FACE RECOGNITION WITH OPENCV

Seminar (IT290) Report

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

By

MADHAVA CHARY RAVULA (201IT235)

SHIVAM REVANSIDDH KHOSE(201IT257)

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE -575025

MAY, 2022

DECLARATION

We hereby *declare* that the *Seminar (IT290) Report* entitled **FACE RECOGNITION WITH OPENCV** which is being submitted to the National Institute of Technology Karnataka Surathkal, in partial fulfillment of the requirementsfor the award of the Degree of Bachelor of Technology in the department of Information Technology, is a ***bonafide report of the work carried out by us.*** The material contained in thisseminar report has not been submitted to any University or Institution for the award of any degree.

MADHAVA CHARY RAVULA (201IT235)

SHIVAM REVANSIDDH KHOSE (201IT257)

Signature of the Student

Place: NITK, Surathkal

Date: 06/05/2022

<u>**CERTIFICATE**</u>

This is to certify that the Seminar entitled **“FACE RECOGNITION WITH OPENCV”** has been presented by Madhava Chary Ravula (201IT235)&Shivam Revansiddh Khose (201IT257), students of IV semester B.Tech. (IT), Department of Information Technology, NationalInstitute of Technology Karnataka Surathkal, during the even semester of the academic year2021-2022, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology.

Guide Name

Signature of the Guide with Date

Place: NITK, Surathkal

Date: 06/05/2022

ABSTRACT

The Oxford Dictionary defines a face as the part of a person's head from the forehead to the chin or the corresponding part of an animal. In human interactions, the face is the most significant factor because it contains important information about an individual. All humans will acknowledge people from their faces.

This research paper gives an ideal way of detecting and recognizing human face using OpenCV, and python which is part of deep learning. This report contains the ways in which deep learning an important part of computer science field can be used to determine the face using several libraries in OpenCV along with python. This report will contain a proposed system which will help in the detecting the human face in real time. This implementation can be used at various platforms in machines and smartphones, and several software applications.

This study primarily consists of 3 elements, specifically face detection from the image, feature extraction and storing many reminder images, and recognition. Face finding rule is employed to detect the face from the given image. The foremost helpful and distinctive options of the face image are extracted within the feature extraction part. Face Detection may be challenging because of pictures and video frames will contain advanced background, completely different head poses and occlusion like carrying glasses or scarf. It presents a rule for finding face recognition downside and concatenated into one feature vector that is employed to coach the system to recognize among the prevailing photos with it. Within the testing stage the system takes the face of the image of someone for recognition. Image acquisition, preprocessing, image filtering, feature extraction is just like the learning stage. For classification the options are fed to the trained system. The algorithms can determine the face image from the content and acknowledges it.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter-1

INTRODUCTION

Nowadays advancement of man-made brainpower is effectively creating; they open up tremendous potential outcomes before us. Investigation, gauging, detection went to another level with the utilization of man-made reasoning advancements. As of late, an incredibly encouraging field of research is Computer vision. Face detection is a phase where identifying the faces from the images or video sources. It very well may be utilized for remote distinguishing proof administrations for security in regions, for example, banking, transportation, law requirement, and electrical businesses. Despite huge varieties in visual upgrades because of evolving condition, maturing, and interruptions like whiskers, glasses, and haircut changes, this capacity is extremely powerful.

A face recognition program is a software application for verifying a person and identifying him or her with a video or picture from a source. For psychology at least the 1950s and the 1960's, the earlier work on facial recognition can be traced back to engineering literature. Some of the earliest findings include experiments on Darwin's feelings for face expression. With the open source platform Intel called OpenCV, facial recognition can be done quickly and reliably. One way from a face and an image database are the preferred facial features. It is generally compared to biometrics like fingerprints and eye reconnaissance systems, and is used in security systems, thumb recognition systems. The key element analysis using Fisher face algorithms, the Markov model, multilinear subspace learning using tensor representations and the nervously driven dynamic reference matching, etc, were also common recognition algorithms. The Computer-View library for Intel's open-source makes programming easy to use. This provides advanced capabilities such as facial detection, face tracking, facial recognition and a range of ready-to-use methods for artificial intelligence (AI). It has the advantages of being a multi-platform framework; this supports Windows and Macos, as well as Mac OS X recently.

The objective of this project is to provide a simpler and easy method in machine technology. With the help of such a technology one can easily detect the face by the help of dataset in similar matching appearance of a person. The method in which with the help of python and OpenCV in deep learning is the most efficient way to detect the face of the person. This method is useful in many fields such as the military, for security, schools, colleges and universities, airlines, banking, online web applications, gaming etc. this system uses powerful python algorithm through which the detection and recognition of face is very easy and efficient.

The most useful area in which face recognition is important is the biometrics that is used for authentication process which makes the work mor easier. Face recognition is one of the widely used technologies or systems in which it has the potential to perform tasks such as to have records provided in by the dataset in many areas such as the school and colleges attendance systems, it can also be helpful in catching the thieves or the terrorist, can be helpful in the security of common people and the much needed security areas in the country. Face recognition can be used by the government to verify the voters list, find missing persons, find the population or census, immigration process, also provide security over internet scams protecting Ecommerce and highly used in the medicine and healthcare range. This brings in a very high demand or a real time face recognition system for several uses for the people and government. Providing such excellent systems there would be ease in several activities.

Maintenance of attendance is incredibly necessary altogether at the institutes for checking the performance of employees/students. Some institutes take attending manually using paper or file-based approaches and a few have adopted ways of automatic attendance using some biometric techniques. The recent methodology for taking attendance is by calling out the name or roll number of the scholar to record attendance. It's a long and less efficient method of marking attending as a result of as we all know the info written within the paper typically may be lost or is less accurate as a result of students often mark every other's attending proxy. Therefore, to unravel these issues and avoid errors, we recommend computerizing this method by providing a system that records and manages student’s attending mechanically with no need for lecturers' interference. Every biometric system consists of the enrollment process during which distinctive features of someone are stored within the database and then there are processes of identification and verification. These 2 processes compare the biometric feature of someone with previously stored biometric details captured at the time of enrollment. Biometric templates are of many varieties like Fingerprints, Eye Iris, Face, Signature, and Voice. Our system uses the face recognition approach for the automated Attendance Management System. By considering this truth our system is going to be quicker and correct in marking the attendance of individual students. Face recognition consists of two steps, in the first step faces are detected in the image and then these detected faces are compared with the database for verification. Face detection is employed to find the position of face region and face recognition is employed for marking the attendance. The info can store the faces of scholars. Once the face of the scholar matches with one in each of the faces kept within the database then the attendance is recorded.

Chapter-2

LITERATURE REVIEW

Different journal papers and research papers are analyzed and reviewed.

1)Face Detection and Recognition using OpenCV and Python: International Research Journal of Engineering and Technology (IRJET) ,2020 was analyzed and it explains the basic overview of the major techniques used in the face recognition system that apply mostly to the front face of the human being. The methods include neural networks, hidden Markov model, face matching done geometrically and template matching. Eigenface is one of the most widely used methods in face recognition and detection which are broadly called as the principle components in mathematical terms. The eigenvectors are ordered to represent different amounts of the variations in the faces. Neural networks are highly used in the face recognition and detection systems. An ANN (artificial neural network) Was used in face recognition which contained a single layer Which shows adaptiveness in crucial face recognition systems. The face verification is done using a double layer of WISARD in neural networks. Graph matching is other option for face recognition. The object as well as the face recognition can be formulated using graph matching performed by optimization of a matching function.

2)“Facial Recognition using OpenCV”, Research Gate article by Shervin Emami and Valentin Petruț, explains common ways like a holistic matching technique, feature extraction technique, and hybrid ways are used. The paper relies on analysis in face recognition. The paper provides readers with an additional sturdy understanding of face recognition ways & applications. Face recognition systems establish people by their face photos. Face recognition systems establish the presence of an accredited person rather than merely checking whether or not or not a sound identification or secret's being utilized or key's being employed or whether or not the user is aware of the key personal identification numbers(pins) or passwords. The face recognition system directly compares the face photos of individuals and does not use ID numbers to differentiate one from the others. Once the very best two matched faces are extremely just like the query face image, manual review is needed to create certain they're so different persons therefore on eliminating duplicates.

Chapter-3

TECHNICAL DISCUSSION

3.1 Methodology

Firstly the image is imported by giving the area of the picture. At that point, the image is changed from RGB to Grayscale because it is anything but difficult to distinguish faces in the grayscale. From that point forward, picture control is utilized, in which the resizing, editing, obscuring, and honing of the pictures is done if necessary. The following stage is picture division, which is utilized for form location or sections the different items in a solitary picture so the classifier can rapidly distinguish the articles and faces in the image. The following step is to use an algorithm. The calculation is used for finding the area of the human countenances in a casing or picture. All human faces share some all-inclusive properties of the human face like the eyes area is more obscure than its neighbor pixels and the nose district is more splendid than the eye locale.

The haar-like calculation is likewise utilized for highlight determination or highlight extraction for an article in a picture, with the assistance of edge location, line identification, focus detection for recognizing eyes, nose, mouth, and so forth in the image. It is utilized to choose the fundamental highlights in a picture and concentrate these highlights for face detection. The subsequent stage is to give the directions of x, y, w, h which makes a square shape enclose the image to show the area of the face or we can say that to show the locale of interest in the picture. After this, it can make a square shape enclose the zone of interest where it identifies the face. There are additionally numerous other identification strategies that are utilized together for recognition, for example, grin detection, eye location, squint detection, and so on. We used some tools to build our system. Without the help of these tools, it would not be possible to make it done. Here we will discuss the most important one.

The libraries we have used in our project are

1. OpenCV:

We used OpenCV 3 dependency for python 3. OpenCV is a library where there are lots of image processing functions available. This is a very useful library for image processing. Even one can get the expected outcome without writing a single code. The library is cross-platform and free for use under the open-source BSD license. Example of some supported functions are given below:

● Derivation: Gradient/laplacian computing, contours delimitation

● Hough transforms: lines, segments, circles, and geometrical shapes detection

● Histograms: computing, equalization, and object localization with a back-projection algorithm

● Segmentation: thresholding, distance transform, foreground/background detection, watershed segmentation

● Filtering: linear and nonlinear filters, morphological operations

● Cascade detectors: detection of a face, eye, car plates

● Interest points: detection and matching

● Video processing: optical flow, background subtraction, camshaft (object tracking)

● Photography: panoramas realization, high definition imaging (HDR), image inpainting

2.Cmake:

Cmake is used to control the software compilation process using simple platform and compiler independent configuration files, and generate native makefiles and workspaces that can be used in the compiler environment of your choice.

The suite of CMake tools were created by Kitware in response to the need for a powerful, cross-platform build environment for open-source projects such as ITK and VTK.

3.Dlib

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge.

4.Face Recognition

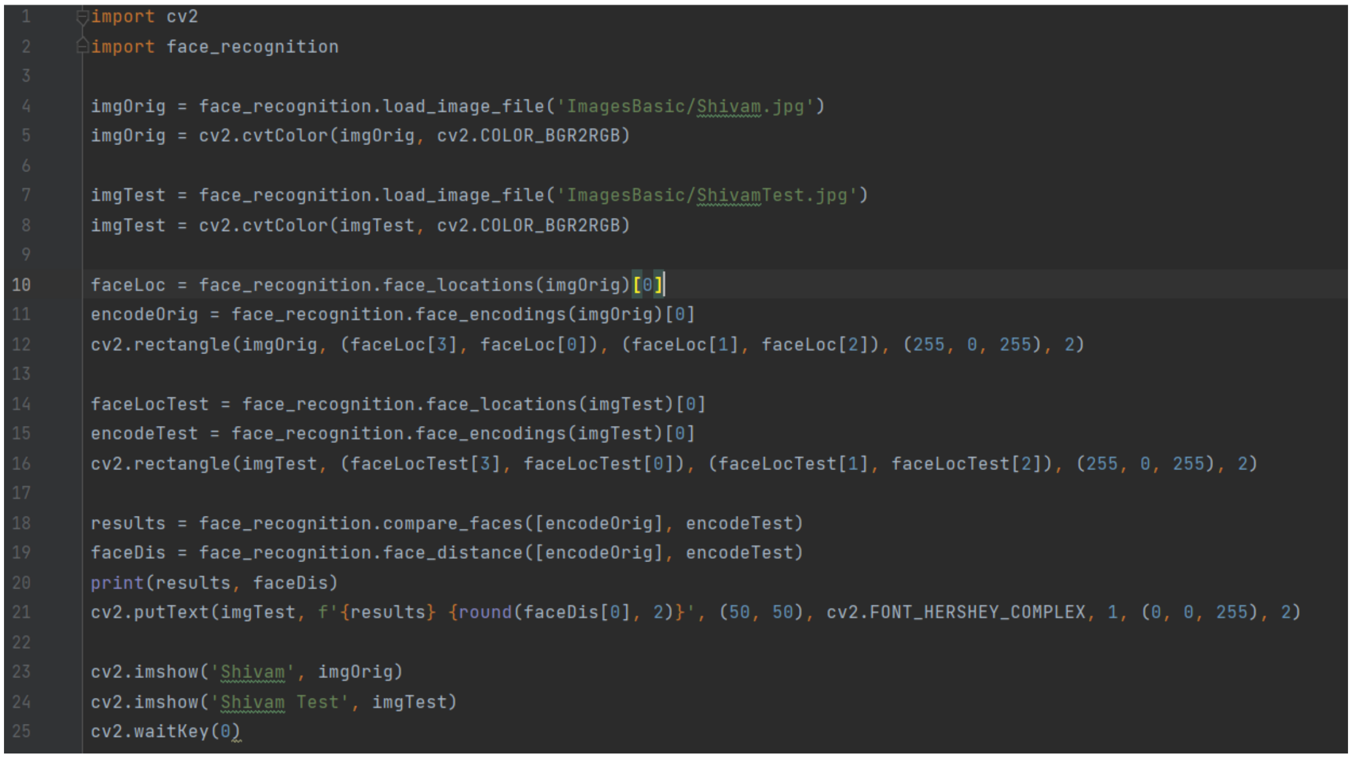
Face Recognition is a technology in computer vision. In Face recognition / detection we locate and visualize the human faces in any digital image. It is a subdomain of Object Detection, where we try to observe the instance of semantic objects. Recognize and manipulate faces from Python or from the command line with the world’s simplest face recognition library. Built using dlib's state-of-the-art face recognition. built with deep learning.

5.Numpy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
It is a third-party library (i.e. it is not part of Python's standard library) that facilitates numerical computing in Python by providing users with a versatile N-dimensional array object for storing data, and powerful mathematical functions for operating on those arrays of numbers.

3.2 Implementation

We have used two python i.e. .py code files and two datasets of images that is image basic and project images.First pyhton file is basic.py which basically compares the two input faces .First of all we load the imags in our variables .We are converting BGR to RGB format as opencv library demand .After that we encode the images and draw recatangle around face.Then we compare their measurements measured by different python libraries.

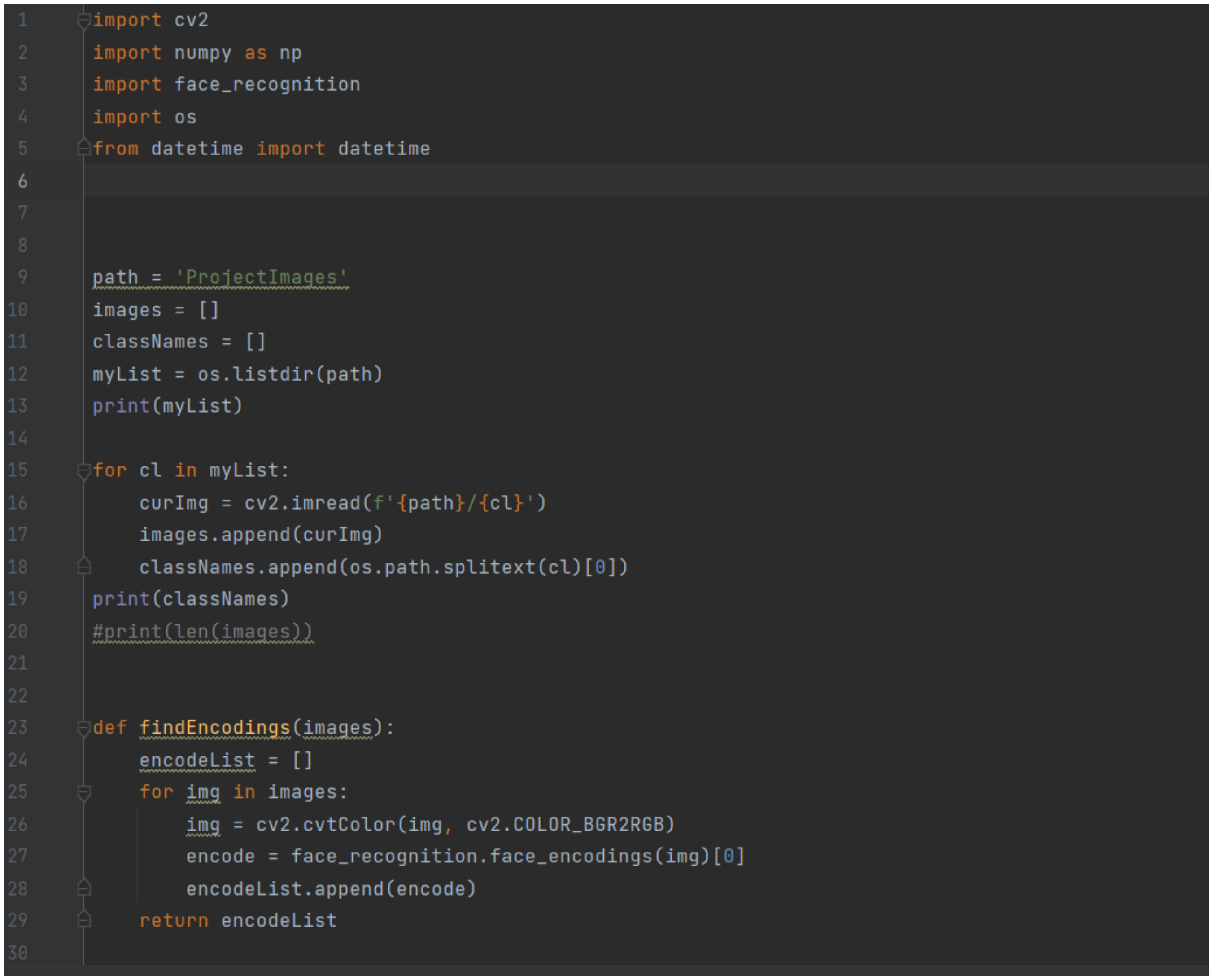


```python
import cv2
import face_recognition

imgOrig = face_recognition.load_image_file('ImagesBasic/Shivam.jpg')
imgOrig = cv2.cvtColor(imgOrig, cv2.COLOR_BGR2RGB)

imgTest = face_recognition.load_image_file('ImagesBasic/ShivamTest.jpg')
imgTest = cv2.cvtColor(imgTest, cv2.COLOR_BGR2RGB)

faceLoc = face_recognition.face_locations(imgOrig)[0]
encodeOrig = face_recognition.face_encodings(imgOrig)[0]
cv2.rectangle(imgOrig, (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([encodeOrig], encodeTest)
faceDis = face_recognition.face_distance([encodeOrig], 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('Shivam', imgOrig)
cv2.imshow('Shivam Test', imgTest)
cv2.waitKey(0)
```

Fig 1:basic.py

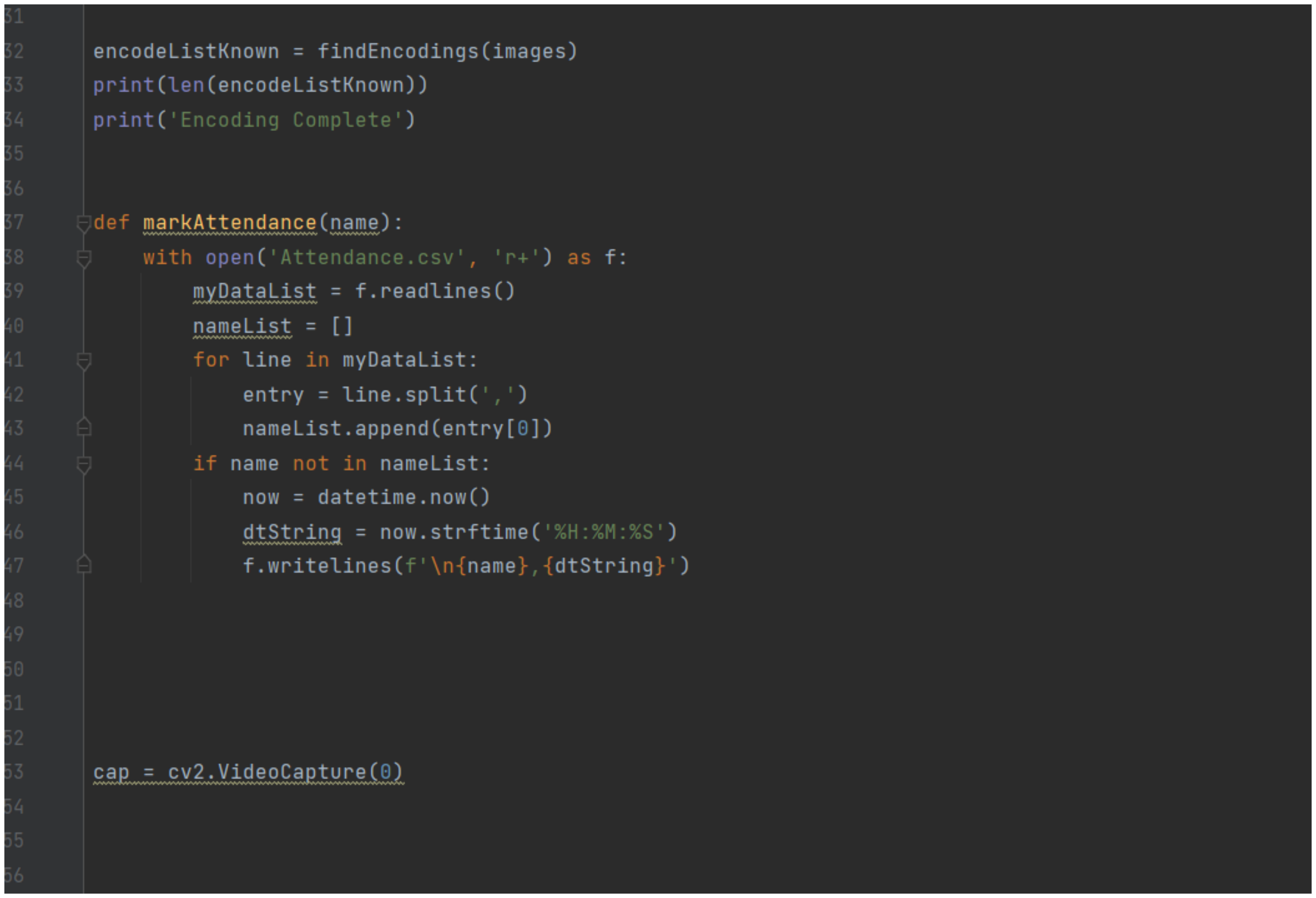
After comparing the images result will be shown on console as well as on test image as we are using puttext function of cv2 library.

After that we have performed small application on this face recognition where we are giving input image from capturing the image from webcam . After encoding the images it comopares its parameters with the images present in the dataset and finds images distances and print the output as name of images present in dataset .



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



path = 'ProjectImages'
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)
#print(len(images))


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

```

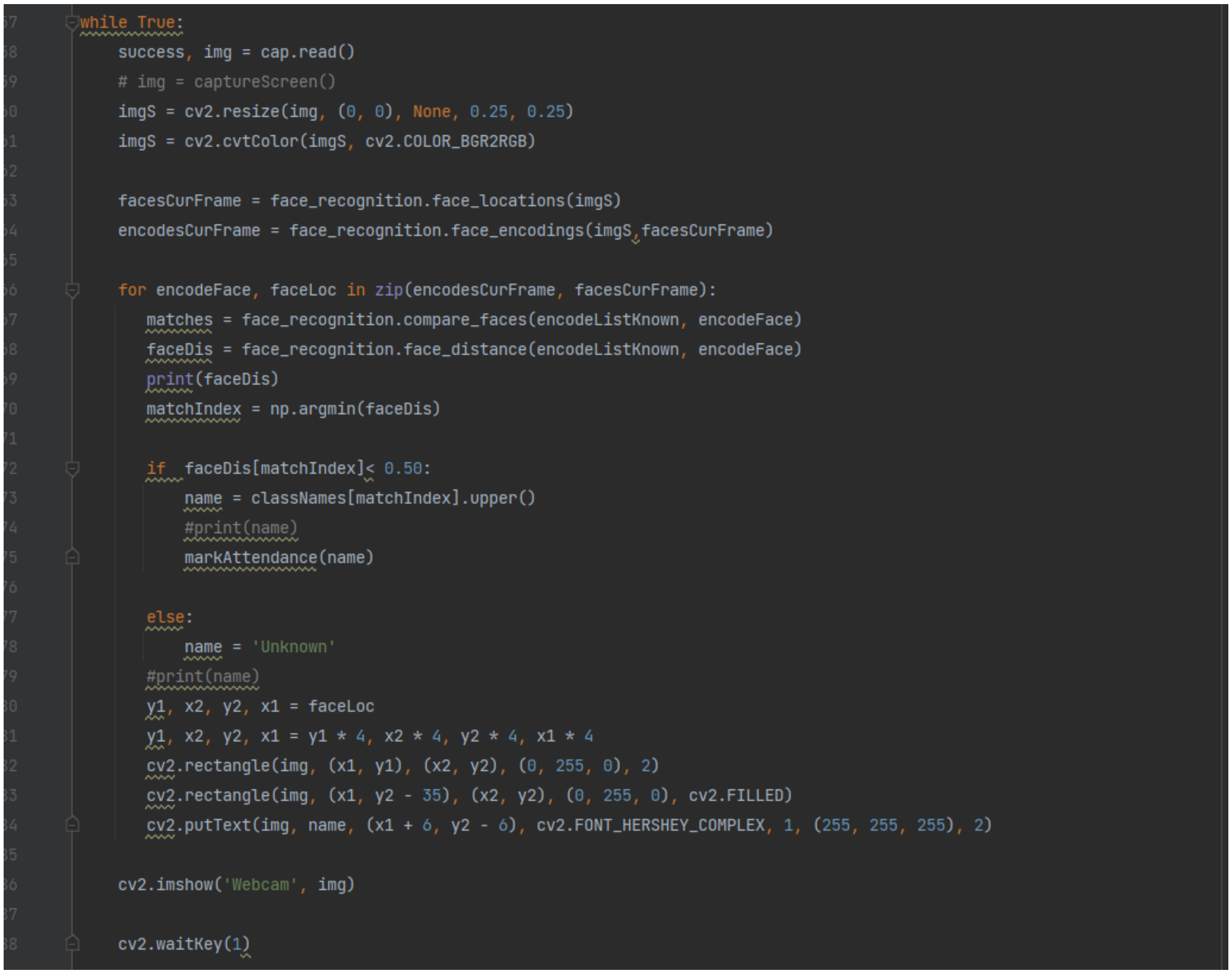
Fig 2:Project.py Part-1



```python

encodeListKnown = findEncodings(images)
print(len(encodeListKnown))
print('Encoding Complete')


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}')





cap = cv2.VideoCapture(0)



```

Fig 3:Project.py Part-2

As we can see here we are saving all the images in the dataset in the list then finding its encoding measurements . Here we are creating the the function markAttendance function to store information of the input recognized image as name and time and storing it in the attendance file .



```python
while True:
    success, img = cap.read()
    # 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)
    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  faceDis[matchIndex]< 0.50:
            name = classNames[matchIndex].upper()
            #print(name)
            markAttendance(name)

        else:
            name = 'Unknown'
        #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)

    cv2.imshow('Webcam', img)

    cv2.waitKey(1)
```

Fig 4:Project.py Part-3

Here we are comparing the measurements of the captured images with images in the dataset and printing the best result that is having atleast face distance.

3.3 Experimental Results

In this section we are discussing about the results through the output screenshots of the project.As a result of first python file that is basic.py we get the ouptupt in the format of true or false as we compare the images.

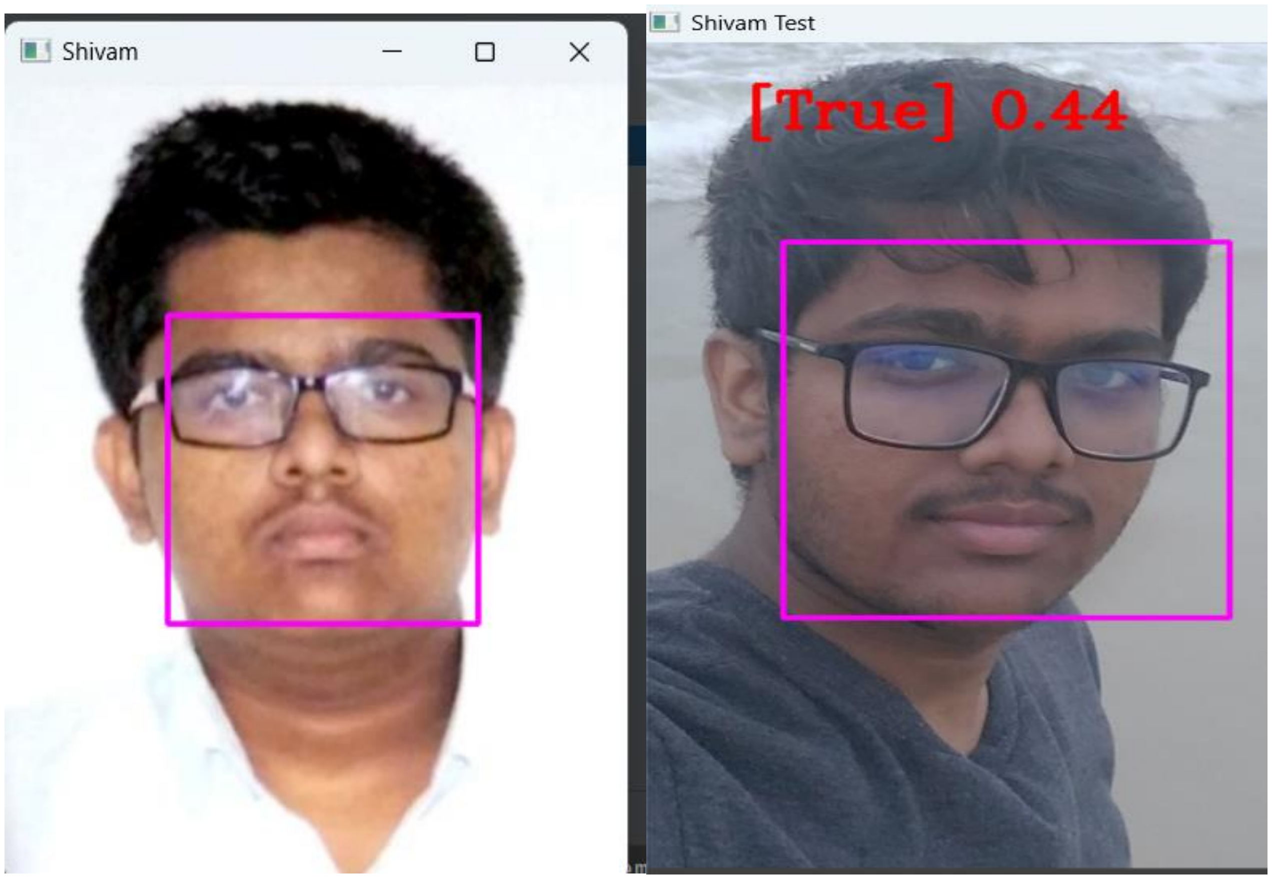


Fig 5:Result for basic.py

For our second pyhton file we are capturing the input image from webcam and after compairing we are printing the result on the face as name stored in our database .

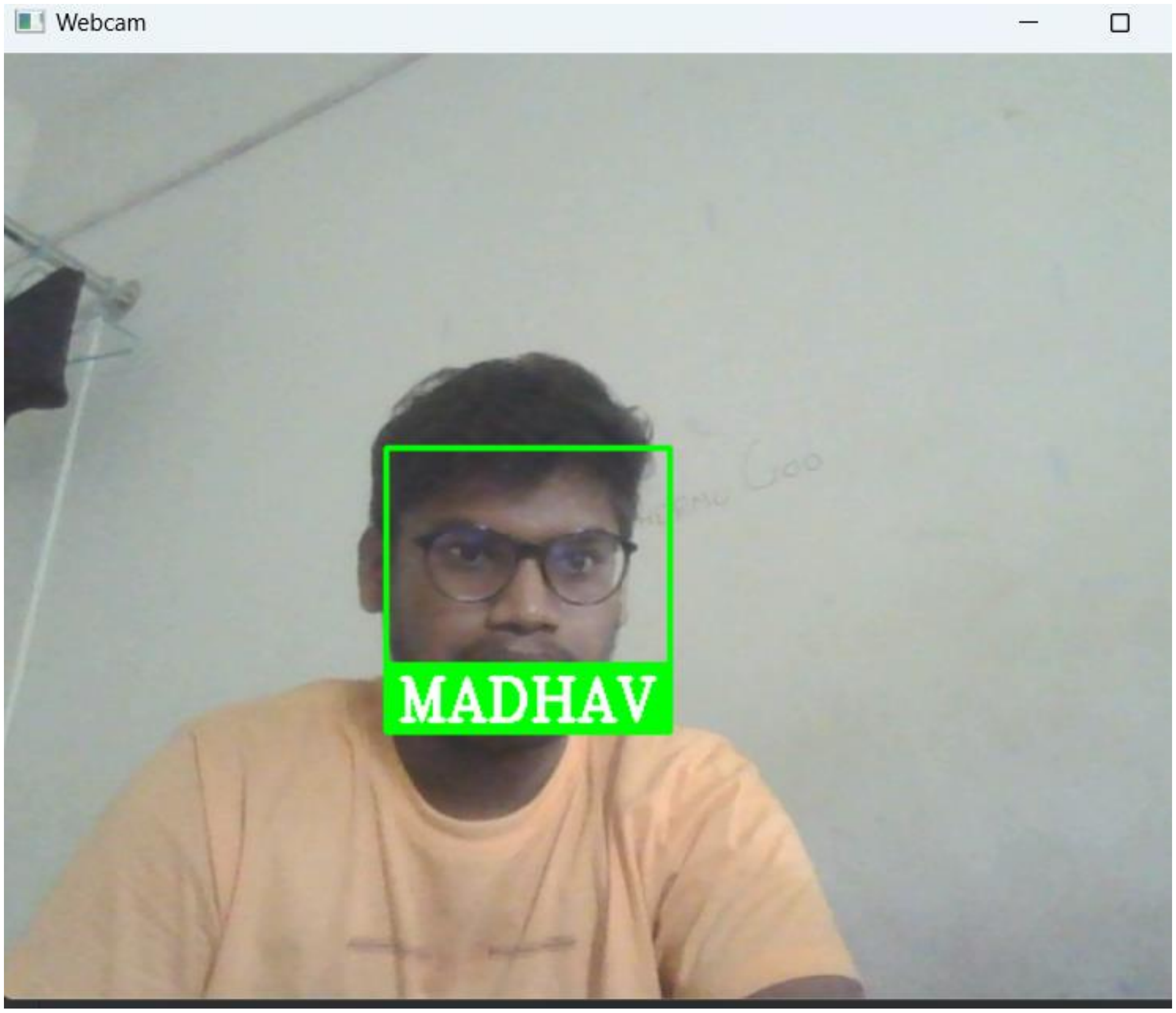
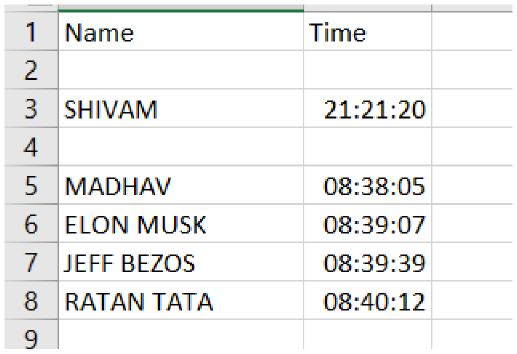


Fig 6:Result for Project.py

After recognition of faces we are storing the name and the time when it is encountered in the csv file



1	Name	Time
2		
3	SHIVAM	21:21:20
4		
5	MADHAV	08:38:05
6	ELON MUSK	08:39:07
7	JEFF BEZOS	08:39:39
8	RATAN TATA	08:40:12
9		

Fig 7:Attendance

Chapter-4

Conclusion and Future Work

4.1 Conclusion

Face recognition systems are currently associated with many top technological companies and industries making the work of face recognition easier. The use of python programming and OpenCV makes it an easier and handy tool or system which can be made by anyone according to their requirement. The proposed system discussed in this project will be helpful for many as it is user friendly and cost_ efficient system. Hence by the use of python and OpenCV the face recognition system can be designed for various purposes.

A face detection and recognition system would certainly speed up the process of checking student attendance in comparison to other biometrics authentication methods and in the right circumstances it would be able to match their accuracy. Nowadays there are a wide variety of software, whether it is a Face API like Microsoft's or a library like OpenCV, that makes face detection and recognition accessible and reliable and is constantly improving.

4.2 Future Work

In the field of face recognition, there are several important problems. One challenge is to plan the picture before use of the tool. Of example, it is one of the ability to identify if the image is a man or a woman, so that an image can be identified by one or two classifiers and trained with such an individual. It may be possible to obtain consistency in classification when multiple spaces are spectrum-scale individuals in one section. This turns out to be a collection of peculiarity for men and one for women. The new classification network is yet another piece of work by developing it in future. Instead of the resulting image projection into face space, you can create your network from the data as an output. Perhaps by learning face projection, it improves the accuracy of the neural network classifier.

References

1) Face Detection and Recognition using OpenCV and Python. Tejashree Dhawle, Urvashi Ukey, Rakshandha Choudante : International Research Journal of Engineering and Technology (IRJET) ,2020.

2) Face Detection and Recognition using OpenCV. Maliha Khan ; Shaveta Khepra ; Rani Astya ; Sudeshna Chakraborty : 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS).

3)Shervin EMAMI and Valentin Petrut SUCIU, "Facial Recognition using OpenCV", ResearchGate article.

4) Divyarajsinh N. Parmar and Brijesh B. Mehta, "Face Recognition Methods & Applications", an article in International Journal of Computer Applications in Technology.