MAJOR PROJECT

Auto Attendance using Face Recognition

GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY

UNIVERSITY SCHOOL OF INFORMATION COMMUNICATION AND TECHNOLOGY

Under the supervision of

Mentor:

Prof. Navin Rajpal

Submitted by:

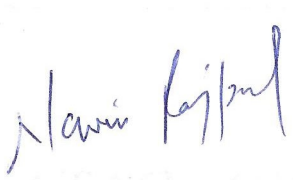
Vikrant Singh Jamwal

B.TECH. – IT 8th Sem

03116401517

CERTIFICATE

This is to certify that the project entitled "**Auto attendance using Face Recognition**" submitted by "**Vikrant Singh Jamwal**" to University School of Information, Communication and Technology, GGSIPU in partial fulfilment of the requirement for the award of the degree of B. Tech in Information Technology. This project work carried out by him under my guidance. The project fulfils the requirement as per the regulation of this institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university to the best of my knowledge and belief.



Date: 02/07/2021

Prof. Navin Rajpal
USICT, GGSIPU
New Delhi

DECLARATION

This is to certify that the work being presented in the project entitled "**Auto attendance using Face Recognition**" submitted by undersigned student of Bachelors in **Information Technology** to University School of Information Communication and Technology in the fulfilment for award of the degree of Bachelors in **Information Technology** is a record of my own work carried out by me under guidance and supervision of **Professor Navin Rajpal**. I certify the content of this work has not submitted elsewhere for award of any other degree.

Date: 30/06/2021

Signature of Student

Vikrant Singh Jamwal

B.Tech.-IT,8th semester

(03116401517)

ACKNOWLEDGMENT

I would like to express my sincere gratitude towards **Prof. Pravin Chandra** , Dean, University School of Information & Technology, GGSIPU for giving me this opportunity of working on this project and **Dr. Sanjay Kumar Malik**, coordinator, who provided a wide vision of scope of the project which was very important to me.

I am also thankful to my mentor, **Prof. Navin Rajpal** for his most valuable and significant guidance throughout the course of the project and in elaborating our view of studying the project's details and getting the right vision for its implementation.

At last, I would like to thank the Department of Information Technology - USICT, GGSIPU, that has played an important role in strengthening my career by providing full cooperation and encouragement throughout the tenure of the project.

Contents

Introduction

Attendance management system is a necessary tool for taking attendance in any environment where attendance plays an important part. However, most of the ways of measuring attendance is time consuming and inefficient. Some institutions use paper based approach and others have adopted automated methods such as fingerprint biometric techniques. This research is aimed at developing a less intrusive, more efficient and cost effective automatic attendance system using face detection and recognition.

As the technology is significantly growing every second, this project aims at updating the old system of recording the attendance to the more efficient way by detection and recognition of the face and creating a complete list of all the people present at the event.

Face recognition technology has also been used for both verifying and identifying the students in a classroom. This research is aimed at developing a less intrusive, cost effective and more efficient automated attendance management system.

Problem Statement

Attendance marking in a classroom during a lecture is not only a difficult job but also time consuming. Due to high number of students present during the lecture there will always be a probability of proxy or missed attendance. Marking the attendance with conventional methods has been a challenge for teachers. The need of saving time and creating an automatic techniques of marking attendance is very important for the future of efficient work.

In recent years, the problem of managing the attendance manually is being tackled by new technology including bio metric checks or fingerprints. However, these techniques lack in reliability. In this project an automatic attendance marking and management system is to be created using face detection and recognition.

Idea behind the project

Main idea behind the project is divided into four parts:

1. Video capture:

   The video of the class can be captured from the platform where the meeting is occurring. This video is then moved to the detection folder inside the software.

   OpenCV provides the VideoCapture() function which is used to work with the Camera or by using a file from storage by specifying the index. (0) is for camera and (1) is for a file path in the storage.

   First we have to import cv2 library to get the window capture function in our program. To get this we have to install OpenCV in our python files.

   We can do the following task:

   -Read video, display video, and save video.

   Read (`ret, frame = vid.read()`)- reads the video present the storage.

   Display (`cv2.imshow()`)-displays the video by opening it in a window.

   Save (`cap.release()`)- releases the video and saves it.

   -Capture from the camera and display it.

   ```
   vid = cv2.VideoCapture(0)
   ```

   This line of code is used to open camera and view the vision in a small window.

2. Face Detection:

First the face is detected using Haar-Cascade and OpenCV by detecting the smile, eyes and nose.

Haar Cascade algorithm for detection is one of the old but efficient face detection algorithms present with us. Haar-Cascade Features were used to detect faces, nose, eyes and also lips.

Basic idea of this algorithm is to get the sum of all the pixels in an image lying on the dark area and the sum of all the pixels of an image lying on the light area of the Haar feature.

Haar Cascade is present in the CV2 folder in the python directory and can be imported just by specifying the absolute path of the file in the cascade function.

-cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
This line of code classifies the cascade in the program.

The Algorithm looks for specific Haar feature of a face. This detection feature takes the image and converts it into a 24X24 window and puts each Haar feature to that window pixel by pixel. First, the algorithm requires a lot of positive images or we can say image of a face and negative images or image excluding the face to train the classifier. Then, these features are being extracted.

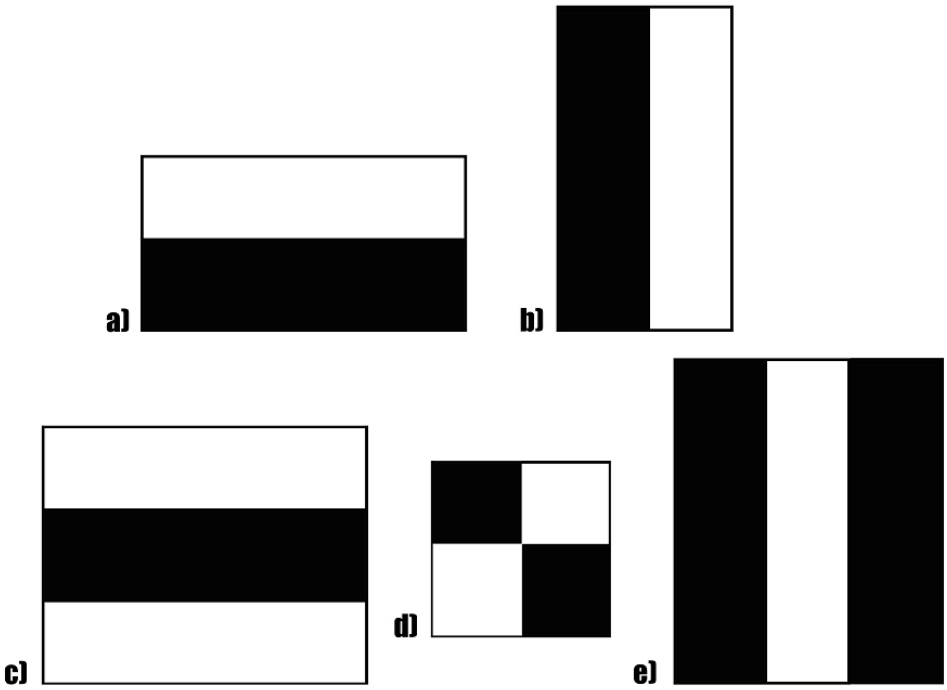


Fig.1 a) and b) are edge features c) and e) are line features and d) is centre surround feature.

Features are numerical values determined from images that are used to differentiate between different images, every feature is single valued by calculating the difference between the sum of white area and sum of black area in the haar cascade triangle.

Feature = Σ (pixels in black part) - Σ (pixels in white part).

This formula is used to find out the face features in numbers and these number are to be stored somewhere to recognise the faces later by comparing the to these numbers. Hence these numbers are saved in a YML document with a specific ID for each face.

3. Face recognition:

After detection the face is then compared to the images present in the database. Multiple frames of the captured videos are used to compare with the images of the students in the database.

Local Binary Pattern (LBP) is an effective texture based operator which give the threshold value of the neighbouring pixels and consider the result as a binary number.

It was found to be a powerful feature for texture classification till date. When LBP(Local binary patern) is combined with Histogram of oriented gradients, it improves the detection considerably on some data.

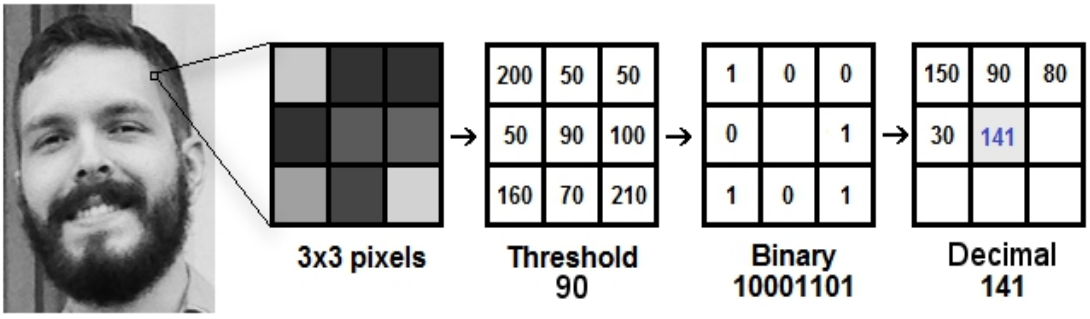


Fig.2 Threshold of neighbouring pixels in LBP

It can easily be available in the OpenCV version 2 and above and can be accessed using a simple line of code.

-cv2.face.LBPHFaceRecognizer_create()

The LBPH uses 4 parameters:

1) Radius: radius is used to build the local binary pattern and represents the radius around the middle pixel. It is mostly set to 1.
2) Neighbours: the number of points to build the local binary pattern and is mostly set to 8.
3) Grid X: It is the number of cells present in the horizontal direction. The more cells, the better the grid, the higher the dimensionality of the resulting feature. It is mostly set to 8.
4) Grid Y: It is the number of cells present in the vertical direction. The more cells, the better the grid, the higher the dimensionality of the resulting feature. It is mostly set to 8.

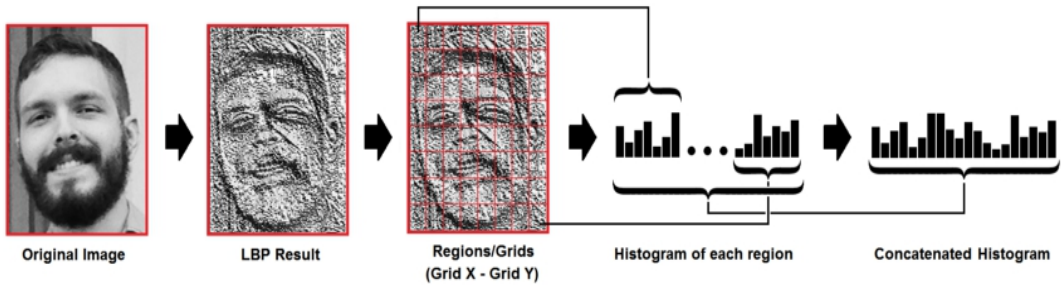


Fig.3 Based on the image above, we can extract the histogram of each region.

As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0~255) representing the occurrences of each pixel intensity.

Then, we need to concatenate each histogram to create a new and bigger histogram. So to find the image which matches with the closest image by comparing their histograms. Output will be ID and Confidence.

4. Data updating:

After recognition of the face, the excel file is accessed using libraries like pandas and csv. The face IDs that were recognised are recorded and marked present in the csv file and then converted back to the excel file.

There are multiple ways of updating the excel file, one of them is by using openpyxl library to create a work station for excel updating.

Table 1 Terms used in openpyxl

Term	Explanation
Spreadsheet or Workbook	A **Spreadsheet** is the main file you are forming or working with.
Worksheet or Sheet	A **Sheet** is used to split different kinds of content within the same spreadsheet. A **Spreadsheet** can have one or more **Sheets**.
Column	A **Column** is a vertical line in the file.
Row	A **Row** is a horizontal line in the File.
Cell	A **Cell** is a combination of **Column** and **Row** together making a particular usnit of the file.

Pandas and can also be used to manipulate the data and save the updates in the file present in the storage. Pandas Data Frame will be formed by opening the data from existing storage can be SQL Database or Excel File.

Flow Chart

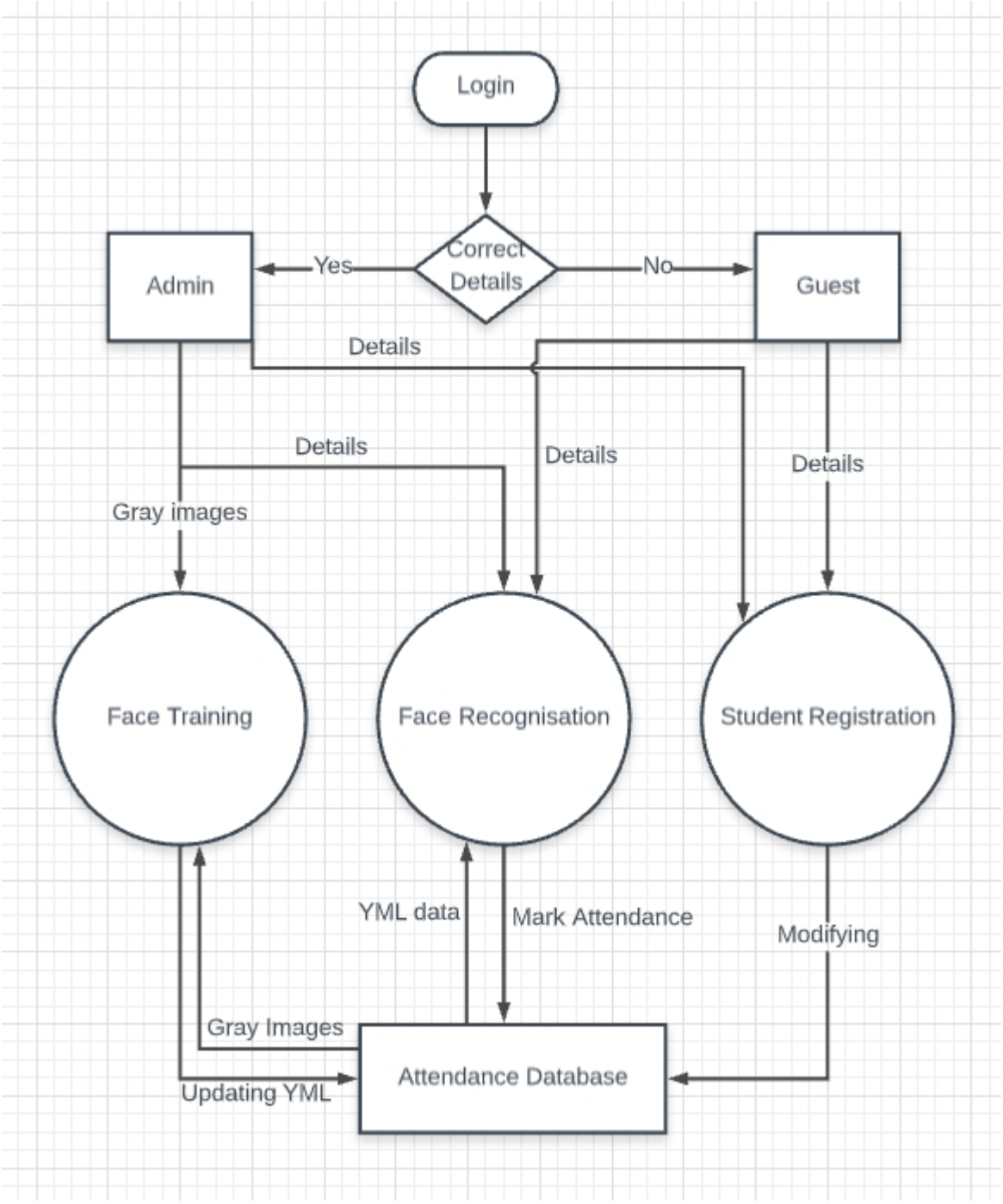


Fig.4 Data Flow Diagram of Auto Attendance System using Face recognition.

Use Case

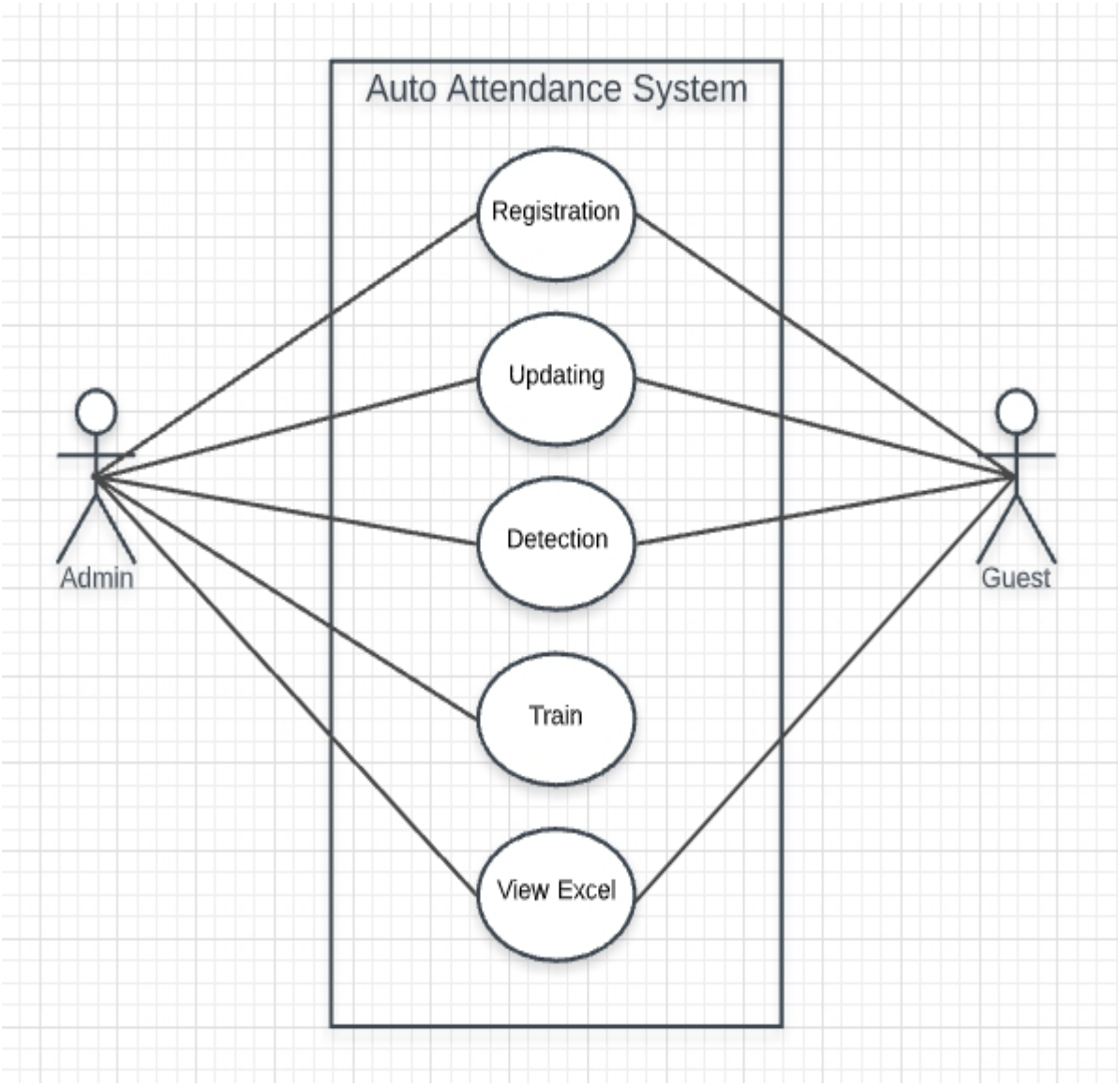


Fig.5 Use case of Auto Attendance System Using Face Recognition

Requirements

Hardware requirement

1. Processor: Intel Atom processor or above
2. Clock speed: 1.2 GHz or faster
3. Storage: 10 GB at least
4. RAM: More than 1GB

Software Requirements

1. Python36
2. Visual Source code
3. Microsoft Excel
4. Windows 7 and above

Libraries Used

1. Tkinter
2. Haar Cascade
3. Pillow
4. Panda
5. NumPy
6. OpenCV

Procedure, Codes and UI

The project is created in Visual Studio Code in Python language. The User Interface of the project is created using Tkinter library of Python and the Excel file is accessed and modified by CSV, Pandas, Numpy and Openpyxl libraries.

Haar-Cascade, OpenCV and LPBH recogniser is used for image and video processing to detect and recognise the faces in the given video sample.

The project is divided into following parts:

- Login screen
    1) Admin
    2) Guest user
- Registration of a student
    1) Register
    2) Update
    3) Delete
- Training the images
- Recognising the face
- Updating the excel
    1) Insert Date
    2) Mark Attendance
- Viewing the excel

Login Screen

The login screen accepts two entries, ID and a Password. There are two types of logging in is provided, one with guest user and the other with admin.

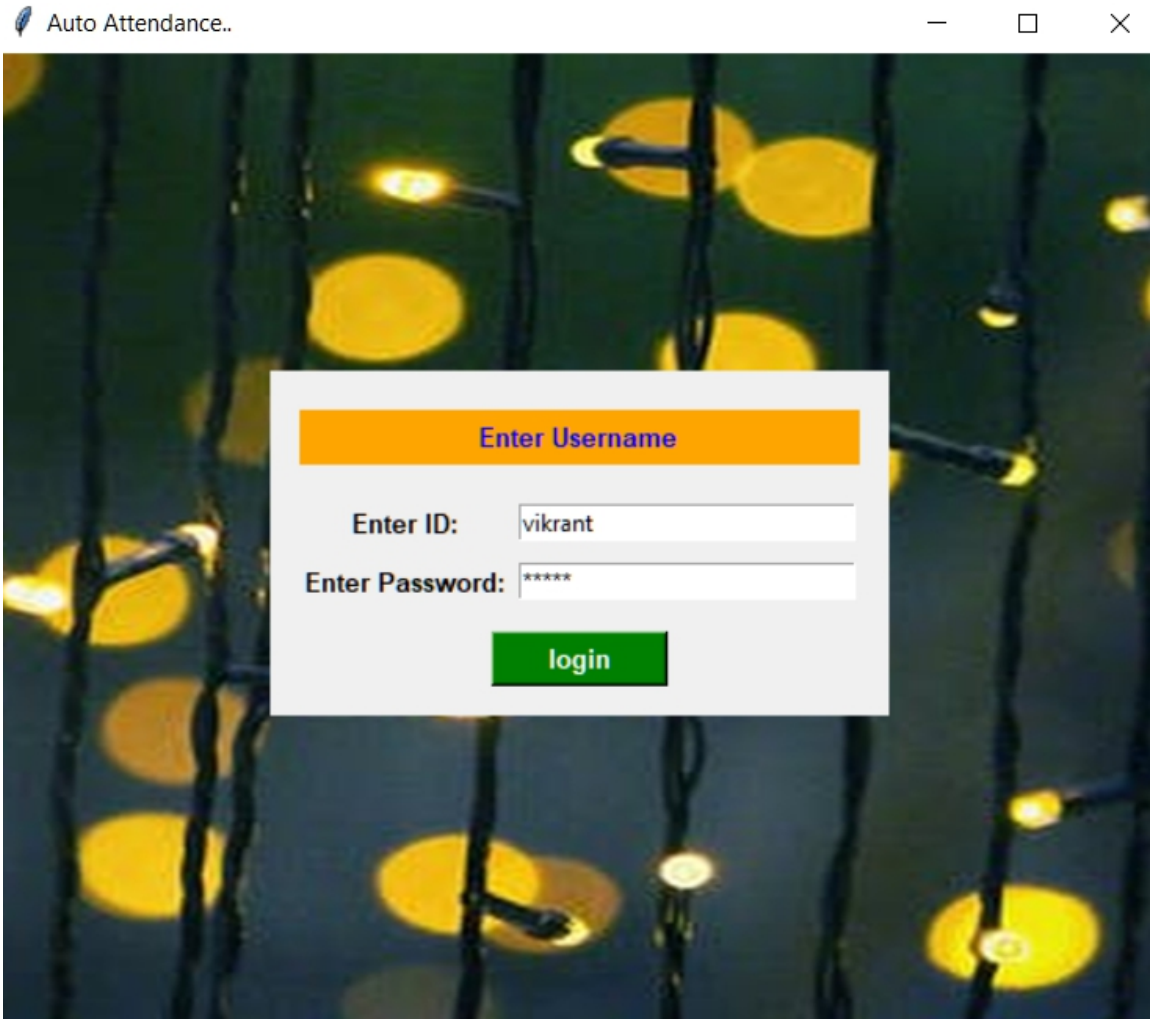
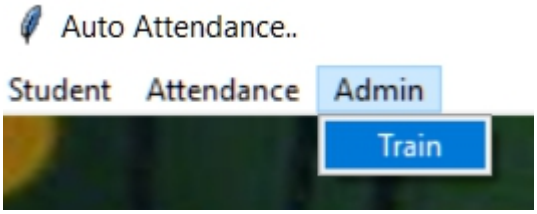


Fig.6 Login Screen

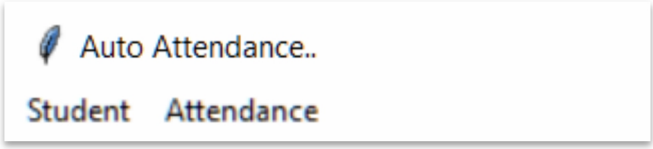
Admin

Admin gets few extra features like Training the images of the students making the program more efficient.



Guest

Guest user doesn't need a user ID or password, they have access to viewing the excel file and detecting the attendance from the video.



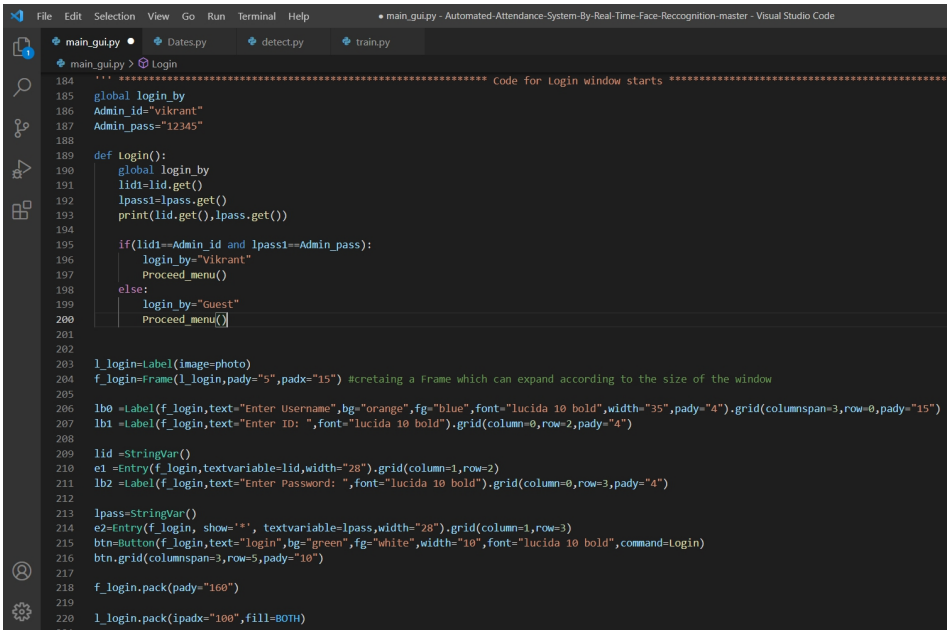


Fig.7 Login Screen code

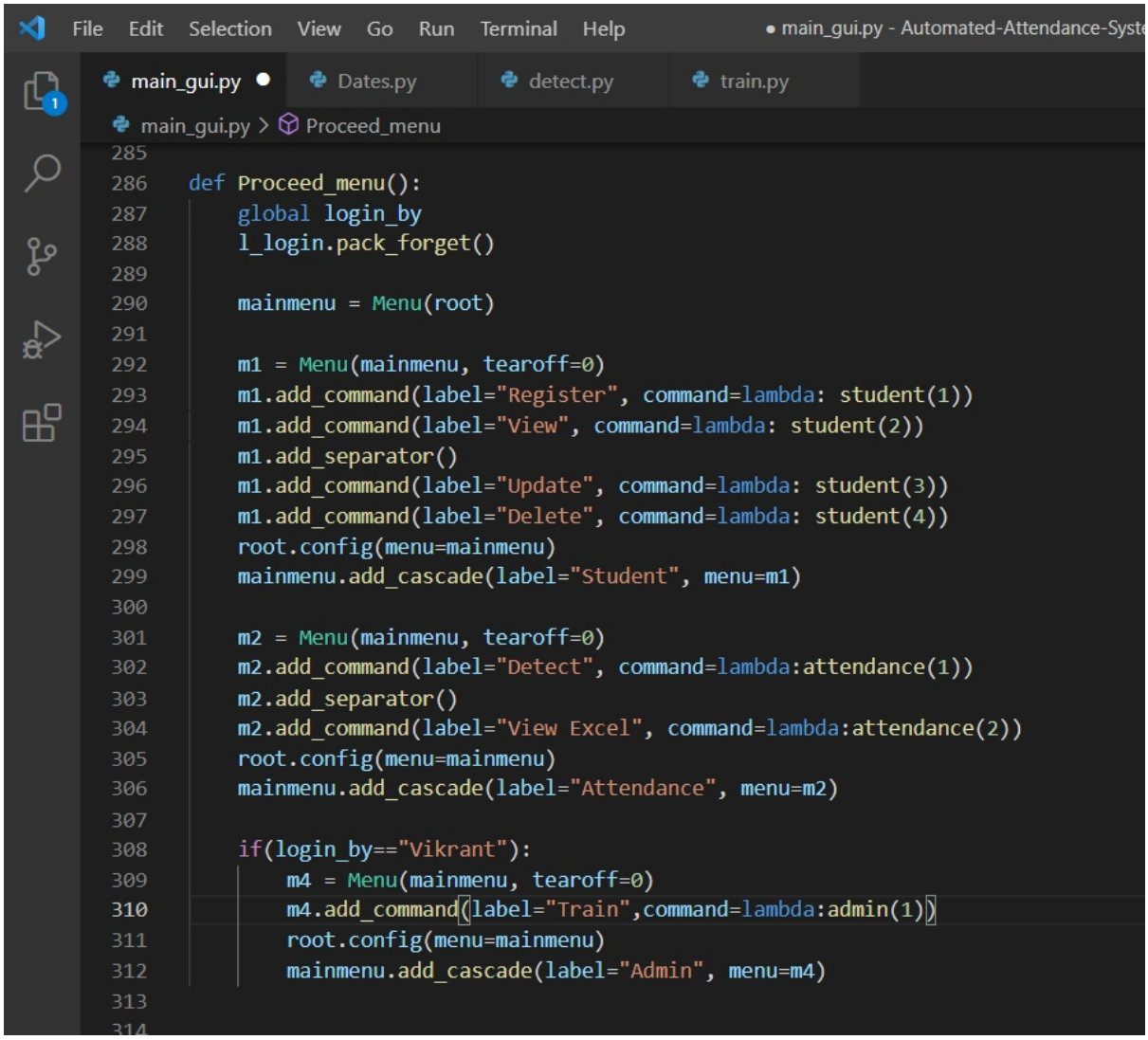


Fig. Proceed menu Code

```
def admin(x):
    if(x==1):
        print("i will train")
        tra.getImageswithId("../gray_images")
```

Fig.9 Train function called by admin

Registration

For registration of new students, their information is gathered and updated in the database file. The file's name is created on the basis of the information sent by the student. This is done to create an efficient database with the students of same class together.

Divided into Register Update and Delete.

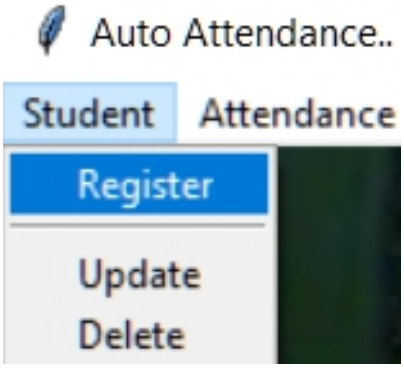


Fig.10 Menu of Register, Update and Delete

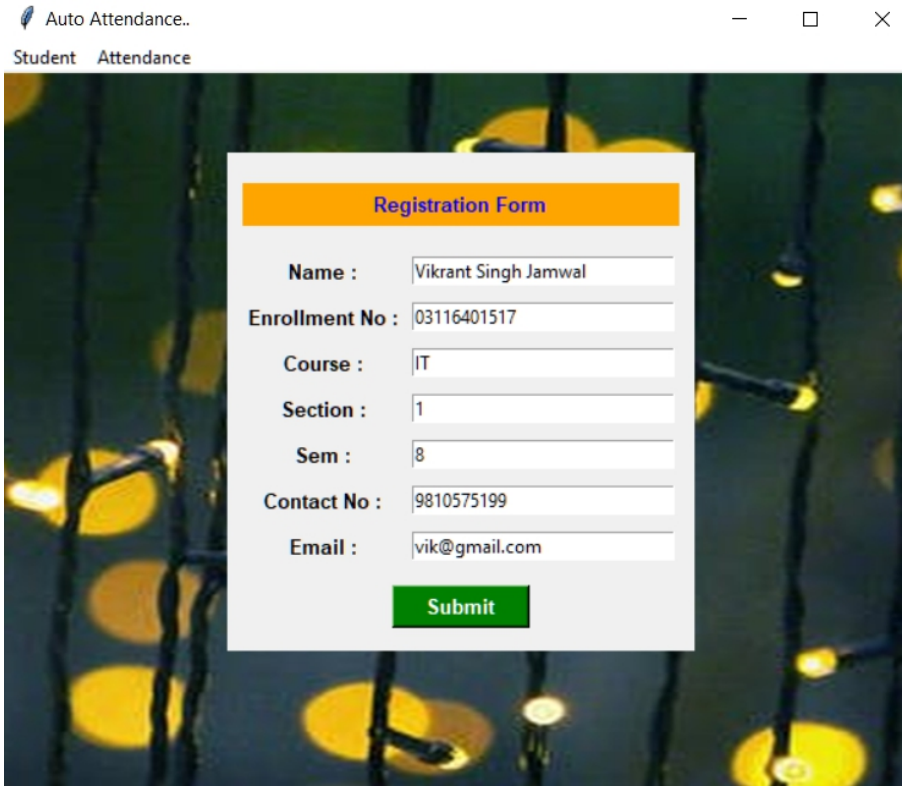


Fig.11 Registration window

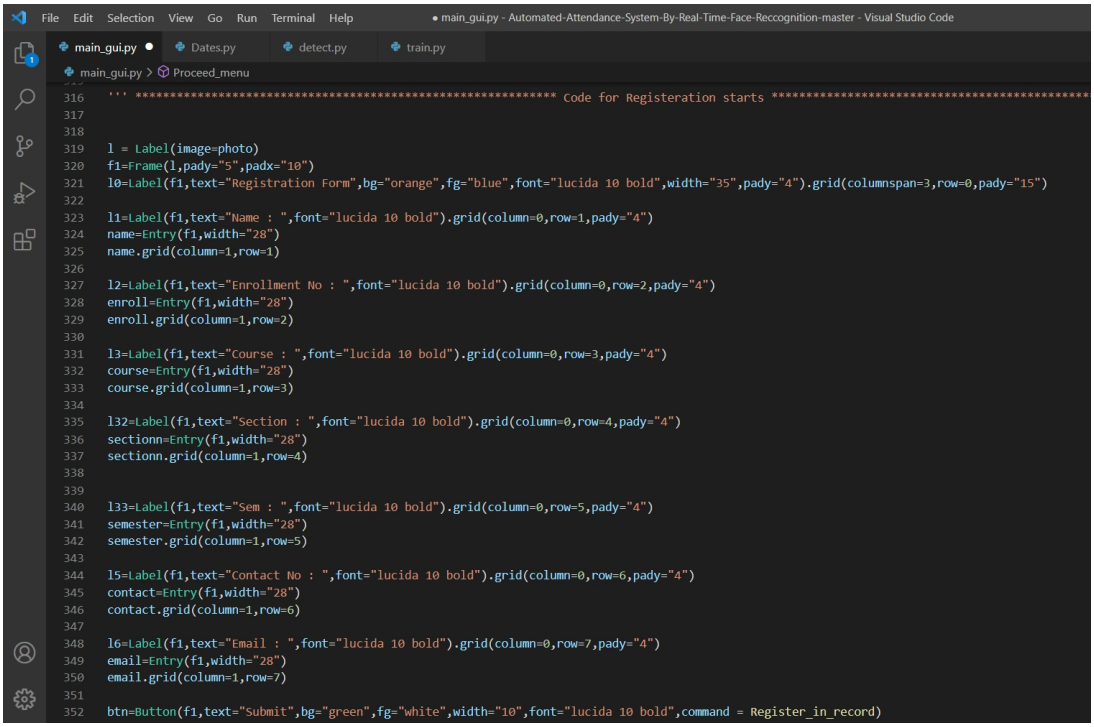


Fig.12 Code for creating UI of the registration window

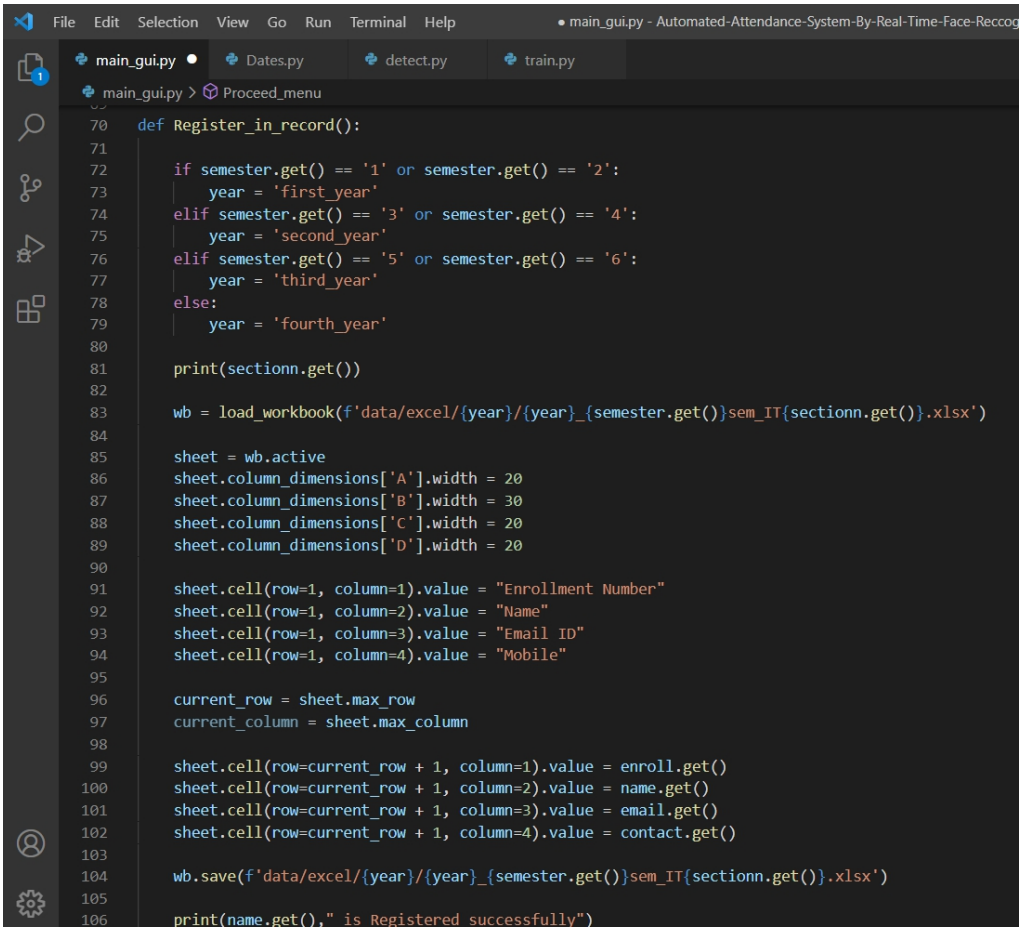


Fig.13 Code for registering in the excel file

Update

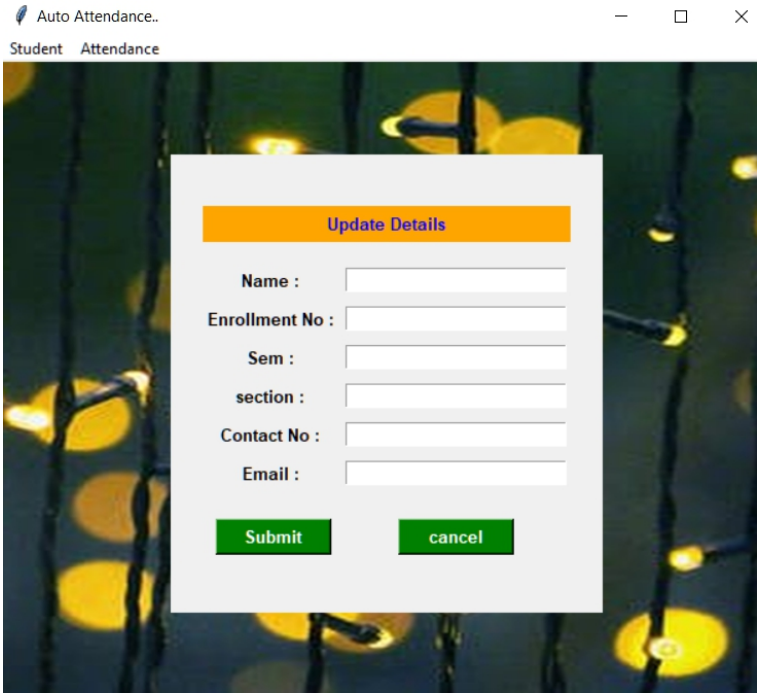


Fig.14 Update window

Delete

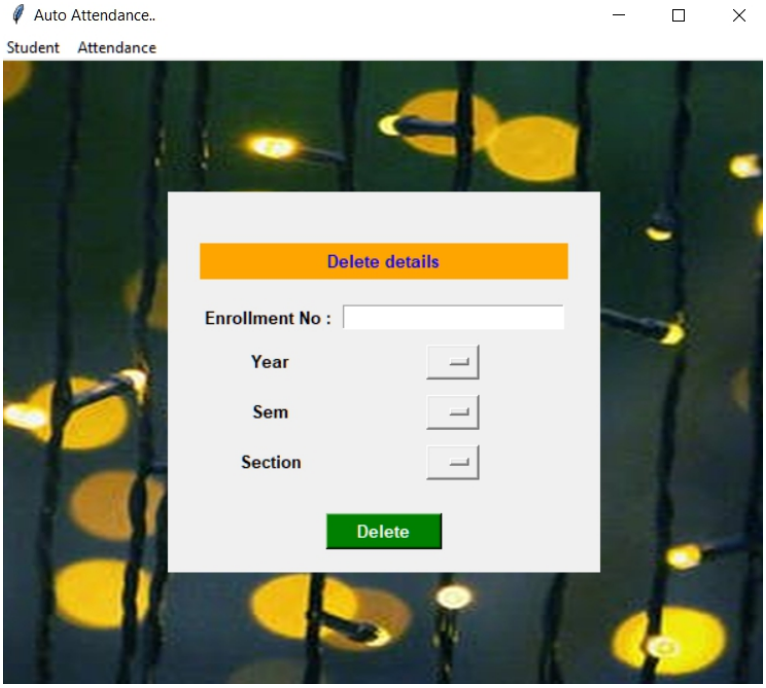


Fig.15 Delete Window

Detect

Detect opens up the given video and starts recognizing the faces seen in the video. As a face is recognised in 20 frames and has a confidence<80, then it is saved in an array which later is updated as attendance in the database.

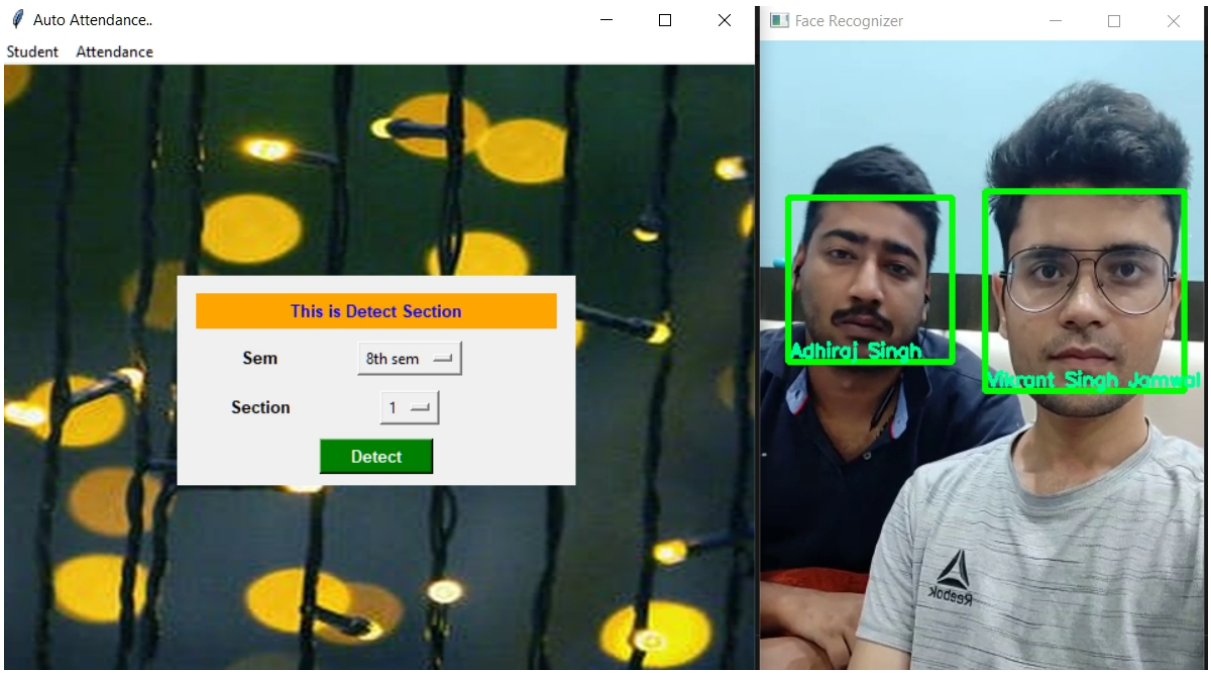


Fig.16 Detect and recogniser window

```
label: 0
confidence: 67.79053179760716
student Recognised :  {'Vikrant Singh Jamwal', 'Adhiraj Singh'} {0, 5}
label: 5
confidence: 65.37350455916452
student Recognised :  {'Vikrant Singh Jamwal', 'Adhiraj Singh'} {0, 5}
label: 0
confidence: 66.10602002899022
student Recognised :  {'Vikrant Singh Jamwal', 'Adhiraj Singh'} {0, 5}
label: 5
confidence: 64.14227643762891
student Recognised :  {'Vikrant Singh Jamwal', 'Adhiraj Singh'} {0, 5}
```

Fig.17 Terminal view of detection of faces

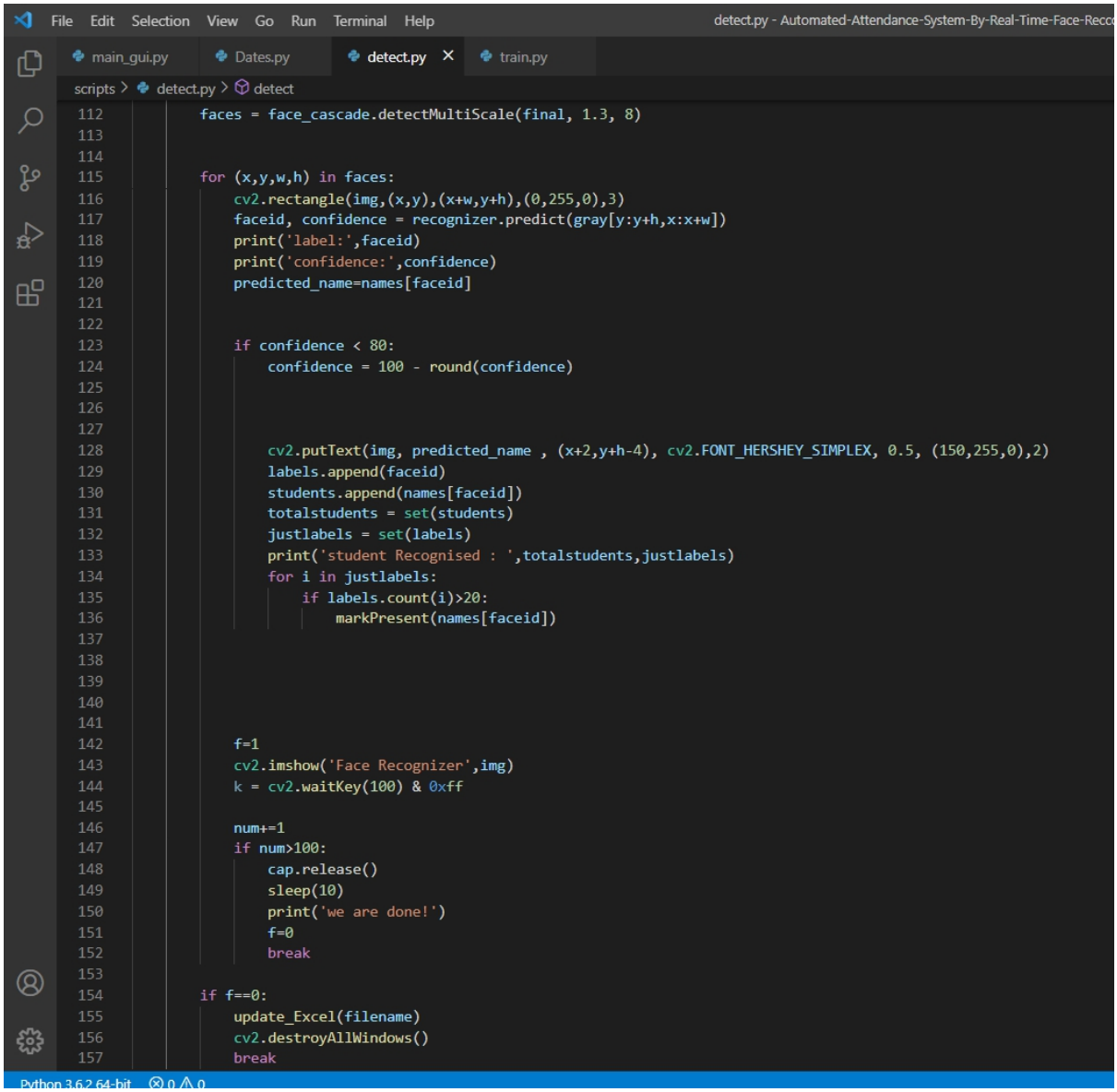


Fig.18 LBPH code for recognising and marking the attendance

Detect function has 4 more functions present in it.

- from_excel_to_csv- Converts excel file to CSV file.
- getData- acquiring the data(names) from the file
- markPresent- to the last column in the database, P is written in front of the name of the student who is recognised by the LBPH recogniser.
- update_Excel- updating excel file from the csv file and saving it.

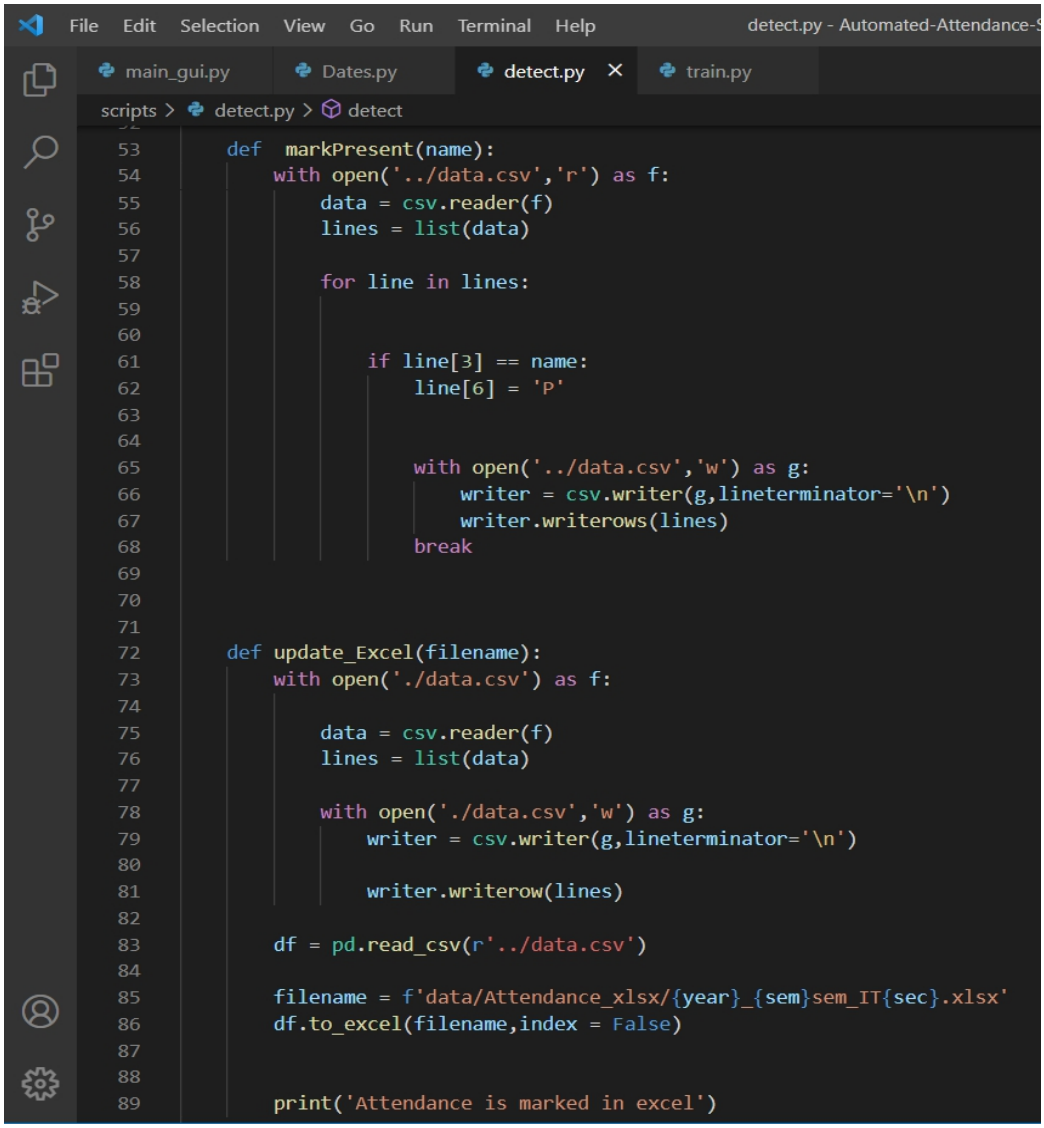


Fig.19 markPresent and update_Excel

```
def from_excel_to_csv():
    filename = f'data/Attendance_xlsx/{year}_{sem}sem_IT{sec}.xlsx'
    df = pd.read_excel(filename)
    df.to_csv('../data.csv')


def getdata():
    with open('../data.csv','r') as f:
        data = csv.reader(f)
        next(data)
        lines = list(data)
        for line in lines:
            names[int(line[0])] = line[3]
```

Fig.20 from_excel_to_csv and getData

Train

Train program is used to create a YML document which saves the threshold of the 8 neighbouring pixels in the image. IDs are created and linked to the face detected using haar-cascade and are saved in the YML document.

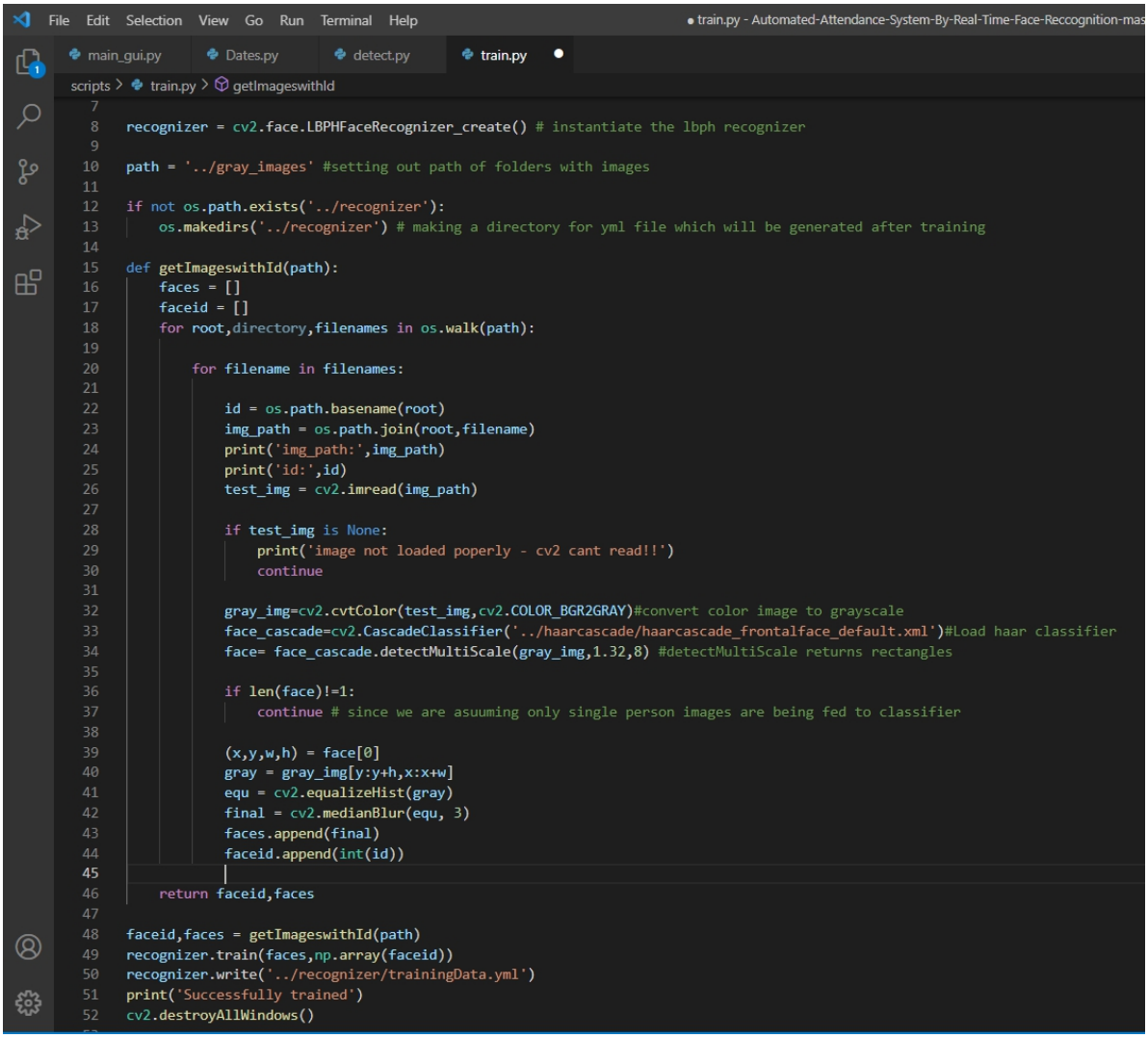


Fig.21 Train program code

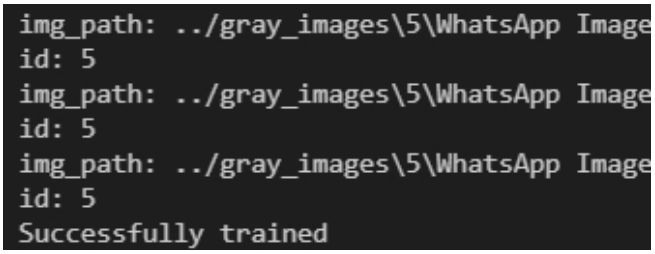


Fig.22 Terminal view

YML document

Contains ID and linked face threshold pixel values.

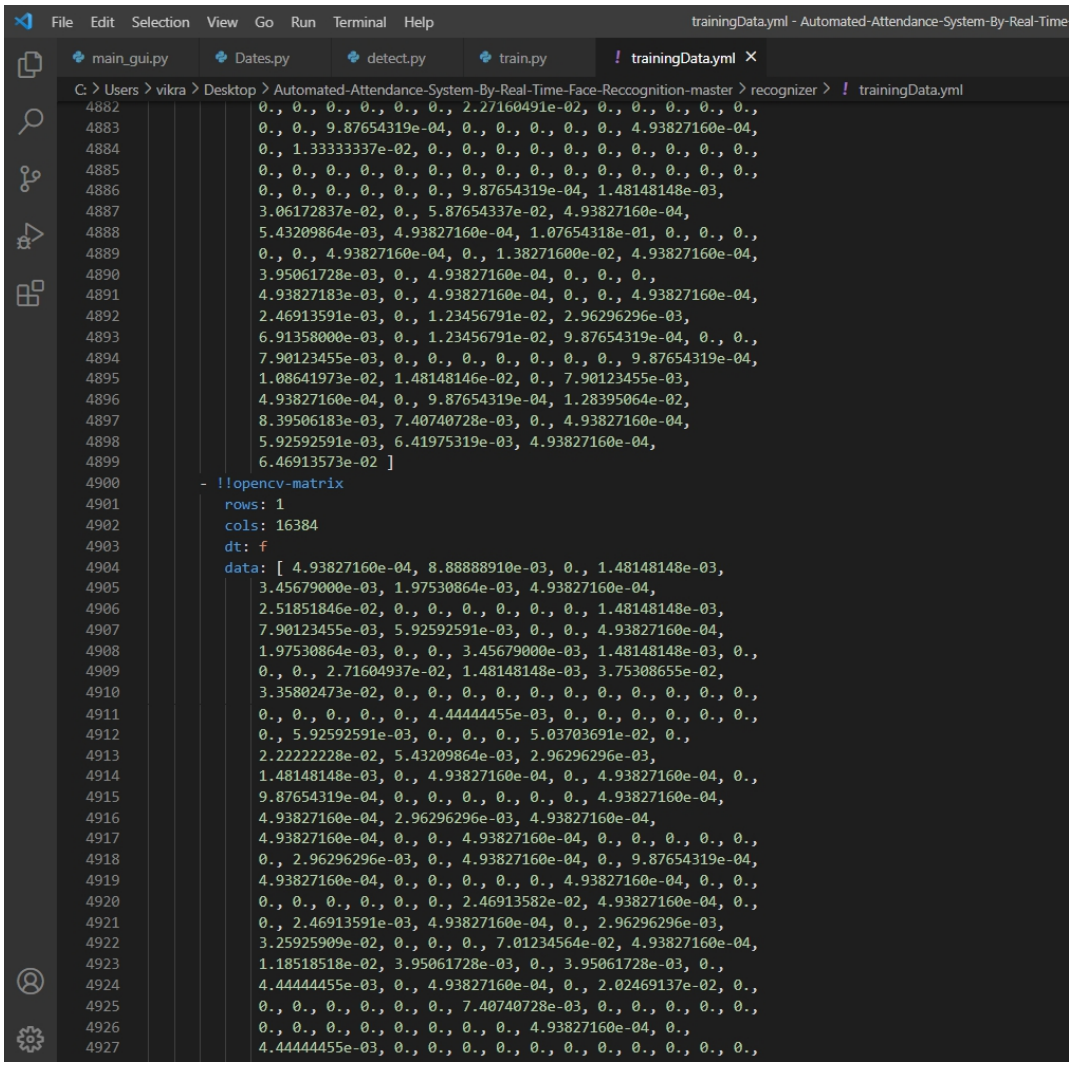


Fig.23 YML document

Pictures of the students are kept in a folder, all the pictures are first converted to gray images and then get detected and saved in YML document.

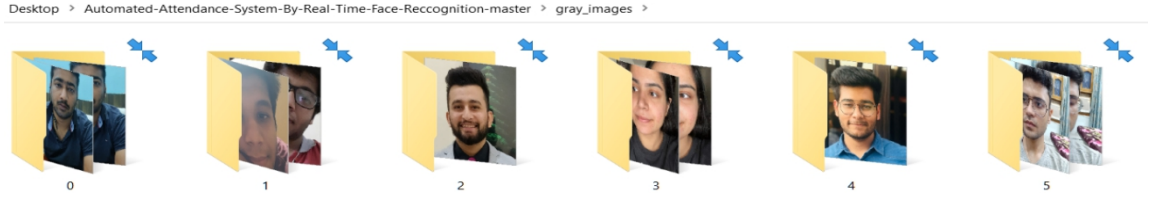


Fig.24 Pictures of the Students

Updating Excel file

After recognising the students, the excel file is updated. The heading attribute is the present Date and the default value under it is A(Absent). As the excel file is updated the P(Present) is written in front of the names of the students recognised in the given video.

Holidays are imported from the python packages to find out that any holiday lies on the present date or not.

Openpyxl is used to create a Workbook to manage and modify the excel file.

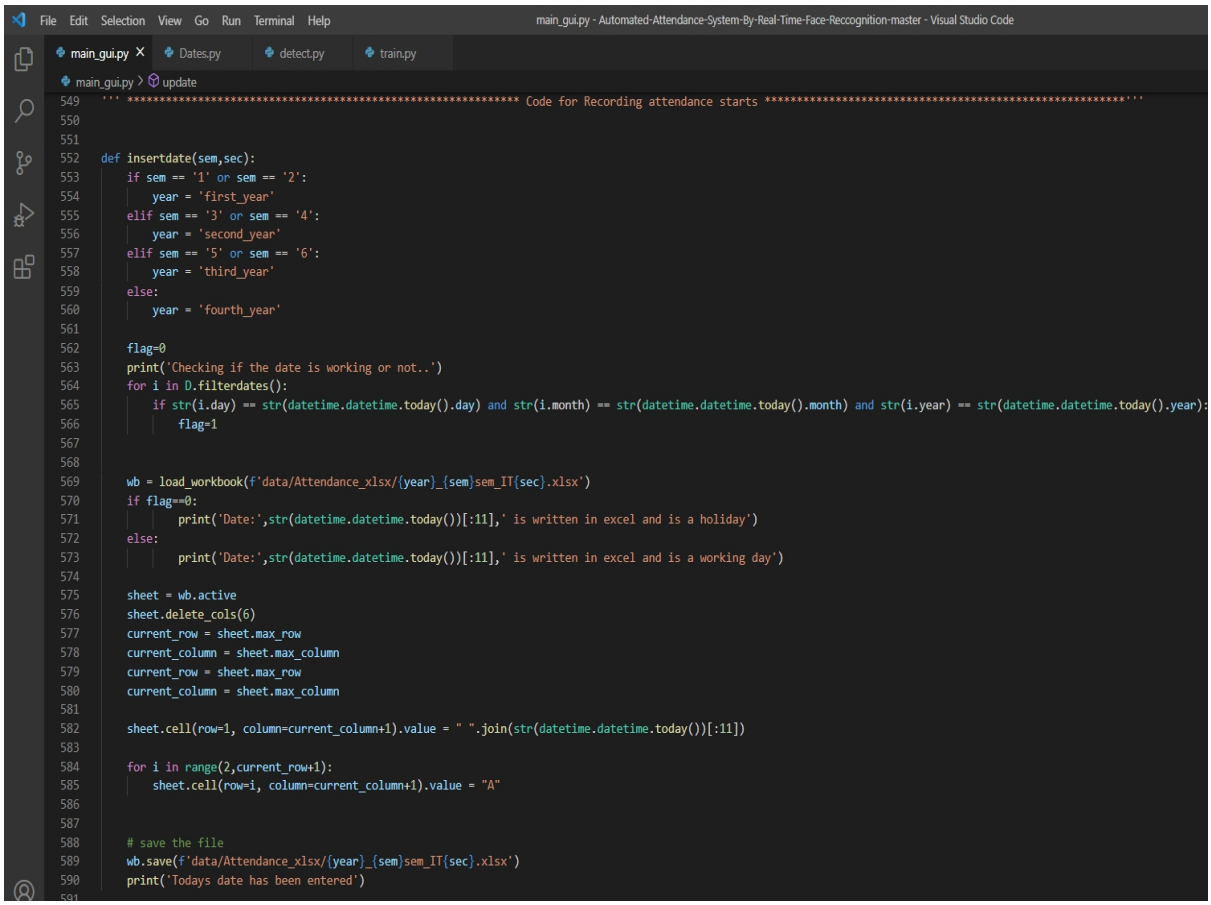


Fig.25 Code for inserting the date in the Excel file.

```
53      def  markPresent(name):
54          with open('../data.csv','r') as f:
55              data = csv.reader(f)
56              lines = list(data)
57
58              for line in lines:
59
60
61                  if line[3] == name:
62                      line[6] = 'P'
63
64
65                      with open('../data.csv','w') as g:
66                          writer = csv.writer(g,lineterminator='\n')
67                          writer.writerows(lines)
68                      break
69
```

Fig.26 Code for marking present in the CSV file.

```
    def update_Excel(filename):
        with open('./data.csv') as f:

            data = csv.reader(f)
            lines = list(data)

            with open('./data.csv','w') as g:
                writer = csv.writer(g,lineterminator='\n')

                writer.writerow(lines)

        df = pd.read_csv(r'../data.csv')

        filename = f'data/Attendance_xlsx/{year}_{sem}sem_IT{sec}.xlsx'
        df.to_excel(filename,index = False)


        print('Attendance is marked in excel')
```

Fig.27 Transferring data from CSV to Excel File.

Viewing The Excel File

View Excel option in the menu bar is used to open the excel file from the software itself.

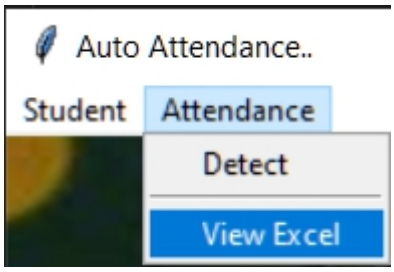


Fig.28 View Excel Option

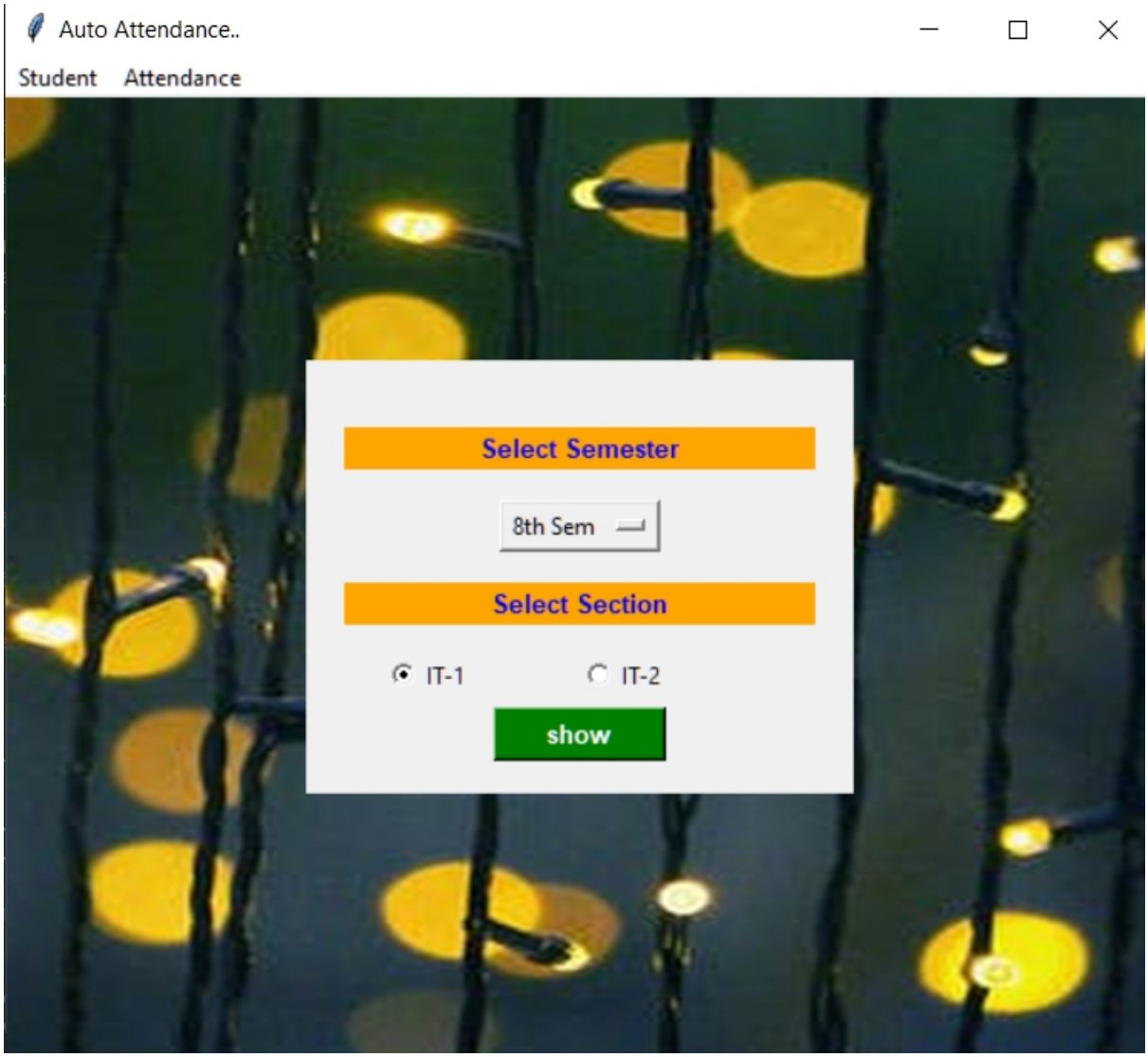
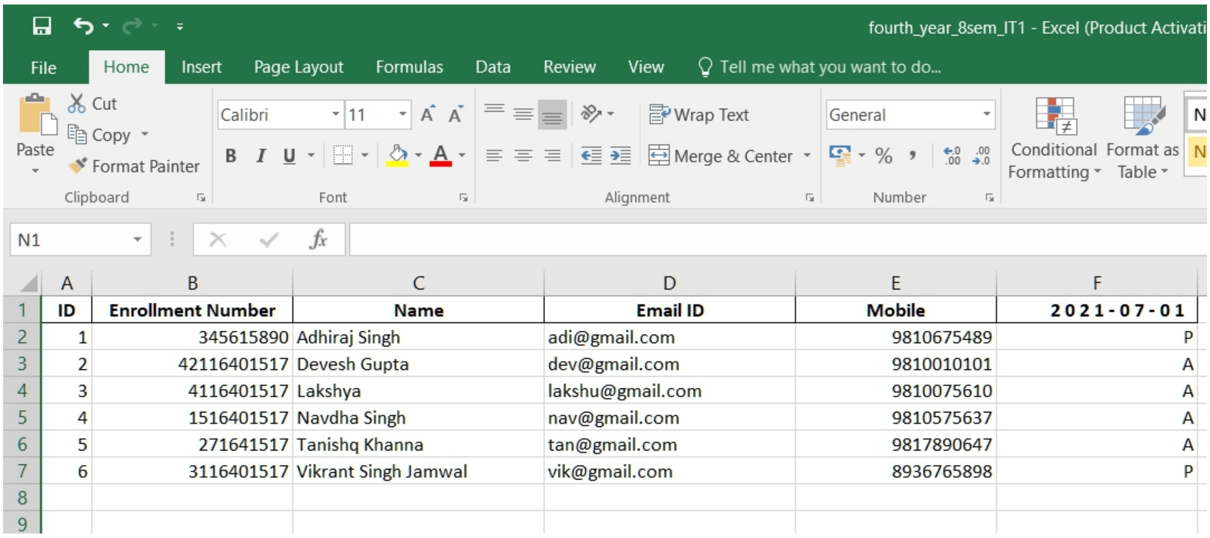
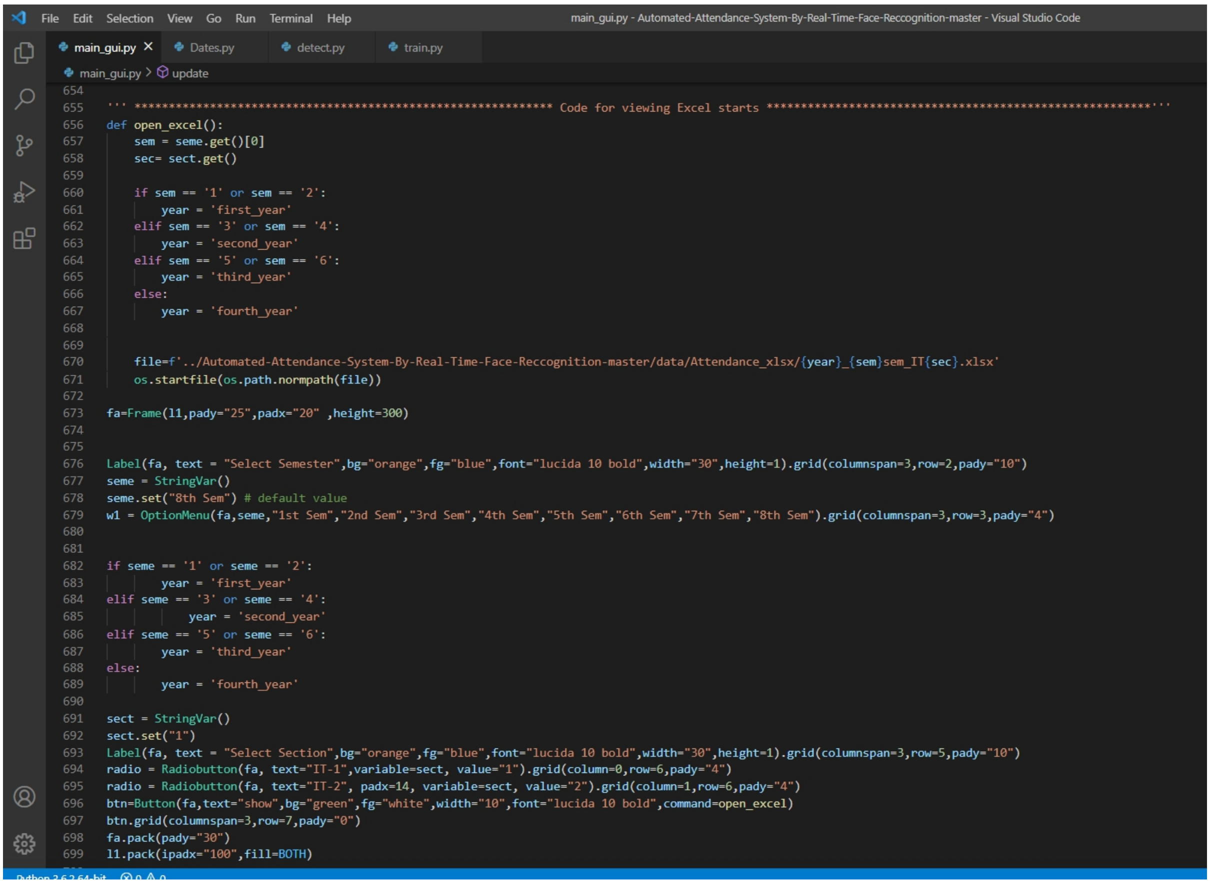


Fig.29 UI for opening the Excel file



ID	Enrollment Number	Name	Email ID	Mobile	2021-07-01
1	345615890	Adhiraj Singh	adi@gmail.com	9810675489	P
2	42116401517	Devesh Gupta	dev@gmail.com	9810010101	A
3	4116401517	Lakshya	lakshu@gmail.com	9810075610	A
4	1516401517	Navdha Singh	nav@gmail.com	9810575637	A
5	271641517	Tanishq Khanna	tan@gmail.com	9817890647	A
6	3116401517	Vikrant Singh Jamwal	vik@gmail.com	8936765898	P

Fig.30 Preview of the Updated Excel file.



```
''' ************************************************************ Code for viewing Excel starts ********************************************************'''
def open_excel():
    sem = seme.get()[0]
    sec= sect.get()

    if sem == '1' or sem == '2':
        year = 'first_year'
    elif sem == '3' or sem == '4':
        year = 'second_year'
    elif sem == '5' or sem == '6':
        year = 'third_year'
    else:
        year = 'fourth_year'


    file=f'../Automated-Attendance-System-By-Real-Time-Face-Reccognition-master/data/Attendance_xlsx/{year}_{sem}sem_IT{sec}.xlsx'
    os.startfile(os.path.normpath(file))

fa=Frame(l1,pady="25",padx="20" ,height=300)


Label(fa, text = "Select Semester",bg="orange",fg="blue",font="lucida 10 bold",width="30",height=1).grid(columnspan=3,row=2,pady="10")
seme = StringVar()
seme.set("8th Sem") # default value
w1 = OptionMenu(fa,seme,"1st Sem","2nd Sem","3rd Sem","4th Sem","5th Sem","6th Sem","7th Sem","8th Sem").grid(columnspan=3,row=3,pady="4")


if seme == '1' or seme == '2':
        year = 'first_year'
elif seme == '3' or seme == '4':
            year = 'second_year'
elif seme == '5' or seme == '6':
        year = 'third_year'
else:
        year = 'fourth_year'

sect = StringVar()
sect.set("1")
Label(fa, text = "Select Section",bg="orange",fg="blue",font="lucida 10 bold",width="30",height=1).grid(columnspan=3,row=5,pady="10")
radio = Radiobutton(fa, text="IT-1",variable=sect, value="1").grid(column=0,row=6,pady="4")
radio = Radiobutton(fa, text="IT-2", padx=14, variable=sect, value="2").grid(column=1,row=6,pady="4")
btn=Button(fa,text="show",bg="green",fg="white",width="10",font="lucida 10 bold",command=open_excel)
btn.grid(columnspan=3,row=7,pady="0")
fa.pack(pady="30")
l1.pack(ipadx="100",fill=BOTH)
```

Fig.31 Code for viewing Excel file

References

1. https://docs.python.org/3/library/tkinter.html
2. www.intel.in
3. https://code.visualstudio.com
4. https://docs.opencv.org/
5. https://www.youtube.com/
6. https://www.w3schools.com/
7. https://stackoverflow.com/

Books

1. ‘Principles of digital image processing’ by Burge and Burger
2. 'Head-First Python' by Paul Barry