AN INDUSTRY ORIENTED MINI PROJECT REPORT

On

AUTHENTICATION BY FACE RECOGNITION USING OPENCV-PYHTON

Submitted in partial fulfillment of the requirements.
For the award of the degree of

BACHELOR OF TECHNOLOGY

In
Electronics & Communication Engineering
By

Nerusu Guru Preetham (Regd No. 17891A04E9)

Rapelli Sai Rajesh (Regd No. 17891A04F3)

Yasarla Nitesh Naidu (Regd No. 17891A04G2)

Under the guidance of

Dr. N. Dinesh Kumar

Guide & Head of the department

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VIGNAN INSTITUTE OF TECHNOLOGY AND SCIENCE

(Affiliated to Jawaharlal Nehru Technological University)
Deshmukhi(V),Pochampally(M), Yadadri-Bhuvanagiri District,Telangana,508284.
Ph No: 08685-226128, 9866399776/861.

2020-2021

AN INDUSTRY ORIENTED MINI PROJECT REPORT

On

AUTHENTICATION BY FACE RECOGNITION USING OPENCV-PYHTON

Submitted in partial fulfilment of the requirements.
For the award of the degree of

BACHELOR OF TECHNOLOGY

In
Electronics & Communication Engineering
By

Nerusu Guru Preetham (Regd No. 17891A04E9)

Rapelli Sai Rajesh (Regd No. 17891A04F3)

Yasarla Nitesh Naidu (Regd No. 17891A04G2)

Under the guidance of

Dr. N. Dinesh Kumar

Guide & Head of the department

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VIGNAN INSTITUTE OF TECHNOLOGY AND SCIENCE

(Affiliated to Jawaharlal Nehru Technological University)
Deshmukhi(V),Pochampally(M), Yadadri-Bhuvanagiri District,Telangana,508284.
Ph No: 08685-226128, 9866399776/861.

2020-2021

DECLARATION

I hereby declare that project entitled "**AUTHENTICATION BY FACE RECOGNITION USING OPENCV-PYHTON**" is bonafide work duly completed by me/us. It does not contain any part of the project or thesis submitted by any other candidate to this or any other institute of the university.

All such materials that have been obtained from other sources have been duly acknowledged.

Nerusu Guru Preetham

Regd No. 17891A04E9

Rapelli Sai Rajesh

Regd No. 17891A04F3

Yasarla Nitesh Naidu

Regd No. 17891A04G2

VIGNAN INSTITUTE OF TECHNOLOGY AND SCIENCE

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the thesis work titled "**AUTHENTICATION BY FACE-RECOGNITION USING OPEN-CV PYTHON**" submitted by **Mr. Nerusu Guru Preetham (Regd No. 17891A04E9), Mr. Rapelli Sai Rajesh (Regd No. 17891A04F3), Mr. Yasarla Nitesh Naidu (Regd No. 17891A04G2)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Electronics & Communication Engineering** to the Vignan Institute of Technology And Science, Deshmukhi is a record of bonafide work carried out by him/her under my guidance and supervision.

The results embodied in this project report have not been submitted in any university for the award of any degree and the results are achieved satisfactorily.

Dr.N.Dinesh Kumar
Guide & Head of The Department

PREFACE

The project "**AUTHENTICATION BY FACE-RECOGNITION USING OPEN-CV PYTHON**".

Chapter 1: A brief introduction of "**AUTHENTICATION BY FACE-RECOGNITION USING OPEN-CV PYTHON**", problem description, proposed solution and advantages. Problem description describes the major challenges and problems faced in order to achieve face recognition. The Proposed solution describes the procedure to achieve the major objective. Advantages describe the benefits of the project.

Chapter 2: Presents the topic literature survey. It explains about the history of face recognition and different approaches made in order to achieve face recognition.

Chapter 3: Presents the methodology of face recognition using opencv-python. It deals with the block diagram of the project, software description and project explanation. The tools and major modules utilised in the project.

Chapter 4: Presents outputs which represent the clock-in, clock-out time and measured time between clock-in and clock-out time and the final spreadsheet.

Chapter 5: Presents the conclusion, future scope and the changes to be made in the project for different purposes.

ACKNOWLEDGEMENT

This project would not have been possible without considerable guidance and support. So, I would like to acknowledge those who have enabled us to complete this seminar.

Firstly, I would like to express my gratitude to our CEO **Mr. Shravan Boyapati**, management and Principal **Dr. Durga Sukumar** for extending his support

Secondly, my sincere thanks to our head of the department, **Dr. N. Dinesh Kumar**, who helped me in selecting the topic, correcting various documents with attention and care. He has taken the pain to go through the project and make necessary correction as and when needed.

Finally, I would also like to thank our staff members that we called upon for assistance since the genesis of this project. Their opinions and suggestions have helped us in realizing this seminar and with their support and sharing ideas during the progress. I also extend my heartfelt thanks to my parents and well wishes.

ABSTRACT

The main purpose of this work is to build a face recognition-base attendance monitoring system for educational institution to enhance and upgrade the current attendance system into more efficient and effective as compared to before.

In this project, the Open CV based face recognition approach has been proposed. This model integrates a camera that captures an input image, an algorithm for detecting face from an input image, encoding and identifying the face then marking the attendance in a spreadsheet. The training database is created by training the system with the faces of the authorized students. The cropped images are then stored in the dataset folder with respective labels.

Some of the places such systems are either being used or could be used Airports, Railway Stations, Banks and Financial Institutions, Stadiums, Public Transport, Government Offices, Business Establishments.

CONTENTS **PAGE NO**

List of figures

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION TO THE PROJECT

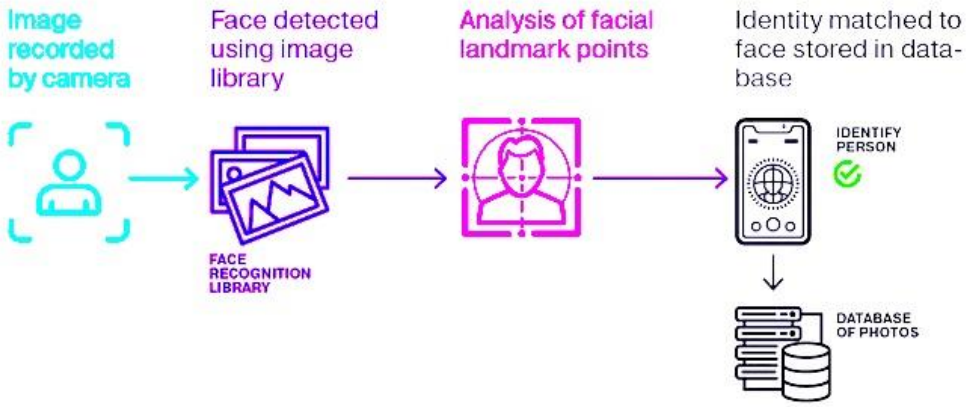

Fig:1 Block diagram

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

The various techniques for marking attendance are:-

1) **Fingerprint Based Recognition System:-** In the Fingerprint based existing attendance system, a portable fingerprint device needs to be configured with the students fingerprint earlier. Later either during the lecture hours or before, the student needs to record the fingerprint on the configured device to ensure their attendance for the day. The problem with this approach is that during the lecture time it may distract the attention of the students.
2) **RFID (Radio Frequency Identification) Based Recognition System:-** In the RFID based existing system, the student needs to carry a Radio Frequency Identity Card with them and place the ID on the card reader to record their presence for the

day. The system is capable of to connect to RS232 and record the attendance to the saved database. There are possibilities for the fraudulent access may occur. Some are students may make use of other students ID to ensure their presence when the particular student is absent or they even try to misuse it sometimes.

3) Iris Based Recognition System:- In the Iris based student attendance system, the student needs to stand in front of a camera, so that the camera will scan the Iris of the student. The scanned iris is matched with data of student stored in the database and the attendance on their presence needs be updated. This reduces the paper and pen workload of the faculty member of the institute. This also reduces the chances of proxies in the class, and helps in maintaining the student records safe. It is a wireless biometric technique that solves the problem of spurious attendance and the trouble of laying the corresponding network.

4)Face Based Recognition System:- The facial recognition technology can be used in recording the attendance through a high-resolution digital camera that detects and recognizes the faces of the students and the machine compares the recognized face with students' face images stored in the database. Once the face of the student is matched with the stored image, then the attendance is marked in attendance database for further calculation. If the captured image doesn't match with the student's face present in the database then the images is marked as unknown. In this system, there are possibilities for the camera to not to capture the image properly or it may miss some of the students from capturing. Amongst the above techniques, Face Recognition is natural, easy to use and does not require aid from the test subject.

It is a series of several related problems which are solved step by step:

1. To capture a picture and discern all the faces in it.

2. Concentrate on one face at a time and understand that even if a face is turned in a strange direction or in bad lighting, it is still the same person.

3. Determine various unique features of the face that can help in distinguishing it from the face of any other person. These characteristics could be the size eyes, nose, length of face, skin colour, etc.

4. Compare these distinctive features of that face to all the faces of people we already know to find out the person's name.

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

1.2 PROBLEM DESCRIPTION:

Recognizing faces in computer vision is a challenging problem. The illumination problem, the pose problem, scale variability, low quality image acquisition, partially occluded faces are some examples of the issues to deal with. Thus, face recognition algorithms must exhibit robustness to variations in the above parameters. The existing techniques do not perform well in cases of different illumination, background or rotation. Thus, there is a need to address the above-mentioned disadvantages. The project aims to design and implement a system which is less sensitive to Illumination, is rotation invariant, scale invariant and robust enough to be implemented in practical applications. Although many face recognition opencv algorithms have been developed over the years, their speed and accuracy balance has not been quite optimal**.** But some recent advancements have shown promise. A good example is Facebook, where they are able to tag you and your friends with just a few images of training and with accuracy as high as 98%.

Face recognition is a series of several problems:-

1.First, look at a picture and find all the faces in it

2.Second, focus on each face and be able to understand that even if a face is turned in a weird direction or in bad lighting, it is still the same person.

3.Third, be able to pick out unique features of the face that you can use to tell it apart from other people like how big the eyes are, how long the face is, etc.

4.Finally, compare the unique features of that face to all the people you already know to determine the person's name.

1.3 PROPOSED SOLUTION :-

The attendance is recorded by using a camera that will stream video of students, detect the faces in the image and compare the detected faces with the images in the dataset and mark the attendance. The attendance gets marked in a spreadsheet including clock-in time and clock-out time. Face recognition is achieved using OpenCV and the basic pipeline used for it is as follows:

1. Finds face in an image.

2. Analyses facial features.

3. Compares against known faces and stores the name of the person in the spreadsheet.

Development of Face Recognition attendance system App is achieved using PYQT5 which is a python GUI module. Here the application takes image data in the dataset folder and it takes the video as input, detects every single person's face frame by frame and provides a clock in button to start measuring the clock in time of the attendee and there is another clock-out button for measuring the time between the clock-in and clock-out of the attendee. Every individual must provide his/her facial data during the clock-in and clock-out time.

1.4 ADVANTAGES OF FACIAL RECOGNITION ATTENDANCE SYSTEM:-

1) An automated time tracking system

A time and attendance system using facial recognition technology can accurately report attendance, absence, and overtime with an identification process that is fast as well as accurate.

2) Labour cost saving

It keeps track of the exact number of hours an employee is working, which can help save the company money. You will never have to worry about time fraud or "buddy punching" with a facial recognition time tracking system.

3) Tighter security

4) Time-saving and reduced contagion

With facial recognition, employees can enter and leave the facility in considerably less time. There is need to touch the surface of the system to clock in or out. This saves time, as well as minimizing the spread of illnesses due to physical contact.

5) Ease of integration

It's customizable and can be easily programmed into your time and attendance system.

CHAPTER-2

LITERATURE SURVEY

“Eigenfaces for recognition’’ (Mathew Turk and Alex Pentland) [1], here they have developed a near-real time computer system that can locate and track a subject’s head, and then recognize the person by comparing characteristics of the face to those of known individuals. The computational approach taken in this system is motivated by both physiology and information theory, as well as by the practical requirements of near-real time performance and accuracy. This approach treats the face recognition problem as an intrinsically two-dimensional recognition problem rather than requiring recovery of threedimensional geometry, taking advantage of the fact that these faces are normally upright and thus may be described by a small set of two-dimensional characteristic views. Their experiments show that the eigenface technique can be made to perform at very high accuracy, although with a substantial “unknown “rejection rate and thus potentially well suited to these applications. The future scope of this project was-in addition to recognizing face, to use eigenface analysis to determine the gender of the subject and to interpret facial expressions.

“Fast face recognition using eigenfaces” (Arun Vyas and Rajbala Tokas) [2], their approach signifies face recognition as a two-dimensional problem. In this approach, face reorganization is done by Principal Component Analysis (PCA). Face images are faced onto a space that encodes best difference among known face images. The face space is created by eigenface methods which are eigenvectors of the set of faces, which may not link to general facial features such as eyes, nose, and lips. The eigenface method uses the PCA for recognition of the images. The system performs by facing pre-extracted face image onto a set of face space that shows significant difference among known face images. Face will be categorized as known or unknown face after imitating it with the present database. From the obtained results, it was concluded that, for recognition, it is sufficient to take about 10% eigenfaces with the highest eigenvalues. It is also clear that the recognition rate increases with the number of training images.

“Face recognition using eigenfaces and artificial neural networks” (Mayank Agarwal, Nikunj Jain, Mr. Manish Kumar and Himanshu Agrawal) [4], this paper presents a methodology for face recognition based on information theory approach of coding and decoding the face image. Proposed methodology is connection of two stages – Feature extraction using principle component analysis and recognition using the feed forward back propagation Neural Network. The algorithm has been tested on 400 images (40 classes). A recognition score for test lot is calculated by considering almost all the variants of feature extraction. The proposed methods were tested on Olivetti and Oracle Research Laboratory (ORL) face database. Test results gave a recognition rate of 97.018%.

CHAPTER-3

SOFTWARE EXPLANATION

Editor & Compiler:- Python & VSCode

Tools:- QT Designer For GUI

3.1 Modules used:- OpenCV, Face_Recognition, CMake, dlib, Numpy, Pandas, PYQT5 ,datetime and csv.

1) OpenCV:-

OpenCV was started at Intel in 1999 by Gary Bradsky, and the first release came out in 2000. Vadim Pisarevsky joined Gary Bradsky to manage Intel's Russian software OpenCV team. In 2005, OpenCV was used on Stanley, the vehicle that won the 2005 DARPA Grand Challenge. Later, its active development continued under the support of Willow Garage with Gary Bradsky and Vadim Pisarevsky leading the project. OpenCV now supports a multitude of algorithms related to Computer Vision and Machine Learning and is expanding day by day.

OpenCV supports a wide variety of programming languages such as C++, Python, Java, etc., and is available on different platforms including Windows, Linux, OS X, Android, and iOS. Interfaces for high-speed GPU operations based on CUDA and OpenCL are also under active development. OpenCV-Python is the Python API for OpenCV, combining the best qualities of the OpenCV C++ API and the Python language.

2) OpenCV-Python:-

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. Python is a general-purpose programming language started by Guido van Rossum that became very popular very quickly, mainly because of its simplicity and code readability.

It enables the programmer to express ideas in fewer lines of code without reducing readability. Compared to languages like C/C++, Python is slower. That said, Python can be easily extended with C/C++, which allows us to write computationally intensive code in C/C++ and create Python wrappers that can be used as Python modules.

This gives us two advantages: first, the code is as fast as the original C/C++ code (since it is the actual C++ code working in background) and second, it easier to code in Python than C/C++. OpenCV-Python is a Python wrapper for the original OpenCV C++ implementation. OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.

Fig:2 OpenCV-python

3) Face_Recognition:-

Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library. Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.

This also provides a simple face_recognition command line tool that lets you do face recognition on a folder of images from the command line. When you install face_recognition, you get a simple command-line program called face_recognition that you can use to recognize faces in a photograph or folder full for photographs. First, you need to provide a folder with one picture of each person you already know. There should be one image file for each person with the files named according to the person who is in the picture. Face recognition can be done in parallel if you have a computer with multiple CPU cores. For example, if your system has 4 CPU cores, you can process about 4 times as many images in the same amount of time by using all your CPU cores in parallel.

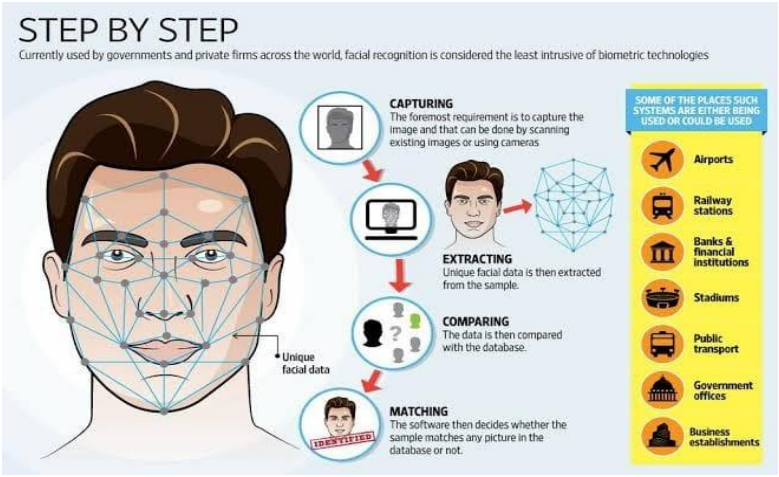

Fig:3 Face_recognition process

4) Dlib:-

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge.

Unlike a lot of open-source projects, this one provides complete and precise documentation for every class and function. There are also debugging modes that check the documented preconditions for functions. When this is enabled it will catch the vast majority of bugs caused by calling functions incorrectly or using objects in an incorrect manner. Good unit test coverage. The ratio of unit test lines of code to library lines of code is about 1 to 4.The library is tested regularly on MS Windows, Linux, and Mac OS X systems. However, it should work on any POSIX system and has been used on Solaris, HPUX, and the BSDs. No other packages are required to use the library. Only APIs that are provided by an out of the box OS are needed. There is no installation or configure step needed before you can use the library. All operating system specific code is isolated inside the OS abstraction layers which are kept as small as possible. The rest of the library is either layered on top of the OS abstraction layers or is pure ISO standard C++.

Fig:4 Dlib

Image Processing with pillow module:-

1.Routines for reading and writing common image formats.

2.Automatic colour space conversion between various pixel types.

3.Common image operations such as edge finding and morphological operations.

4.Implementations of the SURF, HOG, and FHOG feature extraction algorithms.

5.Tools for detecting objects in images including frontal face detection and object pose estimation.

6.High quality face recognition.

Python Pillow - Overview

In today's digital world, we come across lots of digital images. In case, we are working with Python programming language, it provides lot of image processing libraries to add image processing capabilities to digital images.

Some of the most common image processing libraries are: OpenCV, Python Imaging Library (PIL), Scikit-image, Pillow. However, in this tutorial, we are only focusing on Pillow module and will try to explore various capabilities of this module.

Pillow is built on top of PIL (Python Image Library). PIL is one of the important modules for image processing in Python. However, the PIL module is not supported since 2011 and doesn't support python 3.

Pillow module gives more functionalities, runs on all major operating system and support for python 3. It supports wide variety of images such as "jpeg", "png", "bmp", "gif", "ppm", "tiff". You can do almost anything on digital images using pillow module. Apart from basic image processing functionality, including point operations, filtering images using built-in convolution kernels, and color space conversions.

Image Archives

The Python Imaging Library is best suited for image archival and batch processing applications. Python pillow package can be used for creating thumbnails, converting from one format to another and print images, etc.

Image Display

You can display images using Tk PhotoImage, BitmapImage and Windows DIB interface, which can be used with PythonWin and other Windows-based toolkits and many other Graphical User Interface (GUI) toolkits.

For debugging purposes, there is a show () method to save the image to disk which calls the external display utility.

Image Processing

The Pillow library contains all the basic image processing functionality. You can do image resizing, rotation and transformation.

Pillow module allows you to pull some statistics data out of image using histogram method, which later can be used for statistical analysis and automatic contrast enhancement.

Fig:5 Pillow

Threading:-

1.The library provides a portable and simple threading API.

2.A message passing pipe for inter-thread and inter-process communication.

3.A timer object capable of generating events that are regularly spaced in time.

4.Threaded objects.

5.Threaded functions.

6.Parallel for loops.

7.A thread pool with support for futures.

Networking:-

1.The library provides a portable and simple TCP sockets API.

2.An object to help you make TCP based servers.

3.iostream and streambuf objects that enables TCP sockets to interoperate with the C++ iostreams library.

4.A simple HTTP server object you can use to embed a web server into your applications.

5.A message passing pipe for inter-thread and inter-process communication.

6.A tool used to implement algorithms using the Bulk Synchronous Parallel (BSP) computing model.

Graphical User Interfaces:-

1.The library provides a portable and simple core GUI API.

2.Implemented on top of the core GUI API are numerous widgets.

3.Unlike many other GUI toolkits, the entire dlib GUI toolkit is thread safe.

5) CMake:-

CMake is used to control the software compilation process using simple platform and compiler independent configuration files and generate native make files and workspaces that can be used in the compiler environment of your choice.

The suite of CMake tools were created by Kitware in response to the need for a powerful, cross-platform build environment for open-source projects such as ITK and VTK. The CMake python wheels provide CMake 3.18.2. After installing the package using pip, the executables cmake, cpack and ctest will be available in the PATH and can be used to configure and build any projects cmake-python-distributions was initially developed in September 2016 by Jean-Christophe Fillion-Robin to facilitate the distribution of project using scikit-build and depending on CMake.

Fig:6 CMake

6) CSV:-

First of all, what is a CSV?

CSV (Comma Separated Values) is a simple file format used to store tabular data, such as a spreadsheet or database. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. For working CSV files in python, there is an inbuilt module called csv.

In csv modules, an optional dialect parameter can be given which is used to define a set of parameters specific to a particular CSV format. By default, csv module uses excel dialect which makes them compatible with excel spreadsheets.

The so-called CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases. CSV format was used for many years prior to attempts to describe the format in a standardized way in RFC 4180. The lack of a well-defined standard means that subtle differences often exist in the data produced and consumed by different applications. These differences can make it annoying to process CSV files from multiple sources. Still, while the delimiters and quoting characters vary, the overall format is similar enough that it is possible to write a single module which can efficiently manipulate such data, hiding the details of reading and writing the data from the programmer.

The csv module implements classes to read and write tabular data in CSV format. It allows programmers to say, “write this data in the format preferred by Excel,” or “read data from this file which was generated by Excel,” without knowing the precise details of the CSV format used by Excel. Programmers can also describe the CSV formats understood by other applications or define their own special-purpose CSV formats.

The csv module’s reader and writer objects read and write sequences. Programmers can also read and write data in dictionary form using the DictReader and DictWriter classes.

Fig:7 CSV

7) Date time:-

The datetime module supplies classes for manipulating dates and times. While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation.

Date and time objects may be categorized as “aware” or “naive” depending on whether or not they include timezone information. With sufficient knowledge of applicable algorithmic and political time adjustments, such as time zone and daylight saving time information, an aware object can locate itself relative to other aware objects. An aware object represents a specific moment in time that is not open to interpretation. A naive object does not contain enough information to unambiguously locate itself relative to other date/time objects. Whether a naive object represents Coordinated Universal Time (UTC), local time, or time in some other timezone is purely up to the program, just like it is up to the program whether a particular number represents metres, miles, or mass. Naive objects are easy to understand and to work with, at the cost of ignoring some aspects of reality.

Available Types

➔ class datetime.date

An idealized naive date, assuming the current Gregorian calendar always was, and always will be, in effect. Attributes: year, month, and day.

➔ class datetime.time

An idealized time, independent of any particular day, assuming that every day has exactly 24*60*60 seconds. (There is no notion of "leap seconds" here.) Attributes: hour, minute, second, microsecond, and tzinfo.

➔ class datetime.datetime

A combination of a date and a time. Attributes: year, month, day, hour, minute, second, microsecond, and tzinfo.

➔ class datetime.timedelta

A duration expressing the difference between two date, time, or datetime instances to microsecond resolution.

➔ class datetime.tzinfo

An abstract base class for time zone information objects. These are used by the datetime and time classes to provide a customizable notion of time adjustment (for example, to account for time zone and/or daylight-saving time).

➔ class datetime.timezone

A class that implements the tzinfo abstract base class as a fixed offset from the UTC.

8) PYQT5:-

PyQt is one of the most popular Python bindings for the Qt cross-platform C++ framework. PyQt was developed by Riverbank Computing Limited. Qt itself is developed as part of the Qt Project. PyQt provides bindings for Qt 4 and Qt 5. PyQt is distributed under a choice of licences: GPL version 3 or a commercial license. PyQt is available in two editions: PyQt4 which will build against Qt 4.x and 5.x and PyQt5 which will only build against 5.x. Both editions can be built for Python 2 and 3. PyQt contains over 620 classes that cover graphical user interfaces, XML handling, network communication, SQL databases, Web browsing and other technologies available in Qt.The latest iteration of PyQt is v5.11.3. It fully supports Qt 5.11.2.

PyQt4 runs on Windows, Linux, Mac OS X and various UNIX platforms. PyQt5 also runs on Android and iOS. PyQt5 comprises a number of different components. First of all, there are a number of Python extension modules. These are all installed in the PyQt5 Python package and are described in the list of modules. PyQt5 is distributed as a number of source packages and corresponding binary wheels each of which implement one or more logically related extension modules.
PyQt5 also contains a number of utility programs. When PyQt5 is configured a file called PyQt5.api is generated. This can be used by the QScintilla editor component to enable the use of auto-completion and call tips when editing PyQt5 code. The API file is installed automatically if QScintilla is already installed. PyQt5 includes a large number of examples. These are ports to Python of many of the C++ examples provided with Qt. They can be found in the example's directory. Finally, PyQt5 contains the .sip files used by SIP to generate PyQt5 itself. These can be used by developers of bindings of other Qt based class libraries.

Fig:8 PyQt

QT-Designer:-

You will need Python 3 or above, because the others are out dated**.** Qt Designer is a tool for quickly building graphical user interfaces with widgets from the Qt GUI framework. It gives you a simple drag-and-drop interface for laying out components such as buttons, text fields, combo boxes and more. Qt Designer normally ships as a part of Qt Creator. This is Qt's official editor and lets you do a lot more than just graphically design user interfaces. It is a full-fledged and very powerful C++ IDE.

Many people like to use Qt Designer together with Python because it is a dynamic language that lends itself well to rapid prototyping. The easiest way to combine Qt Designer and Python is via the PyQt binding.

To install PyQt, simply enter the following on the command line:

➔ **python3 -m venv venv**
➔ **source venv/bin/activate # or "call venv\Scripts\activate.bat" on Windows**
➔ **python3 -m pip install PyQt5**

Installing QtDesigner

QtDesigner comes with the pip package pyqt5-tools so if you installed it in the previous tutorials, you're good to go.

Otherwise run the following command in your command prompt.

➔ pip install pyqt5-tools

and if that doesn't work you can try:

➔ pip3 install pyqt5-tools.

Flow Chart

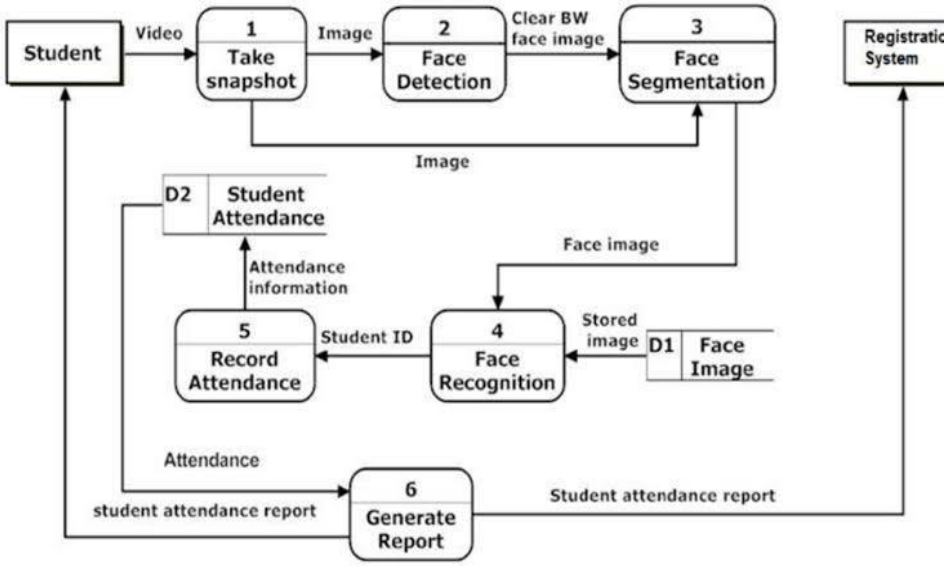

Fig:9 Flow chart

3.2 Code Explanation:-

1.Generating the mainwindow.ui file

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
 <class>Dialog</class>
 <widget class="QDialog" name="Dialog">
  <property name="geometry">
   <rect>
    <x>0</x>
    <y>0</y>
    <width>508</width>
    <height>259</height>
   </rect>
  </property>
  <property name="minimumSize">
   <size>
    <width>508</width>
    <height>259</height>
```

```
 </size>
</property>
<property name="maximumSize">
 <size>
  <width>508</width>
  <height>259</height>
 </size>
</property>
<property name="windowTitle">
 <string>Face Recognition Time Attendance App</string>
</property>
<widget class="QLabel" name="logolabel">
 <property name="geometry">
  <rect>
   <x>0</x>
   <y>0</y>
   <width>500</width>
   <height>133</height>
  </rect>
 </property>
 <property name="text">
  <string/>
 </property>
</widget>
<widget class="QWidget" name="verticalLayoutWidget">
 <property name="geometry">
  <rect>
   <x>10</x>
   <y>160</y>
   <width>491</width>
```

```
 <height>80</height>
</rect>
</property>
<layout class="QVBoxLayout" name="verticalLayout">
 <item>
  <layout class="QHBoxLayout" name="horizontalLayout">
   <item alignment="Qt::AlignHCenter">
    <widget class="QLabel" name="filelabel">
     <property name="font">
      <font>
       <family>Roboto Condensed</family>
       <pointsize>14</pointsize>
       <weight>75</weight>
       <bold>true</bold>
      </font>
     </property>
     <property name="text">
      <string>Face Recognition Time Attendance App</string>
     </property>
    </widget>
   </item>
  </layout>
 </item>
 <item>
  <widget class="QPushButton" name="runButton">
   <property name="font">
    <font>
     <family>Roboto Condensed Light</family>
     <pointsize>13</pointsize>
    </font>
```

```
      </property>
      <property name="text">
       <string>Start</string>
      </property>
     </widget>
    </item>
   </layout>
  </widget>
 </widget>
 <resources>
  <include location="resource.qrc"/>
 </resources>
 <connections/>
</ui>
```

2.Generating the Outputwindow.ui file

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
 <class>OutputDialog</class>
 <widget class="QDialog" name="OutputDialog">
  <property name="geometry">
   <rect>
    <x>0</x>
    <y>0</y>
    <width>951</width>
    <height>591</height>
   </rect>
  </property>
  <property name="minimumSize">
   <size>
```

```
  <width>0</width>
  <height>0</height>
 </size>
</property>
<property name="maximumSize">
 <size>
  <width>1280</width>
  <height>720</height>
 </size>
</property>
<property name="font">
 <font>
  <family>Calibri</family>
  <weight>75</weight>
  <bold>true</bold>
 </font>
</property>
<property name="windowTitle">
 <string>Face Recognition Attendance App</string>
</property>
<widget class="QLabel" name="imgLabel">
 <property name="geometry">
  <rect>
   <x>10</x>
   <y>10</y>
   <width>640</width>
   <height>480</height>
  </rect>
 </property>
 <property name="minimumSize">
```

```
 <size>
  <width>4</width>
  <height>0</height>
 </size>
</property>
<property name="maximumSize">
 <size>
  <width>640</width>
  <height>480</height>
 </size>
</property>
<property name="text">
 <string/>
</property>
</widget>
<widget class="QWidget" name="horizontalLayoutWidget">
 <property name="geometry">
  <rect>
   <x>10</x>
   <y>500</y>
   <width>641</width>
   <height>80</height>
  </rect>
 </property>
 <layout class="QHBoxLayout" name="horizontalLayout">
  <item>
   <widget class="QPushButton" name="ClockInButton">
    <property name="font">
     <font>
      <family>Roboto Condensed Light</family>
```

```
   <pointsize>10</pointsize>
   <weight>75</weight>
   <bold>true</bold>
  </font>
 </property>
 <property name="text">
  <string>Clock In</string>
 </property>
 <property name="checkable">
  <bool>true</bool>
 </property>
</widget>
</item>
<item>
 <widget class="QPushButton" name="ClockOutButton">
  <property name="font">
   <font>
    <family>Roboto Condensed Light</family>
    <pointsize>10</pointsize>
    <weight>75</weight>
    <bold>true</bold>
   </font>
  </property>
  <property name="text">
   <string>Clock Out</string>
  </property>
  <property name="checkable">
   <bool>true</bool>
  </property>
 </widget>
```

```
 </item>
 </layout>
</widget>
<widget class="QWidget" name="gridLayoutWidget">
 <property name="geometry">
  <rect>
   <x>660</x>
   <y>10</y>
   <width>281</width>
   <height>75</height>
  </rect>
 </property>
 <layout class="QGridLayout" name="gridLayout_2">
  <item row="0" column="0">
   <widget class="QLabel" name="label">
    <property name="font">
     <font>
      <family>Roboto Condensed Light</family>
      <pointsize>16</pointsize>
      <weight>75</weight>
      <bold>true</bold>
     </font>
    </property>
    <property name="text">
     <string>Date :</string>
    </property>
   </widget>
  </item>
  <item row="0" column="1">
   <widget class="QLabel" name="Date_Label">
```

```
  <property name="font">
   <font>
    <family>Roboto Condensed Light</family>
    <pointsize>15</pointsize>
    <weight>75</weight>
    <bold>true</bold>
   </font>
  </property>
  <property name="text">
   <string>-</string>
  </property>
 </widget>
</item>
<item row="1" column="0">
 <widget class="QLabel" name="label_2">
  <property name="font">
   <font>
    <family>Roboto Condensed Light</family>
    <pointsize>16</pointsize>
    <weight>75</weight>
    <bold>true</bold>
   </font>
  </property>
  <property name="text">
   <string>Time :</string>
  </property>
 </widget>
</item>
<item row="1" column="1">
 <widget class="QLabel" name="Time_Label">
```

```
   <property name="font">
    <font>
     <family>Roboto Condensed Light</family>
     <pointsize>15</pointsize>
     <weight>75</weight>
     <bold>true</bold>
    </font>
   </property>
   <property name="text">
    <string>-</string>
   </property>
  </widget>
 </item>
</layout>
</widget>
<widget class="QGroupBox" name="groupBox">
 <property name="geometry">
  <rect>
   <x>660</x>
   <y>110</y>
   <width>271</width>
   <height>271</height>
  </rect>
 </property>
 <property name="title">
  <string>Details</string>
 </property>
 <widget class="QWidget" name="verticalLayoutWidget">
  <property name="geometry">
   <rect>
```

```
 <x>30</x>
 <y>30</y>
 <width>141</width>
 <height>211</height>
</rect>
</property>
<layout class="QVBoxLayout" name="verticalLayout">
 <item>
  <widget class="QLabel" name="label_3">
   <property name="font">
    <font>
     <family>Roboto Condensed Light</family>
     <pointsize>12</pointsize>
     <weight>50</weight>
     <bold>false</bold>
    </font>
   </property>
   <property name="text">
    <string>Name : </string>
   </property>
  </widget>
 </item>
 <item>
  <widget class="QLabel" name="label_4">
   <property name="font">
    <font>
     <family>Roboto Condensed Light</family>
     <pointsize>12</pointsize>
     <weight>50</weight>
     <bold>false</bold>
```

```
    </font>
   </property>
   <property name="text">
    <string>Status :</string>
   </property>
  </widget>
 </item>
 <item>
  <widget class="QLabel" name="label_5">
   <property name="font">
    <font>
     <family>Roboto Condensed Light</family>
     <pointsize>12</pointsize>
     <weight>50</weight>
     <bold>false</bold>
    </font>
   </property>
   <property name="text">
    <string>Clocked Time : </string>
   </property>
  </widget>
 </item>
</layout>
</widget>
<widget class="QWidget" name="verticalLayoutWidget_2">
 <property name="geometry">
  <rect>
   <x>120</x>
   <y>30</y>
   <width>131</width>
```

```
  <height>211</height>
 </rect>
</property>
<layout class="QVBoxLayout" name="verticalLayout_2">
 <item>
  <widget class="QLabel" name="NameLabel">
   <property name="font">
    <font>
     <family>Roboto Condensed Light</family>
     <pointsize>12</pointsize>
     <weight>50</weight>
     <bold>false</bold>
    </font>
   </property>
   <property name="text">
    <string/>
   </property>
  </widget>
 </item>
 <item>
  <widget class="QLabel" name="StatusLabel">
   <property name="font">
    <font>
     <family>Roboto Condensed Light</family>
     <pointsize>12</pointsize>
     <weight>50</weight>
     <bold>false</bold>
    </font>
   </property>
   <property name="text">
```

```
   <string/>
  </property>
 </widget>
</item>
<item>
 <layout class="QHBoxLayout" name="horizontalLayout_2">
  <item>
   <widget class="QLabel" name="HoursLabel">
    <property name="font">
     <font>
      <family>Roboto Condensed Light</family>
      <pointsize>12</pointsize>
      <weight>50</weight>
      <bold>false</bold>
     </font>
    </property>
    <property name="text">
     <string/>
    </property>
   </widget>
  </item>
  <item>
   <widget class="QLabel" name="MinLabel">
    <property name="font">
     <font>
      <family>Roboto Condensed Light</family>
      <pointsize>12</pointsize>
      <weight>50</weight>
      <bold>false</bold>
     </font>
```

```
        </property>
        <property name="text">
         <string/>
        </property>
       </widget>
      </item>
     </layout>
    </item>
   </layout>
  </widget>
 </widget>
</widget>
<resources>
 <include location="resource.qrc"/>
</resources>
<connections/>
</ui>
```

3.Resource file for App logo and Icon

```
<RCC>
  <qresource prefix="logo">
    <file>logo.png</file>
  </qresource>
  <qresource prefix="icon">
    <file>icon.png</file>
  </qresource>
</RCC>
```

4.Output Window.py

a. Importing necessary modules required for the project:

```
from PyQt5.QtGui import QImage, QPixmap
from PyQt5.uic import loadUi
from PyQt5.QtCore import pyqtSlot, QTimer, QDate, Qt
from PyQt5.QtWidgets import QDialog, QMessageBox
import cv2
import face_recognition
import numpy as np
import datetime
import os
import csv
```

b. Loading outputwindow.ui and updating time:

```
class Ui_OutputDialog(QDialog):
    def __init__(self):
        super(Ui_OutputDialog, self).__init__()
        loadUi("./outputwindow.ui", self)

        #Update time and date
        now = QDate.currentDate()
        current_date = now.toString('ddd dd MMMM yyyy')
        current_time = datetime.datetime.now().strftime("%I:%M %p")
        self.Date_Label.setText(current_date)
        self.Time_Label.setText(current_time)
        self.image = None
```

c. Encoding known faces from the folder ImagesAttendance:

```
@pyqtSlot()
    def startVideo(self, camera_name):
        """
        :param camera_name: link of camera or usb camera
```

```
        :return:
        """
        if len(camera_name) == 1:
            self.capture = cv2.VideoCapture(int(camera_name))
        else:
            self.capture = cv2.VideoCapture(camera_name)
        self.timer = QTimer(self)  # Create Timer
        path = 'ImagesAttendance'
        if not os.path.exists(path):
            os.mkdir(path)
        # known face encoding and known face name list
        images = []
        self.class_names = []
        self.encode_list = []
        self.TimeList1 = []
        self.TimeList2 = []
        attendance_list = os.listdir(path)
```

d. Mark the encoded faces only with the file name but not with the file extension:

```
for cl in attendance_list:
            cur_img = cv2.imread(f'{path}/{cl}')
            images.append(cur_img)
            self.class_names.append(os.path.splitext(cl)[0])
        for img in images:
            img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
            boxes = face_recognition.face_locations(img)
            encodes_cur_frame = face_recognition.face_encodings(img, boxes)[0]
            # encode = face_recognition.face_encodings(img)[0]
            self.encode_list.append(encodes_cur_frame)
        self.timer.timeout.connect(self.update_frame)  # Connect timeout to the output function
```

```
        self.timer.start(10)  # emit the timeout() signal at x=40ms


    def face_rec_(self, frame, encode_list_known, class_names):
```

e. Marking the Identified face in the Attendance.csv file with clocked in and clocked out time:

```
  def mark_attendance(name):
            """
            :param name: detected face known or unknown one
            :return:
            """
            if self.ClockInButton.isChecked():
               self.ClockInButton.setEnabled(False)
               with open('Attendance.csv', 'a') as f:
                     if (name != 'unknown'):
                        buttonReply = QMessageBox.question(self, 'Welcome ' + name, 'Are you Clocking In?' ,
                                                           QMessageBox.Yes | QMessageBox.No, QMessageBox.No)
                        if buttonReply == QMessageBox.Yes:

                           date_time_string = datetime.datetime.now().strftime("%y/%m/%d %H:%M:%S")
                           f.writelines(f'\n{name},{date_time_string},Clock In')
                           self.ClockInButton.setChecked(False)

                           self.NameLabel.setText(name)
                           self.StatusLabel.setText('Clocked In')
                           self.HoursLabel.setText('Measuring')
                           self.MinLabel.setText('')

                           #self.CalculateElapse(name)
```

```
                    #print('Yes clicked and detected')
                    self.Time1 = datetime.datetime.now()
                    #print(self.Time1)
                    self.ClockInButton.setEnabled(True)
                else:
                    print('Not clicked.')
                    self.ClockInButton.setEnabled(True)
        elif self.ClockOutButton.isChecked():
            self.ClockOutButton.setEnabled(False)
            with open('Attendance.csv', 'a') as f:
                if (name != 'unknown'):
                    buttonReply = QMessageBox.question(self, 'Cheers ' + name, 'Are you Clocking Out?',
                                                       QMessageBox.Yes | QMessageBox.No, QMessageBox.No)
                    if buttonReply == QMessageBox.Yes:
                        date_time_string = datetime.datetime.now().strftime("%y/%m/%d %H:%M:%S")
                        f.writelines(f'\n{name},{date_time_string},Clock Out')
                        self.ClockOutButton.setChecked(False)

                        self.NameLabel.setText(name)
                        self.StatusLabel.setText('Clocked Out')
                        self.Time2 = datetime.datetime.now()
                        #print(self.Time2)

                        self.ElapseList(name)
                        self.TimeList2.append(datetime.datetime.now())
                        CheckInTime = self.TimeList1[-1]
                        CheckOutTime = self.TimeList2[-1]
                        self.ElapseHours = (CheckOutTime - CheckInTime)
```

```
self.MinLabel.setText("{:.0f}".format(abs(self.ElapseHours.total_seconds() / 60)%60) + 'm')

self.HoursLabel.setText("{:.0f}".format(abs(self.ElapseHours.total_seconds() / 60**2)) + 'h')
                    self.ClockOutButton.setEnabled(True)
                else:
                    print('Not clicked.')
                    self.ClockOutButton.setEnabled(True)
```

f. Displaying Identified faces with Black text in a Green frame:

```
faces_cur_frame = face_recognition.face_locations(frame)
    encodes_cur_frame = face_recognition.face_encodings(frame, faces_cur_frame)
    # count = 0
    for encodeFace, faceLoc in zip(encodes_cur_frame, faces_cur_frame):
        match = face_recognition.compare_faces(encode_list_known, encodeFace, tolerance=0.50)
        face_dis = face_recognition.face_distance(encode_list_known, encodeFace)
        name = "unknown"
        best_match_index = np.argmin(face_dis)
        # print("s",best_match_index)
        if match[best_match_index]:
            name = class_names[best_match_index].upper()
            y1, x2, y2, x1 = faceLoc
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.rectangle(frame, (x1, y2 - 20), (x2, y2), (0, 255, 0), cv2.FILLED)
            cv2.putText(frame, name, (x1 + 6, y2 - 6), cv2.FONT_HERSHEY_COMPLEX, 0.5, (255, 255, 255), 1)
        mark_attendance(name)
```

CHAPTER-4

OUTPUTS & RESULTS

Clocked In:-

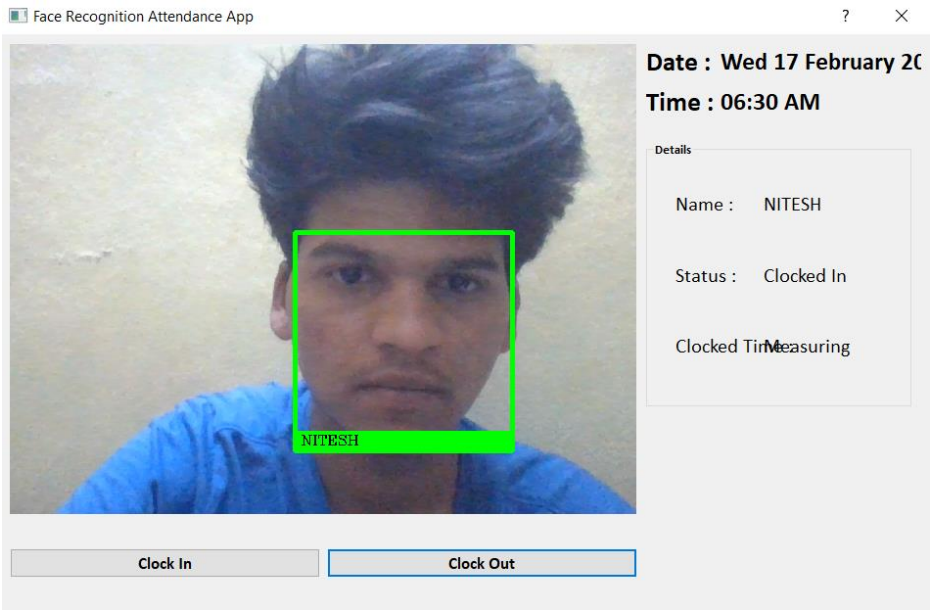

Fig:10 clocked in(Nitesh)

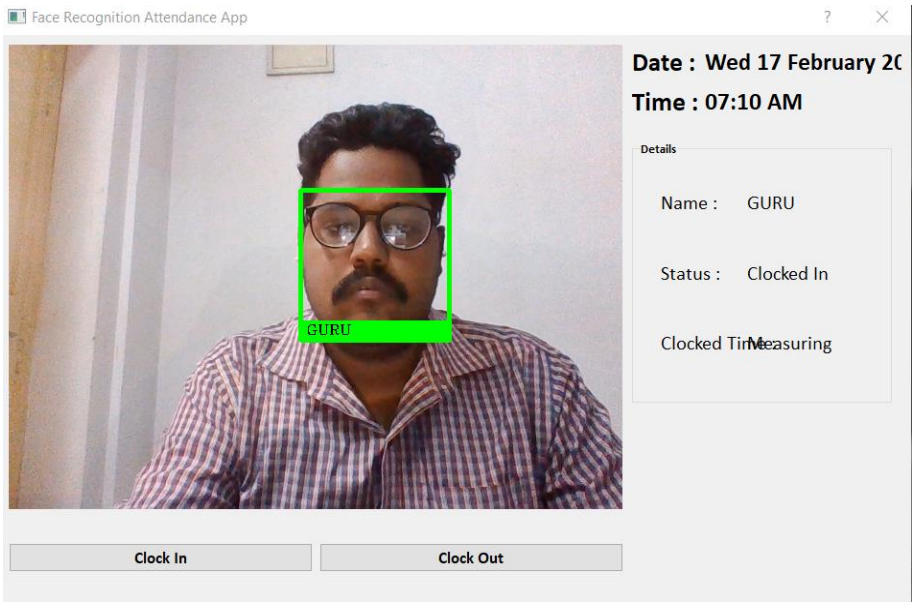

Fig:11 clocked in(Guru)

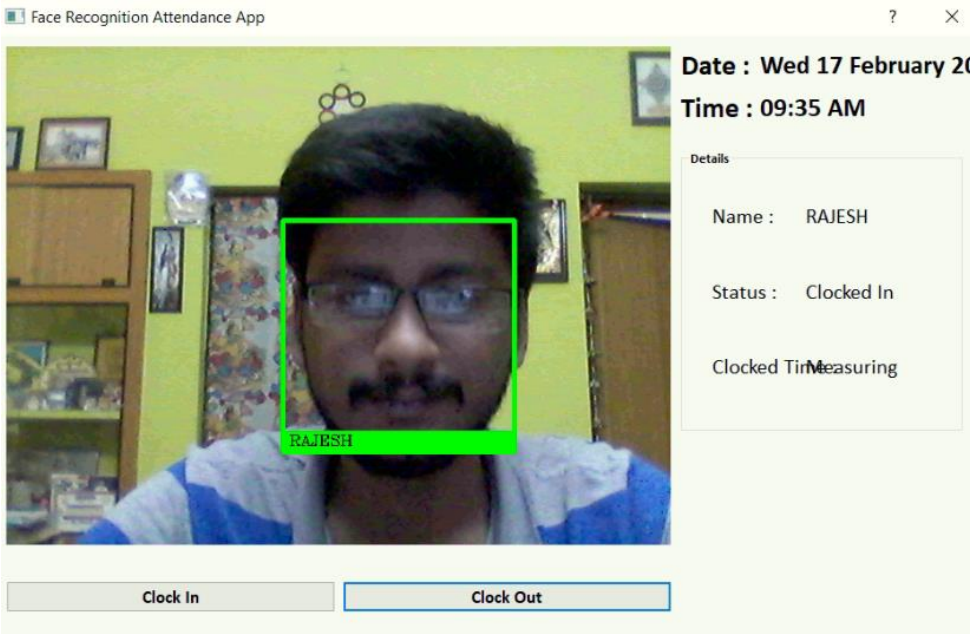

Fig:12 clocked in(Rajesh)

Clocked out & Measured Time Between Clock in & Clock out:

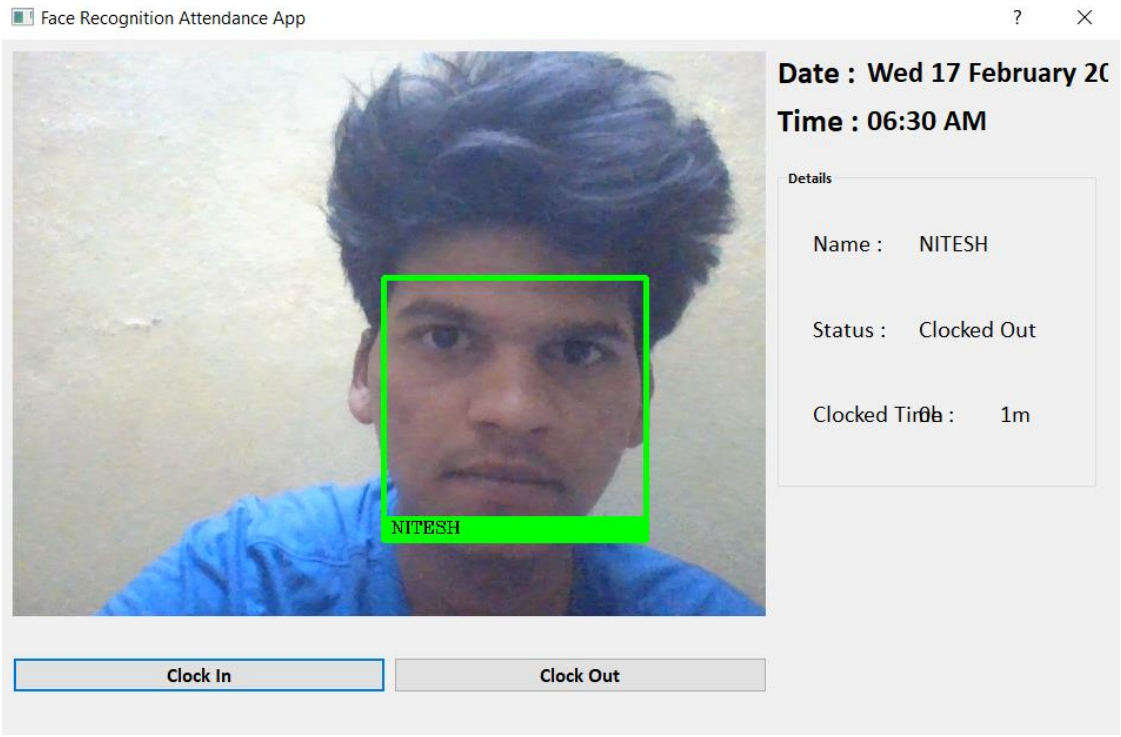

Fig:13 clocked out(Nitesh)

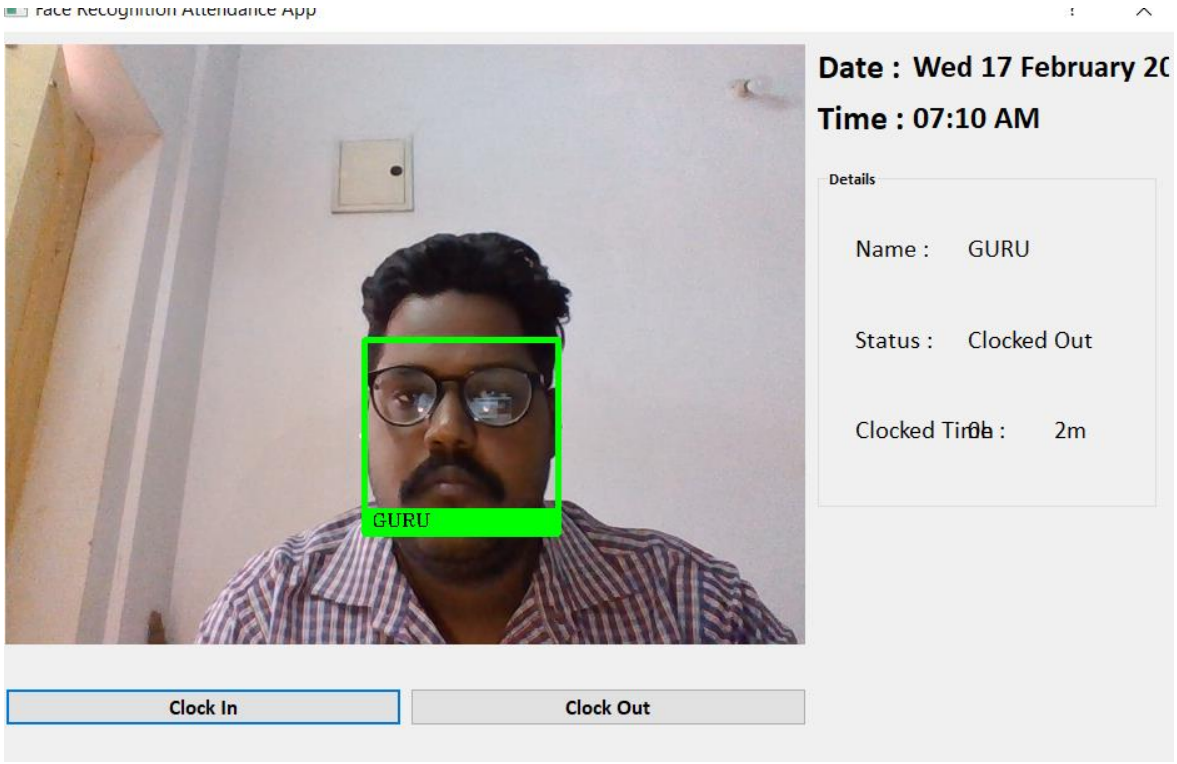

Fig:14 clocked out(Guru)

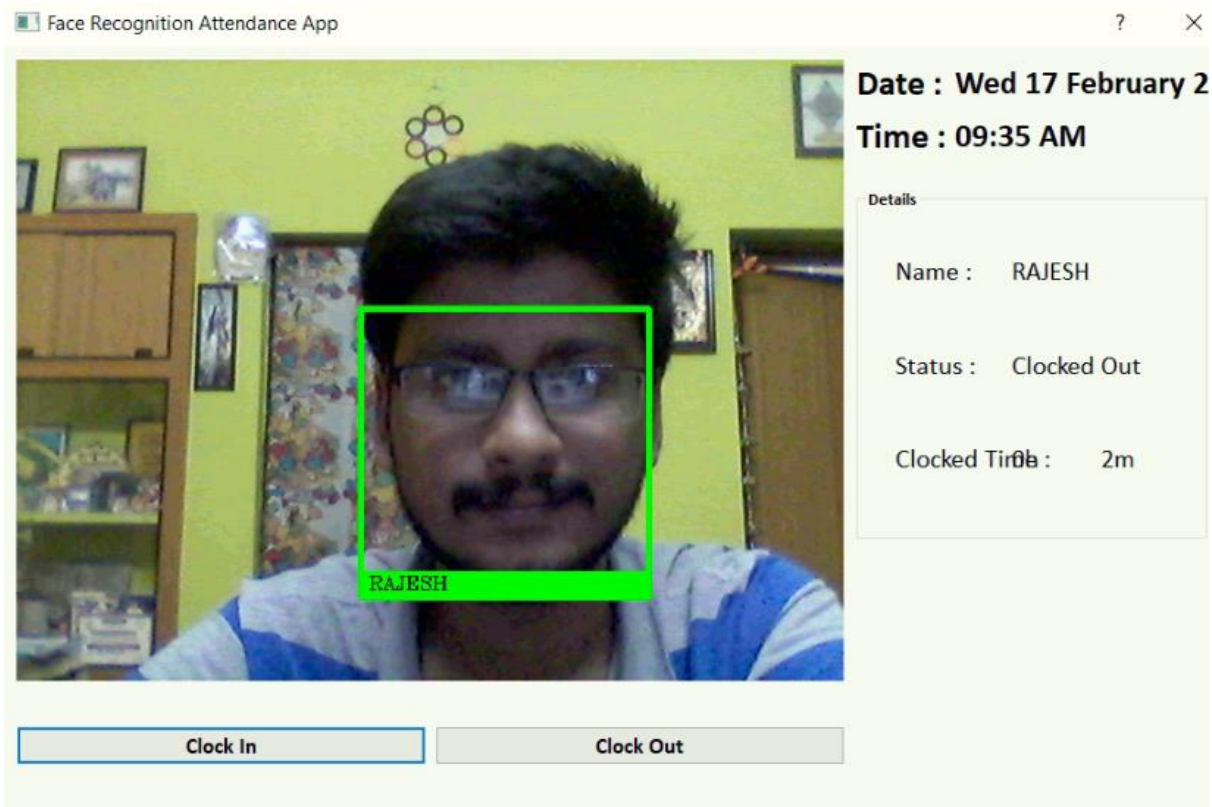

Fig:15 clocked out(Rajesh)

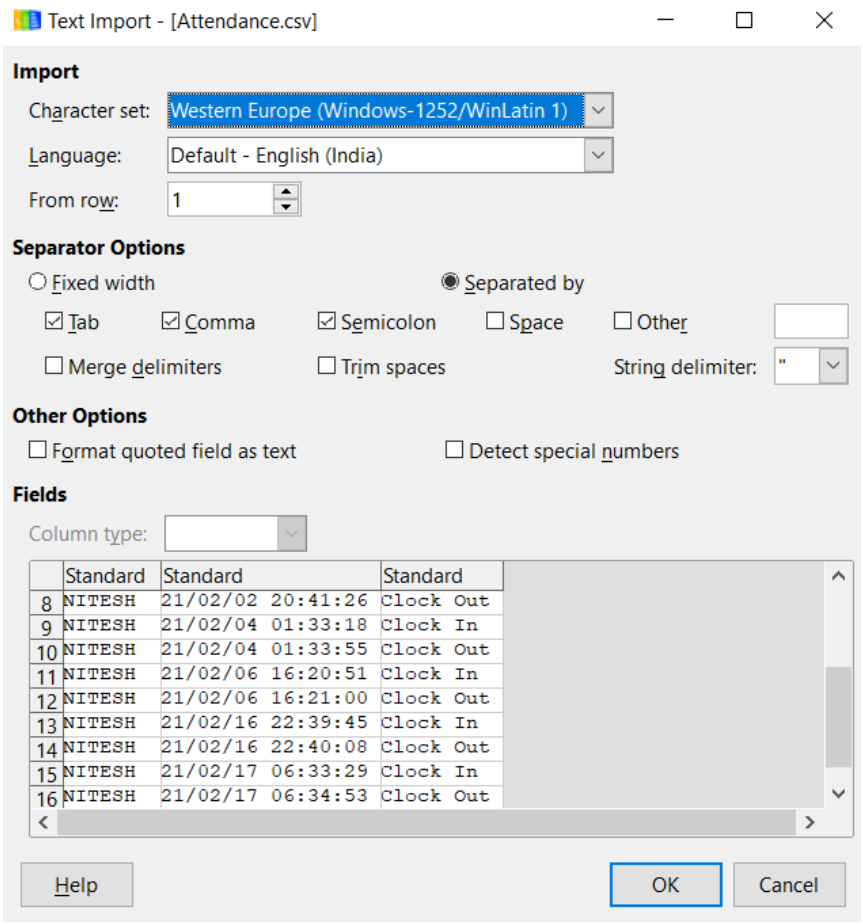

Fig: 16 Attendence.csv

CHAPTER-5

CONCLUSION

In this system we have implemented an attendance system for a lecture, section or laboratory by which lecturer or teaching assistant can record every student's attendance. It saves time and effort, especially if it is a lecture with huge number of students. Auto-mated Attendance System has been envisioned for the purpose of reducing the draw- backs in the traditional (manual) system. This attendance system demonstrates the use of image processing techniques in classroom. This system cannot only merely help in the attendance system, but also improve the goodwill of an institution. The changes will be made in the project based on the requirements of the individual institution.

REFERENCES

[1] M. T. a. A. Pentland, "Eigenfaces For Recognition," Journal of Cognitive Neuroscience, vol. 3, no. 1, 1991.

[2] A. V. a. R. Tokas, "Fast Face Recognition Using Eigen Faces," IJRITCC, vol. 2, no. 11, pp. 3615-3618, November 2014.

[3] Paul Viola and Michael J. Jones, "Robust Real-Time Face Detection," International Journal of Computer Vision, vol. 57, no. 2, pp. 137-154, May 2004.

[4] N. J. M. M. K. a. H. A. Mayank Agarwal, "Face Recognition Using Eigenface aproach," IRCSE, vol. 2, no. 4, pp. 1793-8201, August 2010.

[5] Vinay Hermath, Ashwini Mayakar, "Face Recognition Using Eigen Faces and," IACSIT, vol. 2, no. 4, pp. 1793-8201, August 2010.