VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

A Project Report on

Smart Attendance Management System

Submitted in partial fulfillment of the requirements for the VII Semester of degree of ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi*

Submitted by

Kavya G P **1RN17IS044** **N Sanjana Shree** **1RN17IS058**
Nagashree T S **1RN17IS059**

Under the Guidance of

Mr. Ravi Kumar S G
Designation of the Guide
Department of ISE

Department of Information Science and Engineering

RNS Institute of Technology

Dr. Vishnuvaradhana Road, Rajarajeshwari Nagar post,
Channasandra, Bengaluru-560098

2020-2021

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhana Road, RajarajeshwariNagar post,
Channasandra,Bengaluru - 560098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work phase1 entitled ***SMART ATTENDANCE MANAGEMENT SYSTEM*** has been successfully completed by **Kavya G P (1RN17IS044), N Sanjana Shree(1RN17IS058), Nagashree T S (1RN17IS059)** bonafide students of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during academic year **2020-2021**. The project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

Mr. Ravi Kumar S G	**Dr. Suresh L**	**Dr. M K Venkatesha**
Project Guide	Professor and HoD	Principal
Assistant Professor	Department of ISE	RNSIT
Department of ISE	RNSIT	

External Viva

Name of the Examiners	**Signature with Date**
1. _____	**1.**_____
2. _____	**2.**_____

DECLARATION

We, **KAVYA G P [USN: 1RN17IS044], N SANJANA SHREE [USN: 1RN17IS058], NAGASHREE T S [USN: 1RN17IS059],**students of VII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Project phase1 work entitled ***Smart Attendance Management System*** has been carried out by us and submitted in partial fulfillment of the requirements for the *VII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belgaum* during academic year 2020-2021.

Place : Bengaluru

Date :

KAVYA G P **1RN17IS044**

N SANJANA SHREE **1RN17IS058**

NAGASHREE T S **1RN17IS059**

ABSTRACT

Traditional student attendance marking technique is often facing a lot of trouble. The face recognition student attendance system emphasizes its simplicity by eliminating classical student attendance marking technique such as calling student names or checking respective identification cards. There are not only disturbing the teaching process but also causes distraction for students during exam sessions. Apart from calling names, attendance sheet is passed around the classroom during the lecture sessions. The lecture class especially the class with a large number of students might find it difficult to have the attendance sheet being passed around the class.

Thus, face recognition student attendance system is proposed in order to replace the manual signing of the presence of students which are burdensome and causes students get distracted in order to sign for their attendance. Furthermore, the face recognition based automated student attendance system able to overcome the problem of fraudulent approach and lecturers do not have to count the number of students several times to ensure the presence of the students

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

ACKNOWLEDGMENT

At the very onset we would like to place our gratefulness to all those people who helped us in making the final year project phase 1 a successful one.

Coming up, this final year project phase 1 to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is they who guided me in the right direction.

First of all, we would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, we express sincere gratitude to our Director **Dr. H N Shivashankar and** the Principal **Dr. M K Venkatesha,** for providing us all the facilities.

We are extremely grateful to our own and beloved Professor and Head of Department of Information science and Engineering, **Dr. Suresh L** , for having accepted to patronize us in the right direction with all his wisdom.

We place my heartfelt thanks to **Mr. Ravi kumar S G,** Assistant Professor, Department of Information Science and Engineering for having guided us during the final year project and all the staff members of the department of Information Science and Engineering for helping at all times.

We also thank our final year project coordinator **Dr. Prakasha S ,** Associate Professor, Department of Information Science and Engineering. We would thank our friends for having supported us with all their strength and might. Last but not the least, We thank our parents for supporting and encouraging us throughout. We have made an honest effort in this assignment.

Kavya G P **1RN17IS044**

N Sanjana Shree **1RN17IS058**

Nagashree T S **1RN17IS059**

TABLE OF CONTENTS

List of Figures

ABBREVATIONS

GUI	Graphival User Interface
DFD	Data Flow Diagram
RFID	Radio Frequency Identification
NFC	Near Field Communication
PCA	Principle Component Analysis

Chapter 1

Introduction

1.1 Biometrics

Biometrics is used in the process of authentication of a person by verifying or identifying that a user requesting a network resource is who he, she, or it claims to be, and vice versa. It uses the property that a human trait associated with a person itself like structure of data with the incoming data we can verify the identity of a particular person. There are many types of biometric system like detection and recognition, iris recognitionetc., these traits are used for human identification in surveillance system, criminal identification, face details etc. By comparing the existing fingerprint recognition.

.

1.2 Background

Attendance plays a pivotal role in determining academic performance of children and youth in schools and colleges. The regularity of attendance shows that the students are less likely to engage in delinquent or destructive behavior. Chronic absence increases the risk of school failure and early dropout. Manual maintenance of attendance is inefficient due to the following reasons:

• It takes away a lot of lecture hours

• Prone to proxies or impersonations

To resolve this problem of attendance, many attendance management systems have been introduced in recent years. Jain et. al developed a desktop based application in which students are given attendance by clicking a checkbox next to their name and then by clicking the register button to mark their presence. In 2013, Bhalla et. al have proposed blue-tooth based attendance system. Application software installed in mobile phone enables to register the attendance via blue-tooth connection and transfer the notification to the instructor. Works of propose a system for employee attendance based on the fingerprint. The system compares one fingerprint template with all previously stored in the database. Joardar et. al has developed an attendance system based on the palm dorsal

subcutaneous vein pattern of individuals. However, most of these systems have respective limitations in portability, accessibility, authenticity or cost. So an endeavor to overcome the shortcomings of the respective systems leads to the development of a Smart Attendance Monitoring System (SAMS) based on face recognition. Unlike other biometric and non-biometric means of attendance system, face recognition technology stands tall with its unique advantages. Every student has a separate facial identity and it cannot be faked by mere proxies.

Face recognition is a biometric technique which involves determining if the image of the face of any given person matches any of the face images stored in a database. This problem is hard to solve automatically due to the changes that various factors, such as facial expression, aging and even lighting, can cause on the image. Among the different biometric techniques facial recognition may not be the most reliablebut it has several advantages over the others. It is widely used in various areas such as security and access control, forensic medicine, police controls and in attendance management system. The various techniques for marking attendance are:

1) Signature based System
2) Fingerprint based System
3) Iris Recognition
4) RFID based System
5) Face Recognition

1.3 Face Recognition

Face recognition is a biometric technique which involves determining if the image of the face of any given person matches any of the face images stored in a database. This problem is hard to solve automatically due to the changes that various factors, such as facial expression, aging and even lighting, can cause on the image. Among the different biometric techniques facial recognition may not be the most reliable but it has several advantages over the others. It is widely used in various areas such as security and access control, forensic medicine, police controls and in attendance management system. The various techniques for marking attendance are:

- Signature based System
- Fingerprint based System
- Iris Recognition

- RFID based System
- Face Recognition

Amongst the above techniques, Face Recognition is natural, easy to use and does not require aid from the test subject.[1]. It is a series of several related problems which are solved step by step:

- To capture a picture and discern all the faces in it.
- Concentrate on one face at a time and understand that even if a face is turned in strange direction or in bad lighting, it is still the same person.
- Determine various unique features of the face that can help in distinguishing it from the face of any other person. These characteristics could be the size eyes, nose, length of face, skin colour, etc.
- Compare these distinctive features of that face to all the faces of people we already know to find out the person's name.

Our brain, as a human is made to do all of this automatically and instantaneously. Computers are incapable of this kind of high-level generalization, so we need to teach or program each step of face recognition separately. Face recognition systems fall into two categories: verification and identification. Face verification is a 1:1 match that compares a face image against a template face images, whose identity is being claimed. On the contrary, face identification is a 1:N problem that compares a query face image.

Chapter 2

Literature Survey

NFC Based Mobile Attendance System with Facial Authorization on Raspberry Pi and Cloud Server

Attendance system is a system that is used to track the attendance of a particular person and it is applied in many institutions. However, many systems for taking attendance has drawbacks, such as the traditional way has drawback in the data of the attendance that the list is hard to reuse, a biometric attendance system has drawback of the existence of human error such as fingerprint scans are not acceptable, due to the condition of a wet finger, dirty, too dry or peeling fingertips. In this paper, we propose mobile attendance system with NFC and face authorization to add security feature using Raspberry Pi and afford possibility to store the data in cloud. This paper first review the related works in the field of attendance management, NFC, face authorization, microcomputer and cloud storage. Then, it introduces our methodology and design system structure and plan. The result of this research is the system that reduces the amount of paper usage; eliminate the time and effort wasted in taking attendances by Mobile-based attendance system.

Published in: 2018 6th International Conference on Cyber and IT Service Management (CITSM)

Date of Conference: 7-9 Aug. 2018

Date Added to IEEE *Xplore*: 28 March 2019

Authors:

Siti Ummi Masruroh

Department of Informatics, FST UIN Syarif Hidayatullah, Jakarta, Indonesia

Andrew Fiade

Department of Informatics, FST UIN Syarif Hidayatullah, Jakarta, Indonesia

Imelda Ristanti Julia

Department of Informatics, FST UIN Syarif Hidayatullah, Jakarta, Indonesia

Motion Based Attendence System in Real-Time Environment for Multimedia Application

Modern class rooms are equipped with various smart devices that have supporting software to improve the class rooms. However, it is often seen that the precious time of class room is wasted on taking attendance, or the class may face various interruption due to late entries of students, and this attendance procedure is done repeatedly. Also performance is affected by the presence of the student in theinstitute. Conventional methods for attendance taking are by calling student name or by sensor based card (RFID sensor) or biometric fingerprint based attendance system, but all of these method are inefficient as they can't tell whether the student has attended full class or not. Thus there is need of attendance management system which is totally motion based that will assist the faculties and institute for marking the presence of the student attendance. Motion based attendance system that assist recording and management of attendance that will be used for various institutions and the colleges.

Published in: 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)

Date of Conference: 11-12 Jan. 2018

Date Added to IEEE *Xplore*: 23 August 2018

Authors:

Mayank Yadav

Amity School of Engineering and Technology, Amity University Uttar Pradesh, Department of Computer Science and Engineering, Noida, India.

Anmol Aggarwal

Amity School of Engineering and Technology, Amity University Uttar Pradesh, Department of Computer Science and Engineering, Noida, India.

Nitin Rakesh

Amity School of Engineering and Technology, Amity University Uttar Pradesh, Department of Computer Science and Engineering, Noida, India.

An efficient automated attendance management system based on Eigen Face recognition

A Face recognition system is an application of computer vision and image processing which is capable of performing two major tasks of identifying and verifying a person from an image or a video database. The objective of this paper is to automate the attendance system by integrating the face recognition technology using Eigen Face database and PCA algorithm with Matlab GUI. In Conventional attendance system there are several issues like fake attendance, lot of time consumption, manipulation of attendance, information cannot be secure. There are many limitations in implementing face recognition technologies like Image Quality, Image Size, Face angle, varying intensity of light. In order to overcome these issues various techniques like Illumination Invariant, Histogram equalization, PCA are used. By using this system attendance is updated automatically after comparing the detected face with original Eigen database in Excel sheet integrated with Matlab GUI.

Published in: 2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence

Date of Conference: 12-13 Jan. 2017

Date Added to IEEE *Xplore*: 08 June 2017

Authors:

E. Rekha

Department of Electronics Engineering, Amity University, Dubai, UAE P.

Ramaprasad

Department of Electronics Engineering, School of Engineering and Technology, Manipal University Dubai, UAE.

Attendance management system using hybrid face recognition techniques

Attendance recording of a student in an academic organization plays a vital role in judging students performance. As manual labor involved in this process is time consuming, an automated Attendance Management System (AMS) based on face detection and face recognition techniques is proposed in this paper. The system employs modified Viola-Jones algorithm for face detection, and alignment- free partial face recognition algorithm for face recognition. After successful recognition of a student, the system automatically updates the attendance in the excel sheet. The proposed system improves the performance of existing attendance management systems by eliminating manual calling, marking and entry of attendance in institutional websites.

Published in: 2016 Conference on Advances in Signal Processing (CASP)

Date of Conference: 9-11 June 2016

Date Added to IEEE *Xplore*: 17 November 2016

Authors:

Nazare Kanchan Jayant

Dept of ECE, K.S. Institute of Technology, Bangalore-560062, India .

Surekha Borra

Dept of ECE, K.S. Institute of Technology, Bangalore-560062, India.

Class Attendance Management System Using Face Recognition

We are living in a world where everything is automated and linked online. The internet of things, image processing, and machine learning are evolving day by day. Many systems have been completely changed due to this evolve to achieve more accurate results. The attendance system is a typical example of this transition, starting from the traditional signature on a paper sheet to face recognition. This paper proposes a method of developing a comprehensive embedded class attendance system using facial recognition with controlling the door access. The system is based on Raspberry Pi that runs Raspbian (Linux) Operating System installed on micro SD card. The Raspberry Pi Camera, as well as a 5-inch screen, are connected to the Raspberry Pi. By facing the camera, the camera will capture the image then pass it to the Raspberry Pi which is programmed to handle the face recognition by implementing the Local Binary Patterns algorithm LBPs. If the student's input image matches with the trained dataset image the prototype door will open using Servo Motor, then the attendance results will be stored in the MySQL database. The database is connected to Attendance Management System(AMS) web server, which makes the attendance results reachable to any online connected web browser. The system has 95% accuracy with the dataset of 11 person images.

Published in: 2018 7th International Conference on Computer and Communication Engineering (ICCCE)

Date of Conference: 19-20 Sept. 2018

Date Added to IEEE *Xplore*: 19 November 2018

Authors:

Omar Abdul Rhman Salim

Department of Electrical and Computer Engineering, Faculty of Engineering
International Islamic University Malaysia, Kuala Lumpur, Malaysia.

Rashidah Funke Olanrewaju

Department of Electrical and Computer Engineering, Faculty of Engineering
International Islamic University Malaysia, Kuala Lumpur, Malaysia .

Wasiu Adebayo Balogun

Department of Mechatronics, Lagos State Polytechnic, Ikorodu,, Nigeria.

Student Attendance System in Crowded Classrooms Using a Smartphone Camera

To follow the attendance of students is a major concern in many educational institutions. The manual management of the attendance sheets is laborious for crowded classrooms. In this paper, we propose and evaluate a general methodology for the automated student attendance system that can be used in crowded classrooms, in which the session images are taken by a smartphone camera. We release a realistic full-annotated dataset of images of a classroom with around 70 students in 25 sessions, taken during 15 weeks. Ten face recognition algorithms based on learned and handcrafted features are evaluated using a protocol that takes into account the number of face images per subject used in the gallery. In our experiments, the best one has been FaceNet, a method based on deep learning features, achieving around 95% of accuracy with only one enrollment image per subject. We believe that our automated student attendance system based on face recognition can be used to save time for both teacher and students and to prevent fake attendance.

Published in: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)

Date of Conference: 7-11 Jan. 2019

Date Added to IEEE *Xplore*: 07 March 2019

Authors:

Domingo Mery

Ignacio Mackenney

Esteban Villalobos

Chapter 3

Analysis

The problem of face recognition can be stated as follows: Face Recognition human facial features like the mouth, nose and eyes in a full frontal face image. We will be adapting a multi-step process in order to achieve the goal. To detect the face region we will be using a skin-color segmentation method. Morphological techniques will be adapted to fill the holes that would be created after the segmentation process. From the skeletonization process, a skeleton of the face will be obtained from which face contour points could be extracted. Facial features can be located in the interior of the face contour. We will use several different facial-images to test our method.

3.1 Challenges:

- **Illumination**

  For instance, a slight change in lighting conditions has always been known to cause a major impact on its results. If the illumination tends to vary, then; even if the same individual gets captured with the same sensor and with an almost identical facial expression and pose, the results that emerge may appear quite different.

- **Background**

  The placement of the subject also serves as a significant contributor to the limitations. A facial recognition system might not produce the same results outdoors compared to what it produces indoors because the factors - impacting its performance - change as soon as the locations change. Additional factors, such as individual expressions, aging etc. contribute significantly to these variations.

- **Pose**

  Facial Recognition Systems are highly sensitive to pose variations. The movements of head or differing POV of a camera can invariably cause changes in face appearance and generate intra- class variations making automated face recognition across pose a tough nut to crack.

- **Occlusion**

  Occlusions of the face such as beard, moustache, accessories (goggles, caps, mask etc.) also meddle with the evaluation of a face recognition system. Presence of such components make the subject diverse and hence it becomes difficult for the system to operate in a non-simulated environment.

- **Expressions**

  Another significant factor which needs to be taken into account is different expressions of the same individual. Macro and micro expressions find their place on someone's face due to changes in one's emotional state and in the wake of such expressions - which are many - the efficient recognition becomes difficult.

- **Complexity**

  Existing state-of- the-art methods of facial recognition rely on 'too-deep' Convolutional Neural Network (CNN) architecture which are very complex and unsuitable for real-time performance on embedded devices.

  An ideal Face recognition system should be tolerant to variations in illumination, expression, pose and occlusion. It should be scalable to large number of users with need for capturing minimal images during registration while doing away with complex architecture at the same time.

3.2 Problem Statement

Recognizing faces in computer vision is a challenging problem. The illumination problem[3], the pose problem, scale variability, low quality image acquisition, partially occluded faces are some examples of the issues to deal with. Thus face recognition algorithms must exhibit robustness to variations in the above parameters. The existing techniques do not perform well in cases of different illumination, background or rotation. Thus there is a need to address the above mentioned disadvantages. The project aims to design and implement a system which is less sensitive to Illumination, is rotation invariant, scale invariant and robust enough to be implemented in practical applications.

3.3 Objectives

The overall objective is to develop an automated class attendance management system comprising of a desktop application working in conjunction with a mobile application to perform the following tasks:

- To detect faces real time.
- To recognize the detected faces by the use of a suitable algorithm.
- To update the class attendance register after a successful match.
- To design architecture that constitutes the various components working harmoniously.

3.4 Methodology

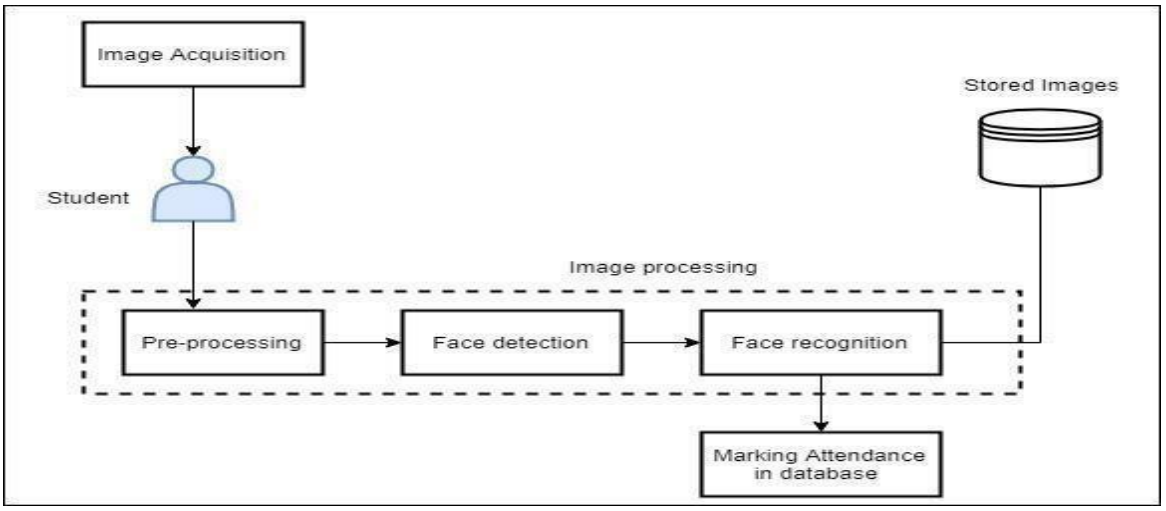

Figure 3.1 : working model of smart attendance system

The System process can be separated into three working modules. They are face representation, feature extraction and classification. The first and foremost task is modelling a face. The way is face is represented determines the next two steps. The image acquired is transformed to match the positions of images already present. In feature extraction the features of the face are mapped as histograms with gradients and they are stored as binary values. The final step is recognizing a familiar face. The system compares the face seen in the camera with records that are already stored. The method proposed in this paper is marking attendance using face recognition technique. As shown in Fig - 1, the attendance is recorded by using a camera that will stream video of students, detect the faces in the image and compare the detected faces with the student database and mark the attendance. The attendance gets marked in a spreadsheet which gets converted into PDF file which is mailed to the concerned e-mail Ids.

The project has two main parts:

- Development of Face Recognition System.
- Development of Attendance System.

Face recognition is achieved using machine learning and the basic pipeline used for it is as follows:

1. Finds face in an image.
2. Analyses facial features.
3. Compares against known faces and makes a prediction.

Development of complete attendance system is achieved using UI and Android application. Here the application takes data like subject details, faculty details, date and time and provides a click to start the attendance. The images of students are clicked and sent to Linux server where python script runs to mark attendance and generate spreadsheet and PDF file which is then mailed.

Possible Outcomes:

- Face Detection
- Attendance in Excel file.
- Weekly report send to mail.
- Individual Attendance marking.

RGB Image to Grayscale:

An RGB image can be viewed as three images(a red scale image, a green scale image and a blue scale image) stacked on top of each other. In MATLAB, an RGB image is basically a M*N*3 array of colour pixel, where each colour pixel is a triplet which corresponds to red, blue and green colour component of RGB image at a specified spatial location. Similarly, A Grayscale image can be viewed as a single layered image. In MATLAB, a grayscale image is basically M*N array whose values have been scaled to represent intensities. In MATLAB, there is a function called rgb2gray() is available to convert RGB image to grayscale image. Here we will convert an RGB image to grayscale image without using rgb2gray() function. Our key idea is to convert an RGB image pixel which a triplet value corresponding to red, blue and green colour component of an image at a specified spatial location to a single value by calculating a weighted sum of all three colour component.

Algorithm for conversion:

- Read RGB colour image into MATLAB environment.
- Extract Red, blue and green colour components from RGB image into 3 different 2-D matrices
- Create a new matrix with the same number of rows and columns as RGB image, containing all zeros.

Convert each RGB pixel values at location (i, j) to grayscale values by forming a weighted sum of the Red, Green, and Blue colour components and assign it to corresponding location (i, j) in new matrix.

3.5 Sofware Component

Open CV

It is a library of programming1functions mainly aimed at real-time1computer vision. It is developed by Intel research center and subsequently supported by1Willow Garage and now maintained by itseez. It is written in C++ and its primary interface is also in C++. Its binding is in Python, Java, and Mat lab. OpenCV runs on a variety of platform i.e. Windows, Linux, and MacOS, openBSD in desktop and Android, IOS and Blackberry in mobile. It is used in diverse purpose for facial recognition, gesture recognition, object identification, mobile robotics, segmentation etc. It is a combination of OpenCV C++ API and Python language. In our project we are using OpenCV version 2 OpenCV is used to gesture control to open a camera and capture the image. It is also used in the image to text and voice conversion technique.

Figure 3.2: Open CV

Chapter 4

System Design

4.1 System Architecture

The Architecture Diagram depicts the overall structure of the software application or model that is to be created or already created architectural diagram. It uses information flow characteristics and maps them into the program structure. The system architecture is shown in the figure 4.1 below.

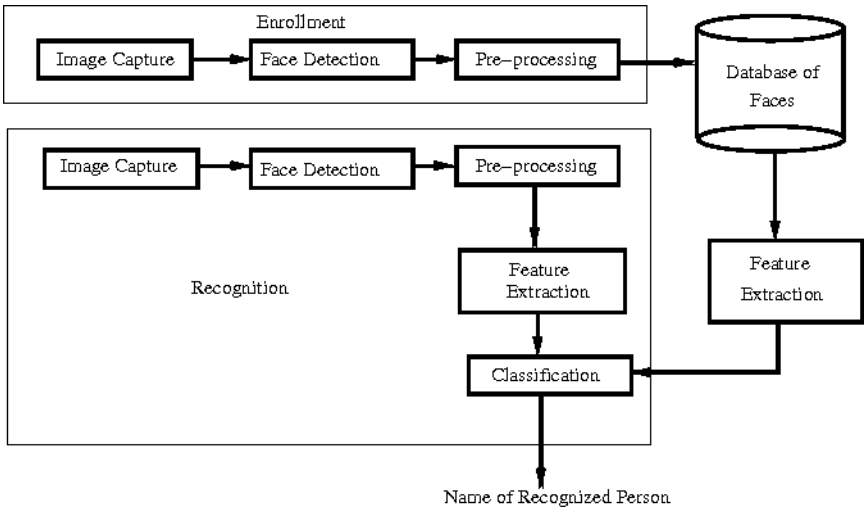

Figure 4.1 Basic System Architecture

The System process can be separated into three working modules. They are face representation, feature extraction and classification. The first and foremost task is modelling a face. The way is face is represented determines the next two steps. The image acquired is transformed to match the positions of images already present.

In feature extraction the features of the face are mapped as histograms with gradients and they are stored as binary values. The final step is recognizing a familiar face. The system compares the face seen in the camera with records that are already stored.The method proposed in this paper is marking attendance using face recognition technique. The attendance is recorded by using a camera that will stream video of students, detect the faces in the image and compare the detected faces with the student database and mark the attendance. The attendance gets marked in a spreadsheet which gets converted into PDF file which is mailed to the concerned e-mail Ids.

4.2 Detailed Design

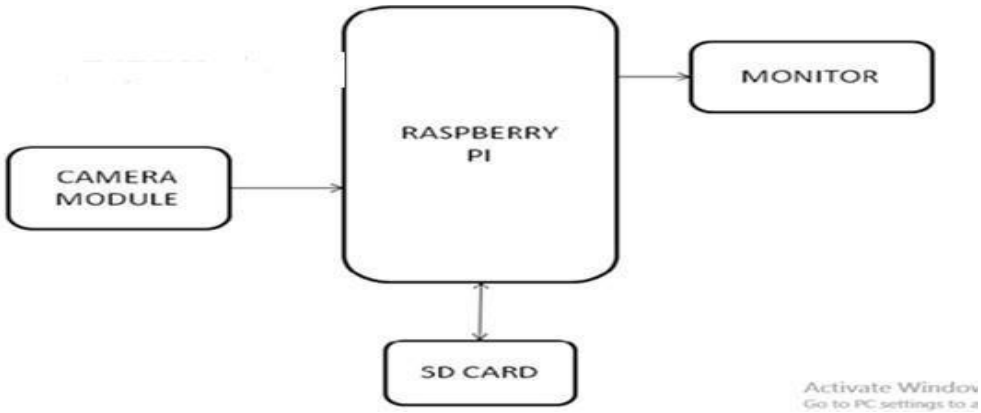

Figure 4.2 Block Diagram

The project has two main parts:

- Development of Face Recognition System.
- Development of Attendance System.

Face recognition is achieved using machine learning and the basic pipeline used for it is as follows:

- Finds face in an image.
- Analyses facial features.
- Compares against known face and make a prediction.

Development of complete attendance system is achieved using UI and Android application. Here the application takes data like subject details, faculty details, date and time and provides a click to start the attendance. The images of students are clicked and sent to Linux server where python script runs to mark attendance and generate spreadsheet and PDF file which is then mailed.

POSSIBLE OUTCOMES:

- Face Detection
- Attendance in Excel file
- Weekly report send to mail
- Individual Attendance marking.

4.3 Data Flow diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

The DFD may be used to perform a system or software at any level of abstraction. In fact, DFDs may be partitioned into levels that represent increasing information flow and functional detail. Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see primarily three levels in the data flow diagram, which are: 0-level DFD, 1-level DFD, and2-level DFD.

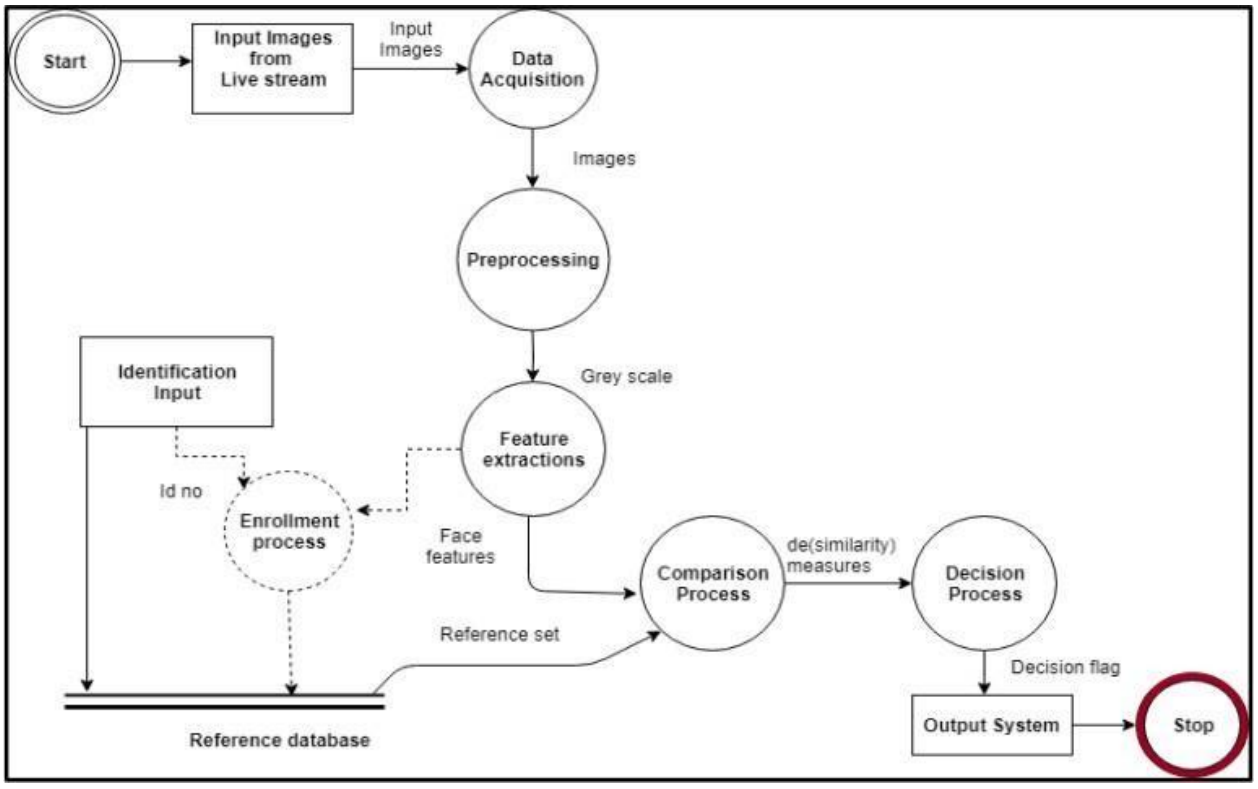

Figure 4.3 Data Flow diagram

4.4 Flow chart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

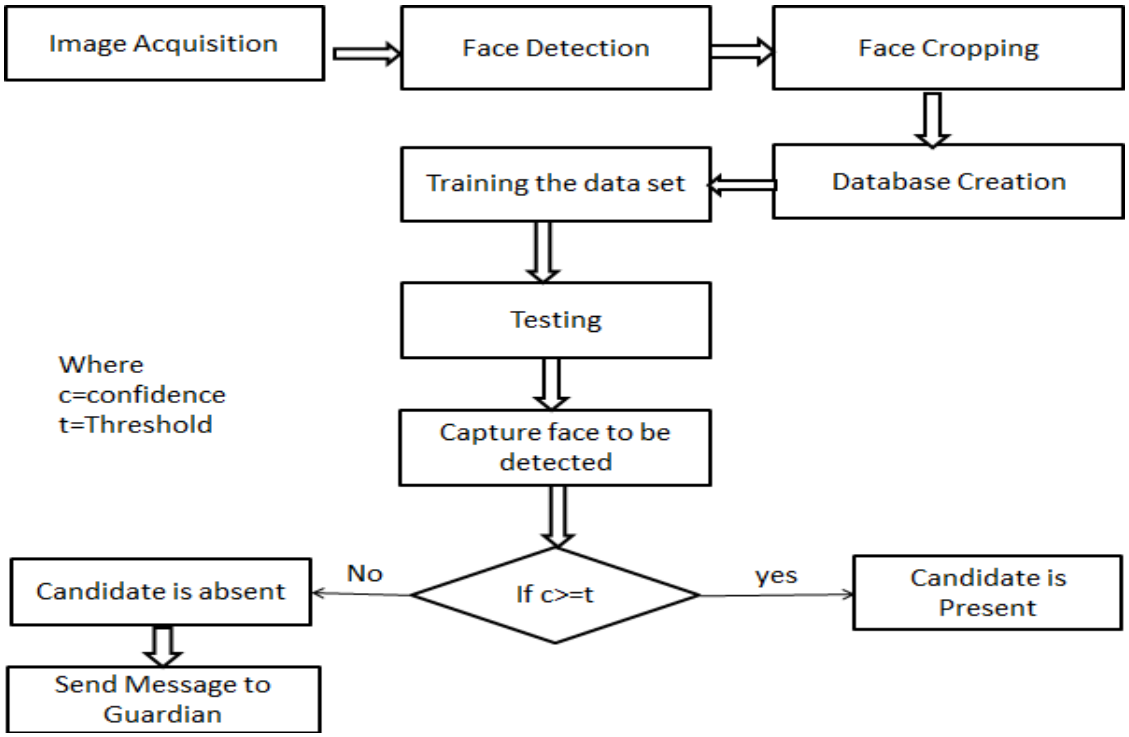

Figure 4.4: Flow Chart of SAMS

The above shown is the flow chart of SAMS where all the different operations performed in this project is shown. The main operations done here are Face Dectection,Training the data set and many more as shown above.The whole workflow of the entire project is confined into a flow chart and representated above.

Chapter 5

Implementation

5.1 Overview of System Implementation

i. RGB Image to Grayscale:

An RGB image can be viewed as three images(a red scale image, a green scale image and a blue scale image) stacked on top of each other. In MATLAB, an RGB image is basically a M*N*3 array of colour pixel, where each colour pixel is a triplet which corresponds to red, blue and green colour component of RGB image at a specified spatial location. Similarly, A Grayscale image can be viewed as a single layered image. In MATLAB, a grayscale image is basically M*N array whose values have been scaled to represent intensities.

In MATLAB, there is a function called rgb2gray() is available to convert RGB image to grayscale image. Here we will convert an RGB image to grayscale image without using rgb2gray() function. Our key idea is to convert an RGB image pixel which a triplet value corresponding to red, blue and green colour component of an image at a specified spatial location to a single value by calculating a weighted sum of all three colour component.

- Read RGB colour image into MATLAB environment.
- Extract Red, blue and green colour components from RGB image into 3 different 2-D matrices
- Create a new matrix with the same number of rows and columns as RGB image, containing all zeros.

Convert each RGB pixel values at location (i, j) to grayscale values by forming a weighted sum of the Red, Green, and Blue colour components and assign it to corresponding location (i, j) in new matrix.

5.2 Algorithms

5.2.1 HAAR CASCADE

Face Detection, a widely popular subject with a huge range of applications. Modern day Smartphones and Laptops come with in-built face detection softwares, which can authenticate the identity of the user. There are numerous apps that can capture, detect and process a face in real time, can identify the age and the gender of the user, and also can apply some really cool filters. The list is not limited to these mobile apps, as Face Detection also has a wide range of applications in Surveillance, Security and Biometrics as well. But the origin of its Success stories dates back to 2001, when Viola and Jones proposed the first ever Object Detection Framework for Real Time Face Detection in Video Footage.

This article is about taking a gentle look on the Viola-Jones Face Detection Technique, popularly known as Haar Cascades, and exploring some of the interesting concepts proposed by them. This piece of work was done long before the Deep Learning Era had even started. But it's an excellent work in comparison to the powerful models that can be built with the modern day Deep Learning

Techniques. The algorithm is still found to be used almost everywhere. It has fully trained models available on GitHub. Haar Cascade is an Object Detection Algorithm used to identify faces in an image or a real time video. The algorithm uses edge or line detection features proposed by Viola and Jones in their research paper "Rapid Object Detection using a Boosted Cascade of Simple Features" published in 2001. The algorithm is given a lot of positive images consisting of faces, and a lot of negative images not consisting of any face to train on them.
The repository has the models stored in XML files, and can be read with the OpenCV methods. These include models for face detection, eye detection, upper body and lower body detection, license plate detection etc. Below we see some of the concepts proposed by Viola and Jones in their research.

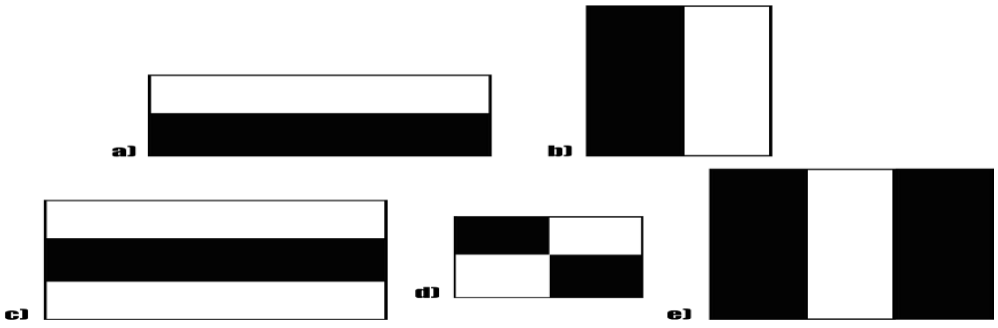

Figure 5.1 A sample of Haar features

The first contribution to the research was the introduction of the haar features shown above. These features on the image makes it easy to find out the edges or the lines in the image, or to pick areas where there is a sudden change in the intensities of the pixels.

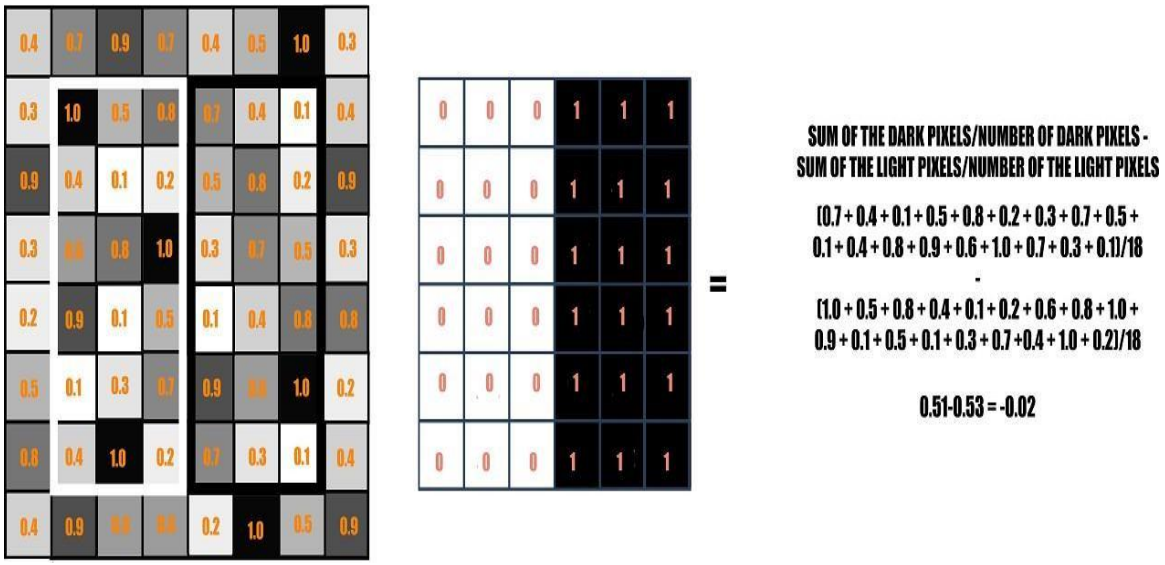

Figure 5.2 Sample of Haar Cascade calculation

The above figure depicts a case where there is a sudden change of pixel intensities moving vertically from the left towards the right in the image.

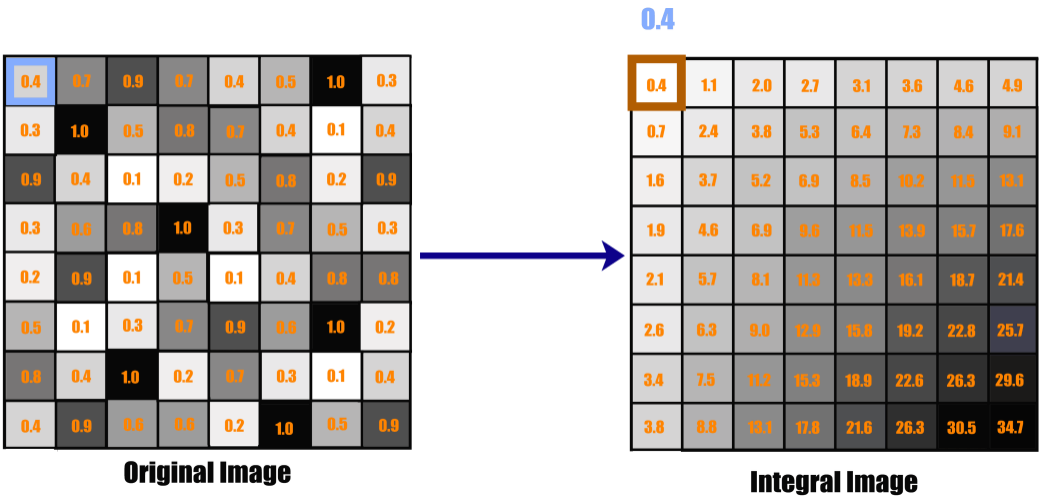

Figure5.3 The above image shows the making of an Integral Image. Each pixel in an Integral image is the sum of all the pixels in its left and above.

To tackle this, they introduced another concept known as The Integral Image to perform the same operation. An Integral Image is calculated from the Original Image in such a way that each pixel in this is the sum of all the pixels lying in its left and above in the Original Image. The calculation of a pixel in the Integral Image can be seen in the above GIF. The last pixel at the bottom right corner of the Integral Image will be the sum of all the pixels in the Original Image.

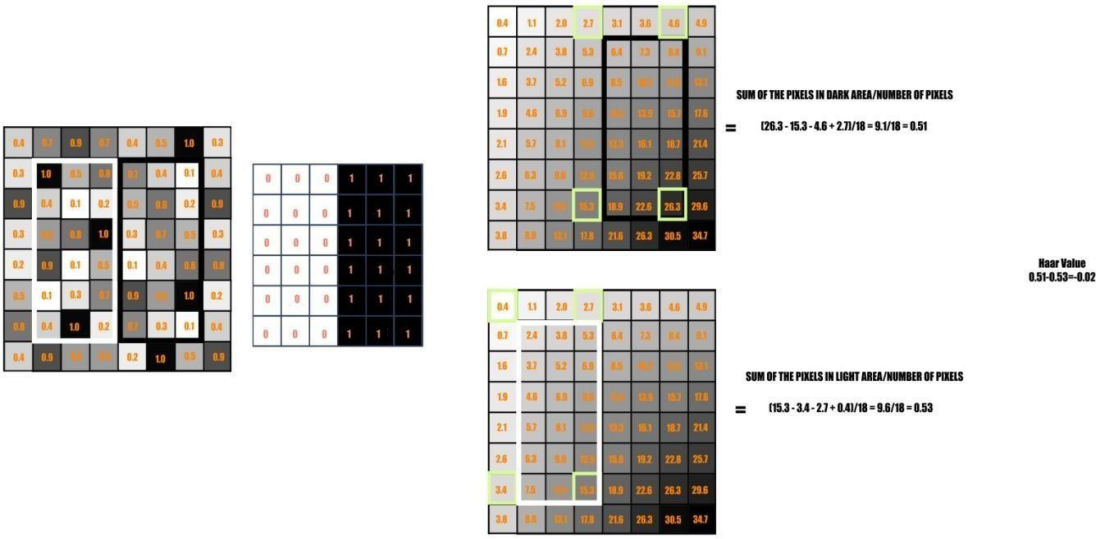

Figure5.4 Integral Image is used here to calculate the haar value.

With the Integral Image, only 4 constant value additions are needed each time for any feature size (with respect to the 18 additions earlier). This reduces the time complexity of each addition gradually, as the number of additions does not depend on the number of pixels enclosed anymore.

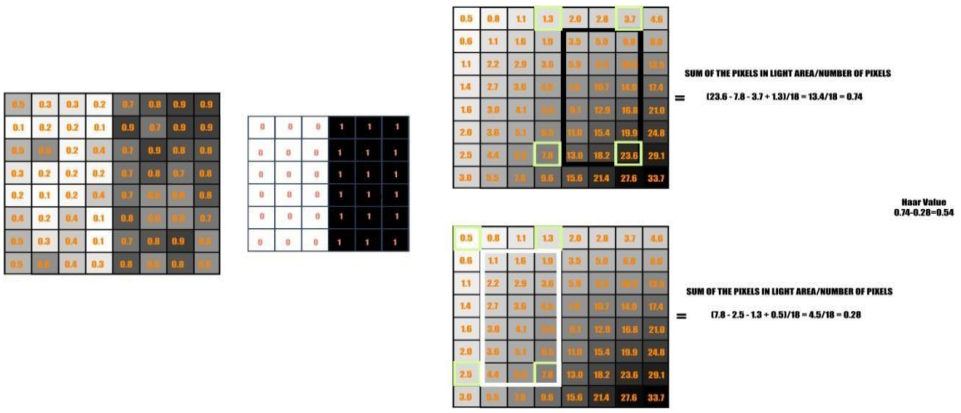

Figure 5.5 Haar calculation from Integral Image. This is a case where there is a sudden change of pixel intensities moving vertically from the left towards the right in the image.

Attentional Cascade

Now comes the Cascading part. The subset of all 6000 features will again run on the training images to detect if there's a facial feature present or not. Now the authors have taken a standard window size of 24x24 within which the feature detection will be running. It's again a tiresome task.

To simplify this, they proposed another technique called The Attentional Cascade. The idea behind this is, not all the features need to run on each and every window. If a feature fails on a particular window, then we can say that the facial features are not present there. Hence, we can move to the next windows where there can be facial features present.

- Features are applied on the images in stages. The stages in the beginning contain simpler features, in comparison to the features in a later stage which are complex, complex enough to find the nitty gritty details on the face. If the initial stage won't detect anything on the window, then discard the window itself from the remaining process, and move on to the next window. This way a lot of processing time will be saved, as the irrelevant windows will not be processed in the majority of the stages.
- The second stage processing would start, only when the features in the first stage are detected in the image. The process continues like this, i.e. if one stage passes, the window is passed onto the next stage, if it fails then the window is discarded.

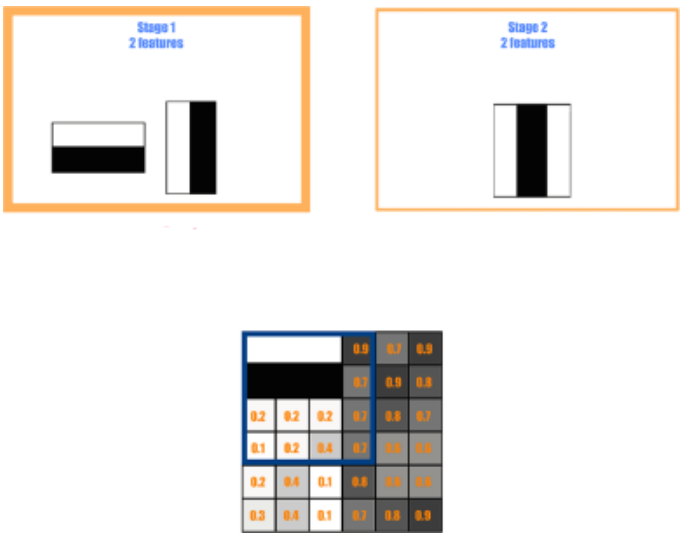

Figure 5.6 A sample 2 stage feature detection

In above figure, the haar features are applied on the image in a 4x4 window. The first stage has 2 simpler features, and the second stage has only 1 complex feature. The first stage is applied first on the 4x4 windows in the image, if it passes, then only the stage is applied.

5.3 Code Snippets

5.3.1 Code for taking of Images

```
def TakeImages():
    Id=(txt.get())
    name=(txt2.get())
    phone=(txt3.get())
    if(is_number(Id) and name.isalpha()):
        cam = cv2.VideoCapture(0)
        harcascadePath = "haarcascade_frontalface_default.xml"
        detector=cv2.CascadeClassifier(harcascadePath)
        sampleNum=0
        while(True):
            ret, img = cam.read()
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = detector.detectMultiScale(gray, 1.3, 5)
            for (x,y,w,h) in faces:
                cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
                #incrementing sample number
                sampleNum=sampleNum+1
                cv2.imwrite("TrainingImage\ "+name +"."+Id +'.'+ str(sampleNum) + ".jpg", gray[y:y+h,x:x+w])
            cv2.imshow('frame',img)
            cv2.imwrite('frame.png',img)
            #wait for 100 miliseconds
            if cv2.waitKey(100) & 0xFF == ord('q'):
                break
            cam.release()
            cv2.destroyAllWindows()
            res = "Images Saved for ID : " + Id +" Name : "+ name
            row = [Id , name, phone]
            with open('StudentDetails\\StudentDetails.csv','a+') as csvFile:
                writer = csv.writer(csvFile)
                writer.writerow(row)
            csvFile.close()
            message.configure(text= res)
        else:
            if(is_number(Id)):
                res = "Enter Alphabetical Name"
                message.configure(text= res)
            if(name.isalpha()):
                res = "Enter Numeric Id"
                message.configure(text= res)
            if(name.isalpha()):
                res = "Enter Numeric No"
                message.configure(text= res)
```

Figure 5.1Code for taking of Images

5.3.2 Traning Images

```
def TrainImages():
    recognizer = cv2.face_LBPHFaceRecognizer.create()#recognizer = cv2.face.LBPHFaceRecognizer_create()#$cv2.createLBPHFaceRecognizer()
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector =cv2.CascadeClassifier(harcascadePath)
    faces,Id = getImagesAndLabels("TrainingImage")
    recognizer.train(faces, np.array(Id))
    recognizer.save("TrainingImageLabel\Trainner.yml")
    res = "Image Trained"#+",".join(str(f) for f in Id)
    message.configure(text= res)
```

Figure 5.2 Code for training Images

5.3.3 Getting of Images and labels

```
def getImagesAndLabels(path):
    #get the path of all the files in the folder
    imagePaths=[os.path.join(path,f) for f in os.listdir(path)]
    #print(imagePaths)

    #create empth face list
    faces=[]
    #create empty ID list
    Ids=[]
    #now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
        #loading the image and converting it to gray scale
        pilImage=Image.open(imagePath).convert('L')
        #Now we are converting the PIL image into numpy array
        imageNp=np.array(pilImage,'uint8')
        #getting the Id from the image
        Id=int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces.append(imageNp)
        Ids.append(Id)
    return faces,Ids
```

Figure 5.3 Code for getting Images and labels

5.3.4 Traking of Images

```
def TrackImages():
    recognizer = cv2.face.LBPHFaceRecognizer_create()#cv2.createLBPHFaceRecognizer()
    recognizer.read("TrainingImageLabel\Trainner.yml")
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath);
    df=pd.read_csv("StudentDetails\StudentDetails.csv")
    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names =  ['Id','Name','Date','Time']
    attendance = pd.DataFrame(columns = col_names)
    now = datetime.datetime.now()
    while True:
        ret, im =cam.read()
       # im = cv2.flip(im, 0)
        gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
        faces=faceCascade.detectMultiScale(gray, 1.2,5)
        for(x,y,w,h) in faces:
            cv2.rectangle(im,(x,y),(x+w,y+h),(225,0,0),2)
            Id, conf = recognizer.predict(gray[y:y+h,x:x+w])
            flag=0
            if(conf < 60):
                ts = time.time()
                date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
                timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
                aa=df.loc[df['Id'] == Id]['Name'].values
                tt=str(Id)+"-"+aa
                attendance.loc[len(attendance)] = [Id,aa,date,timeStamp]
                Id11=[]
                with open('StudentDetails\\StudentDetails.csv', 'a+') as file:
                    reader = csv.reader(file)
                    print(reader)
```

```
            else:
            Id='Unknown'
            tt=str(Id)
        #if(conf > 40):
            noOfFile=len(os.listdir("ImagesUnknown"))+1
            #if flag ==1:
            cv2.imwrite("ImagesUnknown\\Image"+str(noOfFile) + ".jpg", im[y:y+h,x:x+w])
        cv2.putText(im,str(tt),(x,y+h), font, 1,(255,255,255),2)
    attendance=attendance.drop_duplicates(subset=['Id'],keep='first')
    cv2.imshow('im',im)
    if (cv2.waitKey(1)==ord('q')):
        f=open("Database.txt", 'a')
        if flag ==1:
            flag=0
            f.write(str(now)+'\t'+"ffff"+'\t' +str(count1)+'\t'+str(Idr1)+'\n')
            f.write(str(now)+'\t'+"FFFFFFFF"+'\t' +str(count3)+'\t'+str(Idr2)+'\n')
        f.close()
        print('Done')
        ts = time.time()
        date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
        timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
        Hour,Minute,Second=timeStamp.split(":")
        fileName="Attendance\Attendance_"+date+"_"+Hour+"-"+Minute+"-"+Second+".csv"
        attendance.to_csv(fileName,index=False)
        cam.release()
        cv2.destroyAllWindows()
        #print(attendance)
        res=attendance
        message2.configure(text= res)
        break
```

Figure 5.4 Code for traking Images

5.3.5 Traking of absentees

```
def absent():
    f=open("StudentDetails\\StudentDetails.csv", 'r')
    reader1 = csv.reader(f)
    f1=open("Database_new_ct.txt", 'a')
    pj=0
    sd=0
    count =0
    enter =0
    text=f.readlines()
    #while True:
    print('Enter the Attendance Shhet')
    str1=input()
    #str1=str(str)
    print(str1)
    str1=str1+'.csv'
    print(str1)
    text=str(text)
    i=0
    matched=10
    with open('StudentDetails\\StudentDetails.csv', 'r+') as file:
        reader = csv.reader(file)
        #print(reader)
        next(reader)
        for line in reader:
            print(line)
            with open(str1, 'r+') as file1:
                readerlist = csv.reader(file1)
                next (readerlist)
                print("####")
```

```
        for line1 in readerlist:
        print(line1[0])
        if line[0]==line1[0]:
            print('ID matches')
            #print('name {}'.format(line1[1]))
            matched=1
            #file.seek(14)
        else:
            matched=0
      print('name {}'.format(line1[1]))
        matched=10
    if matched==0:
        print('Absentince Phone{}'.format(line[2]))
        matched=10
        phone1= "9663026928" #str(line[2])
        client.api.account.messages.create(
            to = "+91"+ phone1,
            from_="+12166000705" ,  #+1 210-762-4855"
            body="{} is Absent ".format(line[1]) )
```

Figure 5.11 Code for tracking of absentees

5.3.6 Sending Mail

```
def sendMail(to, subject, text, files=[]):
    assert type(to)==list
    assert type(files)==list
    msg = MIMEMultipart()
    msg['From'] = USERNAME
    msg['To'] = COMMASPACE.join(to)
    msg['Date'] = formatdate(localtime=True)
    msg['Subject'] = subject
    msg.attach( MIMEText(text) )
    for file in files:
        part = MIMEBase('application', "octet-stream")
        part.set_payload( open(file,"rb").read() )
        Encoders.encode_base64(part)
        part.add_header('Content-Disposition', 'attachment; filename="%s"'% os.path.basename(file))
        msg.attach(part)
        server = smtplib.SMTP('smtp.gmail.com:587')
        server.ehlo_or_helo_if_needed()
        server.starttls()
        server.ehlo_or_helo_if_needed()
        server.login(USERNAME,PASSWORD)
        server.sendmail(USERNAME, to, msg.as_string())
        server.quit()
sendMail( ["kavyagp666@gmail.com"],
        "Section A attendance",
        "Dear Parent,\nPlease note that your child has not attended today's class.
         Advice him/her to attend the class regularly.
         Please send the letter regarding the reason for his/her absence. ",
        ["frame.png"] )
```

Figure 5.12 Code for sending mail

Chapter 6

Discussion of Results

In this chapter we briefly discuss on the results obtained in our project in various cases. We attach pictures of the processing each case to explain in an enhanced way for better understanding.

In our project we have used Haar cascade algorithm for image classification and recognition.

Figure 6.1 indicates the GUI for taking Attendance.

It has various fields:

- Student ID: Unique ID or Roll Number given to the student
- Student Name: Name of the student
- Phone Number: Mobile Number or Landline Number
- Updates: Holds the data corresponding to the action being performed.

Attendance: Attendance of the student

It has various Buttons:

- Absent: When this button is clicked the absentees list is sent to a particular Email ID.
- Dataset: When this button is clicked Dataset is populated from the image clicked.
- Training: When this button is clicked training of the images takes place.
- Recognition: When this button is clicked the face of the student is recognized based on training.
- Quit: When this button is clicked the user is exited from the window.

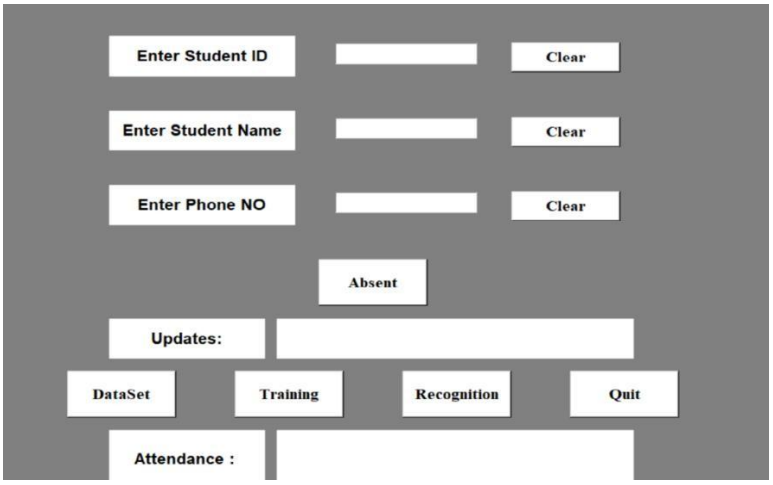

Figure 6.1 GUI for Attendance

The student enters his details such as Student ID, Student Name, Phone Number as shown in the Figure 6.2. Then he clicks on the Dataset button. The inbuilt camera in the Laptop opens and captures the photo of the student as shown in the Figure 6.3. From the image captured the dataset is created as shown in the figure 6.4.

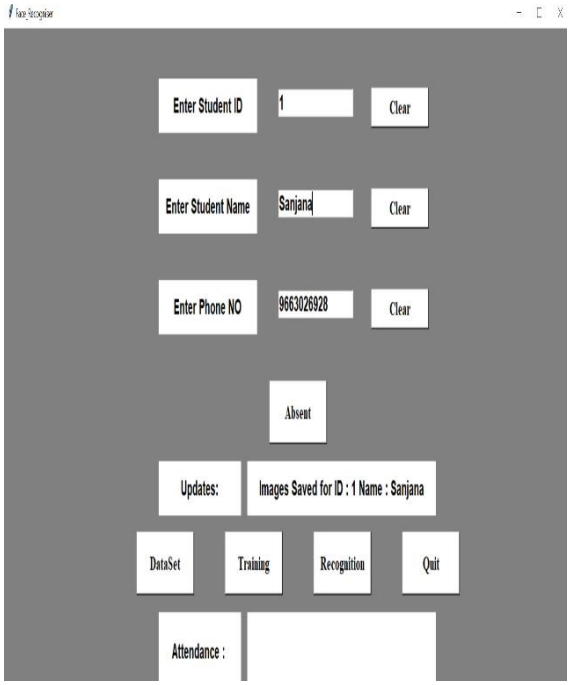

Figure 6.2 Creation of Dataset

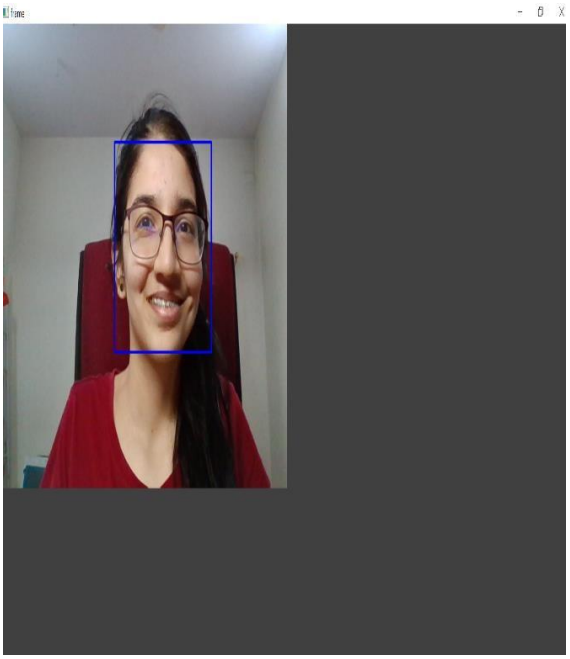

Figure 6.3 : Capturing image for creation of Dataset

Figure 6.4 Dataset

Similarly the details of all the students is stored in the Excel sheet as shown in the Figure 6.5

	A	B	C	D
1	Id	Name	phone	
2	1	Sanjana	9663026918	
3				
4	2	Kavya	9972576764	
5				
6	3	Nagashree	9482041796	
7				
8	4	Sirisha	9986782689	
9				
10	5	Sagar	9845921093	
11				
12	6	Arpita	9972576763	
13				
14	7	Pratik	9972571234	
15				
16	8	Supriya	9972984512	
17				
18	9	Mourya	9628159891	
19				
20	10	Dhanush	9528749892	

StudentDetails

Figure 6.5: Student details

Then the training of the dataset takes place when we click on Training button as shown in the Figure 6.6

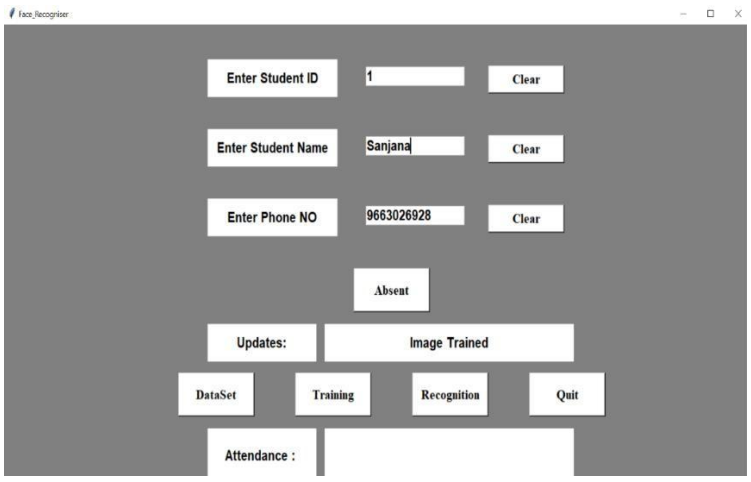

Figure 6.6 Training of DataSet

After the Training of the image is done.The Student attendance can be taken and updated in the attendance sheet. The student's face is recognized as shown in the figure 6.7. The attendance is also updated as shown in Figure 6.8.

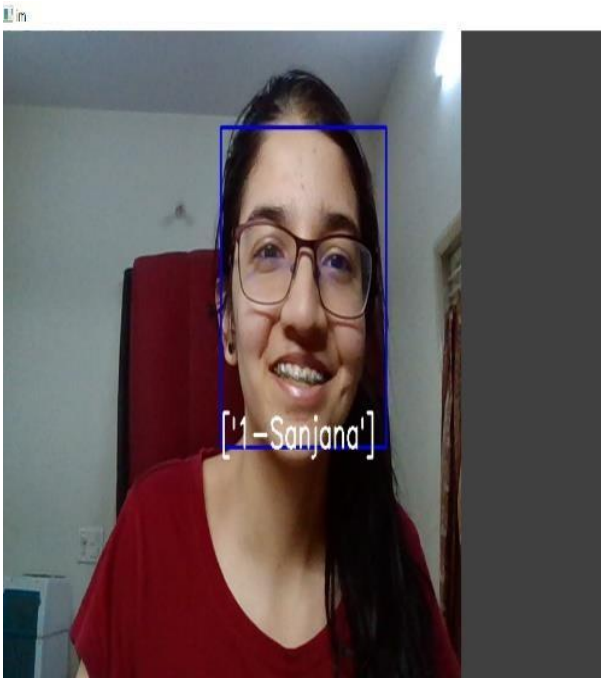

Figure 6.7: Recognition of student

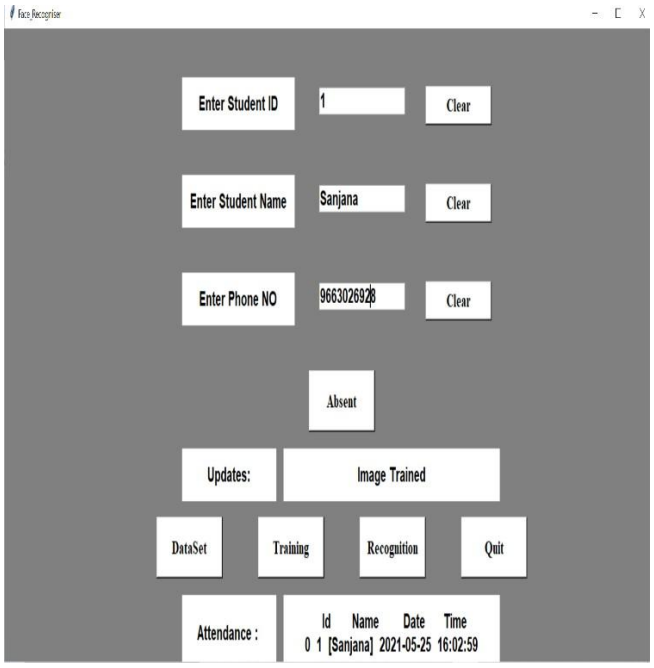

Figure 6.8 Attendance updation

It is important to notify the subject teachers or HOD regarding the absence of their child.In our project the notification e-mail is sent to the corresponding teacher as shown in the Fig 6.9. The attendance is updated in the excel sheet as shown in the Fig 6.11

Figure 6.9 Absentee's Mail

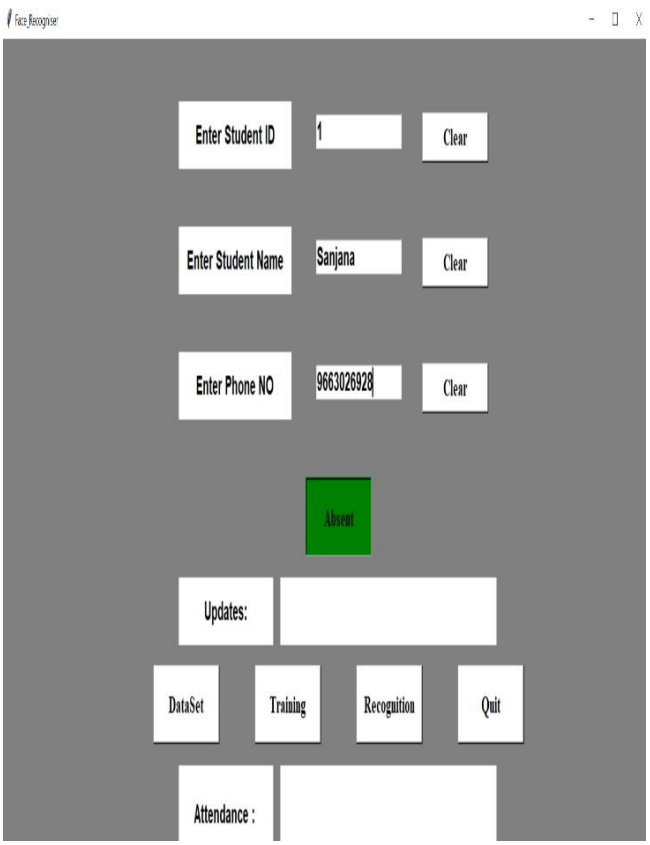

Figure 6.10:Mark as Absent

	A	B	C	D	E
1	Id	Name	Date	Time	
2	2	['Kavya']	########	16:18:57	
3	1	['Sanjana'	########	16:19:13	
4					

Figure 6.11: Attendance in Excel Sheet

After the student's attendance is taken we can exit from the screen using Quit button as shown in the Fig 6.12.

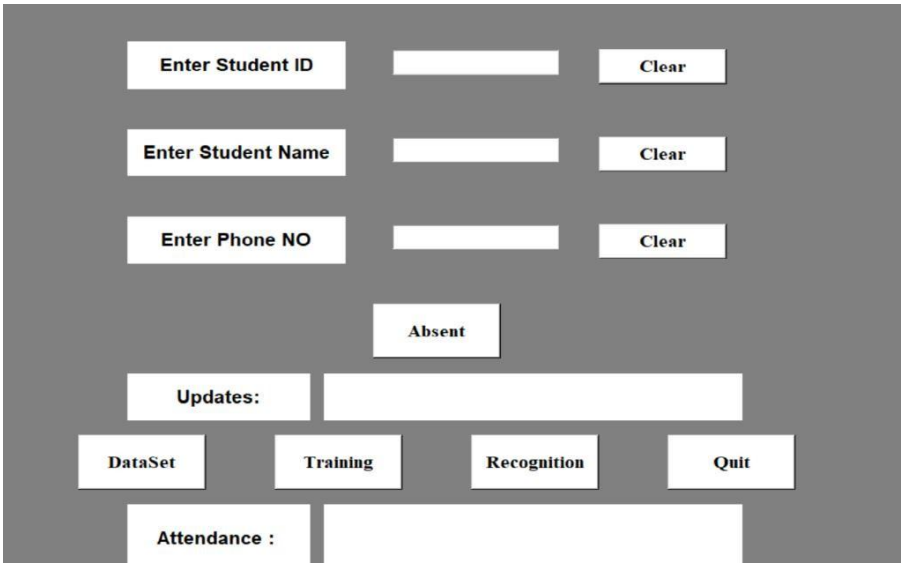

Figure 6.12: Quit

Chapter 7

Conclusion and Future Work

Finally in student attendance management system, the outcome of all the hard work done for attendance management system is here. It is a software which helps the user to work with the attendance, fees update, course update and messages etc.This software reduces the amount of manual data entry and gives greater efficiency. The User Interface of it is very friendly and can be easily used by anyone. It also decreases the amount of time taken to write details and other modules.

All the details about students, teachers and their other tasks can only be seen by the verified users. This Attendance Management System is a solution to all the problems related to the attendance, message, fee status, courses taken by the teachers and the students etc.We have used face recognition concept to mark the attendance of student and make the system better. The system performs satisfactory in different poses and variations. In future this system need be improved because these system sometimes fails to recognize students from some distance, also we have some processing limitation, working with a system of high processing may result even better performance of this system.

At the end, we can say that this software is performing all the tasks accurately and is doing the work for which it is made and this system can be implemented in N number of colleges and schools.

Chapter 8

References

[1]. Shireesha Chintalapati; M. V. Raghunadh, "Automated Attendance Management System Based On Face Recognition Algorithms", 2013 IEEE International Conference on Computational Intelligence and Computing Research.

[2]. Phichaya Jaturawat; Manop Phankokkruad, "An Evaluation of Face Recognition Algorithms and Accuracy based on Video in Unconstrained Factors", 2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)

[3]. Abhishek Jha: ABES Engineering College, Ghaziabad, "Class Room Attendance System Using Facial Recognition System", The International Journal of Mathematics, Science, Technology and Management (ISSN : 2319-8125) Vol. 2 Issue 3

[4]. S. SAYEED, J. HOSSEN, S.M.A. KALAIARASI, V. JAYAKUMAR, I. YUSOF, A. SAMRAJ, "Real-Time Face Recognition For Attendance Monitoring System" Journal of Theoretical and Applied Information Technology 15th January 2017. Vol.95. No.1

[5]. Ashish Choudhary, Abhishek Tripathi, Abhishek Bajaj, Mudit Rathi, and B.M Nandini, "Attendance System Using Face Recognition", International Journal of Modern Trends in Engineering and Research (IJMTER) Volume 03, Issue 04, [April– 2016] ISSN (Online):2349–9745; ISSN (Print):2393-8161

[6]. Preeti Mehta, Dr. Pankaj Tomar, "An Efficient Attendance Management System based on Face Recognition using Matlab and Raspberry Pi 2", May 2016 International Journal of Engineering Technology Science and Research (IJETSR) ISSN 2394 – 3386 Volume 3, Issue 5

[7]. G.Lakshmi Priya1, M.Pandimadevi 1, G.Ramu Priya1, P.Ramya, "Implementation of Attendance Management System using SMART-FR", November 2014 International Journal of Advanced Research in Computer and Communication Engineering ISSN (Online) : 2278-1021 ISSN (Print) : 2319-5940 Vol. 3, Issue 11.

[8]. Samuel Lukas, Aditya Rama Mitra, Ririn Ikana Desanti, Dion Krisnadi, "Student Attendance System in Classroom Using Face Recognition Technique", 2016 International Conference on Information and Communication Technology Convergence (ICTC), IEEE Conference Publications Pages: 1032 - 1035.