CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

Project Report

Face Detection and Recognition Student Attendance System

Team Members

	Student Name	Student ID
1	Karan Chandak	20BCS3928
2	Sonu Saurav	20BCS3952
3	Siddharth Singh	20BCS3933
4	Aryan	20BCS3912
5	Sushant Bisht	20BCS6148

Table of Contents

Abstract:

Abstract Uniqueness or individuality of an individual face is the representation of one's identity. In this project face of an individual is used for the purpose of attendance making automatically. Attendance of the student is very important for every college, universities and school. Conventional methodology for taking attendance is by calling the name or roll number of the student and the attendance is recorded. Time consumption for this purpose is an important point of concern. Assume that the duration for one subject is around 60 minutes or 1 hour & to record attendance takes 5 to 10 minutes. For every tutor this is consumption of time. To stay away from these losses, an automatic process is used in this project which is based on image processing. In this project face detection and face recognition is used. Face detection is used to locate the position of face region and face recognition is used for marking the understudy's attendance. The database of all the students in the class is stored and when the face of the individual student matches with one of the faces stored in the database then the attendance is recorded.

1.1 Introduction:

The **Attendance Management System Project in Python** is a fully functional desktop application developed in Python that covers all of the features that IT students and computer-related courses will require for their college projects or assignments. These can be helpful articles and projects that you are looking for. This **Attendance Management System in Python with MYSQL Database** is one of the best systems for schools that were having trouble keeping track of their students' attendance. This system helps a lot with managing and storing real-time data and capturing real-time attendance. This **Attendance Management System in Python** is quite useful, and the concept and logic of the project are simple to grasp. Attendance is prime important for both the teacher and student of an educational organization. So, it is very important to keep record of the attendance. The problem arises when we think about the traditional process of taking attendance in class room. Calling name or roll number of the student for attendance is not only a problem of time consumption but also it needs energy. So, an automatic attendance system can solve all above problems. There are some automatic attendances making system which are currently used by much institution. One of such system is biometric technique and RFID system. Although it is automatic and a step ahead of traditional method it fails to meet the time constraint. The student has to wait in queue for giving attendance, which is time taking. This project introduces an involuntary attendance marking system, devoid of any kind of interference with the normal teaching procedure. The system can be also implemented during exam sessions or in other teaching activities where attendance is highly essential. This system eliminates classical student identification such as calling name of the student, or checking respective identification cards of the student, which can not only interfere with the ongoing teaching process, but also can be stressful for students during examination sessions. In addition, the students have to register in the database to be recognized. The enrolment can be done on the spot through the user-friendly interface.

1.2. Problem Statement

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 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 does not have to count the number of students several times to ensure the presence of the students. The paper proposed by Zhao, W et al. (2003) has listed the difficulties of facial identification. One of the difficulties of facial identification is the identification between known and unknown images. In addition, paper proposed by Pooja G.R et al. (2010) found out that the training process for face recognition student attendance system is slow and time-consuming. In addition, the paper proposed by Priyanka Wagh et al. (2015) mentioned that different lighting and head poses are often the problems that could degrade the performance of face recognition-based student attendance system. Hence, there is a need to develop a real time operating student attendance system which means the identification process must be done within defined time constraints to prevent omission. The extracted features from facial images which represent the identity of the students have to be consistent towards a change in background, illumination, pose and expression. High accuracy and fast computation time will be the evaluation points of the performance. In our world, Biometric face recognition is used everywhere. It is an advance, and sensible identification system that can identify a person by facial features. It uses a digital camera to capture the image of the face, a

computer for processing and analysis, and an output device for displaying the identification result.

It is straightforward to identify any person from different facial image. The **face recognition system** is a high-speed and reliable technology. This biometric is very safe because it can identify people without any mistakes.

Face recognition is considered the first step forward building biometric is very safe because it can identify people without any mistakes.

Face recognition is considered the step toward building biometric access control-based application scenarios, where biometric features are extracted from the individuals. This system is used in many government offices, firms, banks, and other places.

1.3. Flow Chart

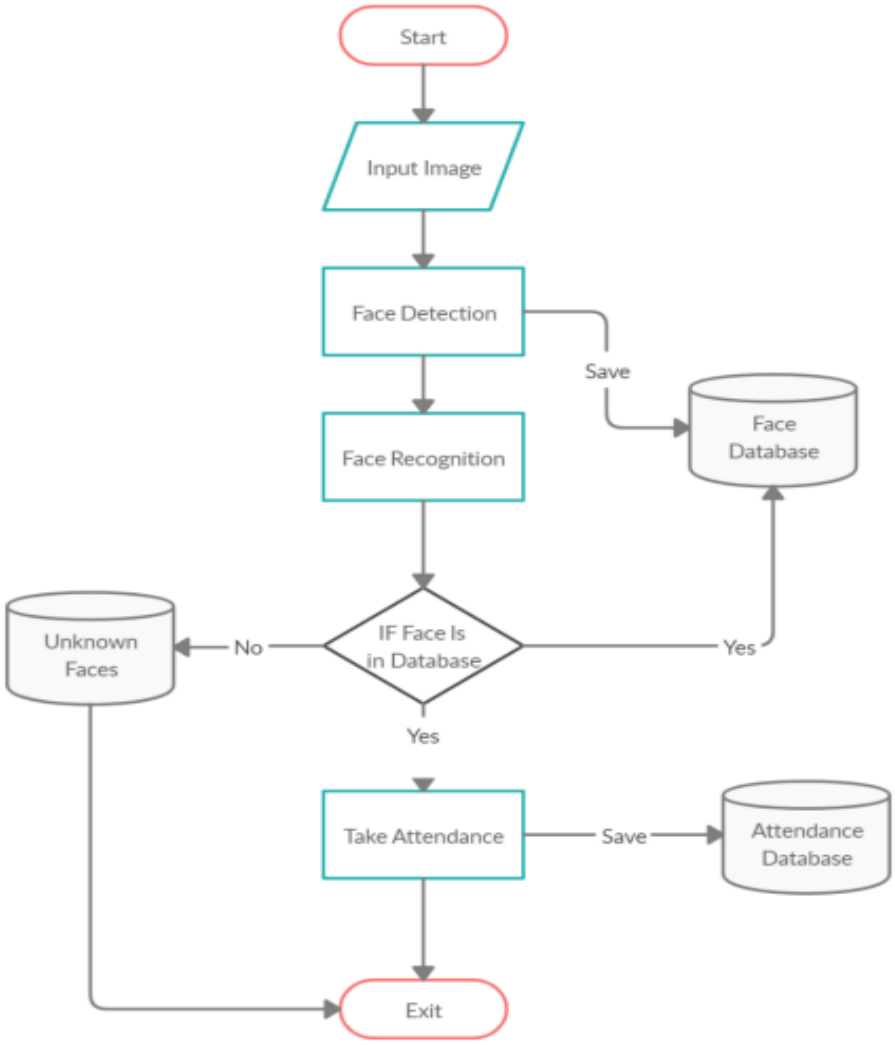


Fig: Flow Chart

1.4. Scope of the project

We are setting up to design a system comprising of two modules. The first module (face detector) is a mobile component, which is basically a camera application that captures student faces and stores them in a file using computer vision face detection algorithms and face extraction techniques. The second module is a desktop application that does face recognition of the captured images (faces) in the file, marks the students register and then stores the results in a database for future analysis

Chapter 2

Literature Review

2.1. Student Attendance System

Arun Katara et al. (2017) mentioned disadvantages of RFID (Radio Frequency Identification) card system, fingerprint system and iris recognition system. RFID card system is implemented due to its simplicity. However, the user tends to help their friends to check in as long as they have their friend's ID card. The fingerprint system is indeed effective but not efficient because it takes time for the verification process so the user has to line up and perform the verification one by one. However, for face recognition, the human face is always exposed and contain less information compared to iris. Iris recognition system which contains more detail might invade the privacy of the user. Voice recognition is available, but it is less accurate compared to other methods. Hence, face recognition system is suggested to be implemented in the student attendance system.

System Type	Advantage	Disadvantages
RFID card system	Simple	Fraudulent usage
Fingerprint system	Accurate	Time-consuming
Voice recognition system		Less accurate compared to Others
Iris recognition system	Accurate	Privacy Invasion

Table 2.1: Advantages & Disadvantages of Different Biometric System[1]

2.2. Digital Image Processing

Digital Image Processing is the processing of images which are digital in nature by a digital computer [2]. Digital image processing techniques are motivated by three major applications mainly:

- Improvement of pictorial information for human perception
- Image processing for autonomous machine application
- Efficient storage and transmission.

2.3. Image Representation in a Digital Computer

An image is a 2-Dimensional light intensity function

f (**x**,**y**) = **r** (**x**,**y**) × **i** (**x**,**y**) - (2.0)

Where, r (x, y) is the reflectivity of the surface of the corresponding image point. i (x,y) Represents the intensity of the incident light. A digital image f(x, y) is discretized both in spatial co-ordinates by grids and in brightness by quantization[3]. Effectively, the image can be represented as a matrix whose row, column indices specify a point in the image and the element value identifies grey level value at that point. These elements are referred to as pixels or pels. Typically following image processing applications, the image size which is used is**256** × **256**, elements, **640** × **480** peels or **1024** × **1024** pixels. Quantization of these matrix pixels is done at 8 bits for black and white images and 24 bits for coloured images (because of the three colour planes Red, Green and Blue each at 8 bits) [4].

2.4. Steps in Digital Image Processing

Digital image processing involves the following basic tasks:

- Image Acquisition - An imaging sensor and the capability to digitize the signal produced by the sensor.
- Preprocessing – Enhances the image quality, filtering, contrast enhancement etc
- Segmentation – Partitions an input image into constituent parts of objects.

- • Description/feature Selection – extracts the description of image objects suitable for further computer processing.
- Recognition and Interpretation – Assigning a label to the object based on the information provided by its descriptor. Interpretation assigns meaning to a set of labelled objects.
- Knowledge Base – This helps for efficient processing as well as inter module cooperation.

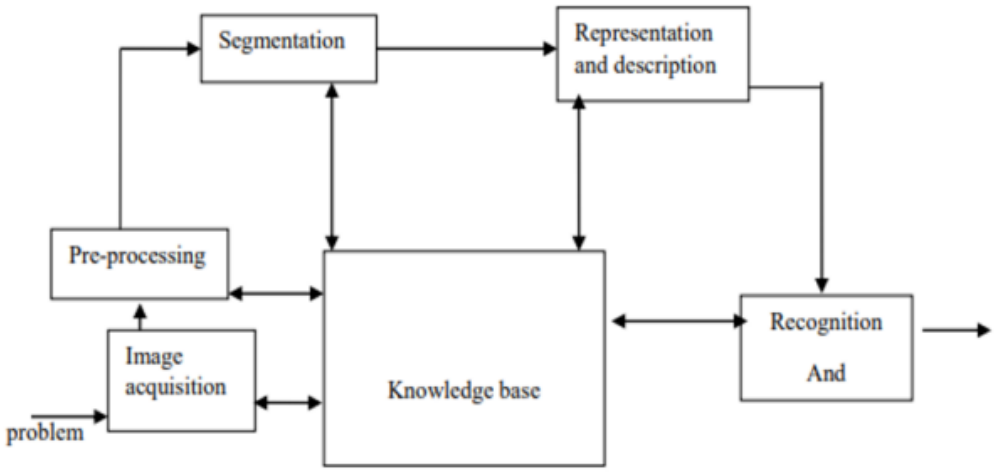


Fig: A diagram showing the steps in digital image processing

2.5. Definition of Terms and History

Face Detection

Face detection is the process of identifying and locating all the present faces in a single image or video regardless of their position, scale, orientation, age and expression. Furthermore, the detection should be irrespective of extraneous illumination conditions and the image and video content[5].

Face Recognition

Face Recognition is a visual pattern recognition problem, where the face, represented as a three dimensional object that is subject to varying illumination, pose and other factors, needs to be identified based on acquired images[6]. Face Recognition is therefore simply the task of identifying an already detected face as

a known or unknown face and in more advanced cases telling exactly whose face it is[7].

Difference between Face Detection and Face Recognition

Face detection answers the question, Where is the face? It identifies an object as a "face" and locates it in the input image. Face Recognition on the other hand answers the question who is this? Or whose face is it? It decides if the detected face is someone known or unknown based on the database of faces it uses to validate this input image[8].It can therefore be seen that face detections output (the detected face) is the input to the face recognizer and the face Recognition's output is the final decision i.e. face known or face unknown.

Face Detection

A face Detector has to tell whether an image of arbitrary size contains a human face and if so, where it is. Face detection can be performed based on several cues: skin color (for faces in color images and videos, motion (for faces in videos), facial/head shape, facial appearance or a combination of these parameters. Most face detection algorithms are appearance based without using other cues. An input image is scanned at all possible locations and scales by a sub window. Face detection is posed as classifying the pattern in the sub window either as a face or a non-face. The face/nonface classifier is learned from face and non-face training examples using statistical learning methods[9]. Most modern algorithms are based on the Viola Jones object detection framework, which is based on Haar Cascades.

Face Detection Method	Advantages	Disadvantages
	illumination changes.	
Ada Boost Algorithm	Need not to have any prior knowledge about face structure.	The result highly depends on the training data and affected by weak classifiers.
SMQT Features and SNOW Classifier Method	1. Capable to deal with lighting problem in object detection. 2. Efficient in computation.	The region contain very similar to grey value regions will be misidentified as face.
Neural-Network	High accuracy only if large size of image were trained.	1. Detection process is slow and computation is complex. 2. Overall performance is weaker than Viola-Jones algorithm.

Table: Advantages & Disadvantages of Face Detection Methods[10]

Viola-Jones Algorithm

Viola-Jones algorithm which was introduced by P. Viola, M. J. Jones (2001) is the most popular algorithm to localize the face segment from static images or video frame. Basically the concept of Viola-Jones algorithm consists of four parts. The first part is known as Haar feature, second part is where integral image is created, followed by implementation of Adaboost on the third part and lastly cascading process.

Fig: Haar Feature

Viola-Jones algorithm analyses a given image using Haar features consisting of multiple rectangles (Mekha Joseph et al., 2016). In the fig shows several types of Haar features. The features perform as window function mapping onto the image. A single value result, which representing each feature can be computed by subtracting the sum of the white rectangle(s) from the sum of the black rectangle(s) (Mekha Joseph et al., 2016)

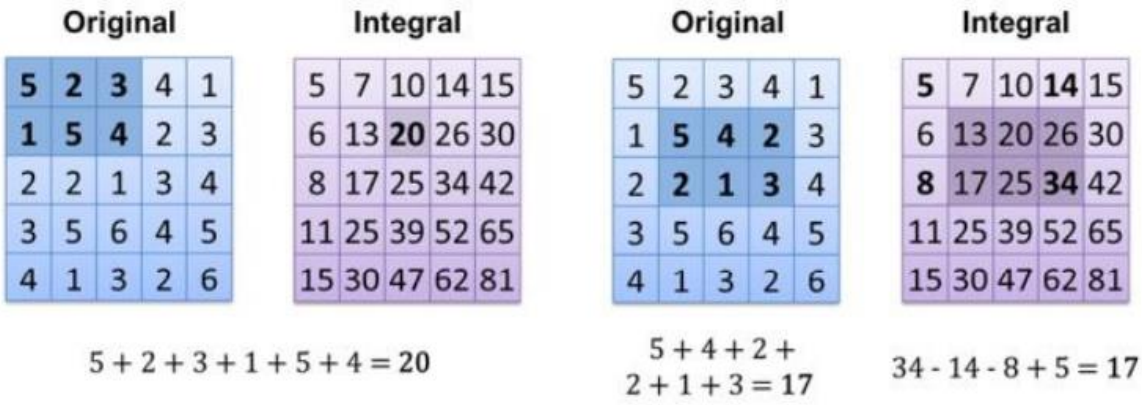


Fig: Integral of Image

The value of integrating image in a specific location is the sum of pixels on the left and the top of the respective location. In order to illustrate clearly, the value of the integral image at location 1 is the sum of the pixels in rectangle A. The values of integral image at the rest of the locations are cumulative. For instance, the value at location 2 is summation of A and B, (A + B), at location 3 is summation of A and C, (A + C), and at location 4 is summation of all the regions, (A + B + C + D) [11]. Therefore, the sum within the D region can be computed with only addition

and subtraction of diagonal at location 4 + 1 − (2 + 3) to eliminate rectangles A, B and C.

Chapter 3

Model Implementation & analysis

3.1. Introduction

Face detection involves separating image windows into two classes; one containing faces (turning the background (clutter). It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin color and facial expression. The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. The face detection task can be broken down into two steps. The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any face or faces within that image as some bounding box with (x, y, width, height). After taking the picture the system will compare the equality of the pictures in its database and give the most related result.

We will use Raspbian operating system, open CV platform and will do the coding in python language.

3.2. Model Implementation

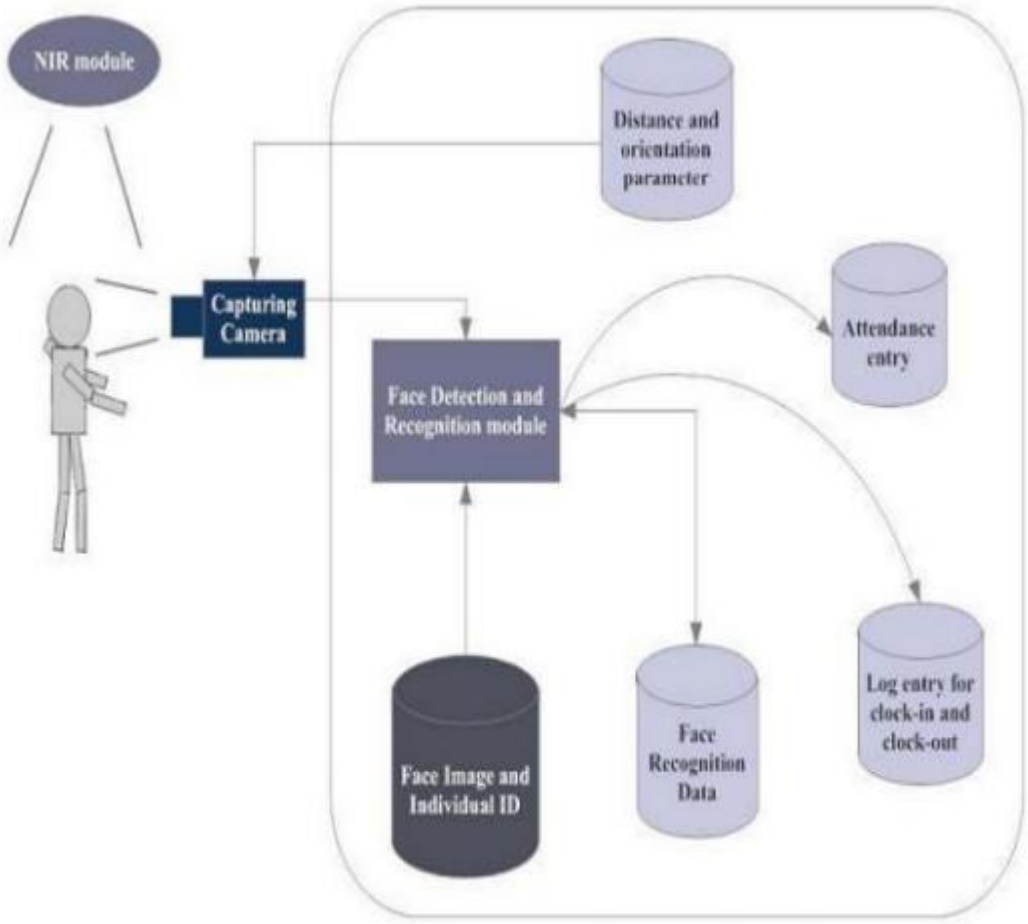


Fig: Model Implement

The main components used in the implementation approach are open source computer vision library (OpenCV). One of OpenCV's goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. OpenCV library contains over 500 functions that span many areas in vision. The primary technology behind Face recognition is OpenCV. The user stands in front of the camera keeping a minimum distance of 50cm and his image is taken as an input. The frontal face is extracted from the image then converted to gray scale and stored. The Principal component Analysis (PCA) algorithm is performed on the images and the eigen values are stored in an xml file. When a user requests for recognition the frontal face is extracted from the captured video frame through the camera. The eigen value is re-calculated for the test face and it is matched with the stored data for the closest neighbour.

3.3. Design Requirements

We used some tools to build the HFR system. Without the help of these tools it would not be possible to make it done. Here we will discuss about the most important one

3.3.1. Software Implementation

1. **OpenCV:** We used OpenCV 3 dependency for python 3. OpenCV is library where there are lots of image processing functions are available. This is very useful library for image processing. Even one can get expected outcome without writing a single code. The library is cross-platform and free for use under the open-source BSD license. Example of some supported functions are given bellow:

- **Derivation:** Gradient / laplacian computing, contours delimitation
- **Hough transforms**: lines, segments, circles, and geometrical shapes detection
- **Histograms**: computing, equalization, and object localization with back projection algorithm.
- **Segmentation:** thresholding, distance transform, foreground / background detection, watershed segmentation.
- **Filtering:** linear and nonlinear filters, morphological operations.
- **Cascade detectors:** detection of face, eye, car plates.
- **Interest points:** detection and matching.
- **Video processing:** optical flow, background subtraction, camshaft (object tracking)
- **Photography**: panoramas realization, high definition imaging (HDR), image inpainting

So, it was very important to install OpenCV. But installing OpenCV 3 is a complex process. How we did it is given below:

```
#!/bin/bash
#Usage : sudo bash./installopencv.bash
echo OpenCV 3.0.0 Raspbian Jessie auto install script - Thomas Cyrix
echo ============================================================
FILE="/tmp/out.$$"
GREP="/bin/grep"
if [ "$(id -u)" != "0" ]; then
   echo "This script must be run as root" 1>&2
   exit 1
fi
echo installing core dependencies ...
apt-get -y install cmake python3-dev python3.4-dev python3-numpy gcc build-essential cmake-curses-gui
echo installing other dependencies ...
apt-get -y install pkg-config libpng12-0 libpng12-dev libpng++-dev libpng3 libpnglite-dev zlib1g-dbg zlib1g
zlib1g-dev pngtools libtiff5-dev libtiff5 libtiffxx0c2 libtiff-tools libeigen3-dev
echo installing helper apps ...
apt-get -y libav-tools
#apt-get -y ffmpeg libavcodec55 libavformat55
apt-get -y install libjpeg8 libjpeg8-dev libjpeg8-dbg libjpeg-progs libavcodec-dev libavformat-dev libgstreamer0.10-0-dbg
libgstreamer0.10-0 libgstreamer0.10-dev libxine2-ffmpeg libxine2-dev libxine2-bin libunicap2 libunicap2-dev swig libv41-0 libv41-dev libpython3.4 libgtk2.0-dev
echo Receving OpenCV 3.0.0 source...
git clone --branch 3.0.0 --depth 1 https://github.com/Itseez/opencv.git
cd opencv
mkdir release
cd release
echo Preparing compilation, may take a long while...
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=$(python3 -c "import sys; print(sys.prefix)") -D PYTHON_EXECUTABLE=$(which python3) ..
echo Compiling Open CV 3.0.0, may take 2 to 36 hours
make -j4
echo Compilation Ok, installing...
make install
cd ../..
rm -r opencv
echo Completed !
echo You now can use OpenCV 3.0.0 in both Python 2 and Python 3 !
```

Fig: Installing OpenCV

We copied this script and place it on a directory on our raspberry pi and saved it. Then through terminal we made this script executable and then ran it.

```
1. Sudo chmod 755 /myfile/pi/installopencv.bash
2. sudo /myfile/pi/installopencv.bash
```

these are the command line we used.

2.Python IDE: There are lots of IDEs for python. Some of them are PyCharm, Thonny, Ninja, Spyder etc. Ninja and Spyder both are very excellent and free but we used Spyder as it feature- rich than ninja. Spyder is a little bit heavier than ninja but still much lighter than PyCharm. You can run them in pi and get GUI on your PC through ssh-Y. We installed Spyder through the command line below.

```
1. sudo apt-get isntall spyder
```

3.3.2. Hardware Implementation

1.Raspberry Pi 3:

1.4GHz 64-bit quad-core processor, dual-band wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and Power-over-Ethernet support (with separate PoE HAT)

Fig: Raspberry Pi 3 Model B+

Specification: The Raspberry Pi 3 Model B+ is the final revision in the Raspberry Pi 3 range.

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

2.Webcam: ELP HD 8Megapixel USB CMOS board camera module adopt Sensor Sony (1/3.2") IMX179 is nice to use in Linux equipment, or that equipment which come with windows, linux, Android system etc.

Fig: Webcam

Specification:

• 1/3.2-inch Sony IMX179 USB webcam

• 8-megapixel high resolution MPEG USB camera

• UVC usb camera, Support windows, linux, Mac with UVC, also for android system.

• Compatible with raspberry pi, Ubuntu, OpenCV, Amcap and many other USB web camera software and hardware.

• Webcam USB with 2.8mm lens

• 38×38/32x32mm mini micro Usb board camera

• USB webcam, well used in many machines, atm machine, medical machine, automatic vending machine, industry machine.

• USB camera module Parameters changable (Brightness, Contrast, Saturation, White Balance, Gamma, Definition, Exposure…)

3.Power Source:

We use Mi 10000 mAH Power Bank for our power sources.

Fig: Power Source

4.Project Machine:

Here is our prototype device.

Fig: Prototype Device

3.4. Experimental Results

The step of the experiments process are given below:

Face Detection:

Start capturing images through web camera of the client side:

Begin:

• Pre-process the captured image and extract face image

• calculate the eigen value of the captured face image and compared with eigen values of existing faces in the database.

• If eigen value does not matched with existing ones,save the new face image information to the face database (xml file).

• If eigen value matched with existing one then recognition step will be done.

End

Face Recognition:

Using PCA algorithm the following steps would be followed in for face recognition:

Begin:

• Find the face information of matched face image in from the database.

• update the log table with corresponding face image and system time that makes completion of attendance for an individual student.

End

This section presents the results of the experiment conducted to capture the face into a grey scale image of 50x50 pixels.

Test data	Expected Result	Observed Result	Pass/ Fail
OpenCAM_CB()	Connects with the installed camera and starts playing.	Camera started.	pass
LoadHaar Classifier()	Loads the HaarClassifier Cascade files for frontal face	Gets ready for Extraction.	Pass
ExtractFace()	Initiates the Paul-Viola Face extracting Frame work.	Face extracted	Pass
Learn()	Start the PCA Algorithm	Updates the facedata. xml	Pass
Recognize()	It compares the input face with the saved faces.	Nearest face	Pass

Here is our data set sample.

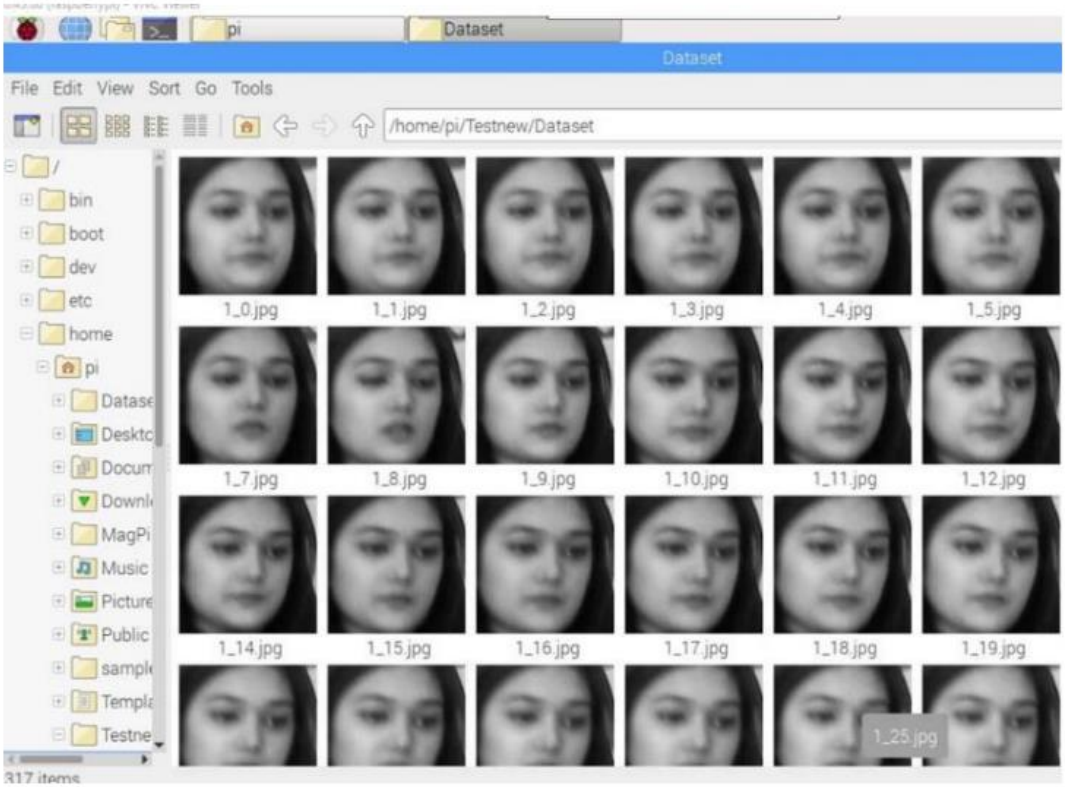
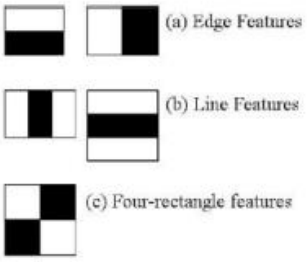


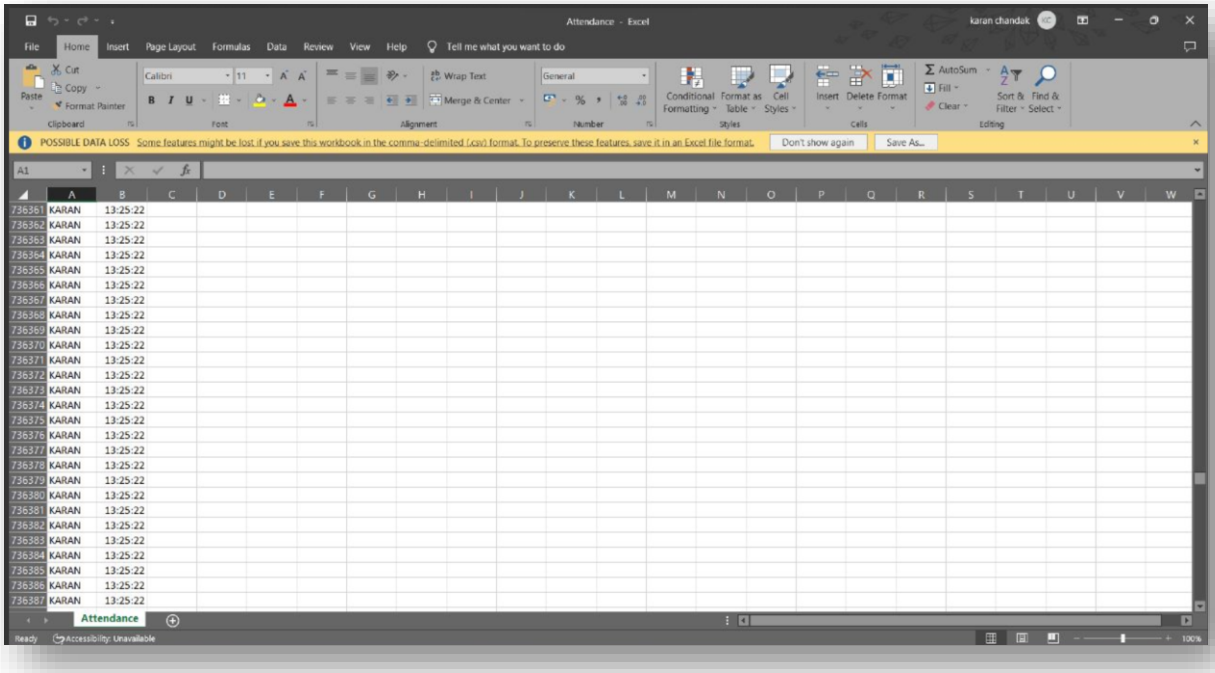
Fig: Dataset sample

Face Orientations	Detection Rate	Recognition Rate
0^0 (Frontal face)	98.7 %	95%
18°	80.0 %	78%
54°	59.2 %	58%
72°	0.00 %	0.00%
90°(Profile face)	0.00 %	0.00%

We performed a set of experiments to demonstrate the efficiency of the proposed method. 30 different images of 10 persons are used in training set. Figure 3

shows a sample binary image detected by the ExtractFace() function using Paul-Viola Face extracting Frame work detection method.





Chapter 4

System Design

4.1. Design constrains

The constraints which were considered while designing on project are following

All our code is written in Python language. First here is our project directory structure and files.

```
1. Dataset: Where all the faces are saved.
2. main.py: Main program file to run the program.
3. dataset.py: Capture images and working on datasets.
4. database.log: To keep track the database events
5. data_set.csv: To save the details of data.
6. data_log.ods: Attendance save.
```

4.1.1. Design Constrain: Engineering Standards

The samples for database should be increase, as to increase the efficiency of detection. Also, the more the expensive the camera, the easier its algorithm is likely detecting the person.

4.1.2. Design Constrain: Environmental

The camera should capture all the students present in the class. Each student present should be seated such that it is visible to camera, so that his/her attendance gets marked easily

4.1.3. Design Constrain: Ethical

The second limitation which is faced include the person appearance by face, which a person changes his/her look and looks different from the picture in the database of the attendance system, then it may be difficult for his/her attendance to be marked.

4.2. Design Methodology

As we mentioned before in (Figure 1.1). The project process is:

- A camera will take continuous stream.
- In Python OpenCV library for vision will be used.
- Convert the RGB image to grayscale image.
- Then perform Machine Vision Algorithm and match with patterns stored in our folder.
- If pattern matches based on the score of how successful, decide to mark attendance or not.

- Update the marked attendance in a measurement file.

4.3. Product Subsystem and components

4.3.1. Product subsytem1: Vison Acquisition

This subsystem is used to acquire continuous stream of video from attached camera. It starts a camera session from desired camera and transmits its image feed to further processing. The feed captured is inline processed and then the next feed is captured.

Chapter 5

Working Plan

5.1. Introduction

In this chapter, we observe the entire work structure, meaning how the scheduling was maintained throughout the developmental phase. We shall also see the financial foundation of this project and furthermore the feasibility study should be also discussed.

5.2. Work Breakdown Structure

In order to develop this system, we gave enormous importance to scheduling because we believed if we want to provide the best of quality in a given period of time then we must give due importance to scheduling

which also helped us to achieve a better result. The figure below focuses the weekly work we had accomplished.

Week No.	Proposed Work
Week-1	Project Proposal Report and Presentation
Week-1	Study related works
Week-1	Study in Python
Week-2	Study related works using OpenCV
Week-2	Study related works using Bluetooth
Week-3	Study related works using processing
Week-3	Study image processing
Week-3	Study image processing
Week-4	Sketching basic structure
Week-4	Prototype design
Week-4	Finalize Prototype design
Week-4	Flexible Box

Week-5	Runnable with basic commands(Input, Output, Turn on, Turn Off)
Week-5	Designing Lookahead table
Week-5	Designing Lookahead table
Week-6	Creating environment for image processing
Week-6	Creating environment for image processing
Week-7	Integrating all together
Week-7	Start coding
Week-8	Coding for basic instructions (Compare, Result, Accuracy measure etc.)
Week-8	Coding for single face detection
Week-9	Single face detection and Compare with database
Week-9	Multiple Face detection and Compare
Week-10	Detecting Multiple face, store and compare with database
Week-10	Attendance collection
Week-10	File Generate base on collective data
Week-10	Daily file generation of attendance

5.3. Gantt Chart

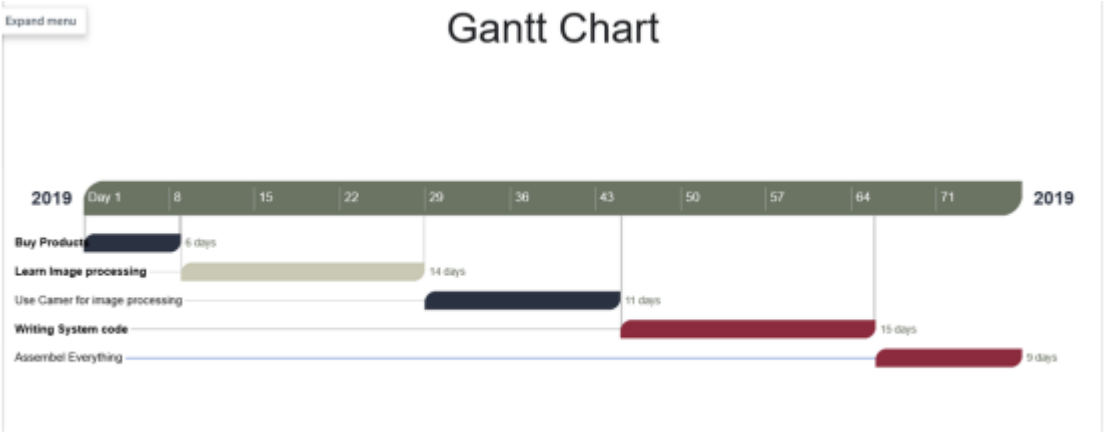


Fig: Gantt Chart

5.4. Financial Plan

Money was required to build the system as we had to buy a lot of components. Breakdown is given bellow:

Item Name	Price
Raspberry Pi	4000
HD Camera	1000
SD Card	500
Total	5500

5.5. Feasibility Study

Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study. “FEASIBILITY STUDY” is a test of system proposal according to its workability, impact of the organization, ability to meet needs and effective use of the resources. It focuses on these major questions:

1. What are the user’s demonstrable needs and how does a candidate System meet them?

2. What resources are available for given candidate system?

3. What are the likely impacts of the candidate system on the organization?

4. Whether it is worth to solve the problem?

During feasibility analysis for on our project, following primary areas of interest are to be considered. Investigation and generating ideas about a new system does the following steps:

Steps in feasibility analysis

1. Form a project team and appoint a project leader.

2. Enumerate potential proposed system.

3. Define and identify characteristics of proposed system.

4. Determine and evaluate performance and cost effectively of each proposed system.

5. Weight system performance and cost data

6. Select the best-proposed system.

7. Prepare and report final project directive to management.

5.6. Summary

To conclude, we discussed the scheduling processes of developing this system. Additionally, we have also identified how feasible the system is through the lens of evaluating using various feasibility studies

Chapter 6

Future Work

6.1. Introduction

This chapter discusses the future scope or the implementation of this robot. To incerease the scope of this device we can add some new features. As technology is becoming more advance it will be mendatory to change the sturctute some day with better replacement and sometimes based on customer requirements.

6.2. Future Scope of Work

There are so many future scope on this project. Some of them are

- Can improve security
- Can use Neural Network for high accuracy
- Can used in big factory or employee attendance
- Can build on fully web base system.

6.3. Summary

This chapter has described the possible future applications of the design. But there are a lot of possibilities with the designed device. The device may need some research for different applications, though the principle of the designed system will remain as it is.

Chapter 7

Result

7.1. Introduction

This chapter of the report contains the results that we achieved throughout the course of using this system.

Results Achieved

From initiation through conclusion of developing this system the following results has been achieved. They are as follows:

- The system can be administered by a non-IT technician.
- The system is market ready for commercial use.
- The system has the capacity to carry up to a thousand faces to recognize.
- The system can serve as much people as they want within an organization.

7.2. Summary

This chapter has covered the different types of results that we have managed to obtain throughout the course of using this system.

Contribution of Team Members

As group, we managed to divide the work between us as shown in (Table 1). We were collaborating and meeting with each other before the COVID-19, however after the pandemic we were doing virtual meetings in order to try our best to work on the project despite what is happening.

TABLE 1: CONTRIBUTION OF TEAM MEMBERS

Aryan	Program Designing
Karan Chandak	Program Developer
Siddharth Singh	Program Developer
Sonu Saurav	Documentation
Sushant	Report

7.3 Project Execution Monitoring

.

Team Meeting

All members discuss their respective progress in their part in daily meeting and prepare a final sheet for the meeting with the advisor.

Testing

After completion of each part mentioned in project plan, proofreading and testing was done for successful verification of the part.

Progress Discussion Meeting

An overall progress discussion meeting was held once a month where current standing of the project is presented to the advisor and everyone including members share their opinion and discuss them, and what amendments are necessary to be added.

8.References

- https://www.researchgate.net/publication/326986115_Face_Detection_and_Recognition_Student_Atten dance_System
- www.ineuron.com