PROJECT REPORT

on

Face Recognition Based Attendance System

(CSE V Semester Mini project)

2021-2022

Submitted to:

Mr. UMANNG GARG

(CC-CSE-A-V-Sem)

Guided by:

Ms. Preeti Choudhary

(Resource Person)

Submitted by:

Mr. HIMANSHU CHAND

Roll. No.: 1918375

CSE-A-V-Sem

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

GRAPHIC ERA HILL UNVERSITY, DEHRADUN

FACE RECOGNITION BASED ATTENDANCE SYSTEM

ABSTRACT

The main purpose of this project is to build a face recognition-based attendance monitoring system for educational institution to enhance and upgrade the current attendance system into more efficient and effective as compared to before. The current old system has a lot of ambiguity that caused inaccurate and inefficient of attendance taking. Many problems arise when the authority is unable to enforce the regulation that exist in the old system. The technology working behind will be the face recognition system. The human face is one of the natural traits that can uniquely identify an individual. Therefore, it is used to trace identity as the possibilities for a face to deviate or being duplicated is low. In this project, face databases will be created to pump data into the recognizer algorithm. Then, during the attendance taking session, faces will be compared against the database to seek for identity. When an individual is identified, its attendance will be taken down automatically saving necessary information into a excel sheet. At the end of the day, the excel sheet containing attendance information regarding all individuals are mailed to the respective faculty.

INTRODUCTION

The main objective of this project is to develop face recognition based automated student attendance system. In order to achieve better performance, the test images and training images of this proposed approach are limited to frontal and upright facial images that consist of a single face only. The test images and training images have to be captured by using the same device to ensure no quality difference. In addition, the students have to register in the database to be recognized. The enrolment can be done on the spot through the user-friendly interface.

The human face is a unique representation of individual identity. Thus, face recognition is defined as a biometric method in which identification of an individual is performed by comparing real-time capture image with stored images in the database of that person. Nowadays, face recognition system is prevalent due to its simplicity and awesome performance. For instance, airport protection systems and FBI use face recognition for criminal investigations by tracking suspects, missing children and drug activities. Apart from that, Facebook which is a popular social networking website implement face recognition to allow the users to tag their friends in the photo for entertainment purposes. Furthermore, Intel Company allows the users to use face recognition to get access to their online account. Apple allows the users to unlock their mobile phone, iPhone X by using face recognition.

MOTIVATION

The motivation behind this project is to simplify the means by which attendance is taken during lectures and how much time it takes. The use of ID cards or manually calling out attendance and writing it down on sheets is not productive and efficient. This system will detect the number of faces on the class and will also identify them from the store database. With the face detection and recognition system in place, it will be easy to tell if a student is actually present in the classroom or not.

WORKING AND METHODOLOGY

Basic Face Matching :-

First, we get the location of where exactly the face is in the image using face_location() method(which gets the outline of the face) on the RGB image. Then face encodings(markings of eyes, nose, mouth, jaws which remain the same for different images of the same person) are taken using face_encodings() function which returns a list containing 128 measurements. Both these two steps are followed for the original and test image. Then a comparison between these two returned lists is done by the function compare_faces() which returns a list of boolean values(True or False). The face distance function gets the value of that by how much the two images differ. The lower the distance the better the matching and vice versa.

```
import cv2

import face_recognition as fr

imgAng = fr.load_image_file('andrew_ng.jpg')

Test = fr.load_image_file('ian_godfellow.jpg')

fLoc = fr.face_locations(imgAng)[0]

encodeAng = fr.face_encodings(imgAng)[0]

fLocTest = fr.face_locations(Test)[0]

encTest = fr.face_encodings(Test)[0]

result = fr.compare_faces([encodeAng],encTest)

faceDist = fr.face_distance([encodeAng],encTest)

print(result,faceDist)
```

```
[True] [0.36569372]     [False] [0.6898802]
```

Building Face Attendance System

Now we are ready to build a realtime face attendance system wherein webcam captured frames will be matched against the existing database images and if the match is found then it’ll store it in a CSV file called ‘Attendance’ along with name and time of capture. Only once the file will store the matched image’s details, if the same image is received again then it’ll not update.

Path setting to the directory containing the image database. Read each image and the images array. Append the filenames into a list called Names and remove the extension.

```
pathlib = 'ImagesAttendance'

images = []

Names = []

myList = os.listdir(pathlib)

print(myList)

for cl in myList:

    currImg = cv2.imread(f'{pathlib}/{cl}')

    images.append(currImg)

    Names.append(os.path.splitext(cl)[0])

print(Names)
```

Finding face encodings of images in the database and keeping them in a list to use later with incoming frames.

```
def DbEncodings(images):
    encList = []
    for image in images:
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        enc = fr.face_encodings(image)[0]
        encList.append(enc)
    return encodeList
```

Capturing video frames

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

Iterating through frames

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

The same process is followed by the first detection face location then getting the face encoding values.

```
    facesInFrame = fr.face_locations(image)
    encodesInFrame = fr.face_encodings(image,facesInFrame)
```

Now the incoming images are tested against the previously-stored encodings. Then the face distance is also computed. Lastly, we call the Attendance function along with the person name who is identified.

```
for encodeFace,faceLoc in zip(encodesInFrame,facesInFrame):
        matchList = fr.compare_faces(encodeKnown,encFace)
        faceDist = fr.face_distance(encodeKnown,encFace)
        match = np.argmin(faceDist)
        if matchList[match]:
            name = Names[match].upper()
            Attendance(name)
```

Reading from attendance file, Storing data(Name and Time of entry) if previously not stored.

```
def Attendance(name):
    with open('Attendance_Register.csv','r+') as f:
        DataList = f.readlines()
        names = []
        for data in DataList:
            ent = data.split(',')
            names.append(ent[0])
        if name not in names:
            curr = datetime.now()
            dt = curr.strftime('%H:%M:%S')
            f.writelines(f'\n{name},{dt}')
```

```
encodeKnown = DbEncodings(images)

print('Encoding Complete')
```

OUTPUT

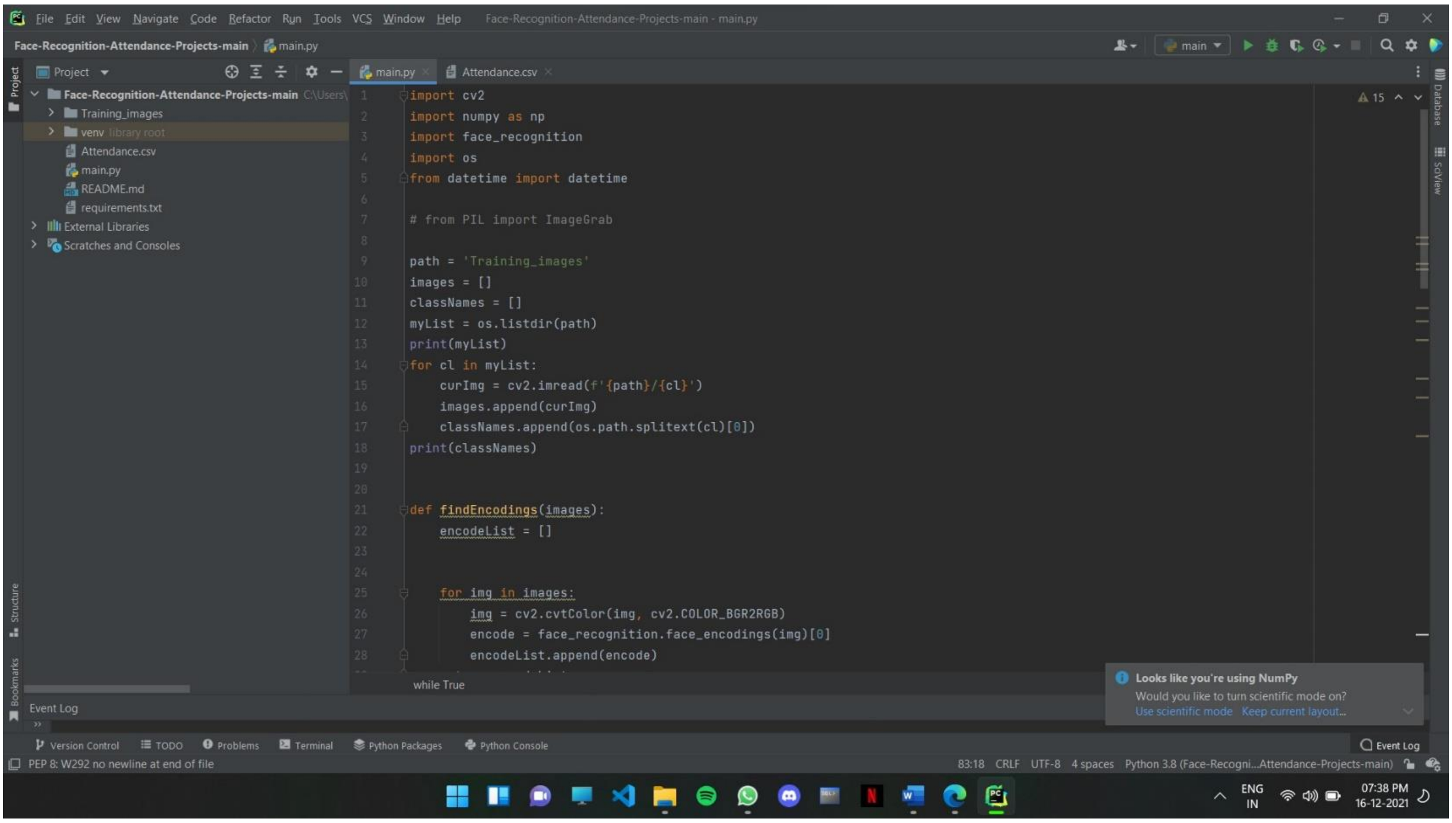
```
['Himanshu', 'Sarthak']

Encoding Complete
```

Attendence_Register.csv

Name	Time
HIMANSHU	15:47:33
SARTHAK	15:55:12

SCREENSHOTS OF THE PROJECT


File Edit View Navigate Code Refactor Run Tools VCS Window Help Face-Recognition-Attendance-Projects-main - main.py
Face-Recognition-Attendance-Projects-main
main.py
Attendance.csv
Project
Face-Recognition-Attendance-Projects-main C:\Users\
Training_images
venv library root
Attendance.csv
main.py
README.md
requirements.txt
External Libraries
Scratches and Consoles
Event Log
Version Control
TODO
Problems
Terminal
Python Packages
Python Console
83:18 CRLF UTF-8 4 spaces Python 3.8 (Face-Recogni...Attendance-Projects-main)
ENG IN
07:38 PM
16-12-2021

```python
import cv2
import numpy as np
import face_recognition
import os
from datetime import datetime

# from PIL import ImageGrab

path = 'Training_images'
images = []
classNames = []
myList = os.listdir(path)
print(myList)
for cl in myList:
    curImg = cv2.imread(f'{path}/{cl}')
    images.append(curImg)
    classNames.append(os.path.splitext(cl)[0])
print(classNames)


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)
```

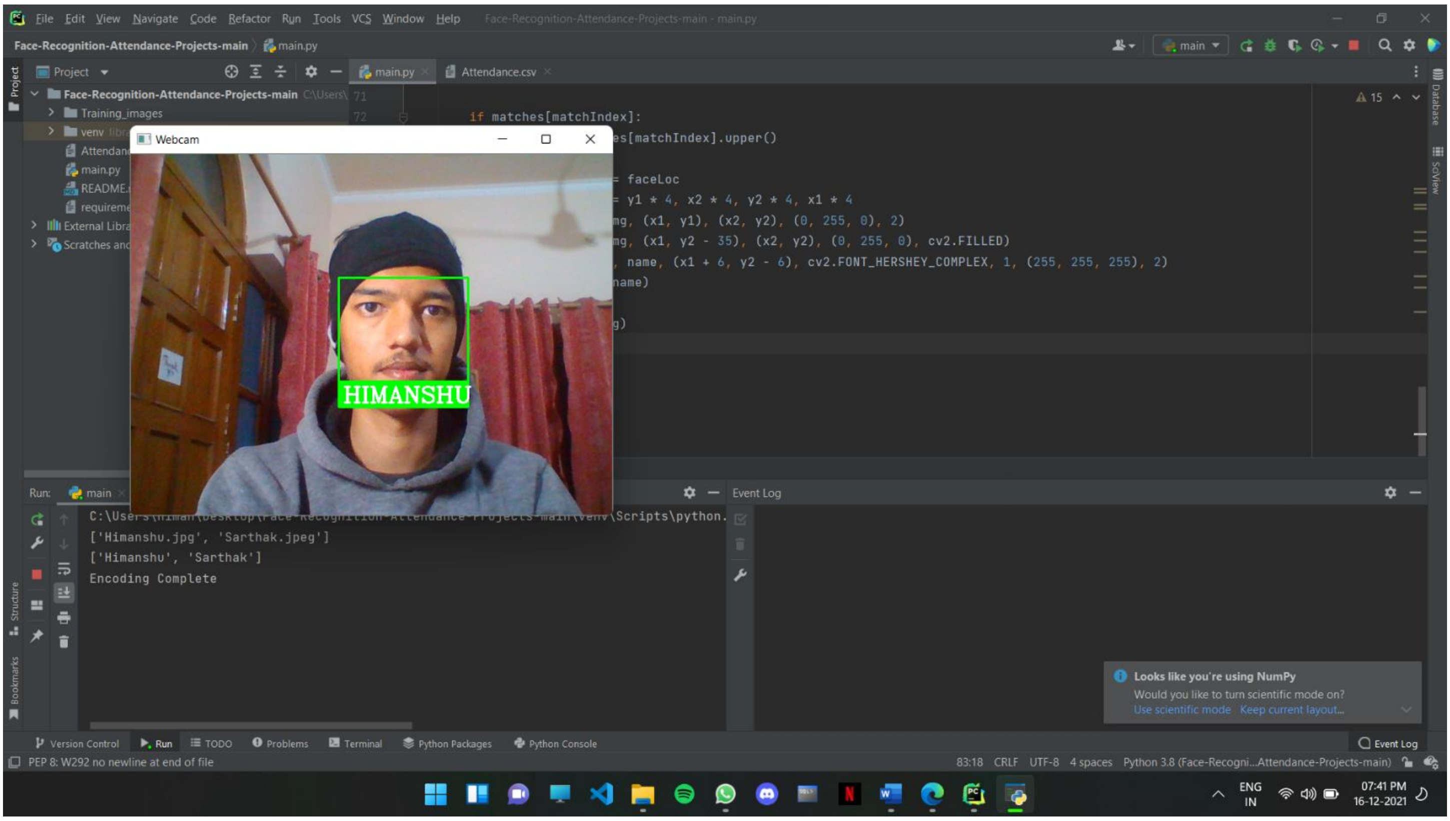
while True

Looks like you're using NumPy

Would you like to turn scientific mode on?

Use scientific mode Keep current layout...

PEP 8: W292 no newline at end of file


File Edit View Navigate Code Refactor Run Tools VCS Window Help Face-Recognition-Attendance-Projects-main - main.py
Face-Recognition-Attendance-Projects-main
main.py
Attendance.csv
Webcam
HIMANSHU
Run: main
Event Log
Version Control
Run
TODO
Problems
Terminal
Python Packages
Python Console
83:18 CRLF UTF-8 4 spaces Python 3.8 (Face-Recogni...Attendance-Projects-main)
ENG IN
07:41 PM
16-12-2021

```python
        if matches[matchIndex]:
es[matchIndex].upper()

= faceLoc
= y1 * 4, x2 * 4, y2 * 4, x1 * 4
mg, (x1, y1), (x2, y2), (0, 255, 0), 2)
mg, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)
, name, (x1 + 6, y2 - 6), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)
name)

g)
```

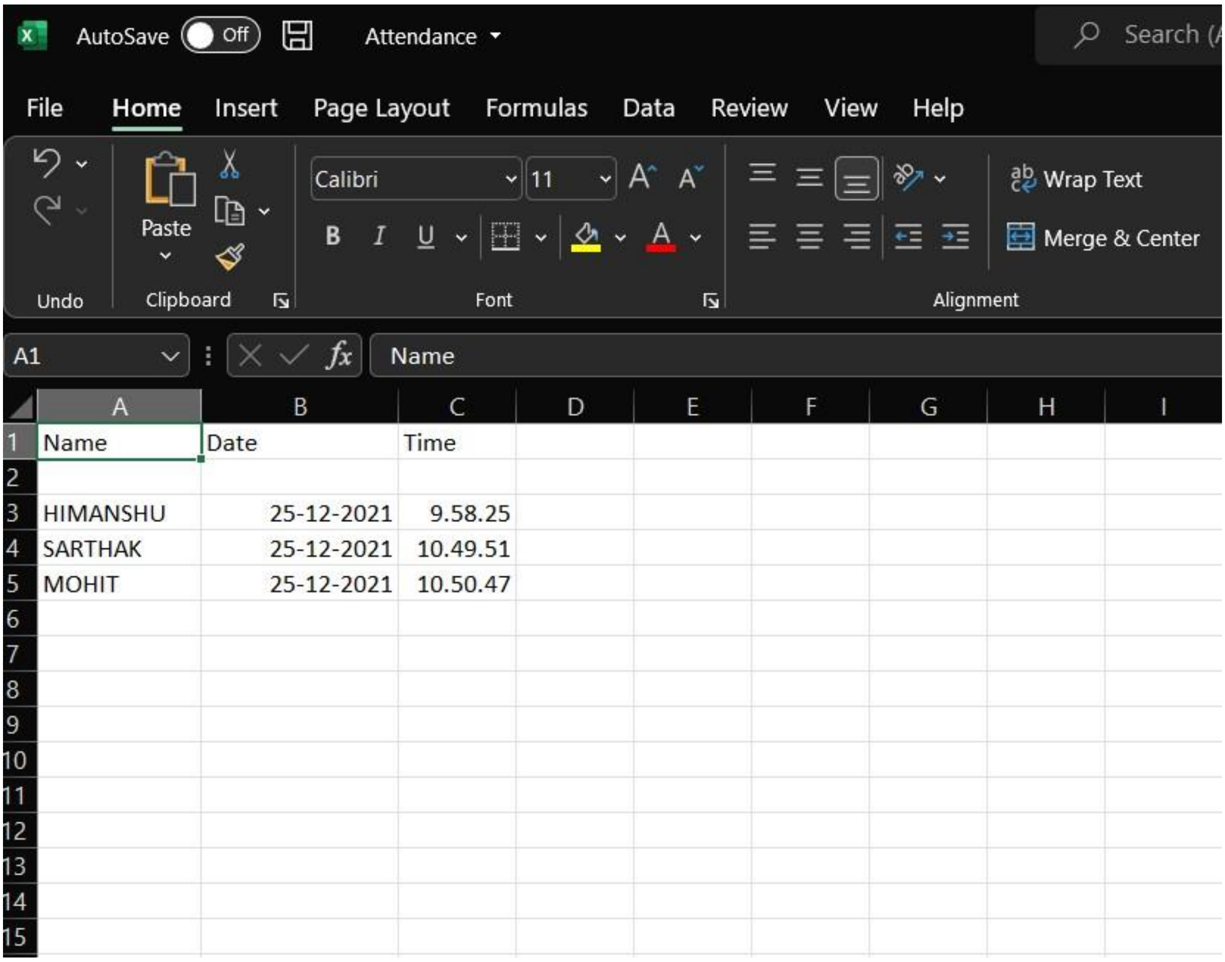
```
C:\Users\himan\Desktop\Face-Recognition-Attendance-Projects-main\venv\Scripts\python.
['Himanshu.jpg', 'Sarthak.jpeg']
['Himanshu', 'Sarthak']
Encoding Complete
```

Looks like you're using NumPy

Would you like to turn scientific mode on?

Use scientific mode Keep current layout...

PEP 8: W292 no newline at end of file



A1 fx Name

	A	B	C
1	Name	Date	Time
2			
3	HIMANSHU	25-12-2021	9.58.25
4	SARTHAK	25-12-2021	10.49.51
5	MOHIT	25-12-2021	10.50.47

REFERENCES

1.-This project is inspired by Adam Geitgey

Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning | by Adam Geitgey | Medium

2.GeeksForGeeks

Opencv Python program for Face Detection - GeeksforGeeks

3.Udemy Machine learning courses

Machine Learning A-Z (Python & R in Data Science Course) | Udemy