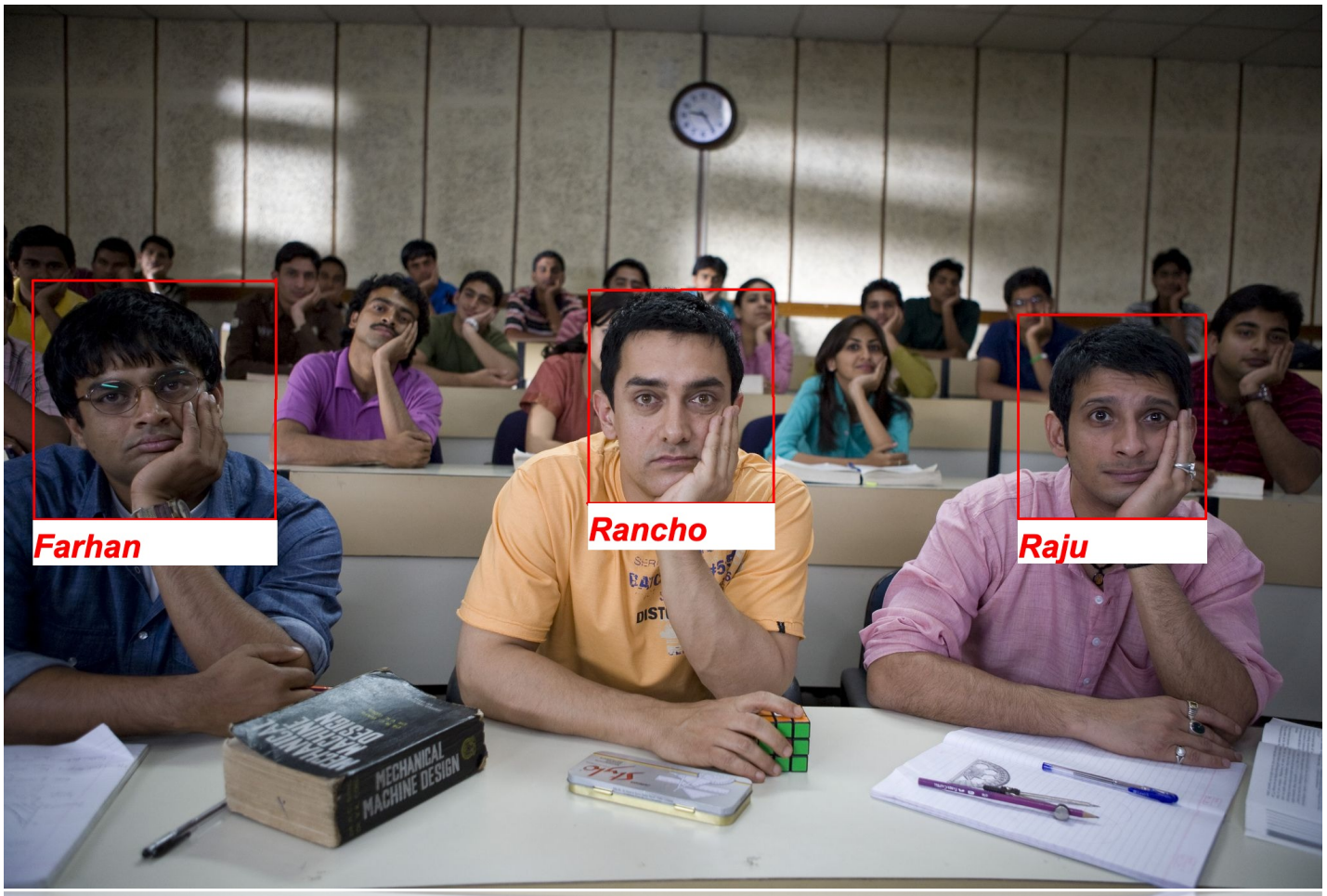
Face Recognition Attendance System For Zoom



ABHIJEET GUPTA AND SHRUTI SINGH

Spring 2021

Index

Content

INTRODUCTION

In present academic system, Zoom class attendance of students' plays a significant role in performance assessment and quality monitoring. The conventional methods practiced in most of the institutions are by calling names or signing on papers, which is highly time-consuming and insecure. Therefore we presents the automatic attendance management system for convenience and data reliability. The system is developed by the integration of ubiquitous components to make a portable device for managing the students' attendance using Face Recognition technology.

PROJECT DESCRIPTION

The project idea of using facial recognition in zoom call or any other web meeting application was inspired by this deep learning course that we took this semester, i.e in Spring 2021. We were familiar with the use of neural networks and training models as well as image classification, as we have done that in our previous assignments during the course, the challenge was to record the attendance and marking it in the excel sheet. So, we researched about the best facial recognition library and came up with the following libraries:

- Cmake
- Dlib
- Facial_recognition
- Opencv

The facial recognition library just needs one image to detect the face, so we fed in one image as our dataset, I.e the front face. Further, the facial recognition has face encodings pre-defined so we extracted our image encodings, and appended it into a list of our dataset. Further we needed to record the screen to detect faces on the zoom application, so to record screen from the pillow library we used ImageGrab library. Since, the image fed in real

time were large we reduced the size and then gave it to encodings list. To test our accuracy, we also incorporated the compare feature, where we can compare our test image and it return the distance value from the given training image. The smaller the value the more accurate our test image is, and vice-versa. Then we created the bounding box for the face to be detected and to display the name of the face recognized by the system. For that we used the face_location function. To save it into an excel, we did it with the help of opencv library where we defined a method where it writes the names of the images recognized line by line with real time date stamp. For this we also used the date time library.

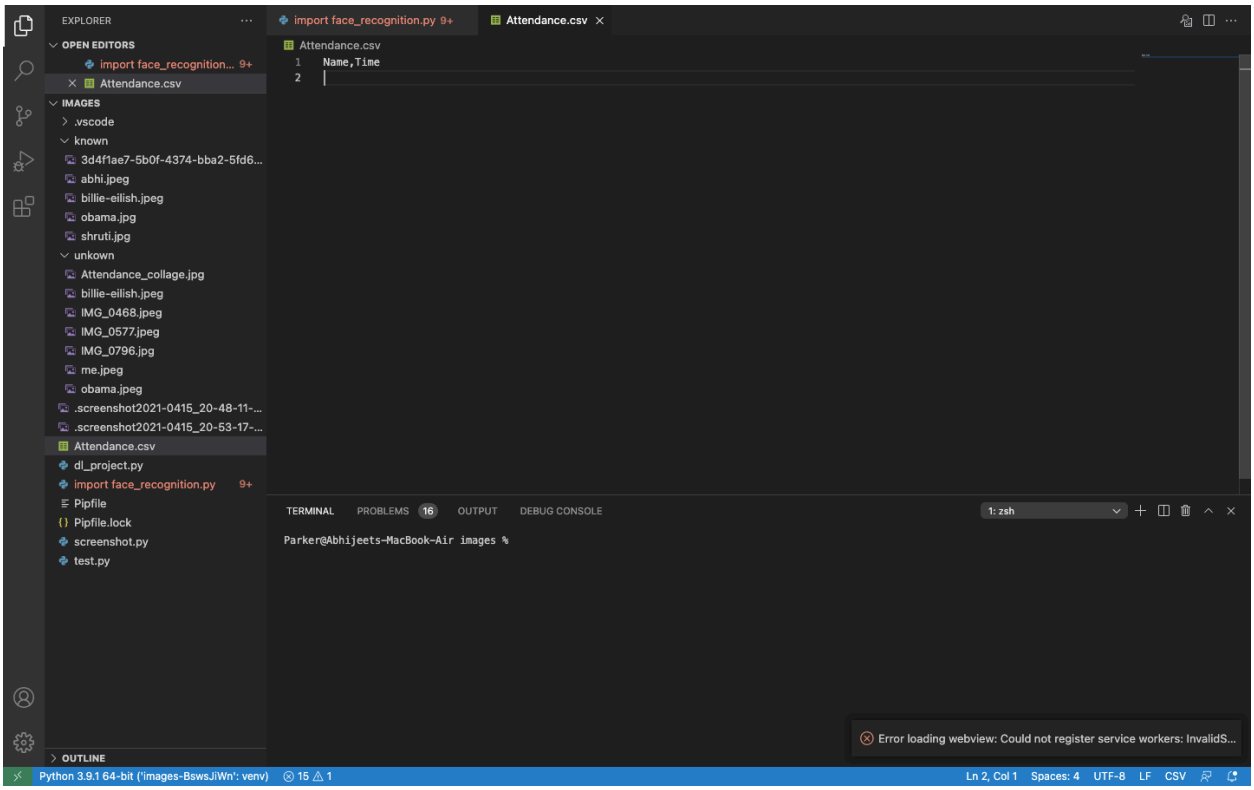
PURPOSE OF THE PROJECT

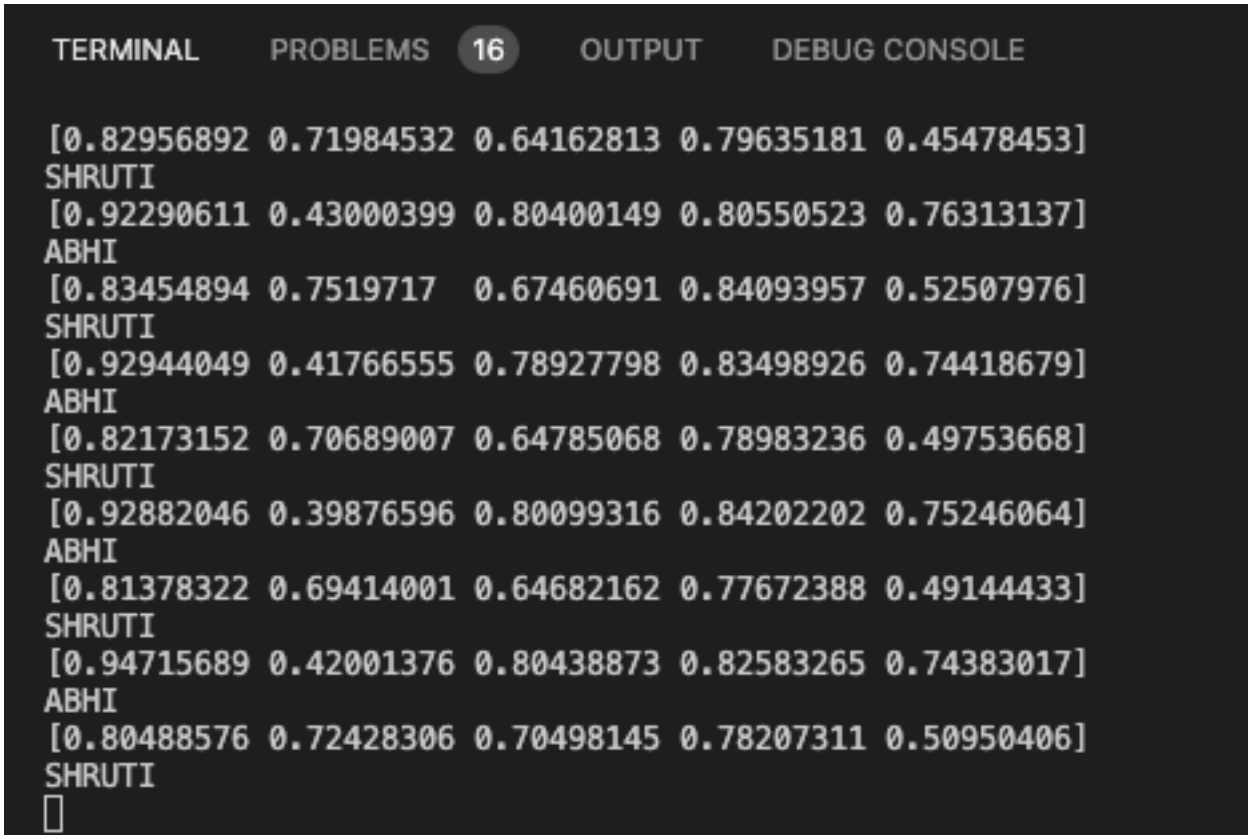
The year 2020 has been hard on all of us. And with that, it has also taught a lot of us. We have learned from our mistakes and trained ourselves to live a safe and healthy life. Keeping that in mind, we dived into 2021 with a more conscious mindset. But the pandemic is still out there and with that comes daily constraints in our lives. While most of us have resumed our lives in a normal manner, there is still a huge population that is still juggling their days behind computer desks. Be it a large corporation managing its employees, or an institution handling its teacher and students. And although we have made much progress sitting behind those desks in the past year, there are still challenges that need some consideration. Being students, we are in similar shoes like the rest of them and that made us think of this project, for ‘ Attendance System using Facial Recognition’. There are several ways to mark attendance but at present, a computer-based student attendance checking system is the most convenient of all. Our objective, in brief, is to facilitate a smooth day-to-day life routine, and this implementation will eradicate the possibility of human error, proxy attendance, and promote time-saving.

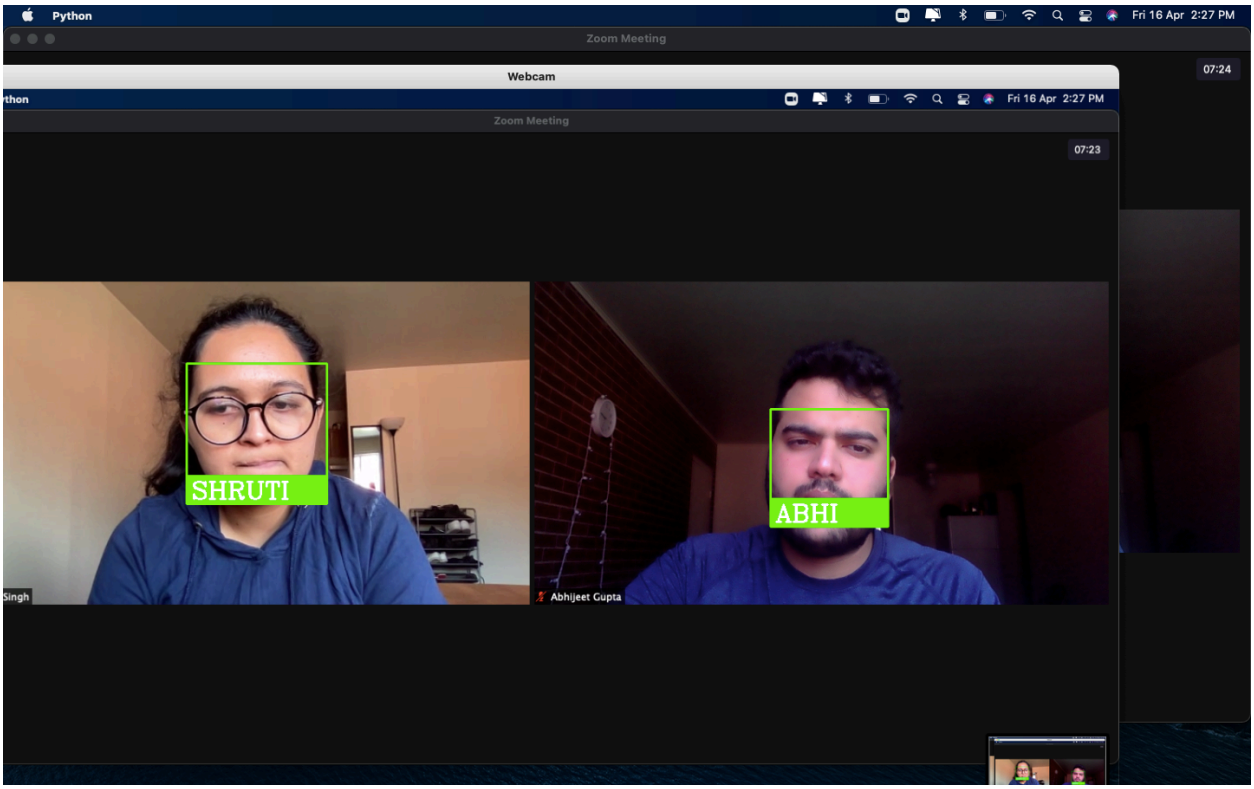
SOURCE OF THE DATA

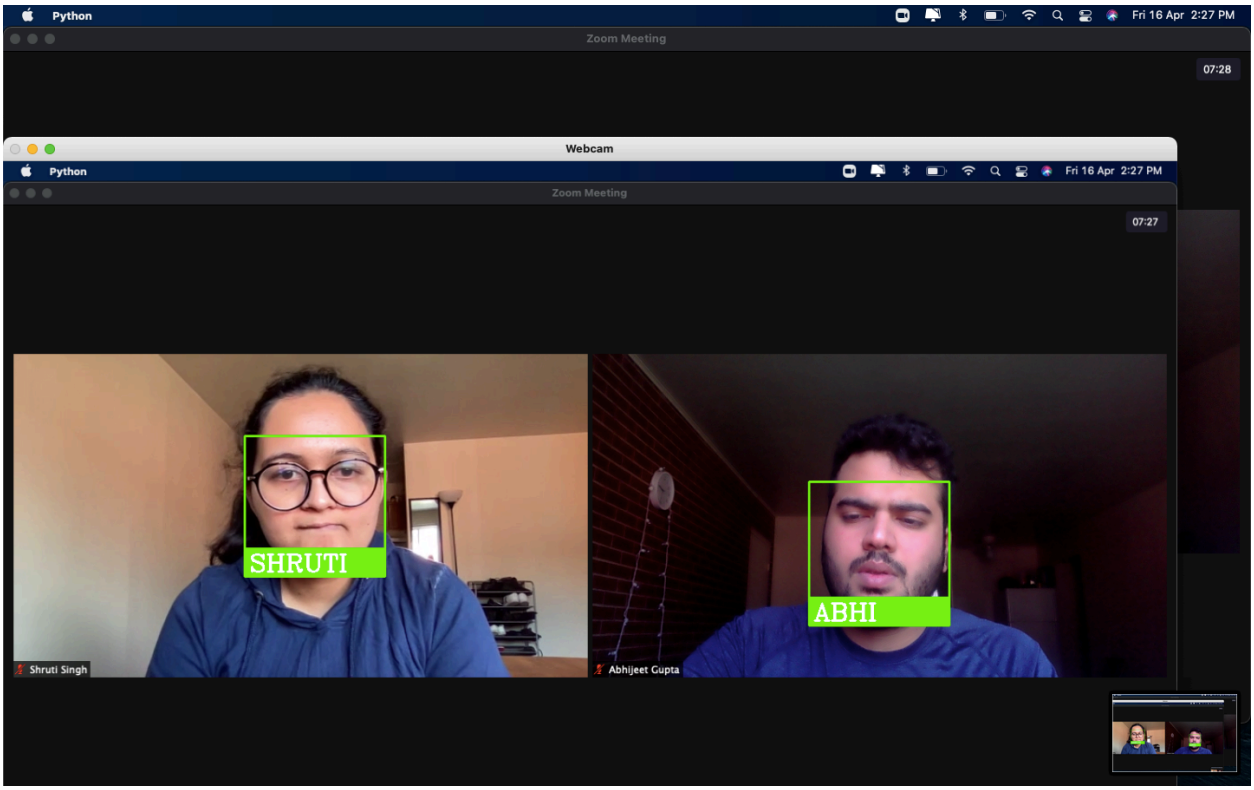
Dlib is morden C++ toolkit contains machine learning algorithm and tools for creating software in C++ to solve the real world problems. In real world we have the limited dataset that's why dlib was the best option we have as it need just one image of the person for the image recognition and its accuracy is also very high. We will be getting one image of the person from the porches which teachers have access to it.

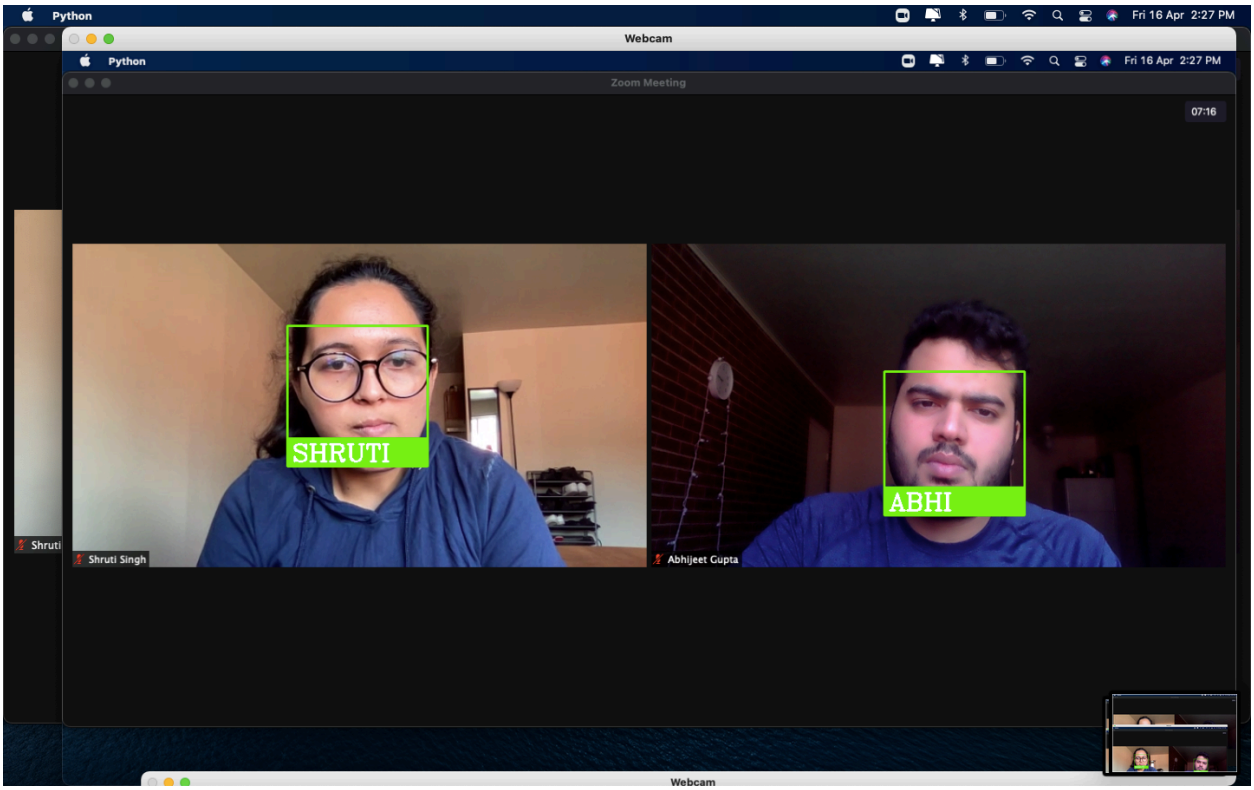
RESULTS

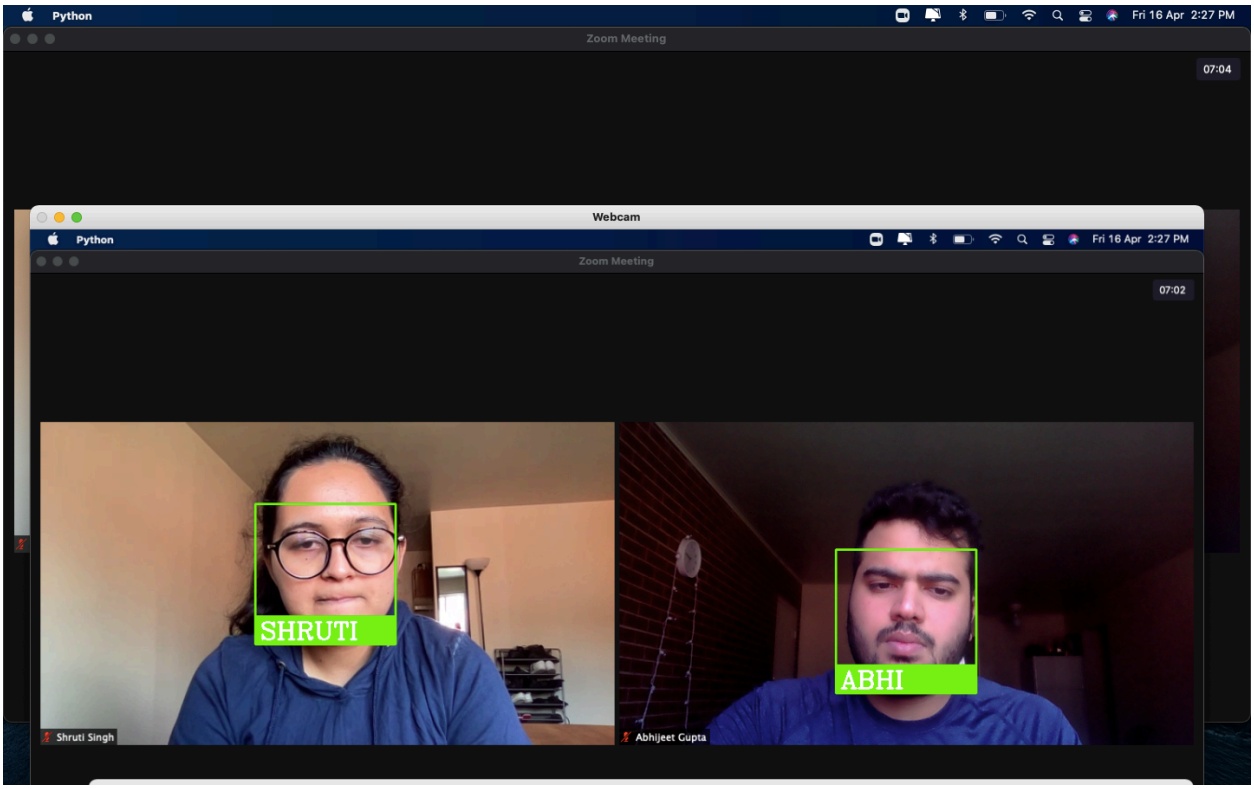

EXPLORER
OPEN EDITORS
import face_recognition... 9+
Attendance.csv
IMAGES
.vscode
known
3d4f1ae7-5b0f-4374-bba2-5fd6...
abhi.jpeg
billie-eilish.jpeg
obama.jpg
shruti.jpg
unkown
Attendance_collage.jpg
billie-eilish.jpeg
IMG_0468.jpeg
IMG_0577.jpeg
IMG_0796.jpg
me.jpeg
obama.jpeg
.screenshot2021-0415_20-48-11-...
.screenshot2021-0415_20-53-17-...
Attendance.csv
dl_project.py
import face_recognition.py 9+
Pipfile
Pipfile.lock
screenshot.py
test.py
OUTLINE
import face_recognition.py 9+
Attendance.csv
Attendance.csv
1 Name,Time
2
TERMINAL
PROBLEMS 16
OUTPUT
DEBUG CONSOLE
1: zsh
Parker@Abhijeets-MacBook-Air images %
Error loading webview: Could not register service workers: InvalidS...
Python 3.9.1 64-bit ('images-BswsJiWn': venv)
15 1
Ln 2, Col 1
Spaces: 4
UTF-8
LF
CSV

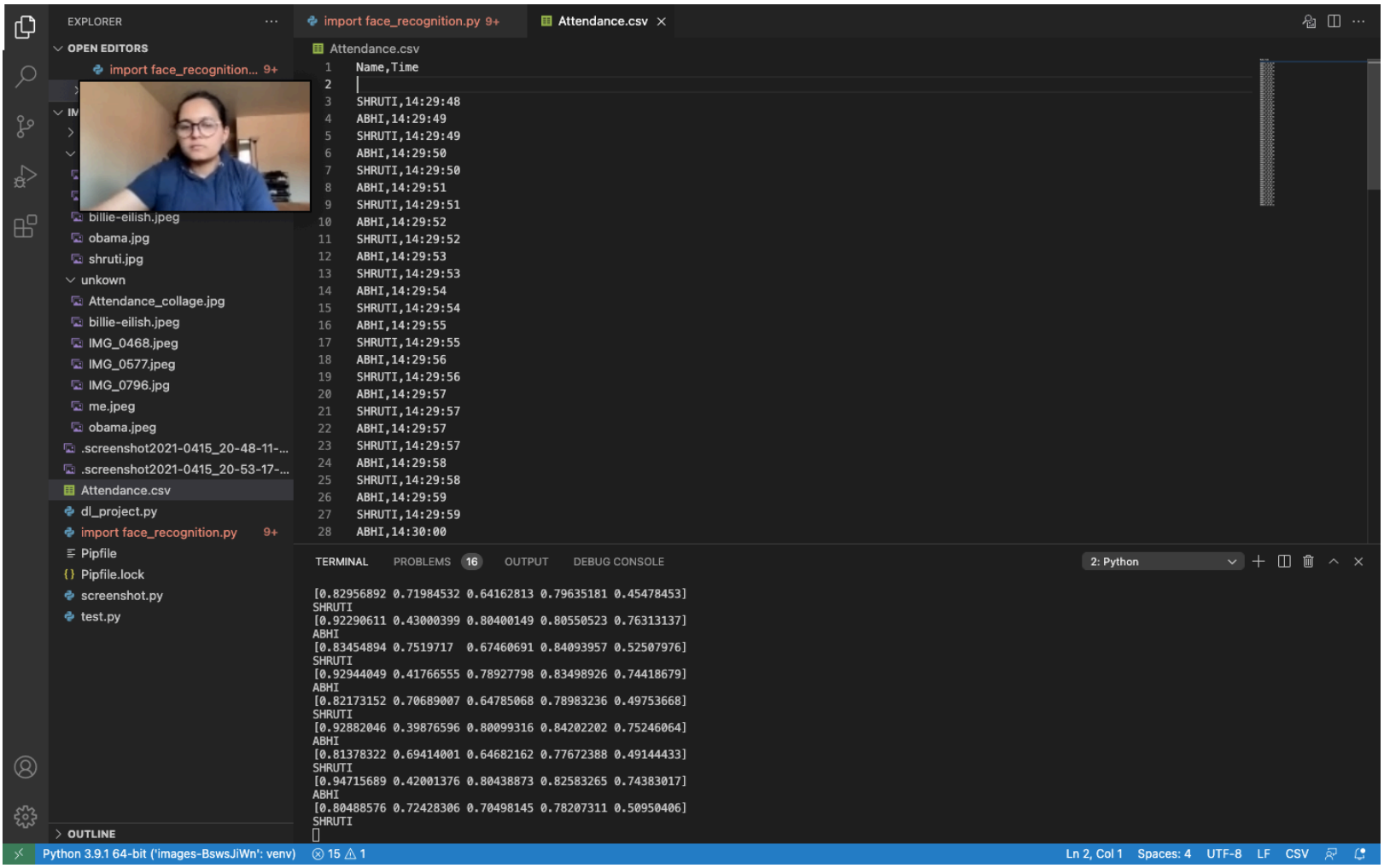

TERMINAL
PROBLEMS 16
OUTPUT
DEBUG CONSOLE
[0.82956892 0.71984532 0.64162813 0.79635181 0.45478453]
SHRUTI
[0.92290611 0.43000399 0.80400149 0.80550523 0.76313137]
ABHI
[0.83454894 0.7519717 0.67460691 0.84093957 0.52507976]
SHRUTI
[0.92944049 0.41766555 0.78927798 0.83498926 0.74418679]
ABHI
[0.82173152 0.70689007 0.64785068 0.78983236 0.49753668]
SHRUTI
[0.92882046 0.39876596 0.80099316 0.84202202 0.75246064]
ABHI
[0.81378322 0.69414001 0.64682162 0.77672388 0.49144433]
SHRUTI
[0.94715689 0.42001376 0.80438873 0.82583265 0.74383017]
ABHI
[0.80488576 0.72428306 0.70498145 0.78207311 0.50950406]
SHRUTI


Python
Fri 16 Apr 2:27 PM
Zoom Meeting
Webcam
07:24
Zoom Meeting
07:23
SHRUTI
ABHI
Abhijeet Gupta


Python
Fri 16 Apr 2:27 PM
Zoom Meeting
07:28
Webcam
Python
Fri 16 Apr 2:27 PM
Zoom Meeting
07:27
SHRUTI
ABHI
Shruti Singh
Abhijeet Gupta


Python
Fri 16 Apr 2:27 PM
Webcam
Python
Fri 16 Apr 2:27 PM
Zoom Meeting
07:16
SHRUTI
ABHI
Shruti Singh
Abhijeet Gupta
Webcam


Python
Fri 16 Apr 2:27 PM
Zoom Meeting
07:04
Webcam
Python
Fri 16 Apr 2:27 PM
Zoom Meeting
07:02
SHRUTI
ABHI
Shruti Singh
Abhijeet Gupta


EXPLORER
OPEN EDITORS
import face_recognition... 9+
billie-eilish.jpeg
obama.jpg
shruti.jpg
unkown
Attendance_collage.jpg
billie-eilish.jpeg
IMG_0468.jpeg
IMG_0577.jpeg
IMG_0796.jpg
me.jpeg
obama.jpeg
.screenshot2021-0415_20-48-11-...
.screenshot2021-0415_20-53-17-...
Attendance.csv
dl_project.py
import face_recognition.py 9+
Pipfile
Pipfile.lock
screenshot.py
test.py
OUTLINE
import face_recognition.py 9+
Attendance.csv
Attendance.csv
1 Name,Time
2
3 SHRUTI,14:29:48
4 ABHI,14:29:49
5 SHRUTI,14:29:49
6 ABHI,14:29:50
7 SHRUTI,14:29:50
8 ABHI,14:29:51
9 SHRUTI,14:29:51
10 ABHI,14:29:52
11 SHRUTI,14:29:52
12 ABHI,14:29:53
13 SHRUTI,14:29:53
14 ABHI,14:29:54
15 SHRUTI,14:29:54
16 ABHI,14:29:55
17 SHRUTI,14:29:55
18 ABHI,14:29:56
19 SHRUTI,14:29:56
20 ABHI,14:29:57
21 SHRUTI,14:29:57
22 ABHI,14:29:57
23 SHRUTI,14:29:57
24 ABHI,14:29:58
25 SHRUTI,14:29:58
26 ABHI,14:29:59
27 SHRUTI,14:29:59
28 ABHI,14:30:00
TERMINAL
PROBLEMS 16
OUTPUT
DEBUG CONSOLE
2: Python
[0.82956892 0.71984532 0.64162813 0.79635181 0.45478453]
SHRUTI
[0.92290611 0.43000399 0.80400149 0.80550523 0.76313137]
ABHI
[0.83454894 0.7519717 0.67460691 0.84093957 0.52507976]
SHRUTI
[0.92944049 0.41766555 0.78927798 0.83498926 0.74418679]
ABHI
[0.82173152 0.70689007 0.64785068 0.78983236 0.49753668]
SHRUTI
[0.92882046 0.39876596 0.80099316 0.84202202 0.75246064]
ABHI
[0.81378322 0.69414001 0.64682162 0.77672388 0.49144433]
SHRUTI
[0.94715689 0.42001376 0.80438873 0.82583265 0.74383017]
ABHI
[0.80488576 0.72428306 0.70498145 0.78207311 0.50950406]
SHRUTI
Python 3.9.1 64-bit ('images-BswsJiWn': venv)
15 1
Ln 2, Col 1
Spaces: 4
UTF-8
LF
CSV

CONCLUSION AND FUTURE WORKS

Even though we faced lot of challenges while making the project our program was able to recognizes the faces through the data set we provided. It only need one front face image of the person to detect and extract features to be able to recognize the same same person in screen recording. To make sure we detected the same person accurately we built the bounding box around the detected face and displayed the name of the person at the bottom. Simultaneously, as a program detected the faces it marked the attendance with name and time on a .csv file.

We also compared the faces detected by our program and displayed the face distance that is the similarity between the two faces the smaller the number the more similar the faces are and vice versa this helped us to ensure that we detected the same person.

The future scope of the project is to integrate it with the zoom so that the professor and student will have the access to see the attendance.

COPY OF THE PROGRAM WITH DOCUMENTS

```
import face_recognition
import cv2
import numpy as np
import os
from PIL import ImageGrab
from datetime import datetime, timedelta


path = 'test_img'
img = []
classNames = []
myList = os.listdir(path)
print(myList)
for cl in myList:

    curImg = cv2.imread(f'{path}/{cl}')
    img.append(curImg)
    classNames.append(os.path.splitext(cl)[0])
print(classNames)

def findEncodings(img):
    encodeList = []
    for img in img:
        img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList


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()
            five = datetime.now() + timedelta(minutes=5)
            dtString = five.strftime('%H:%M:%S')
            f.writelines(f'\n{name},{dtString}')
```

```
encodeListKnown = findEncodings(img)
print('Encoding Complete')



def captureScreen(bbox=None):
    capScr = np.array(ImageGrab.grab(bbox))
    capScr = cv2.cvtColor(capScr, cv2.COLOR_RGB2BGR)
    return capScr

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

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

    for encodeFace,faceLoc in zip(encodeCurFrame,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 = classNames[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)
```

THANKS A LOT FOR GIVING US THE OPPORTUNITY TO DO THIS PROJECT WE HAVE OVERCAME MANY CHALLENGES DURING THE PROJECT AND LEARNED A LOT.

SHRUTI SINGH & ABHIJEET GUPTA

"A BABY LEARNS TO CRAWL WALK AND THEN RUN. WE ARE IN CRAWLING STAGE WHEN IT COMES TO APPLYING DEEP LEARNING" - DAVE WATERS