Chittagong University of Engineering & Technology

Department of Computer Science & Engineering

Course code: CSE-300

Course Title: Software Development Project (Sessional)

Final Report on

Automated Attendance System

Team Member

Sanjida Nuri Pearl (1804023)

Sayeda Tahmina (1804034)

Sanjida Nahid Shorna (1804049)

Sauda Adiv Hanum (1804063)

Course Teacher

Animesh Chandra Roy

Assistant Professor

Dept. of Computer Science & Engineering

Chittagong University of Engineering & Technology

Md. Atiqul Islam Rizvi

Assistant Professor

Dept. of Computer Science & Engineering

Chittagong University of Engineering & Technology

1. Abstract :

Taking attendance is one of the most important jobs that must be completed every day at colleges, institutions, and schools. Most of the time, it is done manually, like calling someone by name or by roll number. We have made a system that detects and recognizes the face as a student attendance system and can be a substitute for the regular paper attendance system. This project meets the requirements for bringing modernization to the way attendance is handled, as well as the criteria for time management. This system is installed in the classroom and student's information, such as name and roll number are trained.

2. Introduction :

This is a project about Facial Recognition-Based Attendance System for Educational Institutions. The process uses a computer application that captures a digital image of an individual's face and compares it to images in a database of stored records. It detects the faces and marks attendance accordingly. This system will prevent unnecessary wastage of time of classes that is usually wasted in the form of class roll calls. This system will be designed using Python.

3. Objective :

The goals we want to achieve by the this project :

1. Reducing time wastage during conventional class attendance
2. Utilizing the latest trends in machine vision to implement a feasible solution for the class attendance system
3. Automating the whole process so that we have a digital environment
4. Preventing fake roll calls as one to one attendance marking is possible only
5. Encouraging the use of technology in daily lives

4. Project Description :

Flowchart :

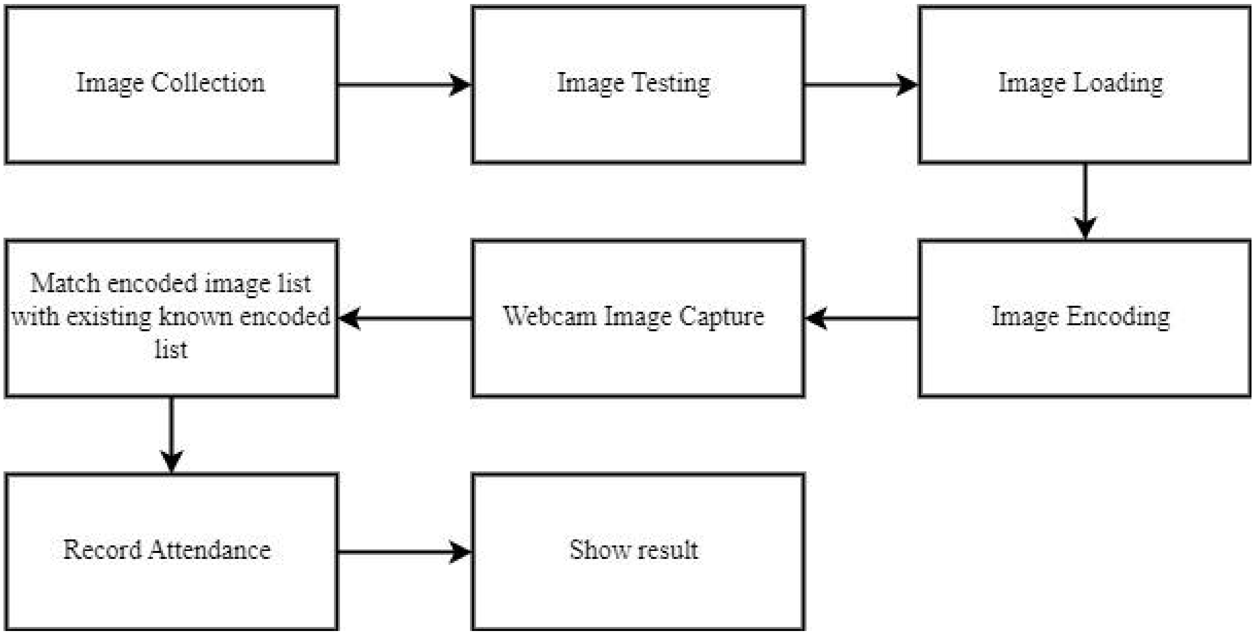


Figure 01 : Flow chart of our workflow

5. Language Required :

5.1. Python :

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.

6. Required Algorithm :

6.1. SVM :

It is a supervised learning technique mostly utilized for classifying data into several categories. A set of label data is used to train SVM. SVMs are utilized in web pages, intrusion detection, face identification, email categorization, gene classification, and handwriting recognition, among other applications. Both classification and regression on linear and non-linear data are supported. Making a straight line between two classes is how a straightforward linear SVM classifier functions. In other words, the data points on one side of the line will all be assigned to one category, while the data points on the other side of the line will be assigned to a different category.

7. Required Package :

7.1. Cmake :

A suite of open-source, cross-platform tools called CMake is used to create, test, and package software. Using straightforward platform and compiler independent configuration files, CMake is used to manage the software compilation process. It also creates native workspaces and makefiles that can be used in any compiler environment. In order to provide an effective, cross-platform build environment for open-source programs like ITK and VTK, Kitware developed the CMake toolkit.

7.2. dlib :

The dlib is used to estimate the location of the coordinates (x, y) that map the facial points on a person's face. It is a landmark's facial detector using pre-trained models. The pre-trained model is used to identify these spots.

7.3. NumPy :

Using NumPy the image is converted into a grayscale image. Also matrix analysis, Fourier transforms, linear algebra, and data science all make use of NumPy. Python lists are significantly slower than NumPy arrays.

7.4. OpenCv :

OpenCV is a sizable open-source library for image processing, machine learning, and computer vision. It now plays a significant part in real-time operation, which is crucial in modern systems. Using it, one may analyze pictures and movies to find faces, objects, and even human handwriting.

7.5. face_recognition :

To identify faces in a photo or photo folder, use the face recognition() function. We must first present a folder containing a single image of each person we already know. For each individual in the photo, there should be a separate image file with their name in the filename.

8. Function Used :

8.1. face_recognition.load_image_file() :

By using this function we load an image from the folder.

8.2. cvtColor() :

Our input image color is in BGR form but the library can detect images in RGB form that’s why it is needed to convert the image from BGR to RGB form.

8.3. cv2.imshow() :

A window containing an image is displayed using the cv2. imshow() technique. The image size is automatically adjusted for the window.

8.4. waitKey() :

waitKey() function of Python OpenCV allows users to display a window for given milliseconds or until any key is pressed. It takes time in milliseconds as a parameter and waits for the given time to destroy the window, if 0 is passed in the argument it waits till any key is pressed.

8.5. face_recognition.face_locations() :

The Histogram of Oriented gradients method is used by the face recognition.face_locations() method to process the NumPy array of the image from load image(). This returns a list with the locations of the faces it finds in the image.

8.6. face_recognition.face_encodings() :

It extracts 128 measurements from an image so that it can differentiate an image from another image.

8.7. cv2.rectangle() :

It is used to draw a rectangle around an image.

8.8. face_recognition.compare_faces() :

By this function two images are compared whether these are the images of the same person or not.

8.9. face_recognition.face_distance() :

By this function distance between two images is found that denotes how similar two images are. The lower the distance, the better the match is.

8.10. cv2.putText() :

In this function, we defined the properties of text results such as color, thickness, font and the distance between two images.

8.11. os.listdir() :

In this function, we send the name of the folder which contains all images of our dataset. And by this we can grab all of the images in our dataset.

8.12. append() :

By this function, we can add an item to the end of the list.

8.13. cv2.VideoCapture() :

It is a function of openCV library that allows working with video by capturing images by webcam.

8.14. read() :

By this function we read images using a webcam.

8.15. cv2.resize() :

This function is used to reduce the real time image size to increase the speed of the process.

8.16. np.argmin() :

By this function we found out the minimum distanced image’s index.

8.17. upper() :

By using this function we convert our result’s text into uppercase letters.

8.18. readlines() :

This function is used to read all the lines that we have in the data. That is because if somebody has already arrived we don’t want to repeat it.

8.19. line.split() :

This is used to mark by which character we want to split the entries based on that character.

8.20. datetime.now() :

By this function we found out the time of student arrival.

8.21. strftime() :

This is used to set the format of the time of student arrival.

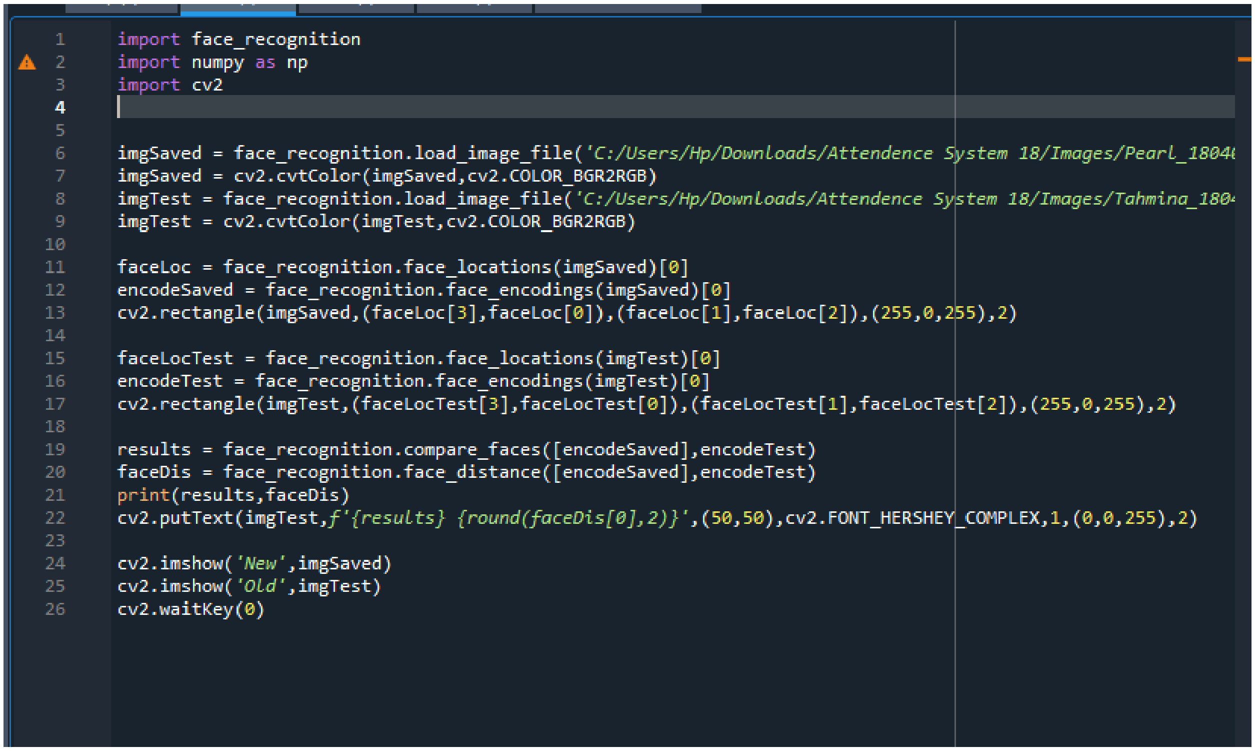
8.22. writelines() :

This is used to write the student name and his/her arrival time.

9. Main Project :

9.1. Picture matching and testing

Here we initially compare the two images to see if our output matches. For example, if two different pictures of the same person are given, the output shows true and if we compare the pictures of two different people, the output shows false.



```
import face_recognition
import numpy as np
import cv2


imgSaved = face_recognition.load_image_file('C:/Users/Hp/Downloads/Attendence System 18/Images/Pearl_1804
imgSaved = cv2.cvtColor(imgSaved,cv2.COLOR_BGR2RGB)
imgTest = face_recognition.load_image_file('C:/Users/Hp/Downloads/Attendence System 18/Images/Tahmina_180
imgTest = cv2.cvtColor(imgTest,cv2.COLOR_BGR2RGB)

faceLoc = face_recognition.face_locations(imgSaved)[0]
encodeSaved = face_recognition.face_encodings(imgSaved)[0]
cv2.rectangle(imgSaved,(faceLoc[3],faceLoc[0]),(faceLoc[1],faceLoc[2]),(255,0,255),2)

faceLocTest = face_recognition.face_locations(imgTest)[0]
encodeTest = face_recognition.face_encodings(imgTest)[0]
cv2.rectangle(imgTest,(faceLocTest[3],faceLocTest[0]),(faceLocTest[1],faceLocTest[2]),(255,0,255),2)

results = face_recognition.compare_faces([encodeSaved],encodeTest)
faceDis = face_recognition.face_distance([encodeSaved],encodeTest)
print(results,faceDis)
cv2.putText(imgTest,f'{results} {round(faceDis[0],2)}',(50,50),cv2.FONT_HERSHEY_COMPLEX,1,(0,0,255),2)

cv2.imshow('New',imgSaved)
cv2.imshow('Old',imgTest)
cv2.waitKey(0)
```

Figure 02 : Code of picture matching and testing

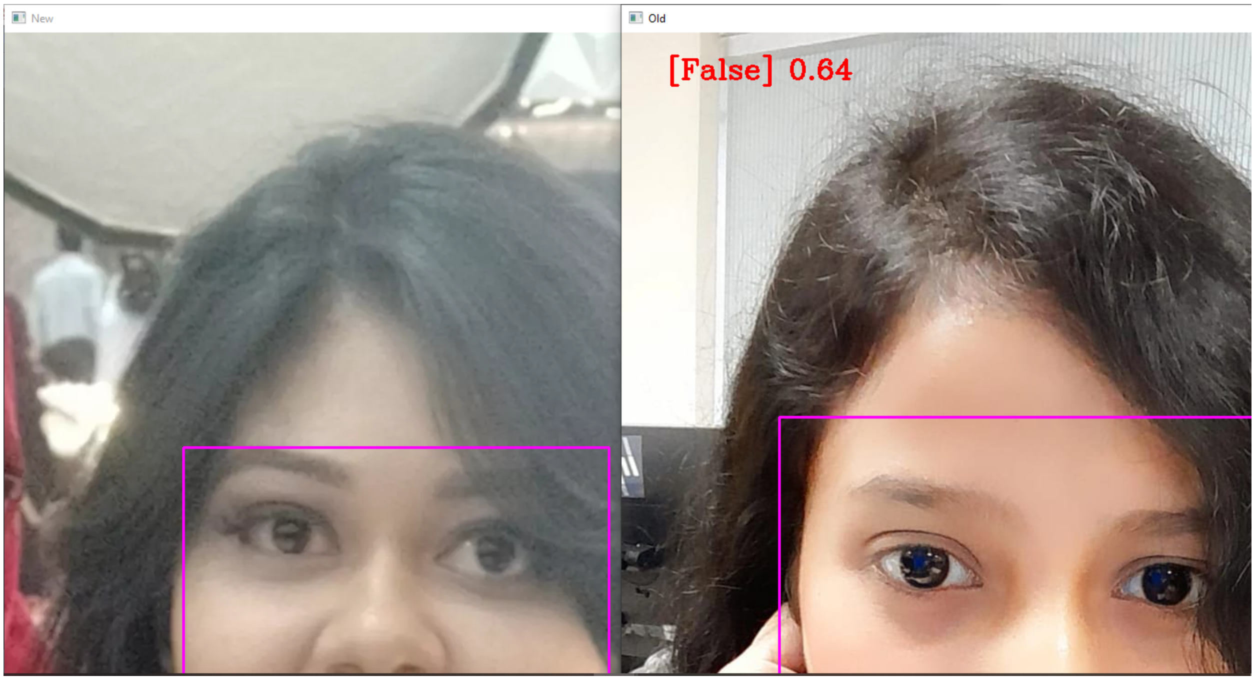
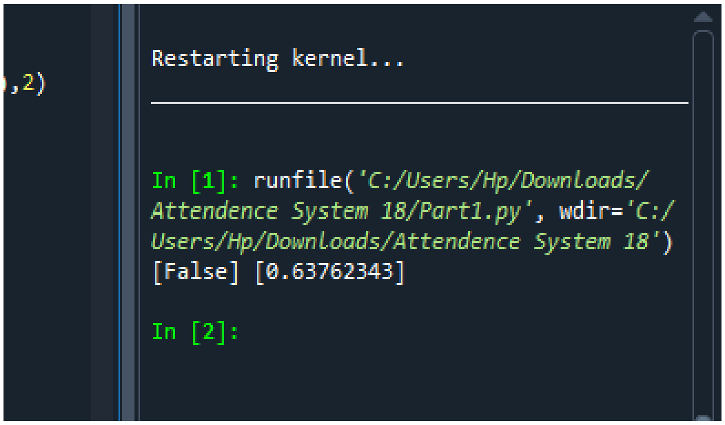


Figure 03 : Output of picture matching and testing

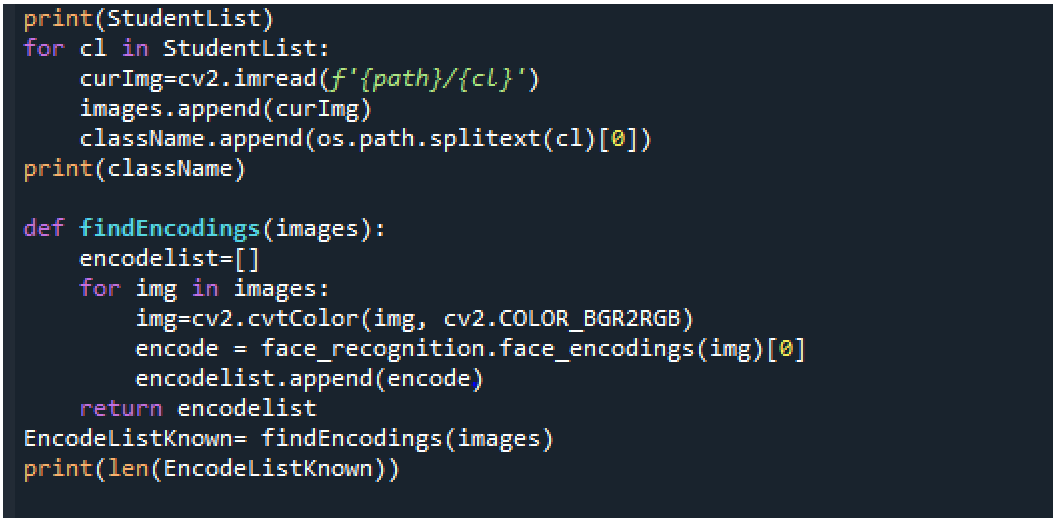


```
,2)
Restarting kernel...

In [1]: runfile('C:/Users/Hp/Downloads/
Attendence System 18/Part1.py', wdir='C:/
Users/Hp/Downloads/Attendence System 18')
[False] [0.63762343]

In [2]:
```

Figure 04 : Output in the kernel of picture matching and testing

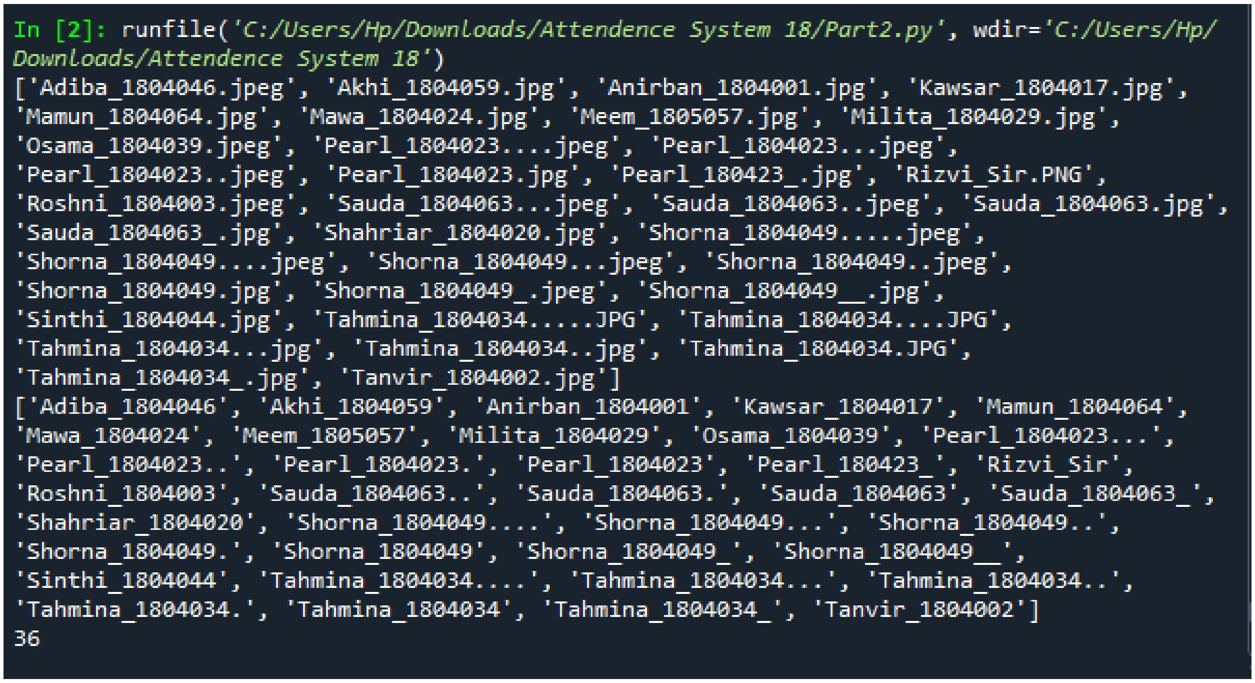
9.2. Picture encoding

Picture encoding is basically a way to represent the face using a set of 128 computer-generated measurements of a face. Here we have made a list of all the names from their image names and have encoded the faces from images.



```
print(StudentList)
for cl in StudentList:
    curImg=cv2.imread(f'{path}/{cl}')
    images.append(curImg)
    className.append(os.path.splitext(cl)[0])
print(className)

def findEncodings(images):
    encodelist=[]
    for img in images:
        img=cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodelist.append(encode)
    return encodelist
EncodeListKnown= findEncodings(images)
print(len(EncodeListKnown))
```

Figure 05 : Code of picture encoding

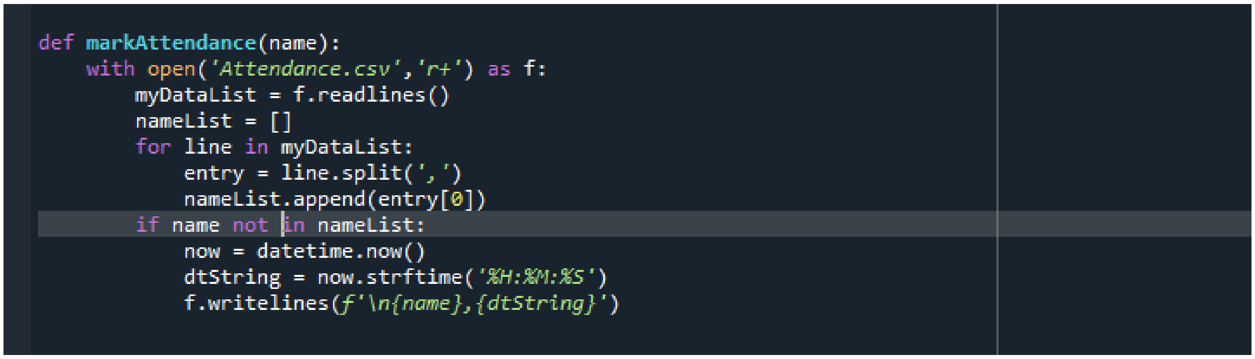


```
In [2]: runfile('C:/Users/Hp/Downloads/Attendence System 18/Part2.py', wdir='C:/Users/Hp/
Downloads/Attendence System 18')
['Adiba_1804046.jpeg', 'Akhi_1804059.jpg', 'Anirban_1804001.jpg', 'Kawsar_1804017.jpg',
'Mamun_1804064.jpg', 'Mawa_1804024.jpg', 'Meem_1805057.jpg', 'Milita_1804029.jpg',
'Osama_1804039.jpeg', 'Pearl_1804023....jpeg', 'Pearl_1804023...jpeg',
'Pearl_1804023..jpeg', 'Pearl_1804023.jpg', 'Pearl_180423_.jpg', 'Rizvi_Sir.PNG',
'Roshni_1804003.jpeg', 'Sauda_1804063...jpeg', 'Sauda_1804063..jpeg', 'Sauda_1804063.jpg',
'Sauda_1804063_.jpg', 'Shahriar_1804020.jpg', 'Shorna_1804049.....jpeg',
'Shorna_1804049....jpeg', 'Shorna_1804049...jpeg', 'Shorna_1804049..jpeg',
'Shorna_1804049.jpg', 'Shorna_1804049_.jpeg', 'Shorna_1804049__.jpg',
'Sinthi_1804044.jpg', 'Tahmina_1804034.....JPG', 'Tahmina_1804034....JPG',
'Tahmina_1804034...jpg', 'Tahmina_1804034..jpg', 'Tahmina_1804034.JPG',
'Tahmina_1804034_.jpg', 'Tanvir_1804002.jpg']
['Adiba_1804046', 'Akhi_1804059', 'Anirban_1804001', 'Kawsar_1804017', 'Mamun_1804064',
'Mawa_1804024', 'Meem_1805057', 'Milita_1804029', 'Osama_1804039', 'Pearl_1804023...',
'Pearl_1804023..', 'Pearl_1804023.', 'Pearl_1804023', 'Pearl_180423_', 'Rizvi_Sir',
'Roshni_1804003', 'Sauda_1804063..', 'Sauda_1804063.', 'Sauda_1804063', 'Sauda_1804063_',
'Shahriar_1804020', 'Shorna_1804049....', 'Shorna_1804049...', 'Shorna_1804049..',
'Shorna_1804049.', 'Shorna_1804049', 'Shorna_1804049_', 'Shorna_1804049__',
'Sinthi_1804044', 'Tahmina_1804034....', 'Tahmina_1804034...', 'Tahmina_1804034..',
'Tahmina_1804034.', 'Tahmina_1804034', 'Tahmina_1804034_', 'Tanvir_1804002']
36
```

Figure 06 : Output of picture encoding

9.3. Mark attendance function

Here we have done the job of attaching a .csv file for students to take their attendance. If we already have a student in our .csv file then it will not insert that in our file and if not the student name and present time will be inserted into our .csv file.

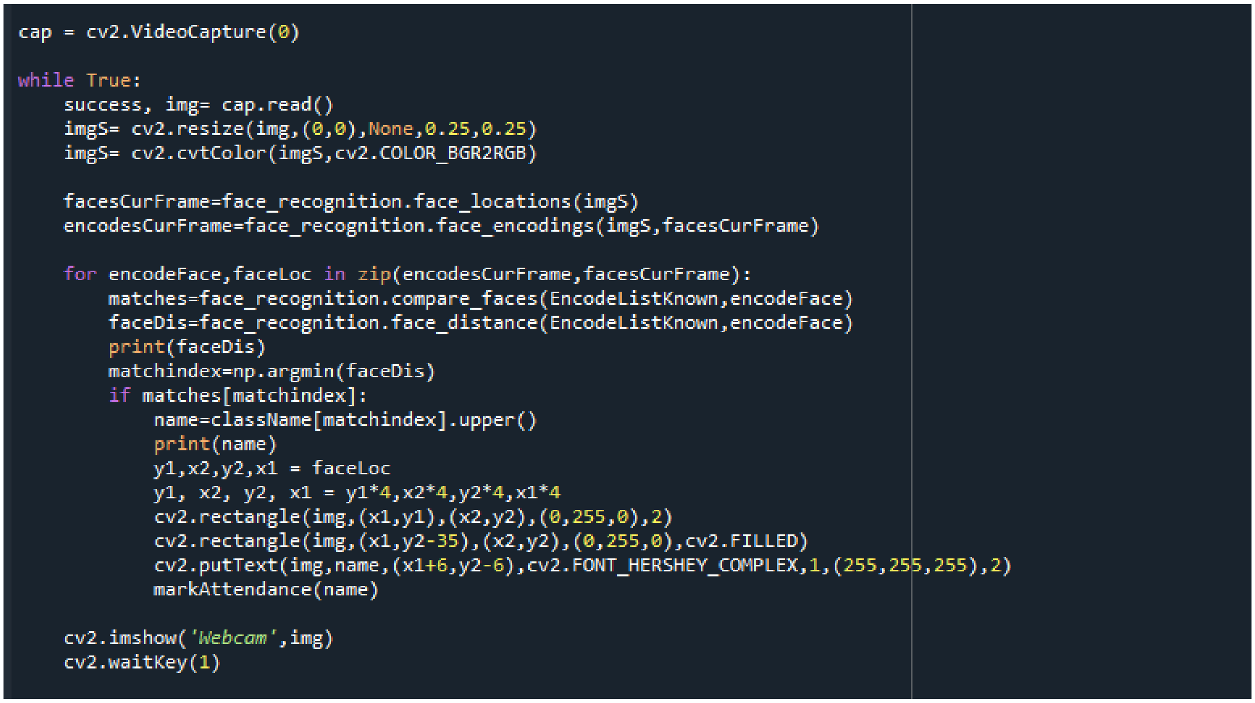


```
def markAttendance(name):
    with open('Attendance.csv','r+') as f:
        myDataList = f.readlines()
        nameList = []
        for line in myDataList:
            entry = line.split(',')
            nameList.append(entry[0])
        if name not in nameList:
            now = datetime.now()
            dtString = now.strftime('%H:%M:%S')
            f.writelines(f'\n{name},{dtString}')
```

Figure 07 : Code of marking attendance function

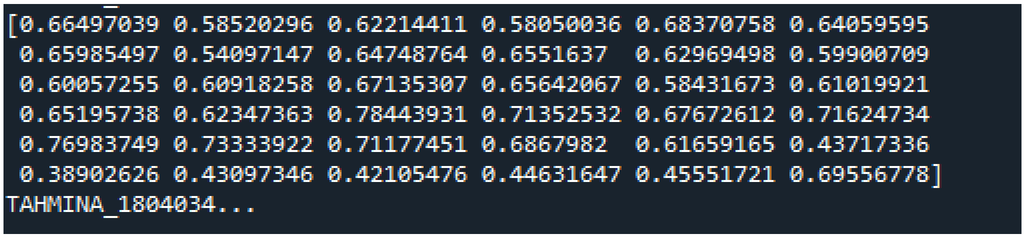
9.4. Webcam picture encoding

In this section we first open our webcam with cv2. Then the webcam captures our picture and resizes it and makes it in rgb format. Then the picture is encoded. This encoded picture is matched with our encoded known picture list. If the encoded picture matches, then we print the class name as our name and put a rectangle around our faces to detect our faces.



```
cap = cv2.VideoCapture(0)

while True:
    success, img= cap.read()
    imgS= cv2.resize(img,(0,0),None,0.25,0.25)
    imgS= cv2.cvtColor(imgS,cv2.COLOR_BGR2RGB)

    facesCurFrame=face_recognition.face_locations(imgS)
    encodesCurFrame=face_recognition.face_encodings(imgS,facesCurFrame)

    for encodeFace,faceLoc in zip(encodesCurFrame,facesCurFrame):
        matches=face_recognition.compare_faces(EncodeListKnown,encodeFace)
        faceDis=face_recognition.face_distance(EncodeListKnown,encodeFace)
        print(faceDis)
        matchindex=np.argmin(faceDis)
        if matches[matchindex]:
            name=className[matchindex].upper()
            print(name)
            y1,x2,y2,x1 = faceLoc
            y1, x2, y2, x1 = y1*4,x2*4,y2*4,x1*4
            cv2.rectangle(img,(x1,y1),(x2,y2),(0,255,0),2)
            cv2.rectangle(img,(x1,y2-35),(x2,y2),(0,255,0),cv2.FILLED)
            cv2.putText(img,name,(x1+6,y2-6),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),2)
            markAttendance(name)

    cv2.imshow('Webcam',img)
    cv2.waitKey(1)
```

Figure 08 : Code of webcam picture encoding

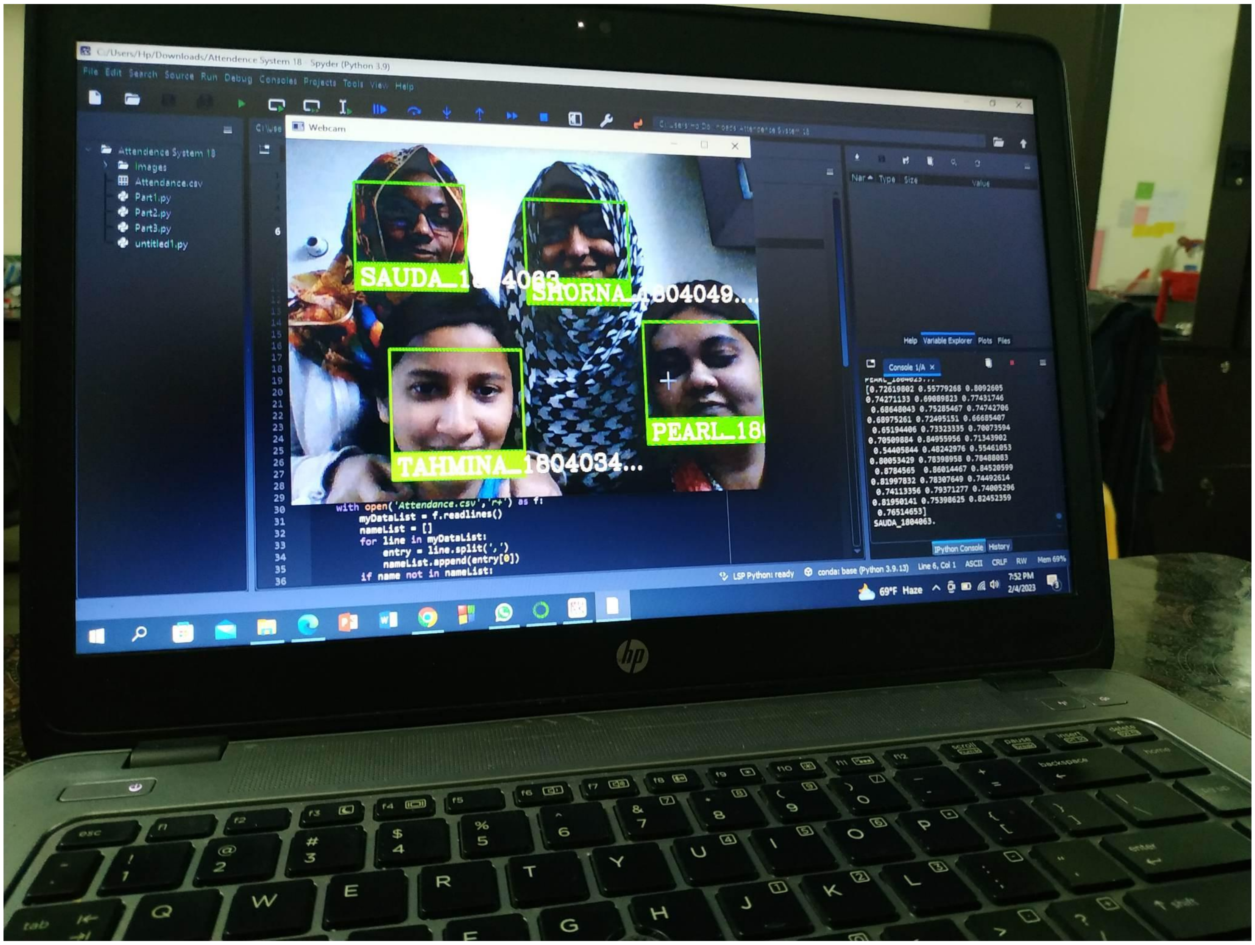


```
[0.66497039 0.58520296 0.62214411 0.58050036 0.68370758 0.64059595
 0.65985497 0.54097147 0.64748764 0.6551637  0.62969498 0.59900709
 0.60057255 0.60918258 0.67135307 0.65642067 0.58431673 0.61019921
 0.65195738 0.62347363 0.78443931 0.71352532 0.67672612 0.71624734
 0.76983749 0.73333922 0.71177451 0.6867982  0.61659165 0.43717336
 0.38902626 0.43097346 0.42105476 0.44631647 0.45551721 0.69556778]
TAHMINA_1804034...
```

Figure 09 : Output of webcam picture encoding

9.5. Result

Here is the picture of our expected result. Here we see the labels with our name. It shows that our system can successfully detect our faces and record our attendance.

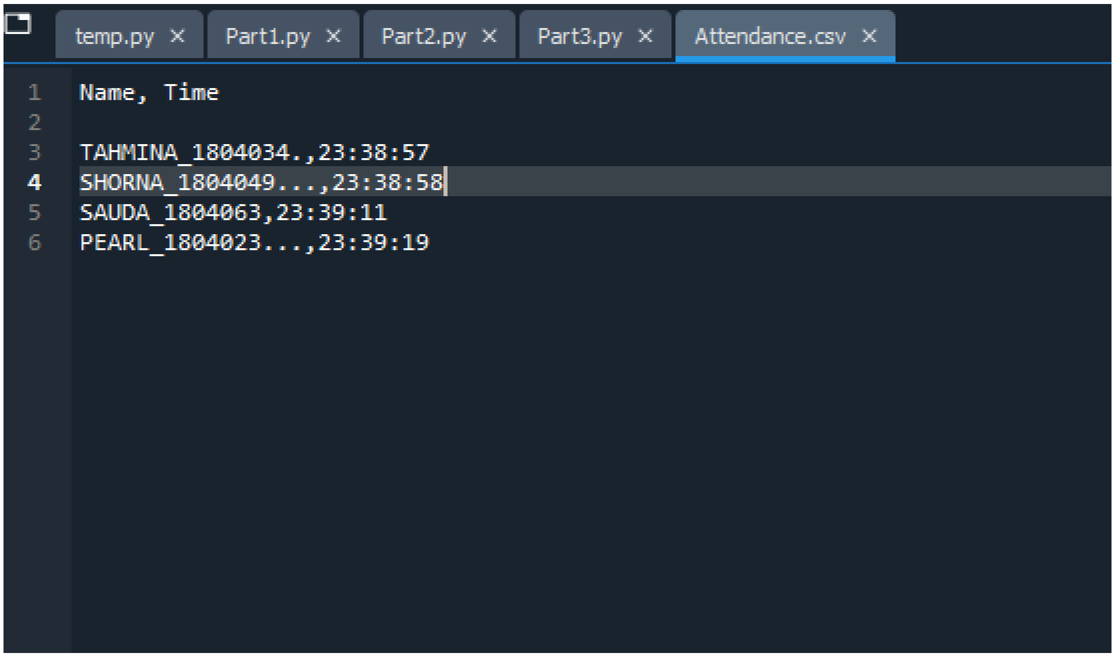


```
with open('Attendance.csv','r+') as f:
    myDataList = f.readlines()
    nameList = []
    for line in myDataList:
        entry = line.split(',')
        nameList.append(entry[0])
    if name not in nameList:
```

Figure 10 : Desired attendance

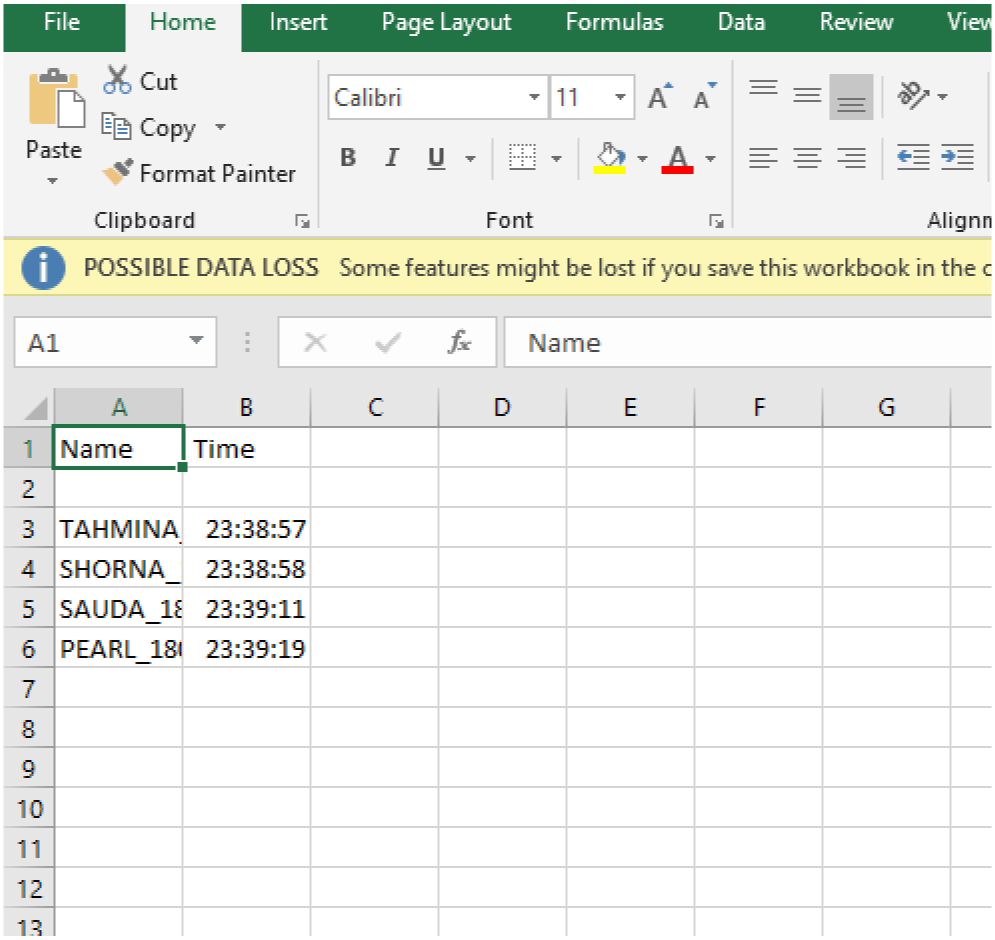
9.6. Record of attendance

Here is the result of our attendance.



```
Name, Time

TAHMINA_1804034.,23:38:57
SHORNA_1804049...,23:38:58
SAUDA_1804063,23:39:11
PEARL_1804023...,23:39:19
```

Figure 11 : Recording our attendance



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the c

A1 | Name

	A	B	C	D	E	F	G
1	Name	Time					
2							
3	TAHMINA_	23:38:57					
4	SHORNA_	23:38:58					
5	SAUDA_18	23:39:11					
6	PEARL_18(	23:39:19					
7							
8							
9							
10							
11							
12							
13							

Figure 12 : Recording our attendance in file

10. Conclusion :

The objective of the class attendance system is to automate the time consuming and error prone attendance system. There are always limitations of every system. One can only have a fixed number of students and provide less freedom to have an intra class attendance system. This means the attendance system for one class can’t be used for the attendance system of another class. One must change programming to do this.The Project experience was tremendous because we learned how a problem can be simplified into smaller tasks and can be achieved successfully. It is the reason why we are able to complete our project.