FACE DETECTION

END TERM REPORT

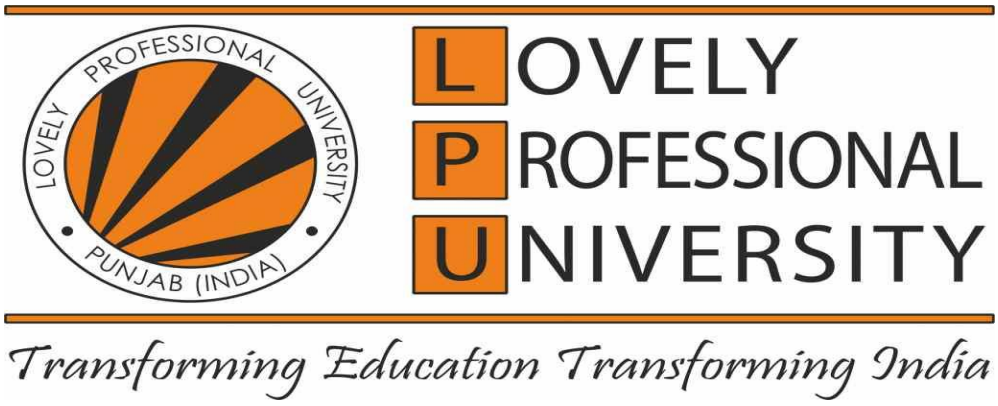
Prabhat Kumar Kushwaha

Divyansh kumar

Jyotish kumar

Roll no. 34,36,53

Section K18HV



Department of Intelligent Systems

School of Computer Science Engineering

Lovely Professional University, Jalandhar

APRIL 2020

Student Declaration

This is to declare that this report has been written by me/us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. I/We aver that if any part of the report is found to be copied, I/we are shall take full responsibility for it.

Name of Student :Prabhat kumar

Roll number: 34

Name of Student Divyansh kumar

Roll number: 36

Name of Student :Jyotish kumar

Roll number: 53

TABLE OF CONTENTS

TITLE **PAGE NO.**

BONAFIDE CERTIFICATE

Certified that this project report “ FACE DETECTION ” is the bonafide work of “Prabhat kumar , Divyansh kumar , and Jyotish kumar” who carried out the project work under my supervision.

Signature of the Supervisor

Dipen Saini

Academic Designation

ID of Supervisor

Department of Supervisor

1.INTRODUCTION

- Face recognition detects faces in the camera's field of view- as many as 15 at the same time and matches them against faces previously stored in the database.
- Faces can be enrolled in the database from existing still images or from the video camera itself.
- Face recognition is the advanced technology due to artificial intelligence created a revolutionized facial recognition system.
- The advancement in technology embedded artificial intelligence to create a smart system able to bring filters of Instagram or Snapchat.
- Face ID automatically adapts to changes in your appearance. From glasses, to cosmetic makeup or even growing facial hair. It is designed to work indoors, outdoors and even in total darkness.

DUE TO AI FACIAL RECOGNITION CAN BE USED IN THE FOLLOWING:

- Real time watchlist detection. Identifying individuals on the spot.
- Safer cities by capturing an image, assessing it and matching it.FR can recognize anyone with an arrest warrant out on the loose.
- Tracking and correlating individual journeys.
- Monitoring real-time data against specific parameters.

ABSTRACT

- In our project, by using OpenCV in python, we will be detecting multiple people's faces along with their labels. We will place some images to train our classifier to recognize multiple people then we will add every person's folder as different labels and then we will be adding their names along with labels. We will be predicting faces realtime via our webcam.
- We will look at scripts to generate hundreds of training images quickly and to resize images as well.

LITERATURE REVIEW

- **ABSTRACT:**
  Approach of face recognition aims to detect faces in still image and sequence image from video that has many methods such as local, global and hybrid approach.

Face perception is currently an active research area in the computer vision literature. The goal of this paper is to present a critical survey of existing literatures on human face detection systems. Face detection is not straightforward. In this paper we presented three methods of face detection, which are commonly used. Our future work wants to solve and create accuracy gain for widely accepted face recognition system.

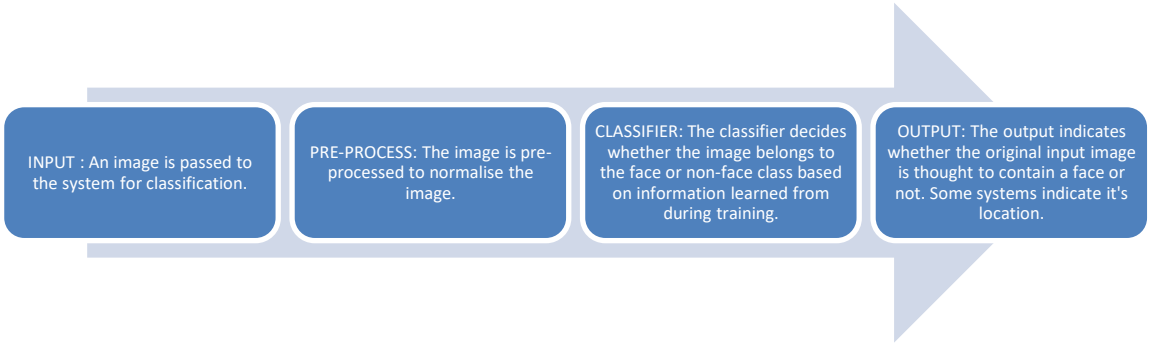
Keyword: face recognition, face detection, integral image, literature review, neural network.

- **INTRODUCTION:**

  Face detection is a dedicated process, not merely an application of the general object recognition process.It is the representation of the most splendid capacities of human vision.

  Automatic face detection is the cornerstone of all applications revolving around automatic facial image analysis including face recognition and verification, face tracking for surveillance, facial behavior analysis, facial attribute recognition, gender/age recognition etc.

- **RELATED WORK:** Face detection is basically an image segmentation problem as the image is to be segmented into two parts: one containing faces and the other representing non-face regions.



- **METHODS OF FACE DETECTION**:

  A) **Skin color based Face detection method**: Detection of skin color in color images is useful technique for face detection. Using skin-color as a feature for tracking a face has several advantages. Color processing is much faster than processing other facial features.

  The HSV model is preferred. The first step is to segment the color image into skin and non-skin region. After segmentation procedure, morphological operators are implemented with a structuring element. Then, the standard deviation of the area is calculated and rectangles are

drawn in the skin regions. If any unwanted rectangles are created, it is then removed.

Advantages: (i) This method is able to correctly locate all faces in the images with almost at right scale.

(ii) More robust to noise and shape variations.

(iii) Accuracy 80-82%

Disadvantages: (i) Moderate falsi detection rate

(ii) Sometimes non face skin color regions are also detected.

B) **Viola Jones Face detection system**: It is an object detection framework which provide robust and competitive object detection rates in real-time proposed in 2001. It is capable of processing images extremely rapidly and achieving high detection rates.

1.Integral Image: It allows the features used by detector to be computed very rapidly.

2.Adaboost algorithm: To select principal features and to train classifiers that would be using them. Aim is to create strong classifier from linear combination of weak classifier.

3.Cascading: The third stage is a combining successively more complex classifiers in a cascade structure which increases the speed of the detector by focusing attention on promising face like regions of the image.

Advantages: (i) Rapid image processing with high detection rates.

(ii) High accuracy

(iii)Very low false positive rate

(iv) also used to detect other objects.

Disadvantages: (i) Less effective on non-frontal face

(ii) Very long training time

C) **Face detection system based on retinal connected neural network**: It examines small windows of an image to decide whether each window contains a face.

1. It first applies a set of neural network-based filters to an image.
2. It uses an arbitrator to combine the outputs.
3. The filters examine each location looking for face.
4. The arbitrator then merges detections from individual filters and eliminate overlapping detections.

   Advantages: (i) This method produces good detection rates.
   (ii) Same algorithm can be applied for detection of car tires and human eyes.

   Disadvantages: (i) It only detects upright faces looking at the cameras.
   (ii) This methodology is complex.

- **CONCLUSION:**

  We came to understand the various challenges faced in face detection. From this review , we can conclude that it is very important to remove background information.

  We also conclude that following things make face detection more complicated.

  1.Different Facial poses.

  2.Complex background.

  3.Varied facial expressions.

  4.Overlapping Faces.

PROPOSED METHODOLOGY

- First, a classifier (namely a cascade of boosted classifiers working with haar-like features) is trained with a few hundred sample views of a particular object (i.e., a face or a car), called positive examples, that are scaled to the same size (say, 20x20), and negative examples - arbitrary images of the same size.

- After a classifier is trained, it can be applied to a region of interest (of the same size as used during the training) in an input image. The classifier outputs a “1” if the region is likely to show the object (i.e., face/car), and “0” otherwise. To search for the object in the whole image one can move the search window across the image and check every location using the classifier. The classifier is designed so that it can be easily “resized” in order to be able to find the objects of interest at different sizes, which is more

efficient than resizing the image itself. So, to find an object of an unknown size in the image the scan procedure should be done several times at different scales.

- The word “cascade” in the classifier name means that the resultant classifier consists of several simpler classifiers (stages) that are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed. The word “boosted” means that the classifiers at every stage of the cascade are complex themselves and they are built out of basic classifiers using one of four different boosting techniques (weighted voting). Currently Discrete Adaboost, Real Adaboost, Gentle Adaboost and Logitboost are supported.

Description of work division in terms of role among the students:

Prabhat Kumar: Work on data set generator part , , we will be detecting multiple people’s faces along with their labels. We will place some images to train our classifier to recognize multiple people then we will add every person’s folder as different labels and then we will be adding their names along with labels.

Divyansh kumar : Work on recognition part which recognizes the person In our project, by using OpenCV in python, we will be detecting multiple people’s faces along with their labels. : It examines small windows of an image to decide whether each window contains a face. We will be predicting faces realtime via our webcam.

Jyotish Kumar: Work on trainer part, After a classifier is trained, it can be applied to a region of interest (of the same size as used during the training) in an input image. The classifier is designed so that it can be easily “resized” in order to be able to find the objects of interest at different sizes.

CONCLUSION

- Not only has this technology facilitated our day-to-day lives and replaced old technology but there are many additional benefits: improved security, process automation, fast and accurate, non-intrusive using real-time video capture.

FACE RECOGNITION FOR PUBLIC SECURITY

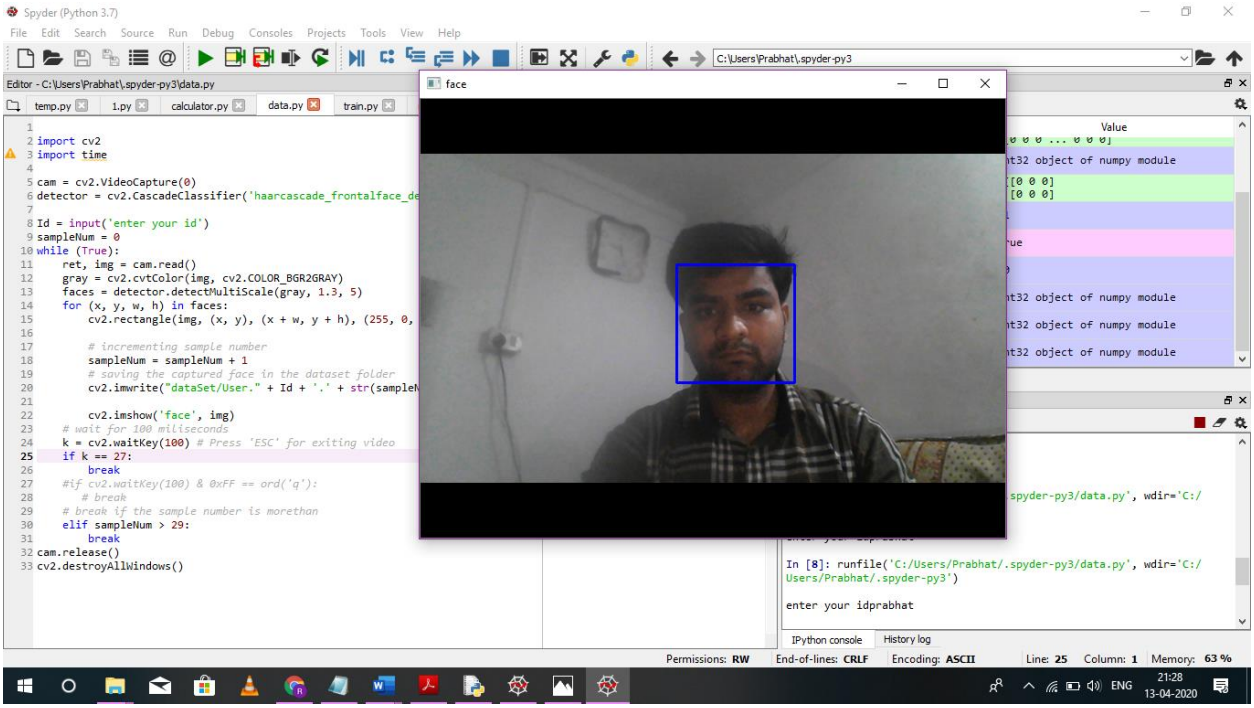
- The use of FR will better the security status around smart cities by taking two distinguished forms: Single and Dual factor authentication factors.

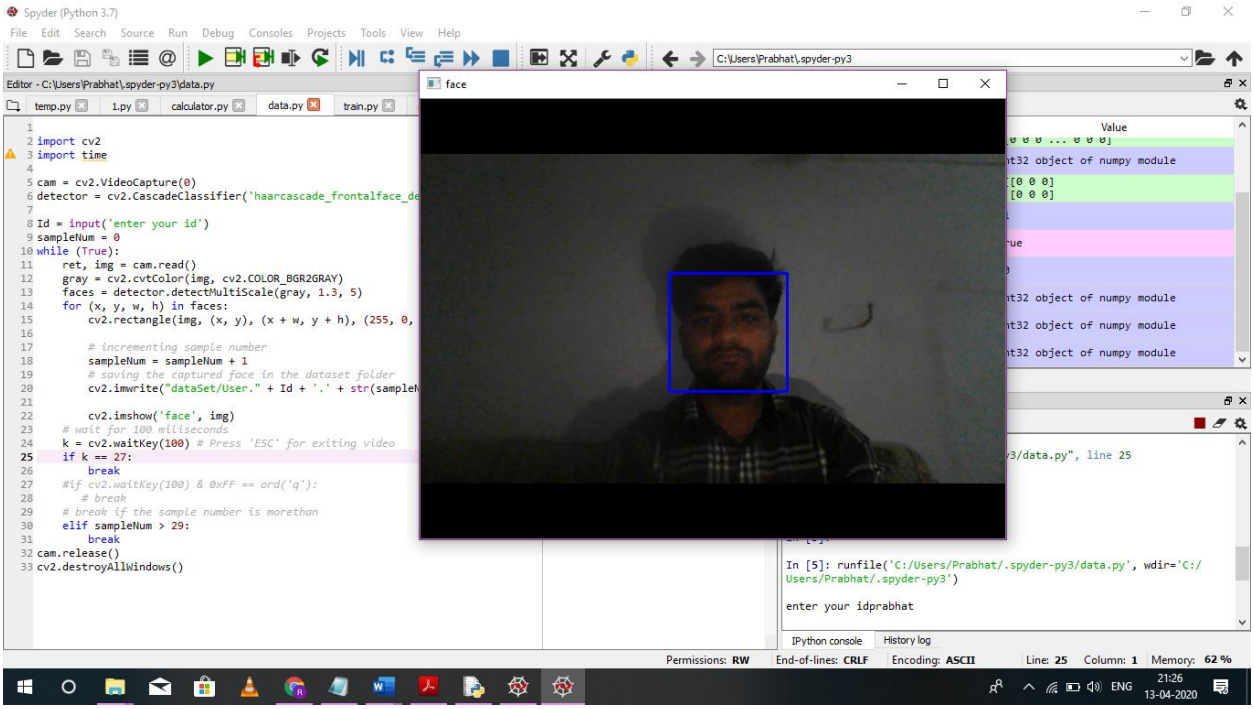
SMART SURVEILLANCE IS BEING USED IN ALL INDUSTRIES

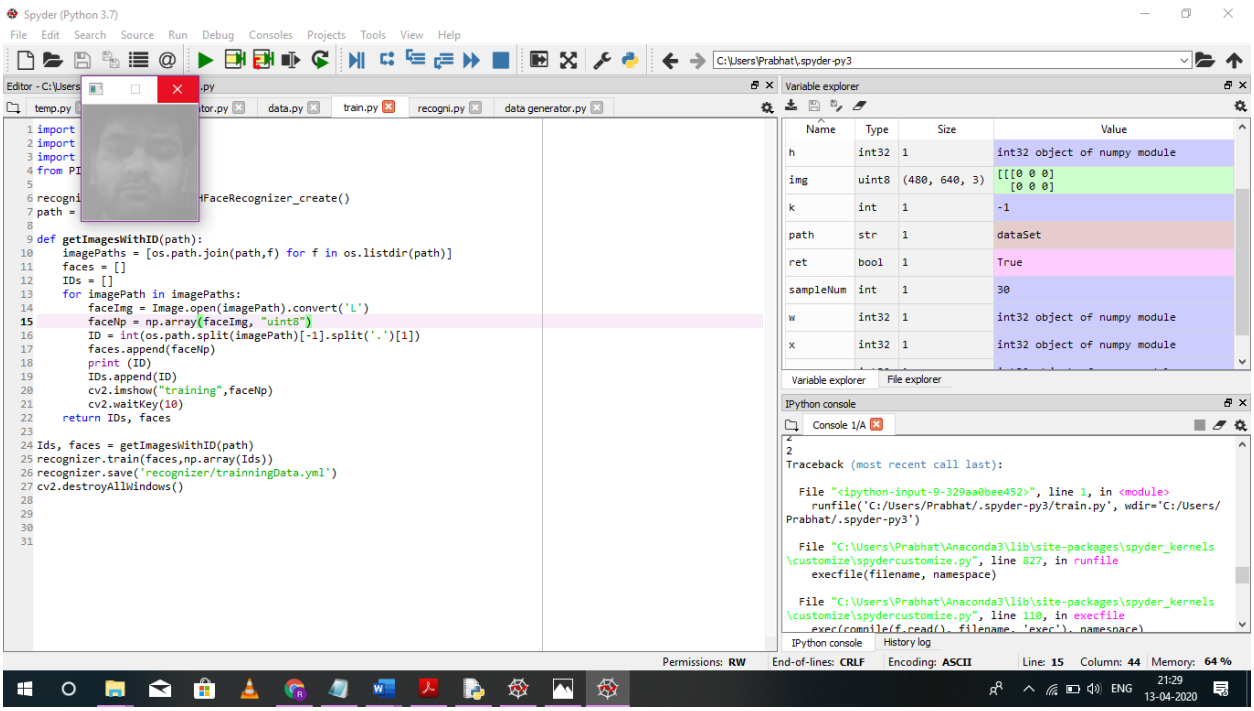
- **RETAIL STORES**: We can use FR to track shopping habits. Making sense of visuals.
- **RELIGIOUS OUTLETS**: Religious Institutions also use FR to gather data on who has been attending the religious ceremonies. This data is used to identify the faces who haven't been showing up so they can be contacted again.
- **EDUCATIONAL INSTITUTIONS**: In various schools, not only on students, but also on teachers, FR is used accurately to track the break timings on teachers without them knowing anything.
- **MEDICAL CENTERS**: Very beneficial uses of FR in the medical field has helped save many lives.Smart engines are able to diagnose genetic disorders that have an effect on facial impressions. FR is generating "BIG DATA" and with this we make sense of collected data.
- **PERSONAL LEVEL**: On Instagram and Snapchat , we can take a nice shot easily without wasting time on taking the right angle and also replace our faces with a celebrity's face with the help of "face transplant effect".

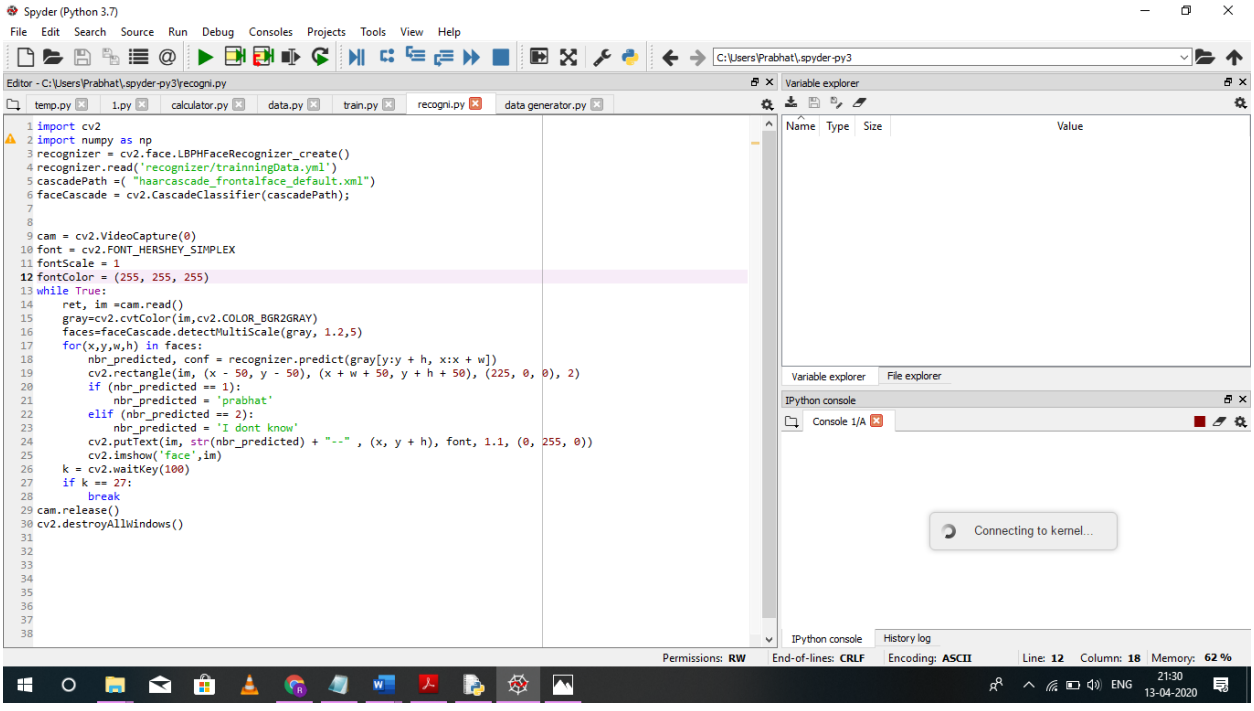
This is only the beginning as it will continue to change the way we live!

Sample output:






Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Prabhat\.spyder-py3
Editor - C:\Users\Prabhat\.spyder-py3\train.py
temp.py
data.py
train.py
recogni.py
data generator.py
1 import
2 import
3 import
4 from PI
6 recogni HFaceRecognizer_create()
7 path =
9 def getImagesWithID(path):
10 imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
11 faces = []
12 IDs = []
13 for imagePath in imagePaths:
14 faceImg = Image.open(imagePath).convert('L')
15 faceNp = np.array(faceImg, "uint8")
16 ID = int(os.path.split(imagePath)[-1].split('.')[1])
17 faces.append(faceNp)
18 print (ID)
19 IDs.append(ID)
20 cv2.imshow("training",faceNp)
21 cv2.waitKey(10)
22 return IDs, faces
24 Ids, faces = getImagesWithID(path)
25 recognizer.train(faces,np.array(Ids))
26 recognizer.save('recognizer/trainningData.yml')
27 cv2.destroyAllWindows()
Variable explorer
Name Type Size Value
h int32 1 int32 object of numpy module
img uint8 (480, 640, 3) [[[0 0 0] [0 0 0]
k int 1 -1
path str 1 dataSet
ret bool 1 True
sampleNum int 1 30
w int32 1 int32 object of numpy module
x int32 1 int32 object of numpy module
Variable explorer File explorer
IPython console
Console 1/A
2
2
Traceback (most recent call last):
File "<ipython-input-9-329aa0bee452>", line 1, in <module>
runfile('C:/Users/Prabhat/.spyder-py3/train.py', wdir='C:/Users/Prabhat/.spyder-py3')
File "C:\Users\Prabhat\Anaconda3\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 827, in runfile
execfile(filename, namespace)
File "C:\Users\Prabhat\Anaconda3\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 110, in execfile
IPython console History log
Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 15 Column: 44 Memory: 64 %
ENG 21:29 13-04-2020


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Prabhat\.spyder-py3
Editor - C:\Users\Prabhat\.spyder-py3\recogni.py
temp.py
1.py
calculator.py
data.py
train.py
recogni.py
data generator.py
1 import cv2
2 import numpy as np
3 recognizer = cv2.face.LBPHFaceRecognizer_create()
4 recognizer.read('recognizer/trainningData.yml')
5 cascadePath =("haarcascade_frontalface_default.xml")
6 faceCascade = cv2.CascadeClassifier(cascadePath);
9 cam = cv2.VideoCapture(0)
10 font = cv2.FONT_HERSHEY_SIMPLEX
11 fontScale = 1
12 fontColor = (255, 255, 255)
13 while True:
14 ret, im =cam.read()
15 gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
16 faces=faceCascade.detectMultiScale(gray, 1.2,5)
17 for(x,y,w,h) in faces:
18 nbr_predicted, conf = recognizer.predict(gray[y:y + h, x:x + w])
19 cv2.rectangle(im, (x - 50, y - 50), (x + w + 50, y + h + 50), (225, 0, 0), 2)
20 if (nbr_predicted == 1):
21 nbr_predicted = 'prabhat'
22 elif (nbr_predicted == 2):
23 nbr_predicted = 'I dont know'
24 cv2.putText(im, str(nbr_predicted) + "--" , (x, y + h), font, 1.1, (0, 255, 0))
25 cv2.imshow('face',im)
26 k = cv2.waitKey(100)
27 if k == 27:
28 break
29 cam.release()
30 cv2.destroyAllWindows()
Variable explorer
Name Type Size Value
Variable explorer File explorer
IPython console
Console 1/A
Connecting to kernel...
IPython console History log
Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 12 Column: 18 Memory: 62 %
ENG 21:30 13-04-2020