



JAYPEE UNIVERSITY OF INFORMATION
TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING

PROJECT REPORT

Python Programming Essentials

Project Topic: Face recognition in Python using OpenCV

Submitted by:

1) Aniket Rawat

2) Himanshu Sharma

Batch: CS37

Roll No. 211315 and 211321.

Submitted to:

Mr. Naveen Jaglan

CONTENTS

- ❖ Introduction
- ❖ Problem Statement
- ❖ Aims and Objectives
- ❖ Technology Stack
- ❖ Significance of the Project
- ❖ Project Features
- ❖ Practical Demo
- ❖ Conclusion

INTRODUCTION

Face recognition is the technique in which the identity of a human being can be identified using one's individual face. Such kind of systems can be used in photos, videos, or in real time machines.

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.

PROBLEM STATEMENT

To develop a face recognition application, using the techniques of OpenCV, which recognizes the facial features of a given person and displays the name above their head.

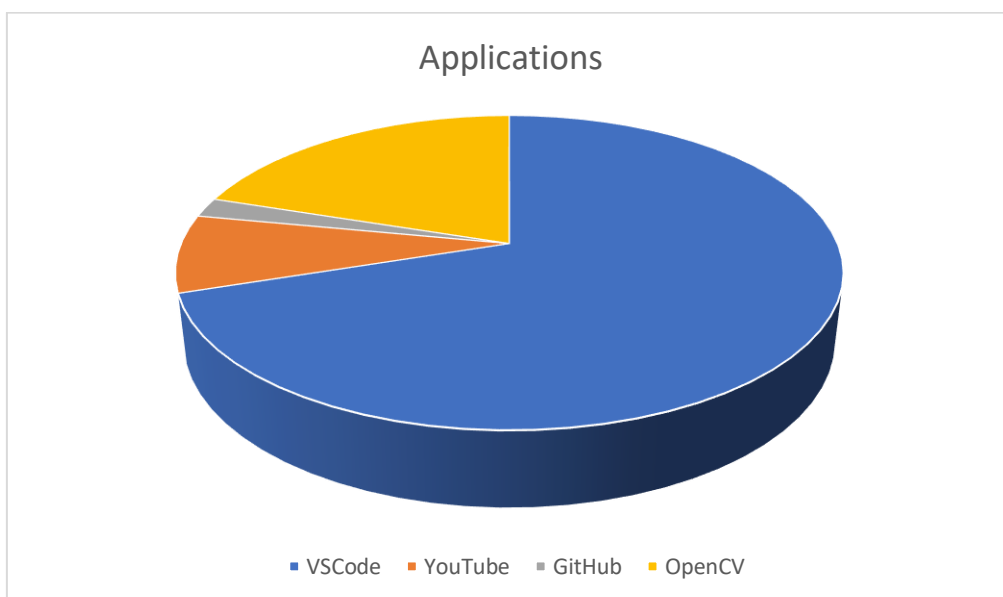
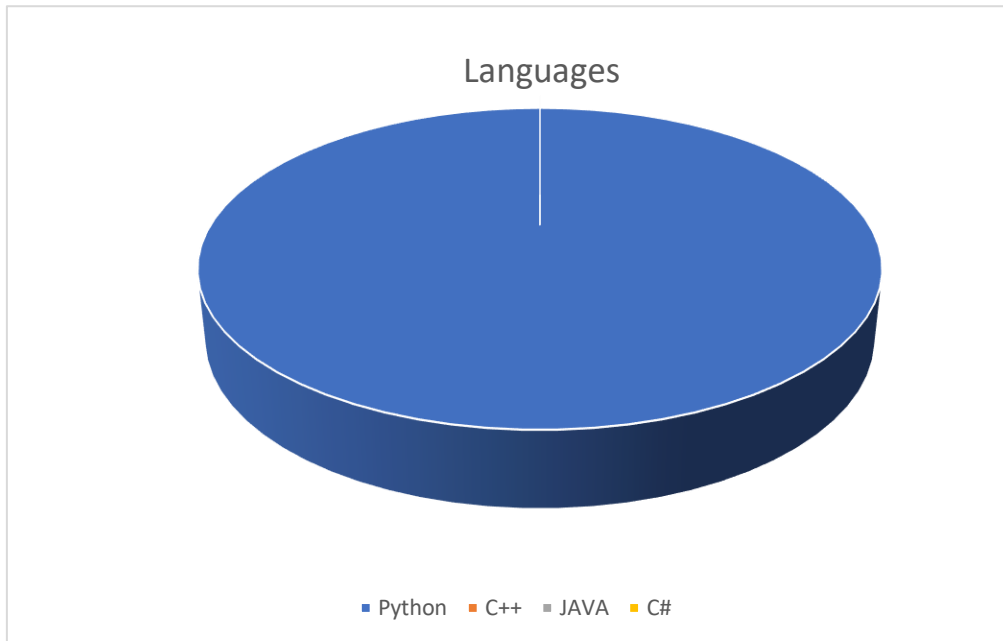
Our main goal is to differentiate between different people by specifying a confidence value which signifies the accuracy of our Python program.

AIMS AND OBJECTIVES

The main aim or objective of this project report is to provide or develop a system that will use the camera of the computer or the system that would detect and recognize the person's face or the face of the individual using the tool in OpenCV called as the Open Face and python programming language in deep learning domain.

This project majorly focuses on the recognition of the faces in a video running or in a captured frame. This frames the person's face in a rectangle and his name appears in a box above the frame.

TECHNOLOGY STACK



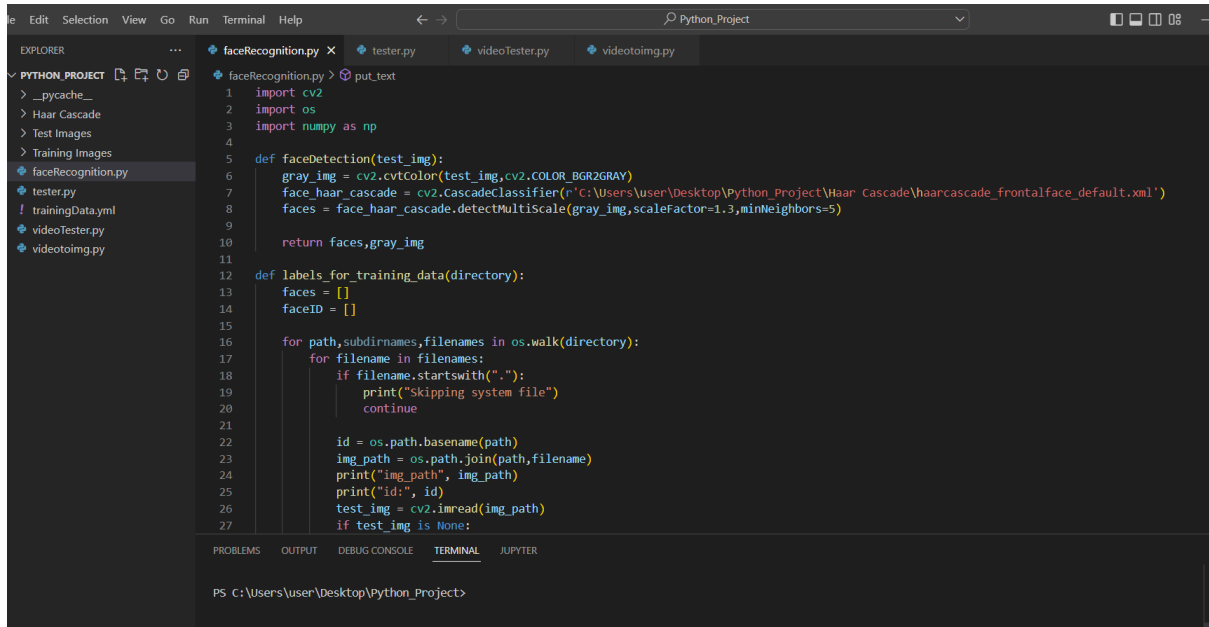
SIGNIFICANCE OF THE PROJECT

The most useful area in which face recognition is important is the biometrics that is used for authentication process which makes the work 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.

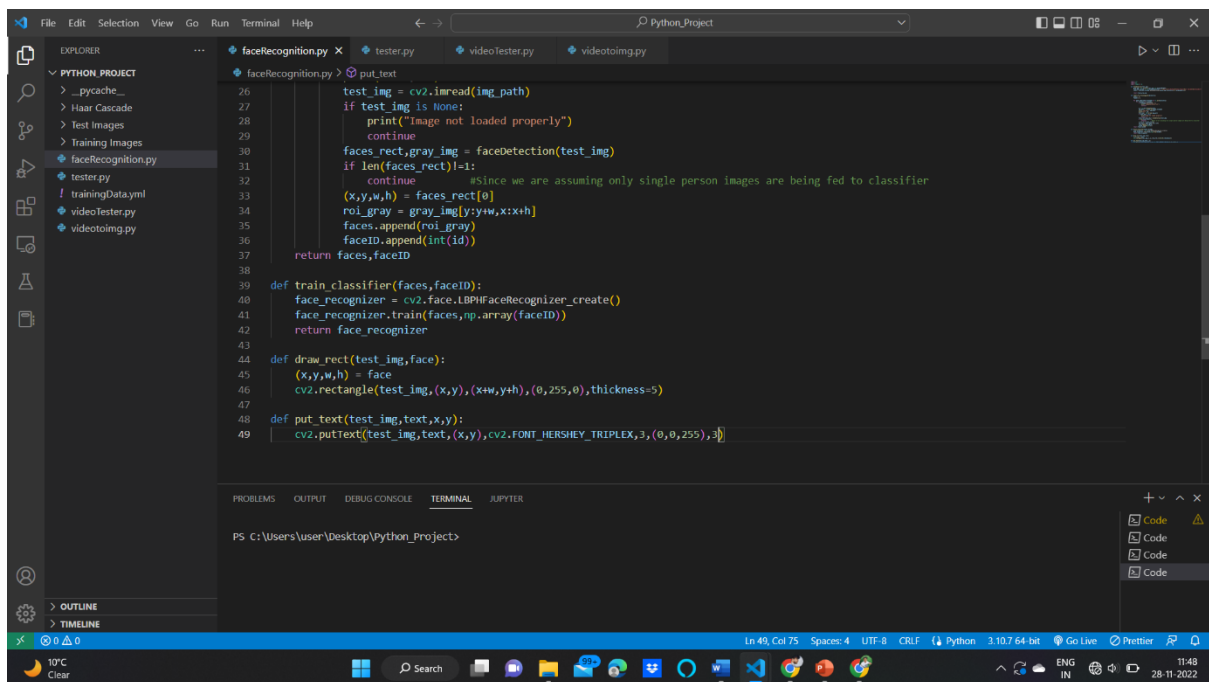
PROJECT FEATURES

faceRecognition.py



This screenshot shows the first 27 lines of the `faceRecognition.py` file in a VS Code editor. The Explorer sidebar on the left shows the project structure with files like `faceRecognition.py`, `tester.py`, `videoTester.py`, and `videotoimg.py`. The main editor area displays the following code:

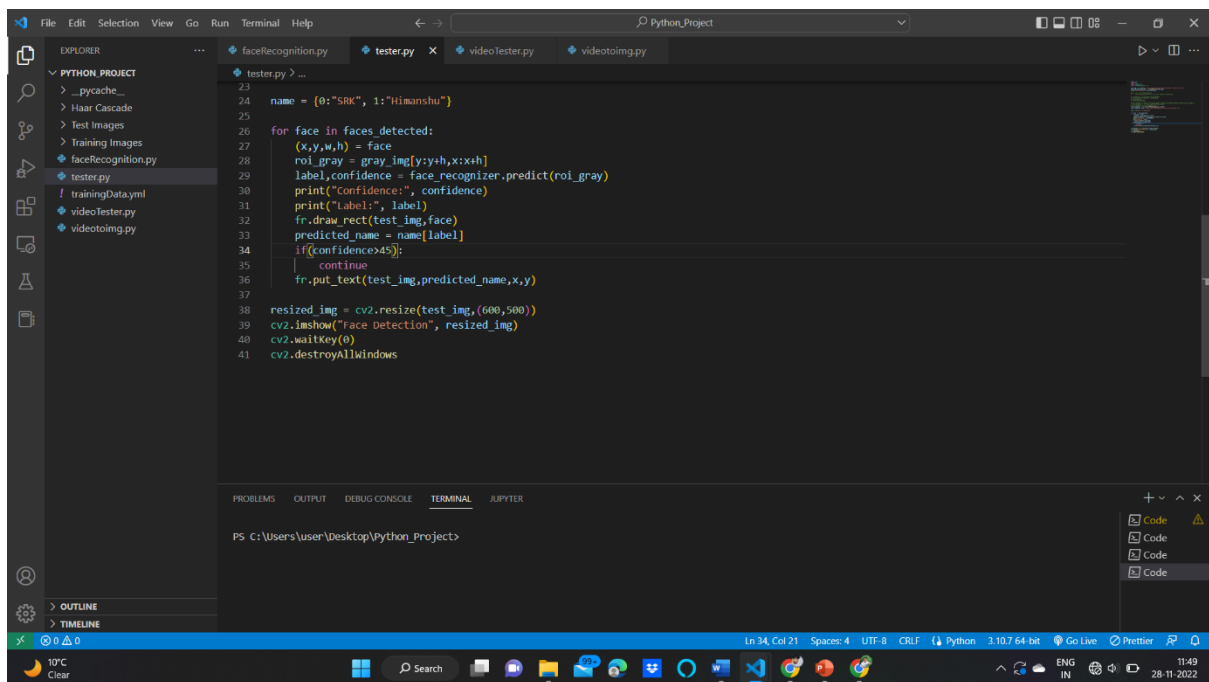
```
1 import cv2
2 import os
3 import numpy as np
4
5 def faceDetection(test_img):
6     gray_img = cv2.cvtColor(test_img, cv2.COLOR_BGR2GRAY)
7     face_haar_cascade = cv2.CascadeClassifier(r"C:\Users\user\Desktop\Python_Project\Haar_Cascade\haarcascade_frontalface_default.xml")
8     faces = face_haar_cascade.detectMultiScale(gray_img, scaleFactor=1.3, minNeighbors=5)
9
10    return faces, gray_img
11
12 def labels_for_training_data(directory):
13     faces = []
14     faceID = []
15
16     for path, subdirname, filenames in os.walk(directory):
17         for filename in filenames:
18             if filename.startswith("."):
19                 print("Skipping system file")
20                 continue
21
22             id = os.path.basename(path)
23             img_path = os.path.join(path, filename)
24             print("img_path", img_path)
25             print("id:", id)
26             test_img = cv2.imread(img_path)
27             if test_img is None:
```



This screenshot shows the continuation of the `faceRecognition.py` file, lines 26 through 49. The code includes logic for loading test images, detecting faces, training a classifier, and drawing bounding boxes and labels on the test image. The VS Code interface and project explorer are consistent with the previous screenshot.

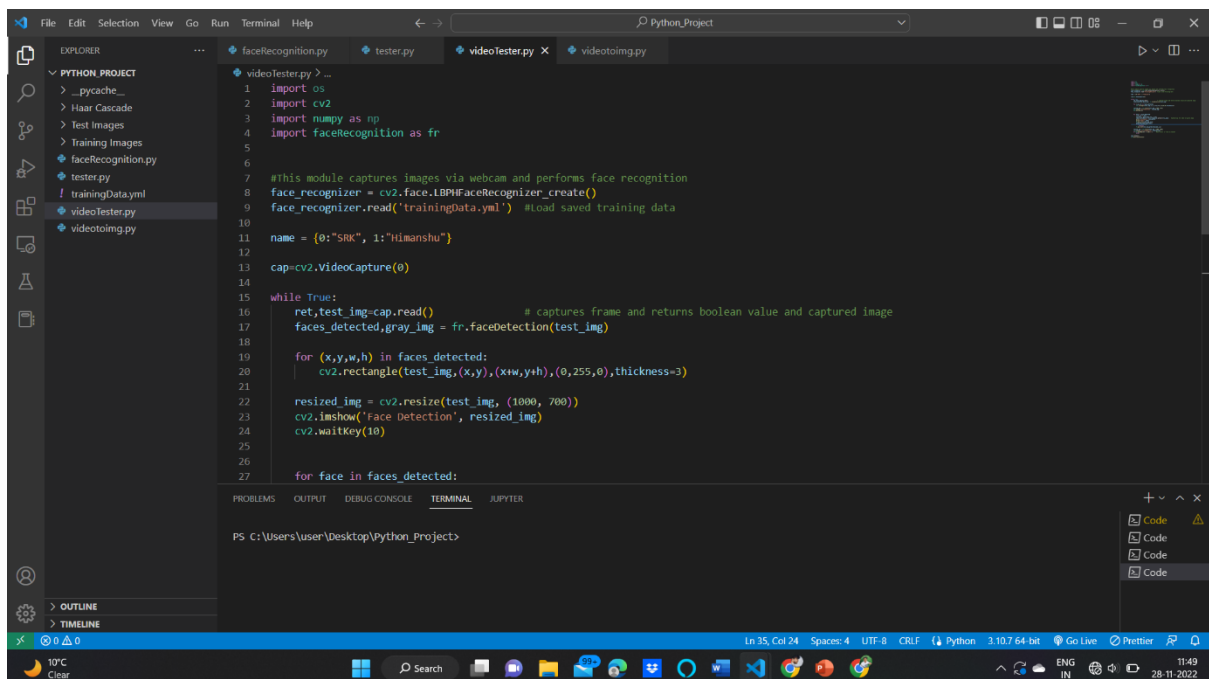
```
26     test_img = cv2.imread(img_path)
27     if test_img is None:
28         print("Image not loaded properly")
29         continue
30     faces_rect, gray_img = faceDetection(test_img)
31     if len(faces_rect) != 1:
32         continue #since we are assuming only single person images are being fed to classifier
33     (x,y,w,h) = faces_rect[0]
34     roi_gray = gray_img[y:y+w, x:x+h]
35     faces.append(roi_gray)
36     faceID.append(int(id))
37     return faces, faceID
38
39 def train_classifier(faces, faceID):
40     face_recognizer = cv2.face.LBPHFaceRecognizer_create()
41     face_recognizer.train(faces, np.array(faceID))
42     return face_recognizer
43
44 def draw_rect(test_img, face):
45     (x,y,w,h) = face
46     cv2.rectangle(test_img, (x,y), (x+w,y+h), (0,255,0), thickness=5)
47
48 def put_text(test_img, text, x, y):
49     cv2.putText(test_img, text, (x,y), cv2.FONT_HERSHEY_TRIPLEX, 3, (0,0,255), 3)
```


tester.py

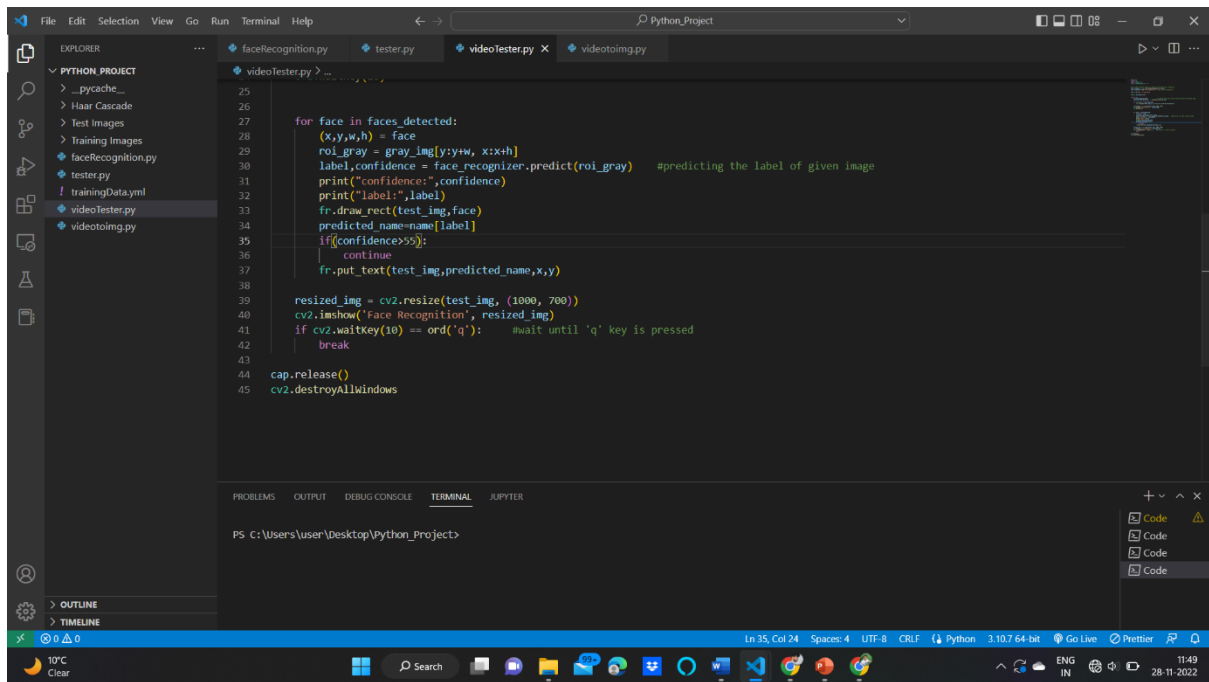


```
23
24 name = {'SRK': 1, 'Himanshu': 2}
25
26 for face in faces_detected:
27     (x,y,w,h) = face
28     roi_gray = gray_img[y:y+h,x:x+h]
29     label, confidence = face_recognizer.predict(roi_gray)
30     print("Confidence:", confidence)
31     print("Label:", label)
32     fr.draw_rect(test_img, face)
33     predicted_name = name[label]
34     if (confidence > 45):
35         continue
36     fr.put_text(test_img, predicted_name, x, y)
37
38 resized_img = cv2.resize(test_img, (600, 500))
39 cv2.imshow("Face Detection", resized_img)
40 cv2.waitKey(0)
41 cv2.destroyAllWindows
```

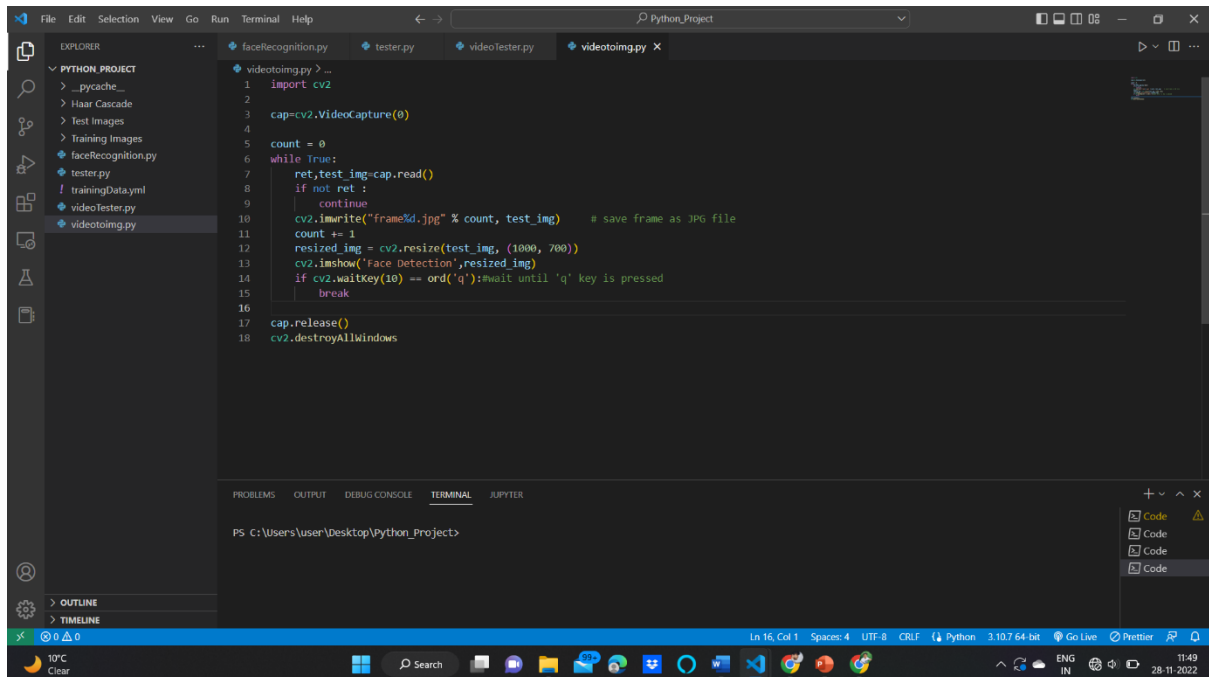
videoTester.py



```
1 import os
2 import cv2
3 import numpy as np
4 import faceRecognition as fr
5
6
7 #This module captures images via webcam and performs face recognition
8 face_recognizer = cv2.face.LBPHFaceRecognizer_create()
9 face_recognizer.read('trainingData.yml') #Load saved training data
10
11 name = {'SRK': 1, 'Himanshu': 2}
12
13 cap=cv2.VideoCapture(0)
14
15 while True:
16     ret, test_img = cap.read() # captures frame and returns boolean value and captured image
17     faces_detected, gray_img = fr.faceDetection(test_img)
18
19     for (x,y,w,h) in faces_detected:
20         cv2.rectangle(test_img, (x,y), (x+w,y+h), (0,255,0), thickness=3)
21
22     resized_img = cv2.resize(test_img, (1000, 700))
23     cv2.imshow("Face Detection", resized_img)
24     cv2.waitKey(10)
25
26     for face in faces_detected:
```



videotoimg.py

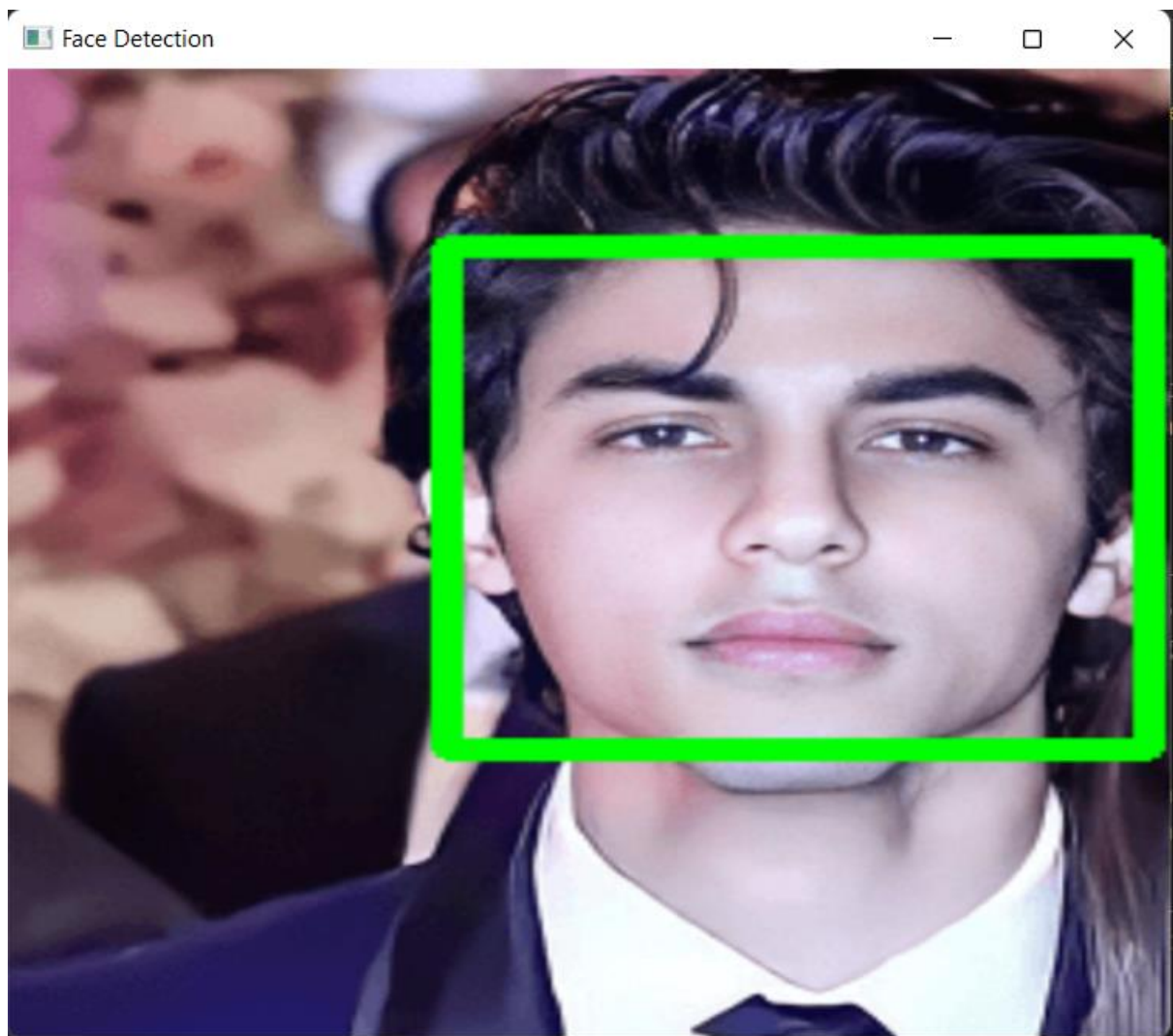


PROJECT DEMO

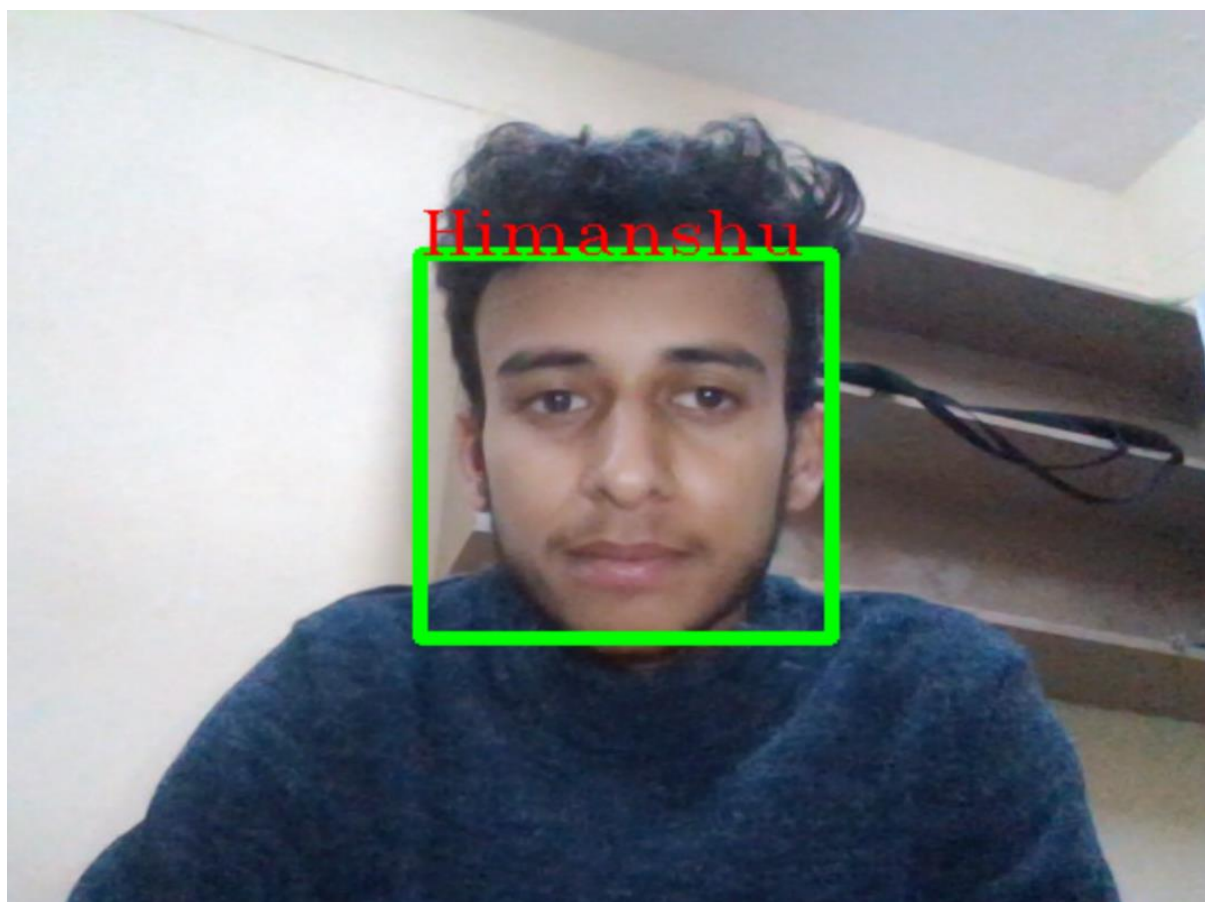
test1.jpg



test2.jpg



Video Capture



CONCLUSION

Face recognition system is a popular study task in the field of image processing and computer vision, owing to its potentially enormous application as well as its theoretical value. This system is widely deployed in many real-world applications such as security, surveillance, homeland security, access control, image search, human-machine, and entertainment. However, these applications pose different challenges such as lighting conditions and facial expressions. This paper highlights the recent research on the 2D or 3D face recognition system, focusing mainly on approaches based on local, holistic (subspace), and hybrid features.