FACE RECOGNITION USING OPENCV AND PYTHON

In this project we are going to study about how to identify the name of a person by identifying the image of a person.

Before we doing this project we want the following softwares:

1. Python 3.7.2
2. Opencv 3.0
3. Numpy 1.9.0

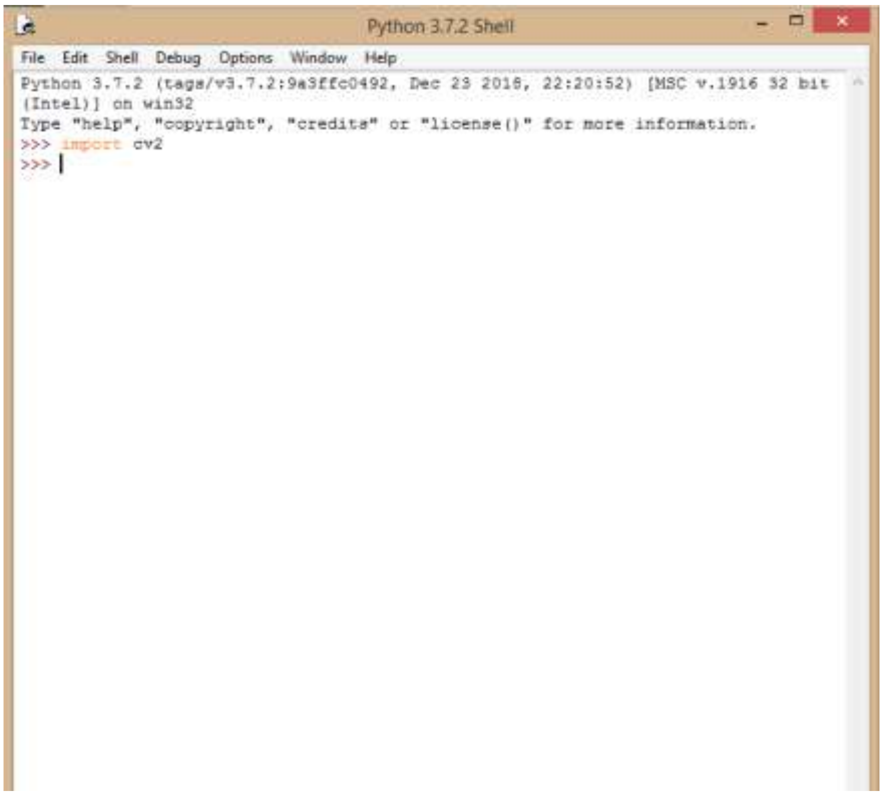
Python is a high level programming language that is used for high level projects such as image processing, prototyping, GUI based desktop applications.

Here we are using python for image processing with the help of its libraries. First of all we want to install the python 3.7.2 you can use the link below given. https://www.python.org/downloads/

After installing the python you have to add the opencv library to python by using pip. That is here I have used the 3.0 version of opencv we can install it in our laptop or pc by using the help of command prompt window. This is the code which we have to type in the command window for installing opencv :

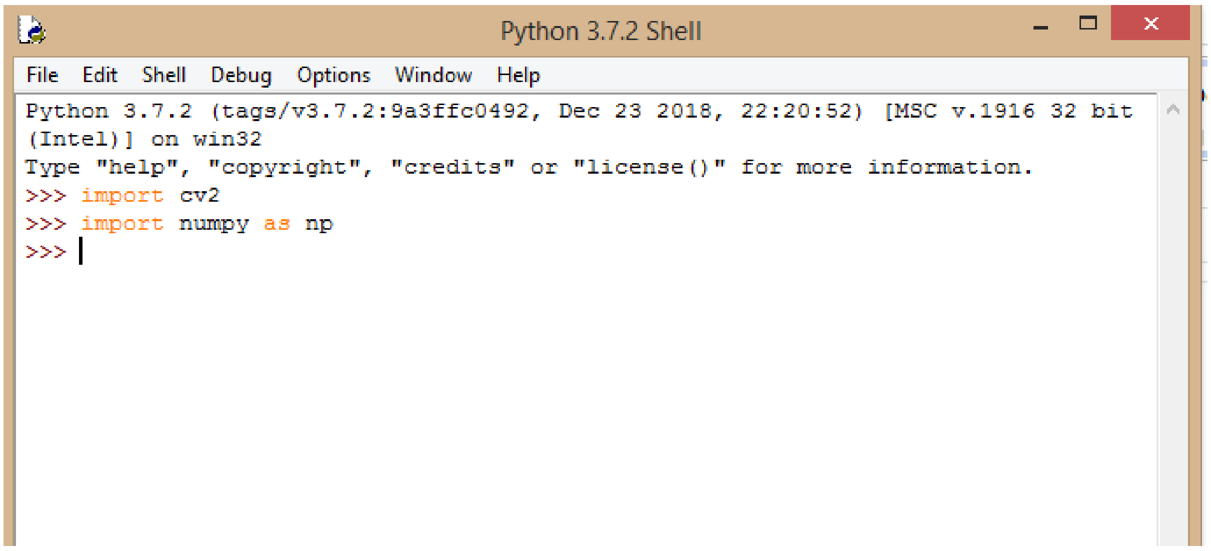
‘pip install opencv-python’.

After the successful installation of opencv we have to check that opencv is being added to python for that we have to open the IDLE python and type ‘import cv2’ if no error is detected we can say that opencv is being added successfully



```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import cv2
>>> |
```

After that is ‘pip install numpy’. For checking that numpy is added to the python we have to type ‘import numpy as Similarly we can install numpy also by using the help of the command window np’ in IDLE python. If no error is detected as below shown then you have successfully added numpy to the python.



```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import cv2
>>> import numpy as np
>>> |
```

Know we have completed the major three operations by installing python and adding our two libraries of python that is opencv and numpy.

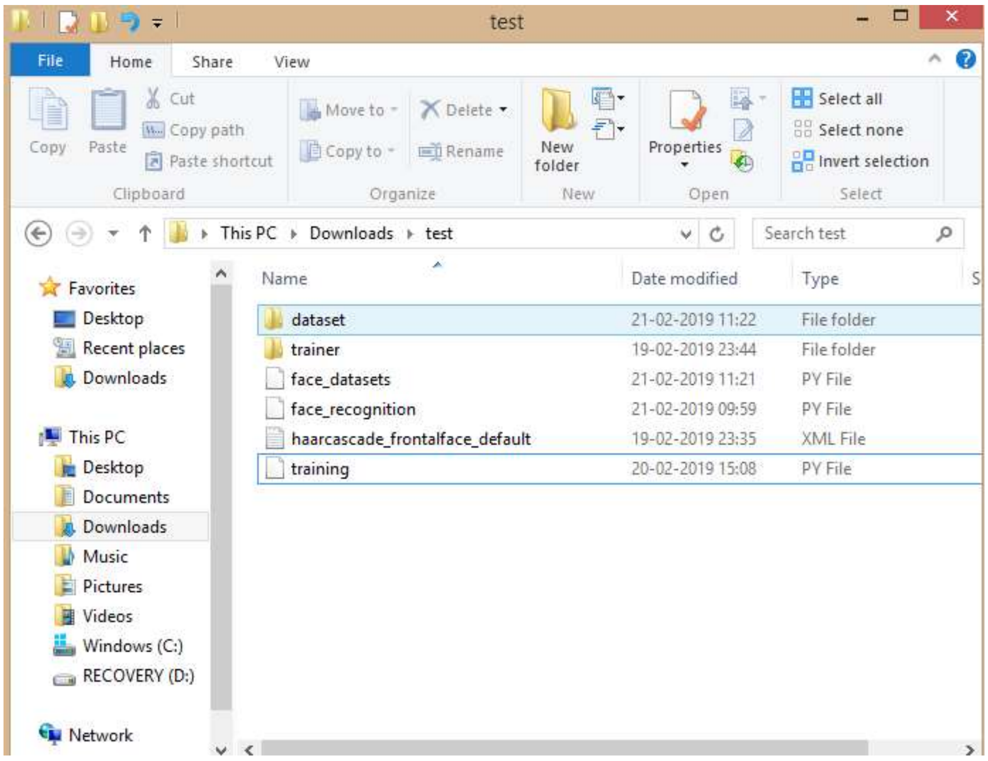
Next is the program section. In here we can divide our program into three sections that is:

1. **FACE DETECTION DATASET**

2. **TRAINING**

3. **FACE RECOGNITION**

We can take the first section that is face detection dataset. Before we start the programs we have to create a two folders in C and name it as 'dataset' and 'trainer'. And also we have to place these two folders together and our three section of programs in a main folder and want to paste a 'haarcascade frontalface default.xml' file also in that main folder like shown below:



Name	Date modified	Type
dataset	21-02-2019 11:22	File folder
trainer	19-02-2019 23:44	File folder
face_datasets	21-02-2019 11:21	PY File
face_recognition	21-02-2019 09:59	PY File
haarcascade_frontalface_default	19-02-2019 23:35	XML File
training	20-02-2019 15:08	PY File

We can come to our first section that is **FACE DETECTION DATASET** in here we are used to capture our images by using the videocapture of our webcam and the images is stored in dataset folder. The program is shown below:

```
# Import OpenCV2 for image processing

import cv2

import os


def assure_path_exists(path):

    dir = os.path.dirname(path)

    if not os.path.exists(dir):

        os.makedirs(dir)


# Start capturing video

vid_cam = cv2.VideoCapture(0)


# Detect object in video stream using Haarcascade Frontal Face

face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')


# For each person, one face id

face_id = 6


# Initialize sample face image

count = 0


assure_path_exists("dataset/")
```

```
# Start looping
while(True):

    # Capture video frame
    _, image_frame = vid_cam.read()

    # Convert frame to grayscale
    gray = cv2.cvtColor(image_frame, cv2.COLOR_BGR2GRAY)

    # Detect frames of different sizes, list of faces rectangles
    faces = face_detector.detectMultiScale(gray, 1.3, 5)

    # Loops for each faces
    for (x,y,w,h) in faces:

        # Crop the image frame into rectangle
        cv2.rectangle(image_frame, (x,y), (x+w,y+h), (255,0,0), 2)

        # Increment sample face image
        count += 1

        # Save the captured image into the datasets folder
        cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count) + ".jpg",
gray[y:y+h,x:x+w])

        # Display the video frame, with bounded rectangle on the person's face
        cv2.imshow('frame', image_frame)
    # To stop taking video, press 'q' for at least 100ms
    if cv2.waitKey(100) & 0xFF == ord('q'):
        break

    # If image taken reach 100, stop taking video
    elif count>100:
        break

# Stop video
vid_cam.release()

# Close all started windows
cv2.destroyAllWindows()
```

We can come to our second section that is **TRAINING** in here we are used to train our images and to store it in the trainer folder as a trainer.xml file the use of training is to identify the images to the computer. The program is shown below:

```
# Import OpenCV2 for image processing
# Import os for file path
import cv2, os

# Import numpy for matrix calculation
import numpy as np
```

```
# Import Python Image Library (PIL)
from PIL import Image

import os

def assure_path_exists(path):
    dir = os.path.dirname(path)
    if not os.path.exists(dir):
        os.makedirs(dir)

# Create Local Binary Patterns Histograms for face recognization
recognizer = cv2.face.LBPHFaceRecognizer_create()

# Using prebuilt frontal face training model, for face detection
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");

# Create method to get the images and label data
def getImagesAndLabels(path):

    # Get all file path
    imagePaths = [os.path.join(path,f) for f in os.listdir(path)]

    # Initialize empty face sample
    faceSamples=[]

    # Initialize empty id
```

```
ids = []

# Loop all the file path
for imagePath in imagePaths:

    # Get the image and convert it to grayscale
    PIL_img = Image.open(imagePath).convert('L')

    # PIL image to numpy array
    img_numpy = np.array(PIL_img,'uint8')

    # Get the image id
    id = int(os.path.split(imagePath)[-1].split(".")[1])

    # Get the face from the training images
    faces = detector.detectMultiScale(img_numpy)

    # Loop for each face, append to their respective ID
    for (x,y,w,h) in faces:

        # Add the image to face samples
        faceSamples.append(img_numpy[y:y+h,x:x+w])

        # Add the ID to IDs
        ids.append(id)
    # Pass the face array and IDs array
    return faceSamples,ids

# Get the faces and IDs
faces,ids = getImagesAndLabels('dataset')

# Train the model using the faces and IDs
recognizer.train(faces, np.array(ids))

# Save the model into trainer.yml
assure_path_exists('trainer/')
recognizer.save('trainer/trainer.yml')
```

After running the program and if no error is coming you can step to the next program of face recognition or if any error is detected you have to install opencv contrib python on the command window that is:

Open the python37-32 folder and click the folder scripts and press shift and right click in the folder you can get an option for using the command window in there we have to type 'pip install opencv-contrib-python'. After the successful completion the program will be executed correctly.

Next we can move on to the **FACE RECOGNITION** program for recognizing or identifying our images by using the help of a webcam. The program is shown below:

```
# Import OpenCV2 for image processing
import cv2

# Import numpy for matrices calculations
import numpy as np
import os

def assure_path_exists(path):
    dir = os.path.dirname(path)
    if not os.path.exists(dir):
        os.makedirs(dir)

# Create Local Binary Patterns Histograms for face recognization
recognizer = cv2.face.LBPHFaceRecognizer_create()

assure_path_exists("trainer/")

# Load the trained mode
recognizer.read('trainer/trainer.yml')

# Load prebuilt model for Frontal Face
cascadePath = "haarcascade_frontalface_default.xml"

# Create classifier from prebuilt model
faceCascade = cv2.CascadeClassifier(cascadePath);

# Set the font style
font = cv2.FONT_HERSHEY_SIMPLEX

# Initialize and start the video frame capture
```

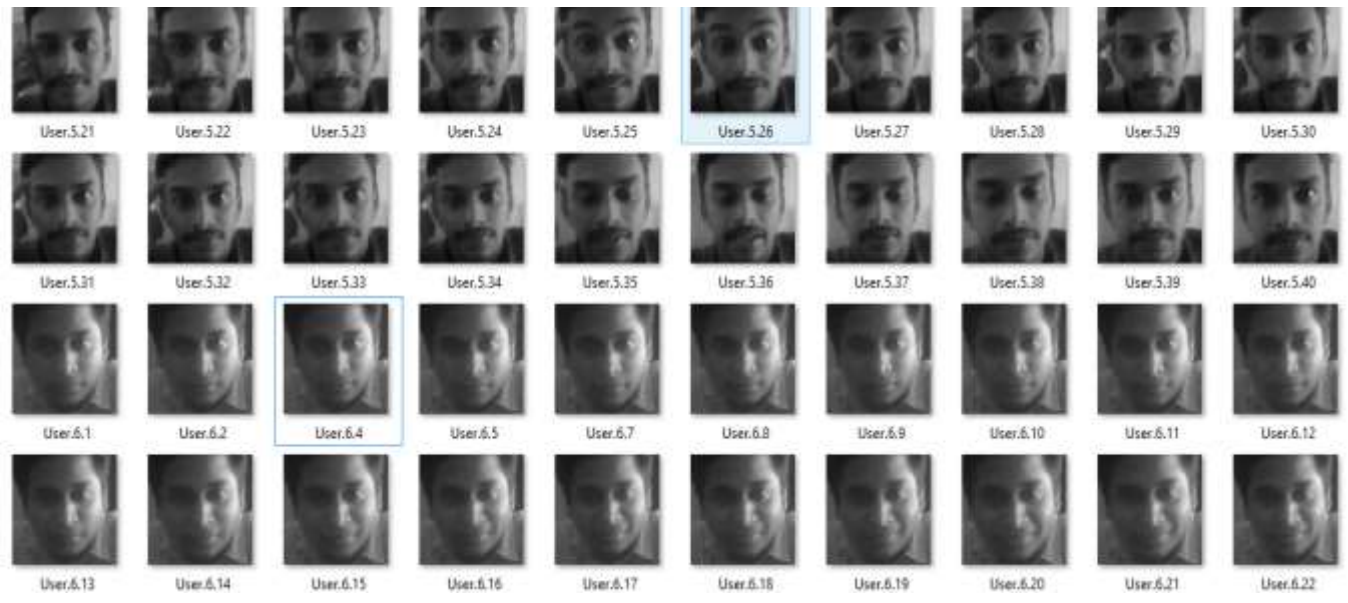
```
cam = cv2.VideoCapture(0)

# Loop
while True:
    # Read the video frame
    ret, im =cam.read()

    # Convert the captured frame into grayscale
    gray = cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)

    # Get all face from the video frame
    faces = faceCascade.detectMultiScale(gray, 1.2,5)

    # For each face in faces
    for(x,y,w,h) in faces:

        # Create rectangle around the face
        cv2.rectangle(im, (x-20,y-20), (x+w+20,y+h+20), (0,255,0), 4)

        # Recognize the face belongs to which ID
        Id, confidence = recognizer.predict(gray[y:y+h,x:x+w])

        # Check the ID if exist
        if(Id == 1):
            Id = "Dais {0:.2f}%".format(round(100 - confidence, 2))
        # Put text describe who is in the picture
        cv2.rectangle(im, (x-22,y-90), (x+w+22, y-22), (0,255,0), -1)
        cv2.putText(im, str(Id), (x,y-40), font, 1, (255,255,255), 3)

    # Display the video frame with the bounded rectangle
    cv2.imshow('im',im)

    # If 'q' is pressed, close program
    if cv2.waitKey(10) & 0xFF == ord('q'):
        break

# Stop the camera
cam.release()

# Close all windows
cv2.destroyAllWindows()
```

Now that we have completed the following sections of the programs we can execute the programs altogether as explained below

First we have to run the FACE DATASET program and the webcam will capture the images and it will be stored into the dataset folder. And we have to run the TRAINING program to train the images and to store that in a .xml file after that you have to run the final program that is FACE RECOGNITION program to identify the images and its corresponding name before that you have to store the name of the images that is been captured in the following program line:

```
# Check the ID if exist
    if(Id == 1):
      Id = "Dais {0:.2f}%".format(round(100 - confidence, 2))
```

In here you can change the name ‘Dais’ and can store the name of the image that has been stored not only that you have to change the ‘Id==1’ to the number which is your image is stored for example:



Here you can see the images stored in different number of user’s like that your images will also be stored in another number so you have to give that number in the place of ‘Id==’

Know that the project is completed you can check in your pc or laptop with the help of these programs and softwares.