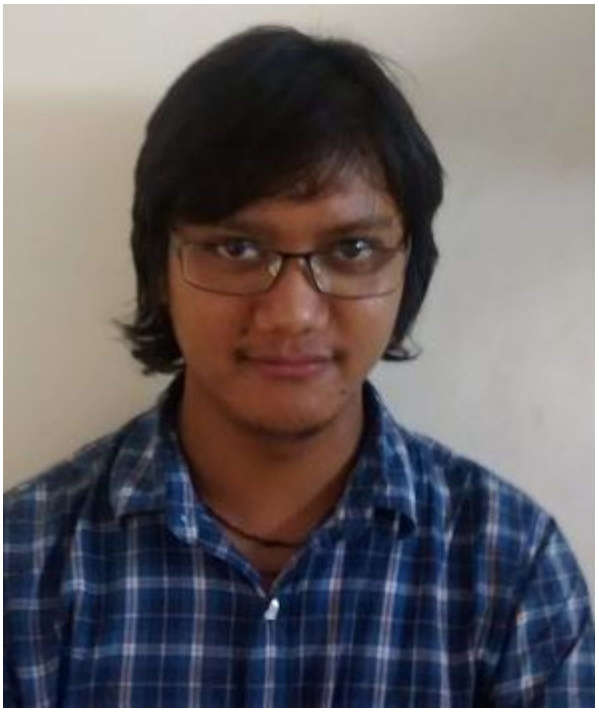


Figure 6.2 Frontal face with spectacles

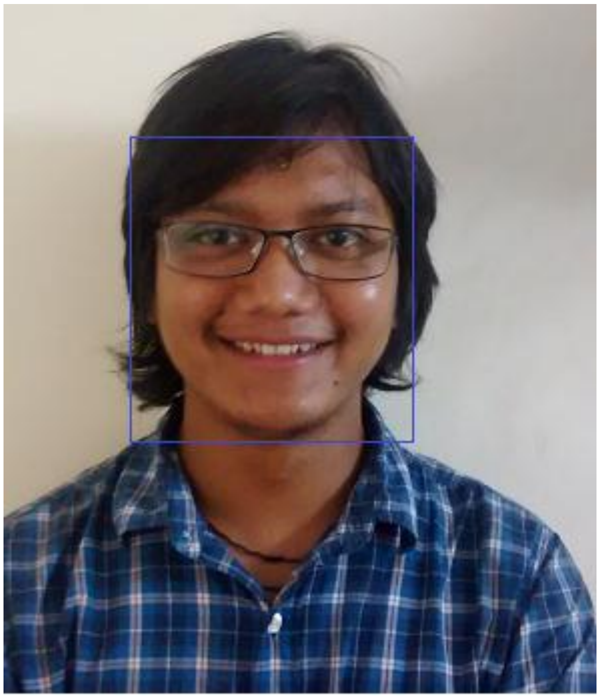


Figure 6.3 Frontal face with spectacles

6.1.3. Case 3: Multiple faces with spectacles

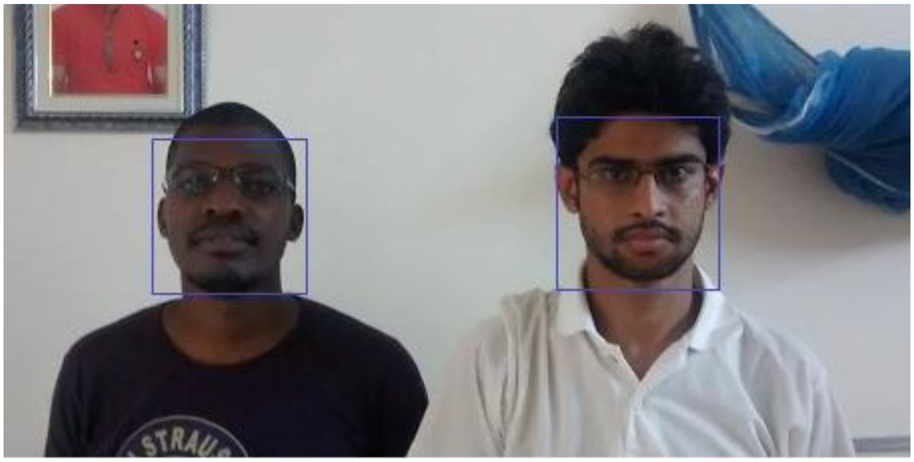


Figure 6.4 Multiple faces with spectacles

6.1.4. Case 4: Multiple faces without spectacles

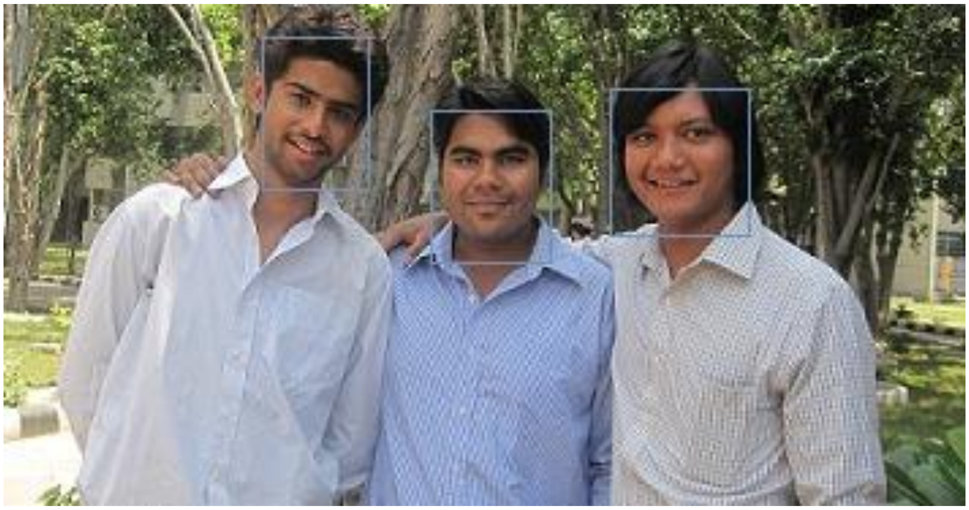


Figure 6.5 Mutliple faces without spectacles

6.1.5. Case 5: Face with cap

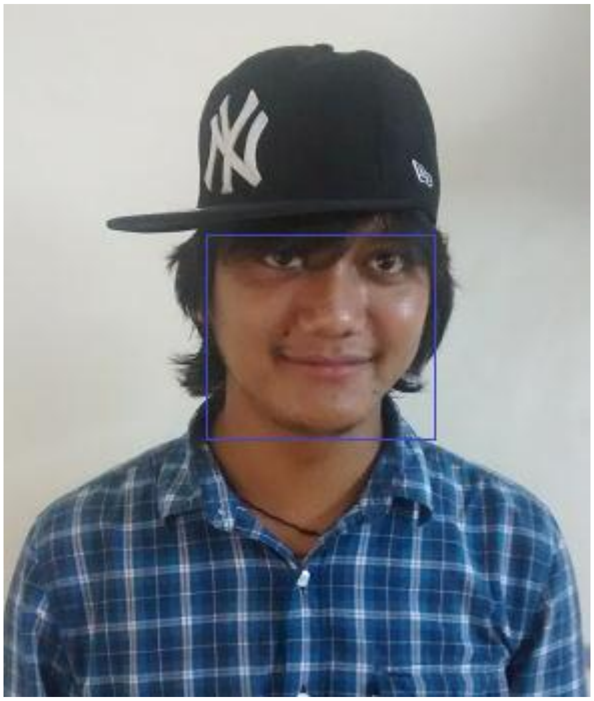


Figure 6.6 Face with cap

6.1.6. Case 6: Face with tilted head

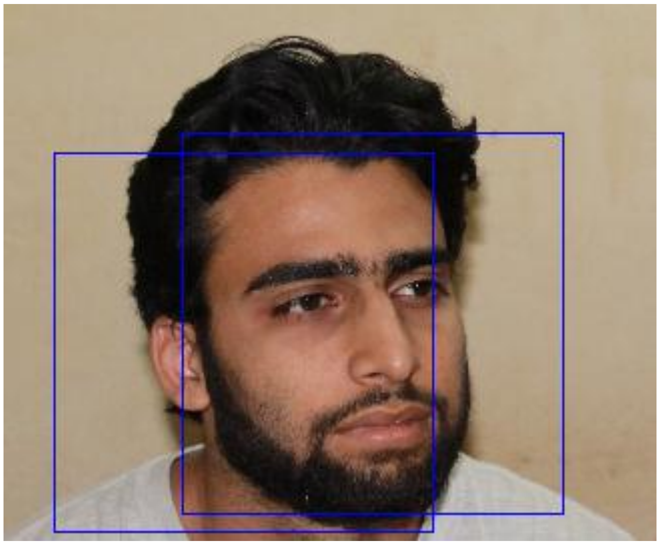


Figure 6.7 Face with tilted head

6.1.7. Case 7: Face with black background

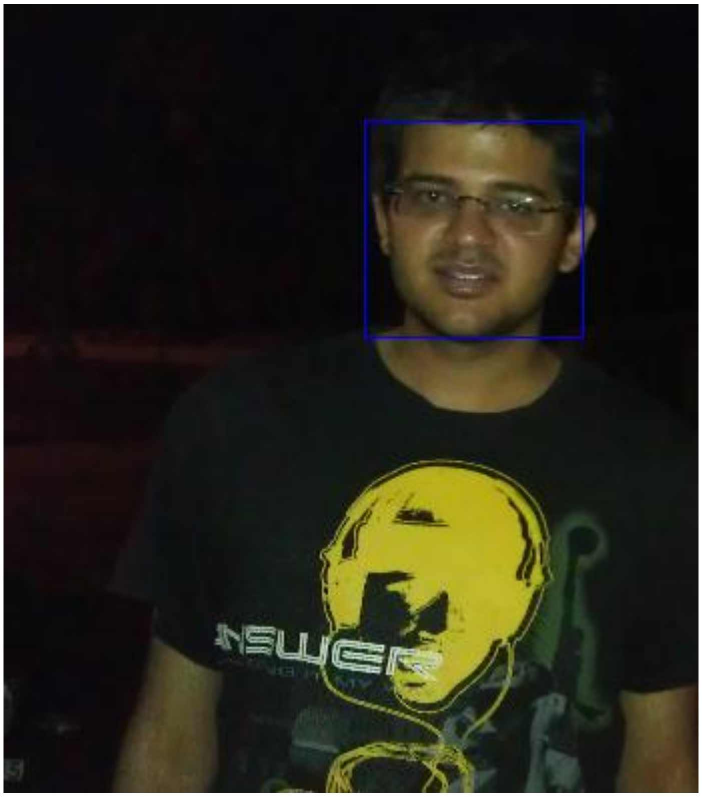


Figure 6.8 Face with black background

6.1.8. Case 8: Upside down face

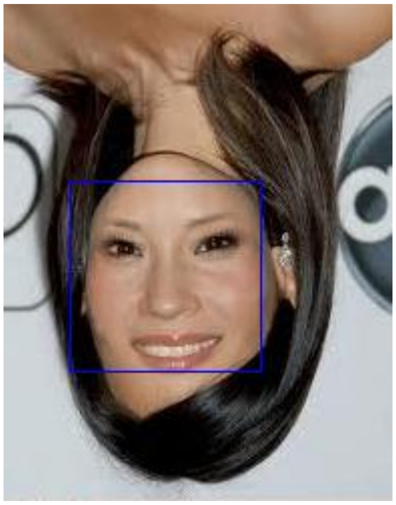
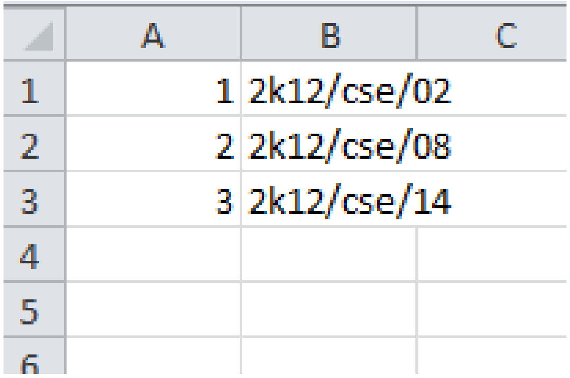


Figure 6.9 Upside down face

6.2. Results of Phase 2 and Phase 3

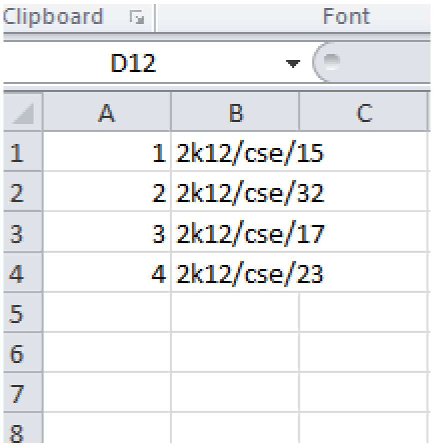
6.2.1. Creation of present excel sheet



	A	B	C
1	1	2k12/cse/02	
2	2	2k12/cse/08	
3	3	2k12/cse/14	
4			
5			
6			

Figure 6.10 present.xls

6.2.2. Creation of absent excel sheet



	A	B	C
1	1	2k12/cse/15	
2	2	2k12/cse/32	
3	3	2k12/cse/17	
4	4	2k12/cse/23	
5			
6			
7			
8			

Figure 6.11 absent.xls

Chapter 7 Conclusion and Future Scope

7.1. Conclusion

I tried to build a reliable automated attendance system using Viola Jones algorithm for face detection and Eigen faces for face recognition.

By observing the performance of our face detector on a number of test images I have noticed a few different failure modes.

The face detector was trained on frontal, upright faces. The faces were only very roughly aligned so there is some variation in rotation both in plane and out of plane. Informal observation suggests that the face detector can detect faces that are tilted up to about ±15 degrees in plane and about ±45 degrees out of plane (toward a profile view). The detector becomes unreliable with more rotation than this.

I have also noticed that harsh backlighting in which the faces are very dark while the background is relatively light sometimes causes failures.

Finally, our face detector fails on significantly occluded faces. If the eyes are occluded for example, the detector will usually fail. The mouth is not as important and so a face with a covered mouth will usually still be detected.

Although PCA is the simplest and fastest approach which can be used for face recognition, but it requires full frontal display of faces. PCA is also very sensitive to lighting conditions and positions of faces and it considers every face in the database as a different image and are not classified in classes.

Considering all the limitations, I would like to conclude that our smart attendance system still requires considerable amount of refinement. Presently it has an accuracy varying between 45 to 55 percent.

7.2. Future Scope of Work

In this project, I have used Viola Jones algorithm for face detection and Eigenfaces for face recognition. These two are the fastest and basic algorithms for both face detection and recognition. Due to the time constraint, I was forced to use these algorithms. As a result, the detection and recognition rate of our project is quite low.

This project has a very good scope of improvement. The detection and recognition rates can be considerably improved by using other algorithms like Linear discriminant analysis (LDA) instead of Principal Component Analysis (PCA) and neural networks.

LDA overcomes several limitations of PCA. It does not consider every face in the database as different images and it considers the classification of images in different classes. Neural network is a very efficient way of carrying out face recognition. The algorithm tries to think like a human brain. It tries to work like the neurons of the human brain, hence the name neural network. In this, the networks can be very effectively and efficiently trained to achieve better results in face recognition. Although it requires more time and computational power, it is a better choice for face recognition.

References

[1] Robust Real-Time Face Detection by Paul Viola and Michael J. Jones

[2] Simple Face-detection Algorithm Based on Minimum Facial Features by Yao-Jiunn Chen and Yen-Chun Lin

[3] Face Recognition Using Principal Component Analysis by S. Thakur

[4] Detecting Faces in Images: A Survey by Ming-Hsuan Yang, David J. Kriegman and Narendra Ahuja

[5] Face Recognition Using Eigenfaces by Matthew A. Turk and Alex P. Pentland

[6] http://www.mathworks.com/matlabcentral/fileexchange/29437-viola-jones-object-detection

[7] http://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework

[8] http://en.wikipedia.org/wiki/Principal_component_analysis

[9] http://www.cs.ait.ac.th/~mdailey/matlab