PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology (Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

By

ARIKATLA GOPI VENKATA SUDHEER

(Reg. No :- 38110046)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119, TAMILNADU

NOVEMBER 2021

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **A. GOPI VENKATA SUDHEER(REG N.O 38110046)** who carried out the project entitled "**ADVANCED FACIAL RECOGNITION ATTENDANCE SYSTEM USING PYTHON**" under my supervision from October 2021 to November 2021.

Internal Guide

Dr.

Ms. M. MAHESWARI,M.E.,Ph.D.,.

Head of the Department

Dr.L. Lakshmanan, M.E.,Ph.D.,.

Dr.S.Vigneshwari M.E., Ph.D.,.

Submitted for Viva voce Examination held on _____

Internal Examiner **External Examiner**

DECLARATION

I, **ARIKATLA GOPI VENKATA SUDHEER (38110046)** hereby declare that the Project Report entitled **ADVANCED FACIAL RECOGNITION ATTENDANCE SYSTEM USING PYTHON** done by me under the guidance of **Ms. M. MAHESWARI M.E.,Ph.D.,.** at Sathyabama institute of science and technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

ARIKATLA GOPI VENKATA SUDHEER

PLACE: **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D.**, **Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D.,** Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. M. MAHESWARI M.E., Ph.D.,.** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

ABSTRACT

The main purpose of this project is to build a face recognition-based attendance monitoring system for educational institution to enhance and upgrade the current attendance system into more efficient and effective as compared to before. The current old system has a lot of ambiguity that caused inaccurate and inefficient of attendance taking. Many problems arise when the authority is unable to enforce the regulation that exist in the old system.

The technology working behind will be the face recognition system. The human face is one of the natural traits that can uniquely identify an individual. Therefore, it is used to trace identity as the possibilities for a face to deviate or being duplicated is low. In this project, face databases will be created to pump data into the recognizer algorithm. Then, during the attendance taking session, faces will be compared against the database to seek for identity. When an individual is identified, its attendance will be taken down automatically saving necessary information into a excel sheet. At the end of the day, the excel sheet containing attendance information regarding all individuals are mailed to the respective faculty.

Keywords- Smart Attendance System,NFC,RFID,OpenCV,Numpy

***In this project we are using libraries like*:**

Tkinter
Firebase
Numpy
Pillow
xlwrite
opencv3

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No
	ABSTRACT	9
	LIST OF FIGURES	12
	LIST OF TABLES	13
	LIST OF ABBREVATIONS	XII
1	**INTRODUCTION**	**1**
	1.1 Problem Statement and Motivation	14
	1.2 Research Objectives	15
	1.3 Project Scope and Direction	15
	1.4 Impact, Significance and contributions	16
	1.5 Historical development prior to the project	16
2.	**AIM AND SCOPE OF THE PRESENT INVESTIGATION**	**17**
	2.1 OUTLINE OF THE PROJECT	17
	2.2 PROBLEM STATEMENT	18

3.	**EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED**	20
	3.1 GENERAL	20
	3.2 OVERVIEW	21
	3.3 LITERATURE	25
4.	**RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS**	**34**
	4.1 RESULT	34
5	**SUMMARY AND CONCLUSIONS**	**38**
	5.1 SUMMARY	38
	5.2 FUTURE WORK	39
	REFERENCES	**40**
	APPENDIX	
	A.METHDOLOGY	40
	B. SOURCE CODE	40

LIST OF FIGURE

FIGURE NO	FIGURE NAME	FIGURE NO
3.2.1	FRONTAL FACE DETECTION USING VIOLA JONES ALGORITHM	21
3.5.1	THE CREATION OF THE FACE DATABASE	22
3.5.2	THE PROCESS OF ATTENDANCE TAKING	23
4.1.1	ATTENDANCE SYSTEM	34
4.1.2	TAKING IMAGES FOR DATA SET	35
4.1.3	ENTER PASSWORD TO SAVE STUDENT PROFILE	35
4.1.4	HELP MENU	35
4.1.5	CHANGE PASSWORD	36
4.1.6	PASSWORD CHANGED	36
4.1.7	CONTACT US	36
4.1.8	PASSWOD ERROR	37
4.1.9	TAKING ATTENDANCE	37
6	FLOW CHART OF THE IMAGE ACQUISITION PROCESS	43

LIST OF TABLES

TABLE NO. TABLE NAME PAGE NO.

1.1

1.2

2.1

3.1

LIST OF ABBREVATIONS

ABBREVATION EXPANSION

CHAPTER 1

INTRODUCTION

This is a project about **Facial Recognition-Based Attendance System for Educational Institutions.** In this chapter, the problem and motivation, research objectives, project scope, project contributions and the background information of the project will be discussed in detail.

1.1 Problem Statement and Motivation

According to the previous attendance management system, **the accuracy of the data**collected is the biggest issue. This is because the attendance might not be recorded personally by the original person, in another word, the attendance of a particular person can be taken by a third party without the realization of the institution which violates the accuracy of the data. For example, student A is lazy to attend a particular class, so student B helped him/her to sign for the attendance which in fact student A didn"t attend the class, but the system overlooked this matter due to no enforcement practiced. Supposing the institution establish an enforcement, it might need to waste a lot of human resource and time which in turn will not be practical at all. Thus, all the recorded attendance in the previous system is not reliable for analysis usage. The second problem of the previous system is where it is **too time consuming**. Assuming the time taken for a student to sign his/her attendance on a 3-4 paged name list is approximately 1 minute. In 1 hour, only approximately 60 students can sign their **attendance which is obviously inefficient and time consuming**. The third issue is with the **accessibility of those information by the legitimate concerned party**. For an example, most of the parents are very concerned to track their child"s actual whereabouts to ensure their kid really attend the classes in college/school. However in the previous system, there are no ways for the parents to access such information. Therefore, evolution is needed to be done to the previous system to improve efficiency, data accuracy and provides accessibility to the information for those legitimate party.

1.2 Research Objectives

In order to solve the drawbacks of the previous system stated in 1.1, the existing system will need to evolve. The proposed system will reduce the paperwork where attendance will no longer involve any manual recording. The new system will also reduce the total time needed to do attendance recording. The new system will acquire individual attendance by means of facial recognition to secure data accuracy of the attendance.

The following are objectives of the project:

▪ To develop a portable Smart Attendance System which is handy and self-powered.

▪ To ensure the speed of the attendance recording process is faster than the previous system which can go as fast as approximately 3 second for each student.

▪ Have enough memory space to store the database.

▪Able to recognize the face of an individual accurately based on the face database.

▪ Allow parents to track their child‟s attendance.

▪ Develop a database for the attendance management system.

▪ Provide a user-friendly interface for admins to access the attendance database andfor non-admins (parents) to check their child‟s attendance by mailing the attendance.

▪ Allow new students or staff to store their faces in the database by using a GUI.

▪ Able to show an indication to the user whether the face- recognition process is successful or not.

1.3 Project Scope and Direction

The main intention of this project is to solve the issues encountered in the old attendance system while reproducing a brand new innovative smart system that can provide convenience to the institution. In this project, an application will be developed which is capable of recognising the identity of each individuals and eventually record down the data into a database system. Apart from that, an excel sheet is created which shows the students attendance and is directly mailed to the respected faculty.

The followings are the project scopes:

▪ The targeted groups of the attendance monitoring system are the students and staff ofan educational institution.

▪ The database of the attendance management system can hold up to 2000

individual"sinformation.

▪ The facial recognition process can only be done for 1 person at a time.

▪ An excel sheet is created which contains the student attendance and is mailed to the respected faculty. ▪ The project has to work under a Wi-Fi coverage area or under Ethernet connection, as the system need to update the database of the attendance system constantly.

▪ The device on which the application is running is powered up by power bank to improve the portability of the application.

1.4 Impact, Significance and contributions

Many attendance management systems that exist nowadays are lack of efficiency and information sharing. Therefore, in this project, those limitations will be overcome and also further improved and are as follows : ▪ Students will be more punctual on attending classes. This is due to the attendance of astudent can only be taken personally where any absentees will be noticed bythe system. This can not only train the student to be punctual as well as avoids any immoral ethics such as signing the attendance for their friends.

▪ The institution can save a lot of resources as enforcement are now done by means oftechnology rather than human supervision which will waste a lot of human resourcefor an insignificant process.

▪ The application can operate on any device at any location as long as there is Wi-Fi coverage or Ethernet connection which makes the attendance system to be portable to be placed at any intended location. For an example, the device can be placed at the entrance of the classroom to take the attendance.

▪ It saves a lot of cost in the sense that it had eliminated the paperwork completely.

▪ The system is also time effective because all calculations are all automated.In short, the project is developed to solve the existing issues in the old attendance system.

1.5 Historical development prior to the project

Back in the years, attendance management system in school/colleges was done bymanual reporting where the student"s attendance were recorded by placing a mark or signature beside their name in a name list to indicate their presence in a particular class. While the staff in the institution will report their attendance through the punch card machine which also have to be done manually. Later on, some of those

attendance systems had evolved into using smart cards to replace signature markings where each students/staff will be required to report their attendance using a smart card embedded with a unique identification chip.

CHAPTER 2

AIM AND SCOPE OF PRESENT INVESTIGATION

2.1 OUTLINEOF THE PROJECT

The main intention of this project is to solve the issues encountered in the old attendance system while reproducing a brand new innovative smart system that can provide convenience to the institution. In this project, an application will be developed which is capable of recognising the identity of each individuals and eventually record down the data into a database system. Apart from that, an excel sheet is created which shows the students attendance and is directly mailed to the respected faculty.

The followings are the project scopes:

▪ The targeted groups of the attendance monitoring system are the students and staff ofan educational institution.

▪ The database of the attendance management system can hold up to 2000 individual‟sinformation.

▪ The facial recognition process can only be done for 1 person at a time.

▪ An excel sheet is created which contains the student attendance and is mailed to the respected faculty. ▪ The project has to work under a Wi-Fi coverage area or under Ethernet connection, as the system need to update the database of the attendance system constantly.

▪ The device on which the application is running is powered up by power bank to improve the portability of the application.

2.2 PROBLEM STATEMENT

Write a Python script to

a) To take attendance using facial recognition

b) Secure the data

c) Easy to access the attendance

d) To search a particular student details from the database

Many attendance management systems that exist nowadays are lack of efficiency and information sharing. Therefore, in this project, those limitations will be overcome and also further improved and are as follows : ▪ Students will be more punctual on attending classes. This is due to the attendance of astudent can only be taken personally where any absentees will be noticed bythe system. This can not only train the student to be punctual as well as avoids any immoral ethics such as signing the attendance for their friends.

▪ The institution can save a lot of resources as enforcement are now done by means oftechnology rather than human supervision which will waste a lot of human resourcefor an insignificant process.

▪ The application can operate on any device at any location as long as there is Wi-Fi coverage or Ethernet connection which makes the attendance system to be portable to be placed at any intended location. For an example, the device can be placed at the entrance of the classroom to take the attendance.

▪ It saves a lot of cost in the sense that it had eliminated the paperwork completely.

▪ The system is also time effective because all calculations are all automated.In short, the project is developed to solve the existing issues in the old attendance system.

Back in the years, attendance management system in school/colleges was done bymanual reporting where the student‟s attendance were recorded by placing a mark or signature beside their name in a name list to indicate their presence in a particular class. While the staff in the institution will report their attendance through the punch card machine which also have to be done manually. Later on, some of those attendance systems had evolved into using smart cards to replace signature markings

where each students/staff will be required to report their attendance using a smart card embedded with a unique identification chip.

CHAPTER 3

ALGORITHMS AND METHODS

3.1 GENERAL

There are some methods which are important for this project. They are:

1) **assure_path_exists(path) :-**

   This method with the files where edit or update can be done with the help of this method. This method contains os functions which enables to perform it's function.

2) **tick() :-**

   This method deals with the time and date. Which are important for attendance.

3) **check_haarcascadefile():-**

   This methods imports "haarcascade_frontalface_default.xml" which is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a Machine Learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

4) **save_pass():-**

   This method is used to save the data which is protected by password which has been set by the admin.

5) **change_pass():-**

   This method is used to change the old password with new password.

6) **psw():-**

   This method provides an error message when we type wrong password.

7) **TakeImages():-**

This method helps to take the images of the students and saves them in dataset. It also stores the images with the name of combination of name and id which helps to recheck with our naked eye if teacher has any doubts with students. It also stores the id and name in an excel sheet named “StudentDetails”.

8) **TrainImages():-**

This method is used to train the dataset with student images.

3.2 OVERVIEW

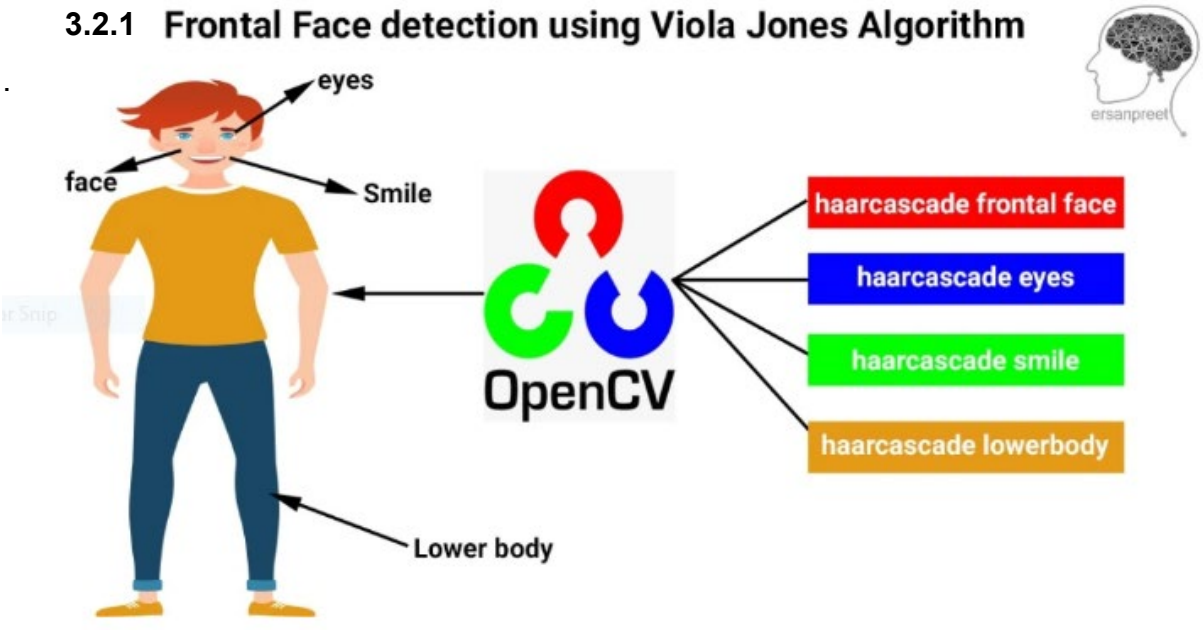
The project main aim is to take the attendance using facial recognition and stores the attendance in an excel sheet.

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, “Rapid Object Detection using a Boosted Cascade of Simple Features” in 2001. It is a Machine Learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it.

3.2.1 Frontal Face detection using Viola Jones Algorithm

.

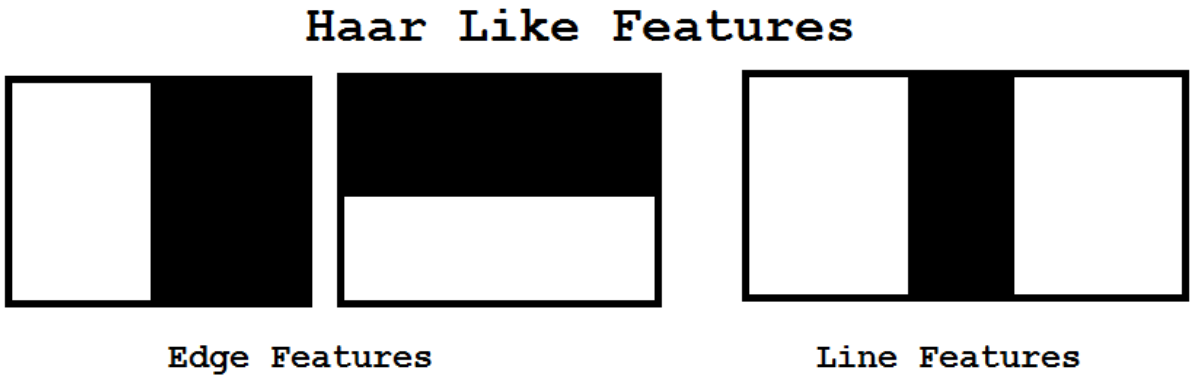


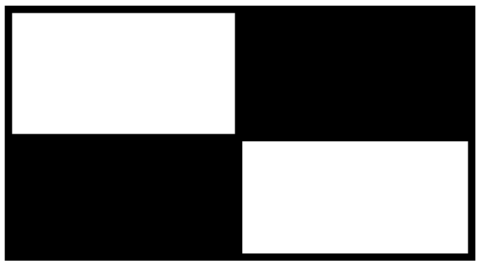
This algorithm can be used to detect **face as well as eye detection with viola jones** will be implemented using **python coding**. Light will be thrown on **haar features, integral image, viola jones algorithm adaboost** training as well as cascading. OpenCV will be used to draw the rectangle on the face as well as eyes. OpenCv as well as haar cascade eye and face xml files will be used to locate the coordinates of both **face as well as eyes**. There will be some discussion on how viola jones trained the algorithm. Limitation of the algorithm will be also discussed and other options will be also presented to overcome former limitations. In short, light will be thrown on deep learning object detection models. I hope readers will gain some useful knowledge while reading this blog and their concepts on computer vision will get a supporting end.

This algorithm was developed by two people named **Paul Viola and Michael Jones**. Developed in **2001**. Even though it was discovered **19 years ago** and deep learning made huge progress in object detection, still it holds a special place in detecting faces, eyes and other haar features. If one has to start learning about detecting objects, I recommend them to learn viola jones first before diving deep into deep learning powerful object detection models. This algorithm is composed of two parts – training as well as detection. It not only detects faces in the images but is applicable to videos also. There is a limitation of this algorithm as it works with only frontal faces which is overcomed by deep learning models such as **ssd, yolo, faster rcnn**. Please refer to this link to see these deep learning models. I hope readers have gained some crucial points from this part. Let us summarize it using the points below.

Haar features and Integral Image

Word **haar** came from Alfred Haar who is an **hungarian** mathematician who developed the **haar** wavelet. Let us see **haar** features from the below diagram and how these are related to **viola jones algorithm**.





haar-like features

These **haar-like features** are developed by **viola and jones** with respect to face in an image. Let us understand these with general sense and then we will apply it to an image.

General understanding of edge and line haar like features

Suppose one is looking at the **floor in a room**. If one looks at the middle of the floor and at the end of the floor, there will be a difference between colors. The floor is brighter everywhere and has darker corner sides. Same is true while one is looking at the table. Edges have different colors with respect to the rest of the table. Same holds true for the image also. Now let us continue this concept with respect to faces in the images.

Applying haar like features to face in the images

If one looks at the lips, there is a change of colors between upper and lower lips, and one can identify line features there. If one looks carefully at the eyebrow, there is contrast between forehead and eyebrow and hence edge feature can be identified

there. Similarly with visualization one can locate line features in the eyes and edge features in the nose.

How these features helps viola jones in detecting different parts of face

From the above discussion, I hope **haar like features** are cleared to readers. Now the question is how these features help viola and jones in detecting different parts of the face. Let us start with the nose. One side of the nose will be brighter while the other side of the nose will be darker while one looks at the nose in an image. This implies edge feature is present there. Left side and right side of the nose are represented by the pixel value while talking with respect to computer vision terms. Difference in the average of pixels on the left and right side will provide a threshold for the detection of the nose in the face and the whole image. This helps in the training phase where the nose is detected by an algorithm using threshold. Same holds true for eyes, lips, jaws and other facial features. We will see further that during training, viola and jones took both face and non-face images. Threshold helps to differentiate between face and non-face during training and hence lies the significance of haar like features.

What is the relationship between integral image with calculations involved in threshold

I hope readers are with me and understand each and everything with ease. If you are facing some difficulty in understanding the concepts, take a deep breath and start reading again. You may also do some outing for some time and come back again and start reading fresh minded. If you look at heading above this heading to identify any facial part first of all haar like feature is to be chosen and calculations on pixels are done to identify it as well as its position in the image. To decrease the number of computations involved in the calculation of threshold and haar like feature, integral image is used.I am not going deep into its concept, this is because a lot is to be covered in this tutorial but if readers are interested in knowing about it, they may purchase cheap price course on Deep learning and Computer Vision A-ZTM: OpenCV, SSD and GANs by Hadelin and Kirill. Great teachers and huge appreciation from my side also.

Training Viola Jones Algorithm

Algorithm is trained on both face and non face images. They fed **4960 manually**

labelled images along with **9544 non-facial images** so that it can distinguish between face and non face. During training images are scaled down to **24 * 24** while predictions features are scaled up. There is a reason why images are scaled to 24 * 24 during training. For a 24*24 windows of image, there are **162336 features**. Hence scaling is a necessity. This means that possibility of such rectangle are present in 24*24 image. Hence training will be very expensive. To overcome this, **Adaboosting** comes into play. It takes a couple of features out of **162336**, multiples them with the weights. Weights decide how important features is an facial features and come before other features. All these features are called weak features and when these are added these forms strong features and this process is called ensemble. In this way training will not be expensive and done within a limited time. Let us see the below equation to understand this.

F(X) = a1f1(x) + a2f2(x) + a3f3(x) + …….

f1(x) is weak classifier 1 and it comes before others because its weights is large compared to others.

F(X) is strong classifier and this process is called ensemble

Cascading

This is the process of speeding up the process. In the previous heading we studied that strong classifiers is made using weak classifiers which are arranged according to their importance which implies that most important features are placed ahead of others. In cascading if the most important feature is not present in the sub-window, then sub-window is rejected. If that feature is present, then look for the second important feature. If it is present then go for the third one else reject the sub-window. This process is called cascading.

3.3 LITERATURE

3.3.1 Attendance System Using NFC Technology with Embedded Camera on Mobile Device

According to research journal “Attendance System Using NFC (Near Field Communication) Technology with Embedded Camera on Mobile Device” (Bhise, Khichi, Korde,Lokare, 2015). The attendance system is improved by using NFCtechnology and mobile application. According to the research paper, each

student is given a NFC tag that has a unique ID during their enrolment into the college. Attendance of each class will then be taken by touching or moving these tags on the lecturer mobile phone. The embedded camera on the phone will then capture the student"s face to send all the data to the college server to do validation and verification. The advantages of this method is where the NFC is simple to use, and the speed of connection establishment is very high. It indeed speeds up the attendance taking process a lot. However, this system couldn"t automatically spot the violation when the NFC tag is not personally tagged by the original owner. Apart from that, the convenience of the system which uses the mobile phone as the NFC reader was actually an inconvenience to the lecturer. Imagine if the lecturer had forgotten to bring their mobile phones to work, what would be the backup procedure for the attendance to be recorded? Moreover, most of the lecturer will not likely to prefer their personal smart phones to be used in this way due to privacy matter. Hence, unique information about the student like biometrics or face recognition, which is guanine for a student should be used in replacement of the NFC tag. This will ensure attendance to be taken originally by the actual student.

3.3.2 Face Recognition Based Attendance Marking System

The second research journals “Face Recognition Based Attendance Marking System” (SenthamilSelvi, Chitrakala, Antony Jenitha, 2014) is based on the identification of face recognition to solve the previous attendance system"s issues. This system uses camera to capture the images of the employee to do face detection and recognition. The captured image is compared one by one with the face database to search for the worker"s face where attendance will be marked when a result is found in the face database. The main advantage of this system is where attendance is marked on the server which is highly secure where no one can mark the attendance of other. Moreover, in this proposed system, the face detection algorithm is improved by using the skin classification technique to increase the accuracy of the detection process. Although more efforts are invested in the accuracy of the face detection algorithm, the system is yet not portable. This system requires a standalone computer which will need a constant power supply that makes it not portable. This type of system is only suitable for marking staff"s attendance as they

only need to report their presence once a day, unlike students which require to report their attendance at every class on a particular day, it will be inconvenient if the attendance marking system is not portable. Thus, to solve this issue, the whole attendance management system can be developed on an portable module so that it can be work just by executing the python program.

3.3.3 Fingerprint Based Attendance System Using Microcontroller and LabView

The third research journal “Fingerprint Based Attendance System Using Microcontroller and LabView” (Kumar Yadav, Singh, Pujari, Mishra, 2015) proposed a solution of using fingerprint to mark the attendance. This system is using 2 microcontrollers to deal with the fingerprint recognition process. Firstly, the fingerprint pattern will be obtained through a fingerprint sensor, then the information will be transmitted to microcontroller 1. Next microcontroller 1 will pass the information to microcontroller 2 to do the checking with the database that resides in it. After finding a student‟s match, the details are sent to the PC through serial communication to be displayed. This design is good as it accelerates development while maintaining design flexibility and simplifies testing. But again, this system is attached to a PC which make it not portable. Other than that, the database information cannot be accessible easily. Meaning that, for the parents whom are interested in knowing their child‟s attendance cannot easily or conveniently access the information. Therefore, to provide accessibility of the student‟s information to the legitimate concerned party, the information can be uploaded to a web server for easy access. While the authentication for the appropriate access can be enforced through a login screen.

3.3.4 RFID based Student Attendance System

According to the fourth research journal “RFID based Student Attendance System” (Hussain, Dugar, Deka, Hannan, 2014), the proposed solution is almost similar to the first research journal where RFID technology is used to improve the older attendance system. In this system, a tag and a reader is again used as a method of tracking the attendance of the students. The difference between the first journals with this is where attendance‟s information can be accessed through a web portal. It provides

more convenient for information retrieval. Again, this system is imperfect in the sense that, firstly, it is not portable, as the RFID reader can only work when it is connected to a PC. Secondly, the RFID tag is not a guanine information that can uniquely identify a student, thus, resulting in the inaccuracy of the collected attendance information.

In conclusion, a better attendance monitoring system should be developed based on its portability, accessibility and the accuracy of the collected attendance information.

3.4 SYSTEM DESIGN

The design part of the attendance monitoring system is divided into two sections which consist of the hardware and the software part. Before the software The design part can be developed, the hardware part is first completed to provide a platform for the software to work. Before the software part we need to install some libraries for effective working of the application. We install **OpenCV** and **Numpy**through **Python.**

3.4.1 Hardware Development

- Camera Module with good mega pixels.
- Power Supply Cable
- 16Gb Micro SD Card Class 10

3.4.2 Libraries Development

“3.4.2.1 OpenCV”

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision.

The OpenCV project was initially an Intel Research initiative to advance CPU-intensive applications, part of a series of projects including real-time raytracing and 3Ddisplay walls. The main contributors to the project included several optimization experts in Intel Russia, as well as Intel's Performance Library Team.

In the early days of OpenCV, the goals of the project were described as:

- Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.
- Disseminatevision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
- Advance vision-based commercial applications by making portable, performance-optimized code available for free – with a license that did not require code to be open or free itself.

OpenCV's application areas include:

- 2D and 3D feature toolkits
- Egomotion estimation
- Facial recognition system
- Gesture recognition
- Human–computer interaction (HCI)
- Mobile robotics
- Motion understanding
- Object identification
- Segmentation and recognition
- Stereopsis stereo vision: depth perception from 2 cameras
- Structure from motion (SFM)
- Motion tracking Augmented reality

To support some of the above areas, OpenCV includes a statistical machine learning library that contains:

- Boosting
- Decision tree learning
- Gradient boosting trees
- Expectation-maximization algorithm
- k-nearest neighbour algorithm
- Naive Bayes classifier
- Artificial neural networks
- Random forest

- SVM

Versions of OpenCV:

- Deep neural networks (DNN)The first alpha version of OpenCV was released to the public at the IEEE Conference on Computer Vision and Pattern Recognition in 2000, and five betas were released between 2001 and 2005. The first 1.0 version was released in 2006. A version 1.1 "pre-release" was released in October 2008.
- The second major release of the OpenCV was in October 2009. OpenCV 2 includes major changes to the C++ interface, aiming at easier, more type-safe patterns, new functions, and better implementations for existing ones in terms of performance (especially on multi-core systems). Official releases now occur every six months and development is now done by an independent Russian team supported by commercial corporations.
- In August 2012, support for OpenCV was taken over by a non-profit foundation OpenCV.org, which maintains a developer and user site.
- On May 2016, Intel signed an agreement to acquire Itseez, a leading developer of OpenCV.

Programming Language:

There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation. Wrappers in other languages such as C#, Perl, Ch, Haskell, and Ruby have been developed to encourage adoption by a wider audience.

Since version 3.4, OpenCV.js is a JavaScript binding for selected subset of OpenCV functions for the web platform.

Operating System Support:

All of the new developments and algorithms in OpenCV runs on the following desktop operating systems: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD.

OpenCV runs on the following mobile operating systems: Android, iOS, Maemo, BlackBerry 10. The user can get official releases from SourceForge or take the latest sources from GitHub. OpenCV uses CMake.

“3.4.2.2 NumPy”

NumPy is a package that defines a multi-dimensional array object and associated fast math functions that operate on it. It also provides simple routines for linear algebra and fft and sophisticated random-number generation. NumPy replaces both Numeric and Numarray.

Example demonstrating NumPy:

```
from numpy import *
from PIL import Image
ar =
ones((100,100),float3
2) ar = ar * 100 for i in
range(0,100): ar[i,:] =
100 + (i * 1.5) im =
Image.fromarray(ar,"F
")
```

The numpy namespace includes all names under the numpy.core and numpy.lib namespaces as well. Thus, import numpy will also import the names from numpy.core and numpy.lib. This is the recommended way to use numpy.

3.5 SOFTWARE DEVELOPMENT

There are two major system flows in the software development section as shown below:

- The creation of the face database
- The process of attendance taking

Both processes mentioned above are essential because they made up the

backbone of the attendance management system. In this section, the process of both flows will be briefly described. Meanwhile, their full functionality, specific requirements and also the methods/approach to accomplish such objectives will be discussed in the upcoming chapter.

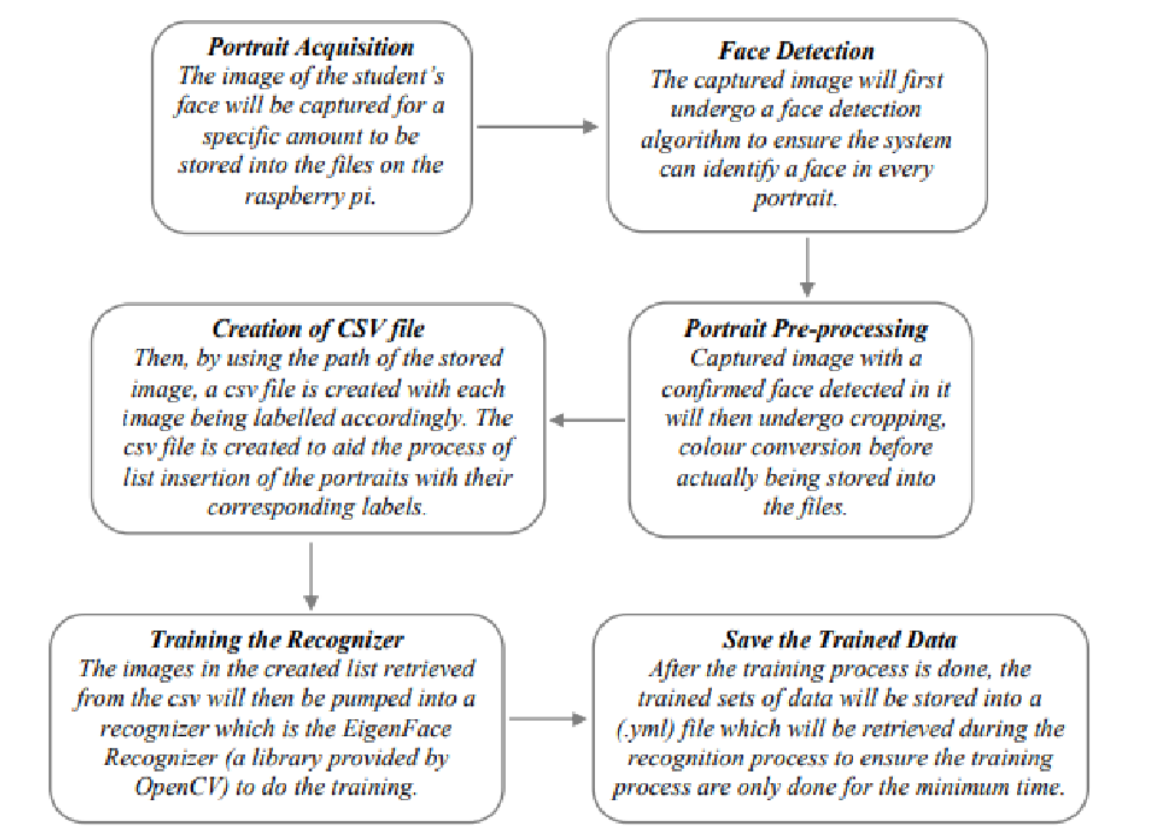


FIG 3.5.1The creation of the face database

The face database is an important step to be done before any further process can be initiated. This is because the face database acts as a comparison factor during the recognition process which will be discussed in later section. In the process above, a csv file is created to aid the process of image labelling because there will be more than one portrait stored for each student, thus, in order to group their portraits under the name of the same person, labels are used to distinguish them. After that, those images will be inserted into a recognizer to do its training. Since the training process is very time consuming as the face database grew larger, the training is only done right after there is a batch of new addition of student‟s portraits to ensure the training is done as minimum as possible.

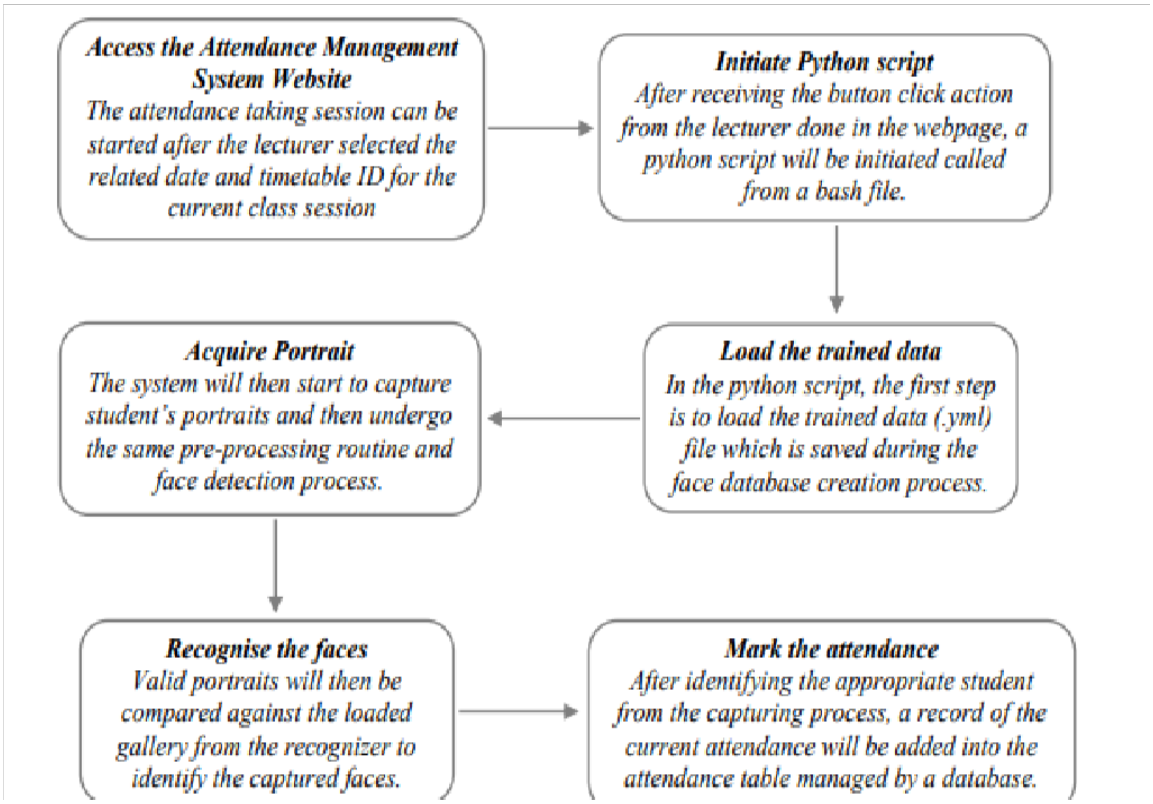


FIG 3.5.2The process of attendance taking

CHAPTER 4

RESULT AND DISCUSSION

4.1 RESULT

The result of this program is an application that enhances the user to store the attendance of their students. Advanced Facial Recognition Attendance System is a technological opportunity for the school, college, university and coaching centre institutions searching for a secure, simple and alternative solution to the conventional paper-based exam results evaluation, reporting and distribution.

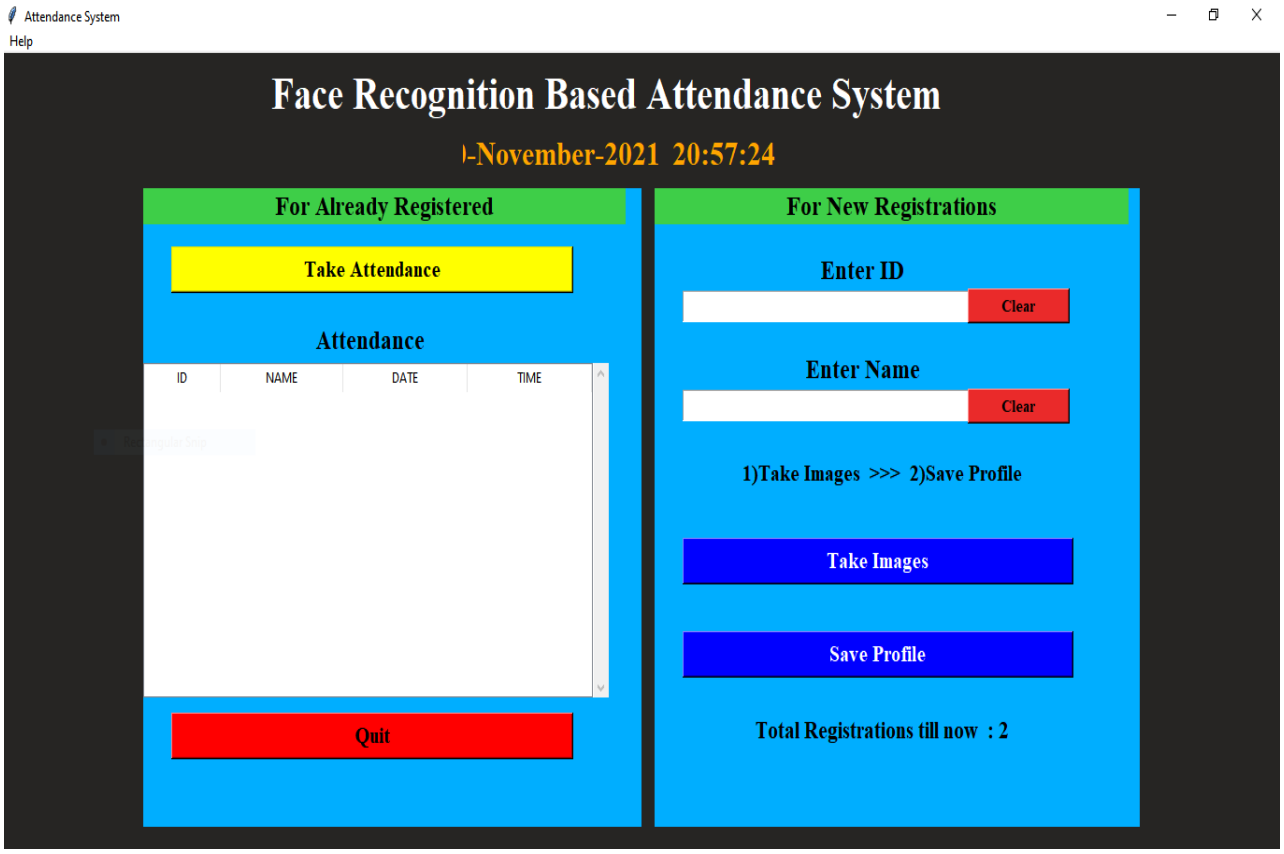


FIG 4.1.1 ATTENDANCE SYSTEM

FIG 4.1.2 TAKING IMAGES FOR DATA SET

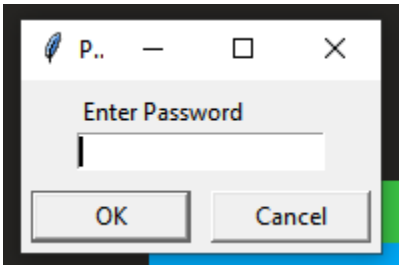


FIG 4.1.3 ENTER PASSWORD TO SAVE STUDENT PROFILE

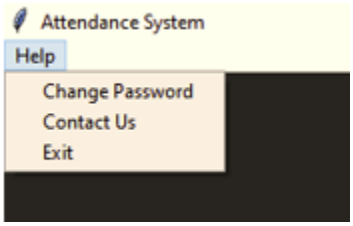


FIG 4.1.4 HELP MENU

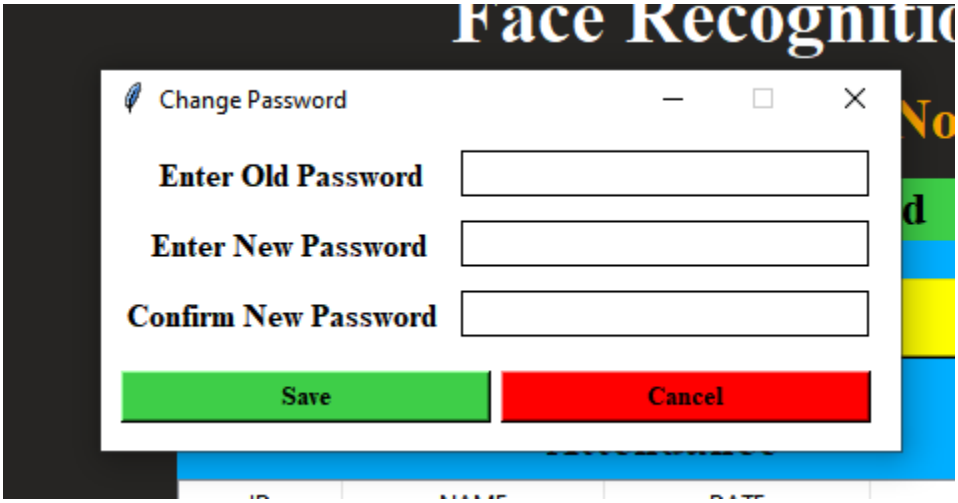


FIG 4.1.5 CHANGE PASSWORD

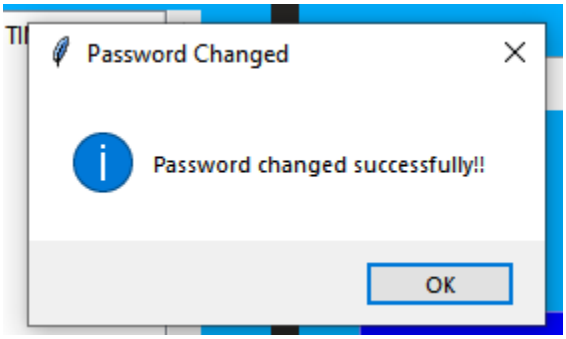


FIG 4.1.6 PASSWORD CHANGED

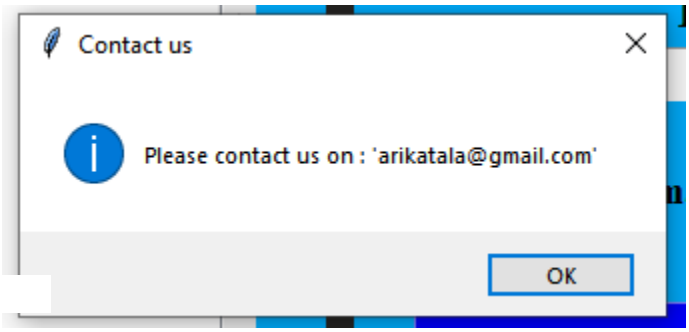


FIG 4.1.7 CONTACT US

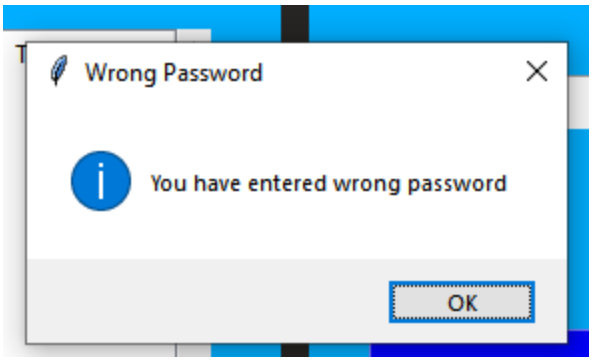


FIG 4.1.8 PASSWOD ERROR

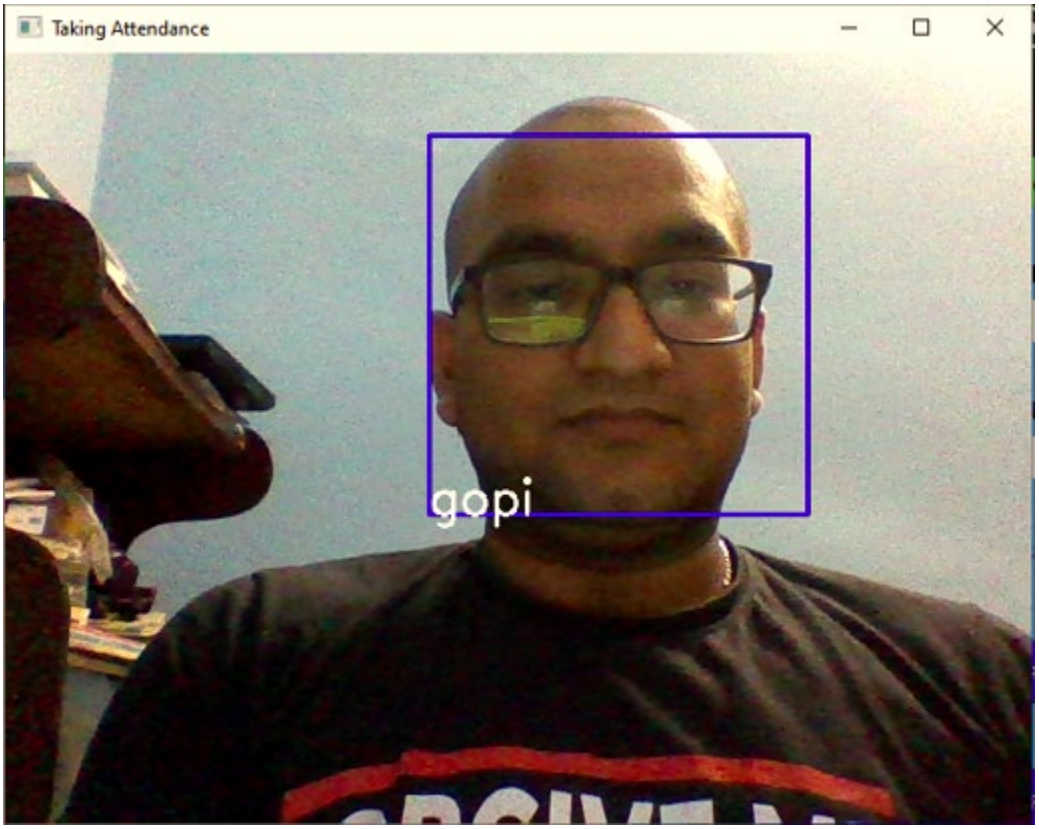


FIG 4.1.9 TAKING ATTENDANCE

CHAPTER 5

SUMMARY AND CONCLUSION

5.1 SUMMARY

The trial has shown conclusively that it is both possible and desirable to use Python as the principal teaching language:

1.It is free(as in both cost and source code)

2.it is trivial to install on a Windows PC allowing students to take their interest further. For many the hurdle of installing a Pascal or C compiler on a Windows machine is either too expensive or too complicated;

3.it is a flexible tool that allows both the teaching of traditional procedural programming and modern OOP; It can be used to teach a large number of transferable skills;

4.it is a real-world programming language that can be *and is* used in academia and the commercial world;

5.it appears to be quicker to learn and, in combination with its many libraries, this offers the possibility of more rapid student development allowing the course to be made more challenging and varied;

6.and most importantly, its clean syntax offers increased understanding and enjoyment for students

7.In conclusion, Python offers the optimum compromise of teachability and applicability.

5.2 FUTURE WORK

Before the development of this project. There are many loopholes in the process of taking attendance using the old method which caused many troubles to most of the institutions. Therefore, the facial recognition feature embedded in the attendance monitoring system can not only ensure attendance to be taken accurately and also eliminated the flaws in the previous system. By using technology to conquer the defects cannot merely save resources but also reduces human intervention in the whole process by handling all the complicated task to the machine. The only cost to this solution is to have sufficient space in to store all the faces into the database storage. Fortunately, there is such existence of micro SD that can compensate with the volume of the data. In this project, the face database is successfully built. Apart from that, the face recognizing system is also working well. At the end, the system not only resolve troubles that exist in the old model but also provide convenience to the user to access the information collected by mailing the attendance sheet to the respected faculty.

REFERENCES

[1]. “Attendance System Using NFC Technology with Embedded Camera on Mobile Device” (Bhise, Khichi, Korde,Lokare, 2015)

[2]. K.SenthamilSelvi, P.Chitrakala, A.AntonyJenitha, "Face Recognition Based Attendance Marking System", IJCSMC, Vol. 3, Issue. 2, February 2014.
[3]. “Fingerprint Based Attendance System Using Microcontroller and LabView” (Kumar Yadav, Singh, Pujari, Mishra, 2015)
[4]. “RFID based Student Attendance System” (Hussain, Dugar, Deka, Hannan, 2014)
[5]. OpenCvDocumentation -https://opencv.org
[6]. Numpy - https://numpy.org

METHODOLOGY

Before the attendance management system can work, there are a set of data needed to be inputted into the system which essentially consist of the individual‟s basic information which is their ID and their faces. The first procedure of portrait acquisition can be done by using the Camera to capture the faces of the individual. In this process the system will first detect the presence of a face in the captured image, if there are no face detected, the system will prompt the user to capture their face again until it meets certain number of portraits which will be 10 required portraits in this project for each student. The decision of storing only 10 portrait per student is due to the consideration of the limited storage space in the raspberry pi because the total amount of students in the university is considered heavy. Then, the images will undergo several pre-processing procedures to obtain a grayscale image and cropped faces of equal sized images because those are the prerequisites of using the EigenFaces Recognizer. Both of the processes mentioned above can be represented in the diagram below.

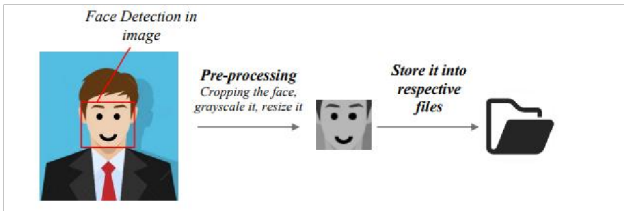


Image Acquisition and Pre-processing procedures

After the images are being processed, they are stored into a file in a hierarchy manner. In this project, all the faces will be stored in a hierarchy manner under the

„database" folder. When expanding through the database folder, there will consist of many sub-folders which each of them will represent an individual where a series of face portrait belonging to the same individual will be stored in that particular sub-folder. The subfolders that represent each individual will be named upon the ID no. of that individual which is unique for every single individual in the institution. The whole process of image retrieval, pre-processing, storing mechanism is done by the script named create_database.py

Hierarchy manner of the face database

After a successful retrieval of facial images into the respective folder, a CSV files created to aid the next process of pumping the faces into the recognizer for the training process. The creation of the CSV file will be done based on a script named create_csv.py. In this project, the content of CSV file will look like the following format:

Structure of the content in the csv file

After having sufficient images in the database, those images will then be inserted into a training mechanism. There are generally 3 different types of training mechanism provided in OpenCV 3.4 which are EigenFaces, FisherFaces, and Local Binary Patterns Histograms (LBPH). The recognizer that will be focused in this project will be the EigenFaces recognizer. The concept behind EigenFaces is simple – it recognizes a particular face by catching the maximum deviation in a face and then turning those identified variations into information to be compared when a new face arrives. In the training process, the csv file will be read to provide the path to all of the images where those images and labels will be loaded into a list variable. Then, the list will be passed into the training function where the training process will take a measurable time to run. The larger the face database, the longer the time will be needed to train those images.

FLOW CHART OF THE IMAGE ACQUISITION PROCESS

The development of the face database is an important phase before any facial recognizing process can be carried out. It acts as a library to compare against with whenever the system wanted to identify a person. In the image retrieval process, the

system will first prompt for an input from the user to enter their ID number. The system will then validate the entered input and then check for duplication in the system. In order to proceed, the entered input must contain only 12 digits of number. Apart from that, the ID inputted have to be a non-registered ID to ensure no duplication. After that, a directory is created for each individual where their portraits will be stored inside of it. It is a compulsory to store 10 - 30 portraits per person in the file. After the acquisition of image is done, the images undergo a pre-processing before storing it into the respective folder.

Flow Chart of the image retrieval process

The above flowchart is only the program flow for the image acquisition process which describes the program flow for the script create_database.py. There are two more python scripts that responsible for the remaining execution which will be explained in the next sub-section.

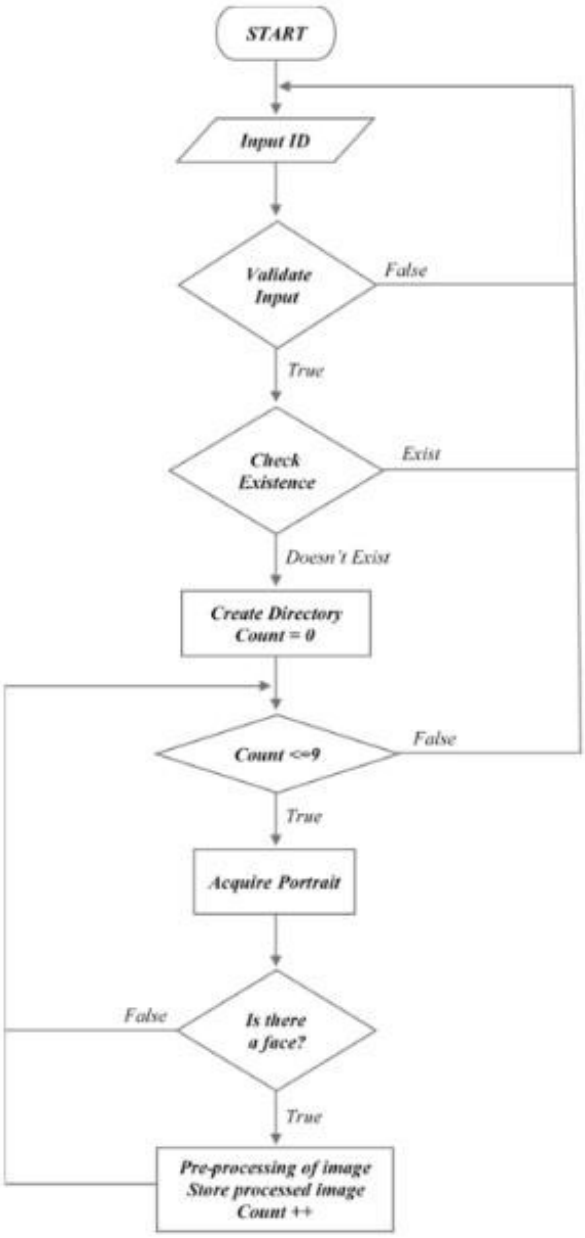


FIG 6 FLOW CHART OF THE IMAGE ACQUISITION PROCESS

SOURCE CODE

```
############################################## IMPORTING
################################################
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox as mess
import tkinter.simpledialog as tsd
import cv2,os
import csv
import numpy as np
from PIL import Image
import pandas as pd
import datetime
import time

############################################## FUNCTIONS
################################################

def assure_path_exists(path):
    dir = os.path.dirname(path)
    if not os.path.exists(dir):
        os.makedirs(dir)

#####################################################################
###############
```

```
def tick():
    time_string = time.strftime('%H:%M:%S')
    clock.config(text=time_string)
    clock.after(200,tick)

##################################################################################

def contact():
    mess._show(title='Contact us', message="Please contact us on :
'arikatala@gmail.com' ")

##################################################################################

def check_haarcascadefile():
    exists = os.path.isfile("haarcascade_frontalface_default.xml")
    if exists:
        pass
    else:
        mess._show(title='Some file missing', message='Please contact us for help')
        window.destroy()

##################################################################################

def save_pass():
    assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
        tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
```

```
    else:
        master.destroy()
        new_pas = tsd.askstring('Old Password not found', 'Please enter a new
password below', show='*')
        if new_pas == None:
            mess._show(title='No Password Entered', message='Password not set!!
Please try again')
        else:
            tf = open("TrainingImageLabel\psd.txt", "w")
            tf.write(new_pas)
            mess._show(title='Password Registered', message='New password was
registered successfully!!')
            return
    op = (old.get())
    newp= (new.get())
    nnewp = (nnew.get())
    if (op == key):
        if(newp == nnewp):
            txf = open("TrainingImageLabel\psd.txt", "w")
            txf.write(newp)
        else:
            mess._show(title='Error', message='Confirm new password again!!!')
            return
    else:
        mess._show(title='Wrong Password', message='Please enter correct old
password.')
        return
    mess._show(title='Password Changed', message='Password changed
successfully!!')
    master.destroy()

####################################################################
```

```
################

def change_pass():
    global master
    master = tk.Tk()
    master.geometry("400x160")
    master.resizable(False,False)
    master.title("Change Password")
    master.configure(background="white")
    lbl4 = tk.Label(master,text='    Enter Old Password',bg='white',font=('times', 12, '
bold '))
    lbl4.place(x=10,y=10)
    global old
    old=tk.Entry(master,width=25 ,fg="black",relief='solid',font=('times', 12, ' bold
'),show='*')
    old.place(x=180,y=10)
    lbl5 = tk.Label(master, text='   Enter New Password', bg='white', font=('times', 12, '
bold '))
    lbl5.place(x=10, y=45)
    global new
    new = tk.Entry(master, width=25, fg="black",relief='solid', font=('times', 12, ' bold
'),show='*')
    new.place(x=180, y=45)
    lbl6 = tk.Label(master, text='Confirm New Password', bg='white', font=('times', 12, '
bold '))
    lbl6.place(x=10, y=80)
    global nnew
    nnew = tk.Entry(master, width=25, fg="black", relief='solid',font=('times', 12, ' bold
'),show='*')
    nnew.place(x=180, y=80)
    cancel=tk.Button(master,text="Cancel", command=master.destroy ,fg="black"
,bg="red" ,height=1,width=25 , activebackground = "white" ,font=('times', 10, ' bold '))
```

```
    cancel.place(x=200, y=120)
    save1 = tk.Button(master, text="Save", command=save_pass, fg="black",
bg="#3ece48", height = 1,width=25, activebackground="white", font=('times', 10, '
bold '))
    save1.place(x=10, y=120)
    master.mainloop()

####################################################################################

def psw():
    assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
        tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
    else:
        new_pas = tsd.askstring('Old Password not found', 'Please enter a new
password below', show='*')
        if new_pas == None:
            mess._show(title='No Password Entered', message='Password not set!!
Please try again')
        else:
            tf = open("TrainingImageLabel\psd.txt", "w")
            tf.write(new_pas)
            mess._show(title='Password Registered', message='New password was
registered successfully!!')
            return
    password = tsd.askstring('Password', 'Enter Password', show='*')
    if (password == key):
        TrainImages()
    elif (password == None):
```

```
        pass
    else:
        mess._show(title='Wrong Password', message='You have entered wrong
password')

######################################################################
####################

def clear():
    txt.delete(0, 'end')
    res = "1)Take Images  >>>  2)Save Profile"
    message1.configure(text=res)


def clear2():
    txt2.delete(0, 'end')
    res = "1)Take Images  >>>  2)Save Profile"
    message1.configure(text=res)

######################################################################
####################

def TakeImages():
    check_haarcascadefile()
    columns = ['SERIAL NO.', '', 'ID', '', 'NAME']
    assure_path_exists("StudentDetails/")
    assure_path_exists("TrainingImage/")
    serial = 0
    exists = os.path.isfile("StudentDetails\StudentDetails.csv")
    if exists:
        with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
            reader1 = csv.reader(csvFile1)
```

```
            for l in reader1:
                serial = serial + 1
        serial = (serial // 2)
        csvFile1.close()
    else:
        with open("StudentDetails\StudentDetails.csv", 'a+') as csvFile1:
            writer = csv.writer(csvFile1)
            writer.writerow(columns)
            serial = 1
        csvFile1.close()
    Id = (txt.get())
    name = (txt2.get())
    if ((name.isalpha()) or (' ' in name)):
        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
                # saving the captured face in the dataset folder TrainingImage
                cv2.imwrite("TrainingImage\ " + name + "." + str(serial) + "." + Id + '.' +
str(sampleNum) + ".jpg",
                            gray[y:y + h, x:x + w])
                # display the frame
                cv2.imshow('Taking Images', img)
            # wait for 100 miliseconds
```

```
            if cv2.waitKey(100) & 0xFF == ord('q'):
                break
            # break if the sample number is morethan 100
            elif sampleNum > 100:
                break
        cam.release()
        cv2.destroyAllWindows()
        res = "Images Taken for ID : " + Id
        row = [serial, '', Id, '', name]
        with open('StudentDetails\StudentDetails.csv', 'a+') as csvFile:
            writer = csv.writer(csvFile)
            writer.writerow(row)
        csvFile.close()
        message1.configure(text=res)
    else:
        if (name.isalpha() == False):
            res = "Enter Correct name"
            message.configure(text=res)

##########################################################################
####################

def TrainImages():
    check_haarcascadefile()
    assure_path_exists("TrainingImageLabel/")
    recognizer = cv2.face_LBPHFaceRecognizer.create()
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector = cv2.CascadeClassifier(harcascadePath)
    faces, ID = getImagesAndLabels("TrainingImage")
    try:
        recognizer.train(faces, np.array(ID))
    except:
```

```
        mess._show(title='No Registrations', message='Please Register someone
first!!!')
        return
    recognizer.save("TrainingImageLabel\Trainner.yml")
    res = "Profile Saved Successfully"
    message1.configure(text=res)
    message.configure(text='Total Registrations till now  : ' + str(ID[0]))

##################################################################
##########################3

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)]
    # 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

##################################################################
```

```
########################

def TrackImages():
    check_haarcascadefile()
    assure_path_exists("Attendance/")
    assure_path_exists("StudentDetails/")
    for k in tv.get_children():
        tv.delete(k)
    msg = ''
    i = 0
    j = 0
    recognizer = cv2.face.LBPHFaceRecognizer_create()  #
cv2.createLBPHFaceRecognizer()
    exists3 = os.path.isfile("TrainingImageLabel\Trainner.yml")
    if exists3:
        recognizer.read("TrainingImageLabel\Trainner.yml")
    else:
        mess._show(title='Data Missing', message='Please click on Save Profile to
reset data!!')
        return
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath);

    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names = ['Id', '', 'Name', '', 'Date', '', 'Time']
    exists1 = os.path.isfile("StudentDetails\StudentDetails.csv")
    if exists1:
        df = pd.read_csv("StudentDetails\StudentDetails.csv")
    else:
        mess._show(title='Details Missing', message='Students details are missing,
please check!')
```

```
    cam.release()
    cv2.destroyAllWindows()
    window.destroy()
while True:
    ret, im = cam.read()
    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)
        serial, conf = recognizer.predict(gray[y:y + h, x:x + w])
        if (conf < 50):
            ts = time.time()
            date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
            timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
            aa = df.loc[df['SERIAL NO.'] == serial]['NAME'].values
            ID = df.loc[df['SERIAL NO.'] == serial]['ID'].values
            ID = str(ID)
            ID = ID[1:-1]
            bb = str(aa)
            bb = bb[2:-2]
            attendance = [str(ID), '', bb, '', str(date), '', str(timeStamp)]

        else:
            Id = 'Unknown'
            bb = str(Id)
        cv2.putText(im, str(bb), (x, y + h), font, 1, (255, 255, 255), 2)
    cv2.imshow('Taking Attendance', im)
    if (cv2.waitKey(1) == ord('q')):
        break
ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
exists = os.path.isfile("Attendance\Attendance_" + date + ".csv")
```

```
    if exists:
        with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:
            writer = csv.writer(csvFile1)
            writer.writerow(attendance)
        csvFile1.close()
    else:
        with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:
            writer = csv.writer(csvFile1)
            writer.writerow(col_names)
            writer.writerow(attendance)
        csvFile1.close()
    with open("Attendance\Attendance_" + date + ".csv", 'r') as csvFile1:
        reader1 = csv.reader(csvFile1)
        for lines in reader1:
            i = i + 1
            if (i > 1):
                if (i % 2 != 0):
                    iidd = str(lines[0]) + '   '
                    tv.insert('', 0, text=iidd, values=(str(lines[2]), str(lines[4]), str(lines[6])))
    csvFile1.close()
    cam.release()
    cv2.destroyAllWindows()

########################################## USED STUFFS
##########################################

global key
key = ''

ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
day,month,year=date.split("-")
```

```
mont={'01':'January',
      '02':'February',
      '03':'March',
      '04':'April',
      '05':'May',
      '06':'June',
      '07':'July',
      '08':'August',
      '09':'September',
      '10':'October',
      '11':'November',
      '12':'December'
      }

######################################### GUI FRONT-END
###############################################

window = tk.Tk()
window.geometry("1280x720")
window.resizable(True,False)
window.title("Attendance System")
window.configure(background='#262523')

frame1 = tk.Frame(window, bg="#00aeff")
frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)

frame2 = tk.Frame(window, bg="#00aeff")
frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)

message3 = tk.Label(window, text="Face Recognition Based Attendance System"
,fg="white",bg="#262523" ,width=55 ,height=1,font=('times', 29, ' bold '))
```

```
message3.place(x=10, y=10)

frame3 = tk.Frame(window, bg="#c4c6ce")
frame3.place(relx=0.52, rely=0.09, relwidth=0.09, relheight=0.07)

frame4 = tk.Frame(window, bg="#c4c6ce")
frame4.place(relx=0.36, rely=0.09, relwidth=0.16, relheight=0.07)

datef = tk.Label(frame4, text = day+"-"+mont[month]+"-"+year+"  |  ",
fg="orange",bg="#262523" ,width=55 ,height=1,font=('times', 22, ' bold '))
datef.pack(fill='both',expand=1)

clock = tk.Label(frame3,fg="orange",bg="#262523" ,width=55 ,height=1,font=('times',
22, ' bold '))
clock.pack(fill='both',expand=1)
tick()

head2 = tk.Label(frame2, text="                    For New Registrations                    ",
fg="black",bg="#3ece48" ,font=('times', 17, ' bold ') )
head2.grid(row=0,column=0)

head1 = tk.Label(frame1, text="                    For Already Registered                    ",
fg="black",bg="#3ece48" ,font=('times', 17, ' bold ') )
head1.place(x=0,y=0)

lbl = tk.Label(frame2, text="Enter ID",width=20  ,height=1  ,fg="black"  ,bg="#00aeff"
,font=('times', 17, ' bold ') )
lbl.place(x=80, y=55)

txt = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold '))
txt.place(x=30, y=88)
```

```
lbl2 = tk.Label(frame2, text="Enter Name",width=20  ,fg="black"  ,bg="#00aeff"
,font=('times', 17, ' bold '))
lbl2.place(x=80, y=140)

txt2 = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold ')  )
txt2.place(x=30, y=173)

message1 = tk.Label(frame2, text="1)Take Images  >>>  2)Save Profile"
,bg="#00aeff" ,fg="black"  ,width=39 ,height=1, activebackground = "yellow"
,font=('times', 15, ' bold '))
message1.place(x=7, y=230)

message = tk.Label(frame2, text="" ,bg="#00aeff" ,fg="black"  ,width=39,height=1,
activebackground = "yellow" ,font=('times', 16, ' bold '))
message.place(x=7, y=450)

lbl3 = tk.Label(frame1, text="Attendance",width=20  ,fg="black"  ,bg="#00aeff"
,height=1 ,font=('times', 17, ' bold '))
lbl3.place(x=100, y=115)

res=0
exists = os.path.isfile("StudentDetails\StudentDetails.csv")
if exists:
    with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
        reader1 = csv.reader(csvFile1)
        for l in reader1:
            res = res + 1
    res = (res // 2) - 1
    csvFile1.close()
else:
    res = 0
message.configure(text='Total Registrations till now  : '+str(res))
```

```
###################### MENUBAR ###################################

menubar = tk.Menu(window,relief='ridge')
filemenu = tk.Menu(menubar,tearoff=0)
filemenu.add_command(label='Change Password', command = change_pass)
filemenu.add_command(label='Contact Us', command = contact)
filemenu.add_command(label='Exit',command = window.destroy)
menubar.add_cascade(label='Help',font=('times', 29, ' bold '),menu=filemenu)

################### TREEVIEW ATTENDANCE TABLE ####################

tv= ttk.Treeview(frame1,height =13,columns = ('name','date','time'))
tv.column('#0',width=82)
tv.column('name',width=130)
tv.column('date',width=133)
tv.column('time',width=133)
tv.grid(row=2,column=0,padx=(0,0),pady=(150,0),columnspan=4)
tv.heading('#0',text ='ID')
tv.heading('name',text ='NAME')
tv.heading('date',text ='DATE')
tv.heading('time',text ='TIME')

###################### SCROLLBAR ###################################

scroll=ttk.Scrollbar(frame1,orient='vertical',command=tv.yview)
scroll.grid(row=2,column=4,padx=(0,100),pady=(150,0),sticky='ns')
tv.configure(yscrollcommand=scroll.set)

###################### BUTTONS ###################################

clearButton = tk.Button(frame2, text="Clear", command=clear  ,fg="black"
```

```
,bg="#ea2a2a"  ,width=11 ,activebackground = "white" ,font=('times', 11, ' bold '))
clearButton.place(x=335, y=86)
clearButton2 = tk.Button(frame2, text="Clear", command=clear2  ,fg="black"
,bg="#ea2a2a"  ,width=11 , activebackground = "white" ,font=('times', 11, ' bold '))
clearButton2.place(x=335, y=172)
takeImg = tk.Button(frame2, text="Take Images", command=TakeImages  ,fg="white"
,bg="blue"  ,width=34  ,height=1, activebackground = "white" ,font=('times', 15, ' bold
'))
takeImg.place(x=30, y=300)
trainImg = tk.Button(frame2, text="Save Profile", command=psw ,fg="white"
,bg="blue"  ,width=34  ,height=1, activebackground = "white" ,font=('times', 15, ' bold
'))
trainImg.place(x=30, y=380)
trackImg = tk.Button(frame1, text="Take Attendance", command=TrackImages
,fg="black"  ,bg="yellow"  ,width=35  ,height=1, activebackground = "white"
,font=('times', 15, ' bold '))
trackImg.place(x=30,y=50)
quitWindow = tk.Button(frame1, text="Quit", command=window.destroy  ,fg="black"
,bg="red"  ,width=35 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
quitWindow.place(x=30, y=450)

###################### END ######################################

window.configure(menu=menubar)
window.mainloop()
```