FACE RECOGNITION ATTENDANCE SYSTEM

UEM
UNIVERSITY OF ENGINEERING & MANAGEMENT

UNDER THE GUIDANCE OF

[Prof. Srilekha Mukherjee
&
Prof. Anirban Chakraborty]

Group - 15

[Department : Computer Science & Information Technology
&
Computer Science & Technology]

GROUP MEMBERS

Shivsundar Bera

Arindam Roy

Soumyajit Patra

Hriteesha Pramanik

Sayandeep Mondal

Swarnadeep Roy

INTRODUCTION

- An attendance system plays a very important role in any organization. Keeping in view the recent development in software and technological fields, a face recognition based attendance system has been planned. This attendance system keeps the face images of all students in a database. During attendance a student is required to present his/her face in front of camera. The system immediately recognizes the image with the image kept in the database and update the attendance register with name, time and date.

DEPLOYMENT OF EXTERNAL LIBRARIES

- Two important freely available external libraries of software are to be imported into Python library. This Python environment will later to be used during development of code. The first one is face-recognition library. Face-recognition code is available in Github. By executing setup python script, face-recognition library can be imported to Python library. The second one is OpenCV/CV2. OpenCV can be directly imported into Python library by using ‘pip’.

LIBRARIES

Most important libraries

1. OpenCV
2. Face-Recognition

These two libraries are used for encoding the database/folder images and camera images. OpenCV/CV2 library encodes the image and returns BGR format of image. But the face-recognition library can read only RGB image. So we must convert the image encoding format from BGR to RGB.

Also use in-built python libraries

1. OS library
2.Numpy library

Import libraries – cv2, numpy, face_recognition, os, datetime

These python libraries are imported at the starting of the code .

```
imageNames[] = Find all the image names with path kept in the ‘images’
directory;
int noOfImages = Get the number of imageNames;

loop (noOfImages times: i) {
   images[i] = Read the binary images into memory;
   personNames[i] = Extract the person name from the path of imageNames;
}

Display the personNames[];
```

→

This code is for listing all image names, with path, into the memory and also for reading binary images.

```
loop (noOfImages times: i) {
    img[i] = Change color format (BGR->RGB) for images[i];
    encodeList[i] = Encode img[i] with the help of face_recognition lib;
}
```

This is the most important area of the code because this code encodes each image in 128 different points.

```
cap = CapTureVideo(Laptop Camera)
```

Capturing the image by camera with the help of OpenCV library API. ‘1’ indicates external camera and ‘0’ indicates laptop camera.

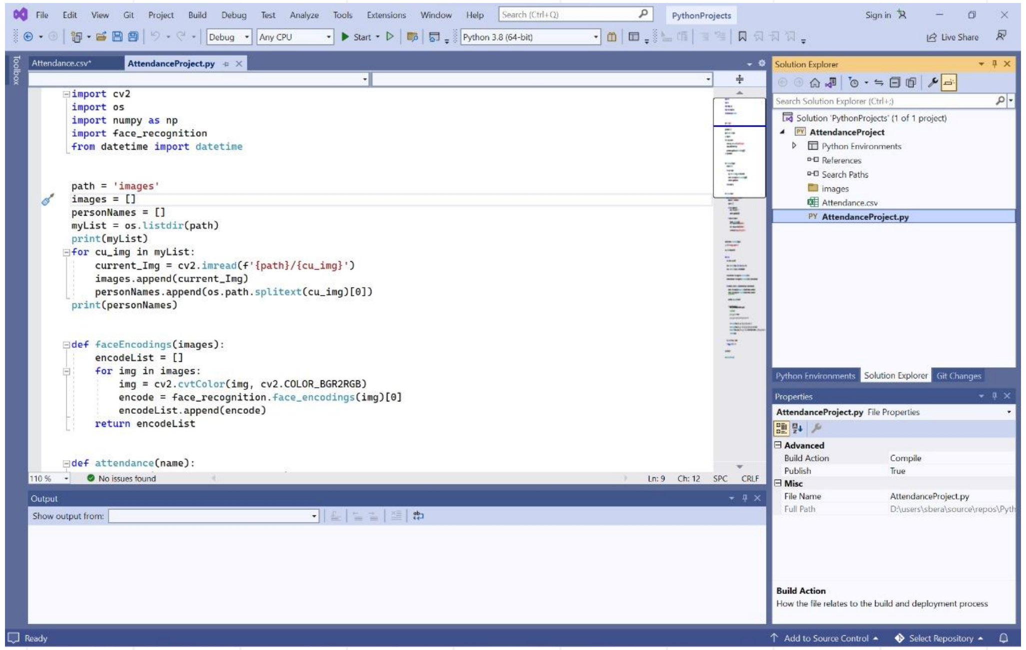
```
loop (true) {
  frame = Read cap into a frame;
  face = Change the color (BGR->RGB)format of frame;
  encodeFace = Encode face with face-recognition lib;
  matches = compare(encodeFace, encodeList);
  If (matches == true) {
    Find matchIndex;
    Display personNames[matchIndex] at the bottom of frame;
    Update attendance file with personName[matchIndex];
  }
}
```

If captured image matches with one of the stored images, image name is displayed at the bottom of the reference frame – waiting to be confirmed by the attendee.

```
// Update attendance file
Get the attendee name;
Read the attendance records(name, time, date) into an array;
If (attendee-name is not in array) {
    Write (name, time, date) into the attendance file;
}
```

Once the matched image is confirmed by the user, attendance sheet is updated with the name of the attendee.

RESULT OF OUR PROJECT



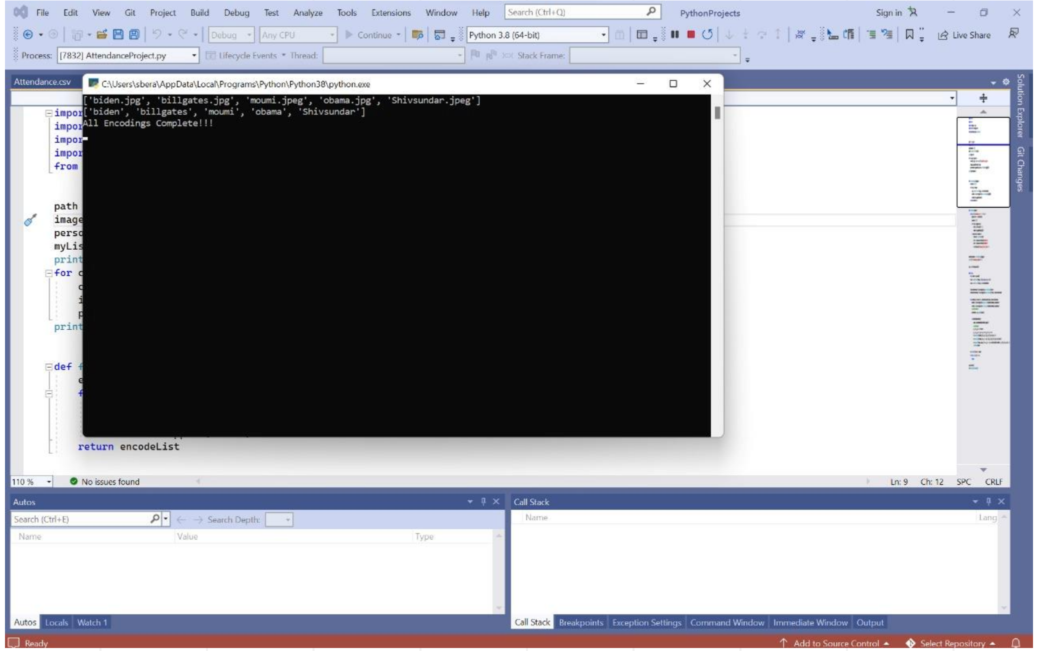
```
import cv2
import os
import numpy as np
import face_recognition
from datetime import datetime


path = 'images'
images = []
personNames = []
myList = os.listdir(path)
print(myList)
for cu_img in myList:
    current_Img = cv2.imread(f'{path}/{cu_img}')
    images.append(current_Img)
    personNames.append(os.path.splitext(cu_img)[0])
print(personNames)


def faceEncodings(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


def attendance(name):
```

AttendanceProject.py File Properties

Advanced
Build Action: Compile
Publish: True

Misc
File Name: AttendanceProject.py
Full Path: D:\users\sbera\source\repos\Pyth

Build Action
How the file relates to the build and deployment process



Process: [7832] AttendanceProject.py

C:\Users\sbera\AppData\Local\Programs\Python\Python38\python.exe

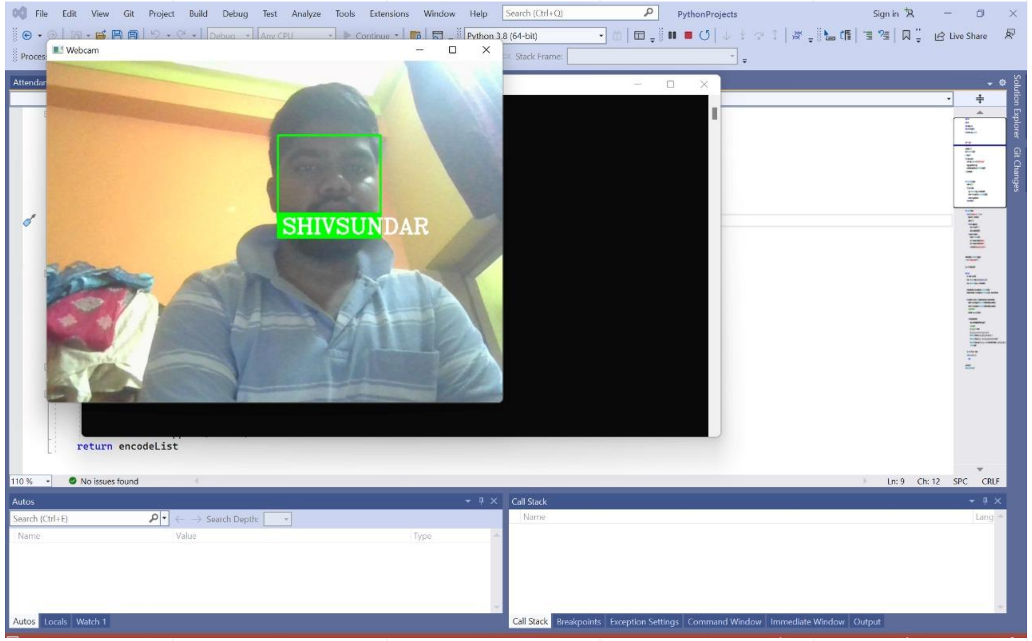
```
['biden.jpg', 'billgates.jpg', 'moumi.jpeg', 'obama.jpg', 'Shivsundar.jpeg']
['biden', 'billgates', 'moumi', 'obama', 'Shivsundar']
All Encodings Complete!!!
```

```
return encodeList
```

Program Code

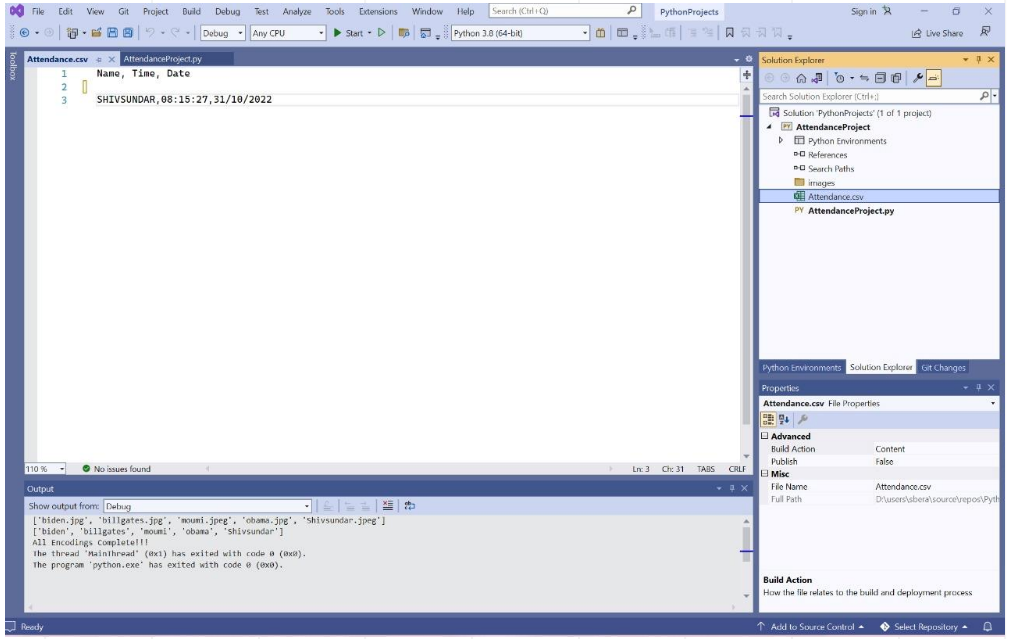
Execution Of Application

RESULT OF OUR PROJECT


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
PythonProjects
Sign in
Python 3.8 (64-bit)
Live Share
Webcam
SHIVSUNDAR
Stack Frame:
No issues found
Ln: 9 Ch: 12 SPC CRLF
Autos
Call Stack
Autos Locals Watch 1
Call Stack Breakpoints Exception Settings Command Window Immediate Window Output

```
return encodeList
```

Capturing And Matching The Image



```
Name, Time, Date

SHIVSUNDAR,08:15:27,31/10/2022
```

Show output from: Debug

```
['biden.jpg', 'billgates.jpg', 'moumi.jpeg', 'obama.jpg', 'Shivsundar.jpeg']
['biden', 'billgates', 'moumi', 'obama', 'Shivsundar']
All Encodings Complete!!!
The thread 'MainThread' (0x1) has exited with code 0 (0x0).
The program 'python.exe' has exited with code 0 (0x0).
```

Properties

Attendance.csv File Properties

Advanced	
Build Action	Content
Publish	False
Misc	
File Name	Attendance.csv
Full Path	D:\users\sbera\source\repos\Pyth

Build Action
How the file relates to the build and deployment process

Update Attendance File

THANKS