Gokaraju Rangaraju Institute of Engineering and Technology

(Autonomous)

Department of Computer Science and Engineering

GR18A4061- Industry oriented Major Project

IV Year

Face Recognition
Review-3

Section : **D**

Batch No : 14

Batch Members:

1.18241A05I8	B. Sonu
2. 18241A05J7	D. Venkata Sai Ram
3. 18241A05L2	L J Atulya Reddy
4. 18241A05M1	P. Praveen Sai Varma

Project guide:

B. Srikanth

Assistant Professor
CSE Department

FACE RECOGNITION

USING

SOFT COMPUTING
&
PERFORMANCE COMPUTING

Introduction

Facial Recognition is a Biometric tool, This biometric technology is alternative for fingerprint recognition, iris recognition, Retinal recognition, Voice recognition.

Face Recognition is a technology of identification of a person based on the facial structures and other characteristics.

Nowadays face recognition technology used in different fields with varies purpose for safety, security and privacy such as **Security check points** at organization: Institutes, Hospitals, Malls, Airports; **Face ID unlock**: Phone, door, locker; **Finding** missing persons and criminals ….

Technologies

Software :

- Operating System: Windows 7,8,10 / Mac OS / Linux
- Language : Python 3
- OpenCV (Open Source Computer Vision Library)
- Dlib (ML Library)

Hardware :

Computer/Laptop with a camera with minimum 2megapixel.

Existing Approach

As Face Recognition used for varies purpose such as unlock system, finding person, registration, attendance... From the research we found, There are different algorithms & approach used in for Facial Recognition.They are all complex and consumes more amount of time to learn the algorithm for implementation and to generate the output. The accuracy in matching with other images is also less.

Following are few existing Systems:

1. **Facenet** – Face Recognition using Tensorflow (link)
2. **Deepface** – Face Analysis using deep learning (link)
3. **Compreface** – Free and open source Face recognition System (link)
4. **Algorithms** : Eigenfaces, Convolutional Neutral Networks(CNN),SURF (link)
5. **Approach** : Class Based, 3D Based, Image comparison based Appoarch

Challenges For Face Recognition

1. Difficult shooting conditions (backlight, low light)

2. Face turned (diagonally, downwards)

3. Sunglasses/masks

4. Aging

5. Facial Hair

6. Makeup

Theses are the some factors where face recognition system fail to recognize the faces

Proposed Approach

In the proposed approach, Face recognition system has three phases which recognize faces in real-time:

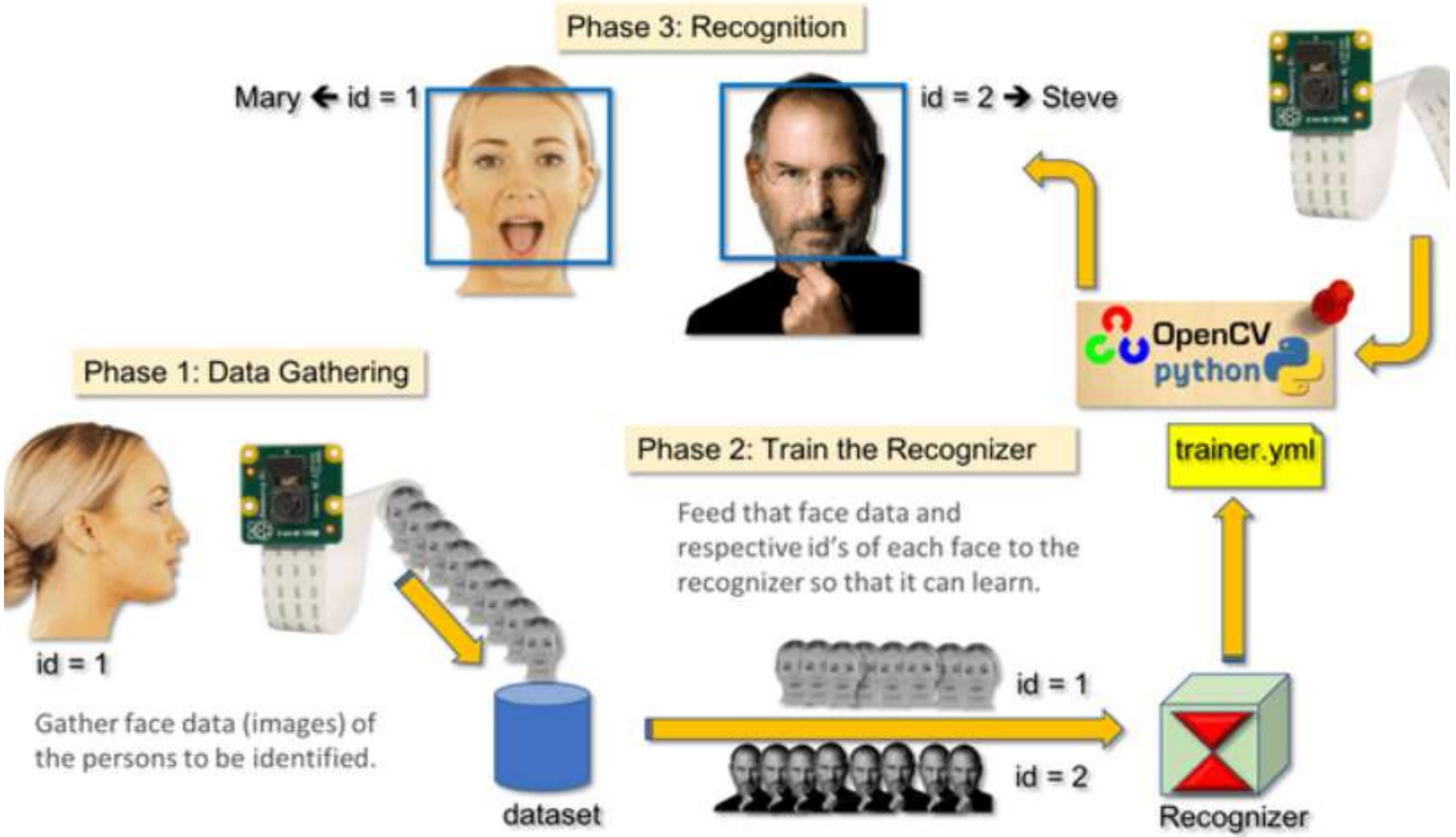
1. Face Detection and Data Gathering

2. Train the Recognizer/ Model

3. Testing Model : Face Recognition/ Identification

Among available Algorithm and Techniques: Eigen faces, CNN and many more. We will consider best and optimized way which is **LBPH (Local Binary Patterns Histograms)**Face recognition Algorithm**,** maintains high facial recognition accuracy and uses less time to train the model

System Architecture

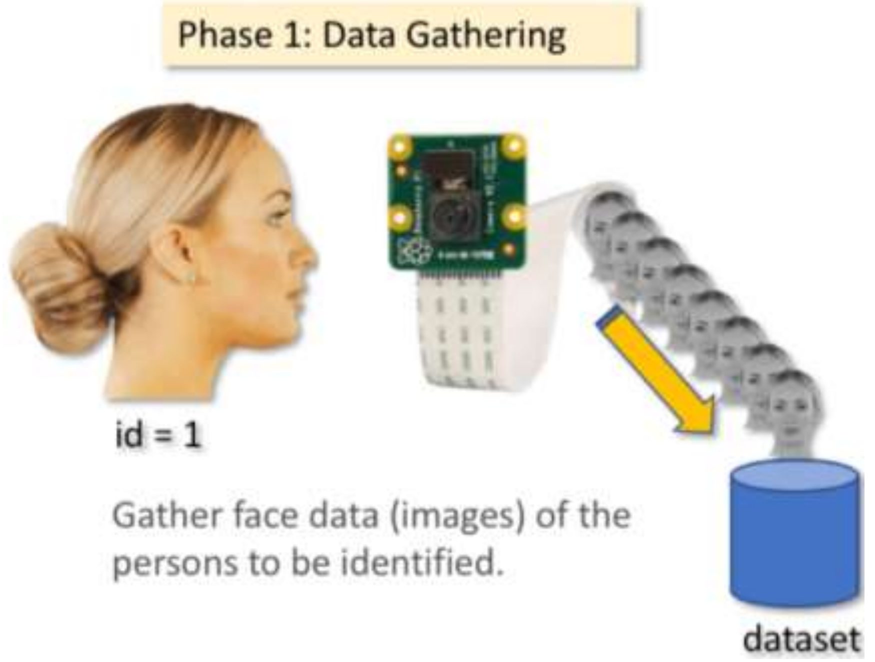

Phase 3: Recognition
Mary ← id = 1
id = 2 → Steve
OpenCV
python
Phase 1: Data Gathering
id = 1
Gather face data (images) of the persons to be identified.
dataset
Phase 2: Train the Recognizer
Feed that face data and respective id's of each face to the recognizer so that it can learn.
id = 1
id = 2
Recognizer
trainer.yml

Modules:

Though based purpose software can vary, the process of facial recognition tends to follow the three basic steps:

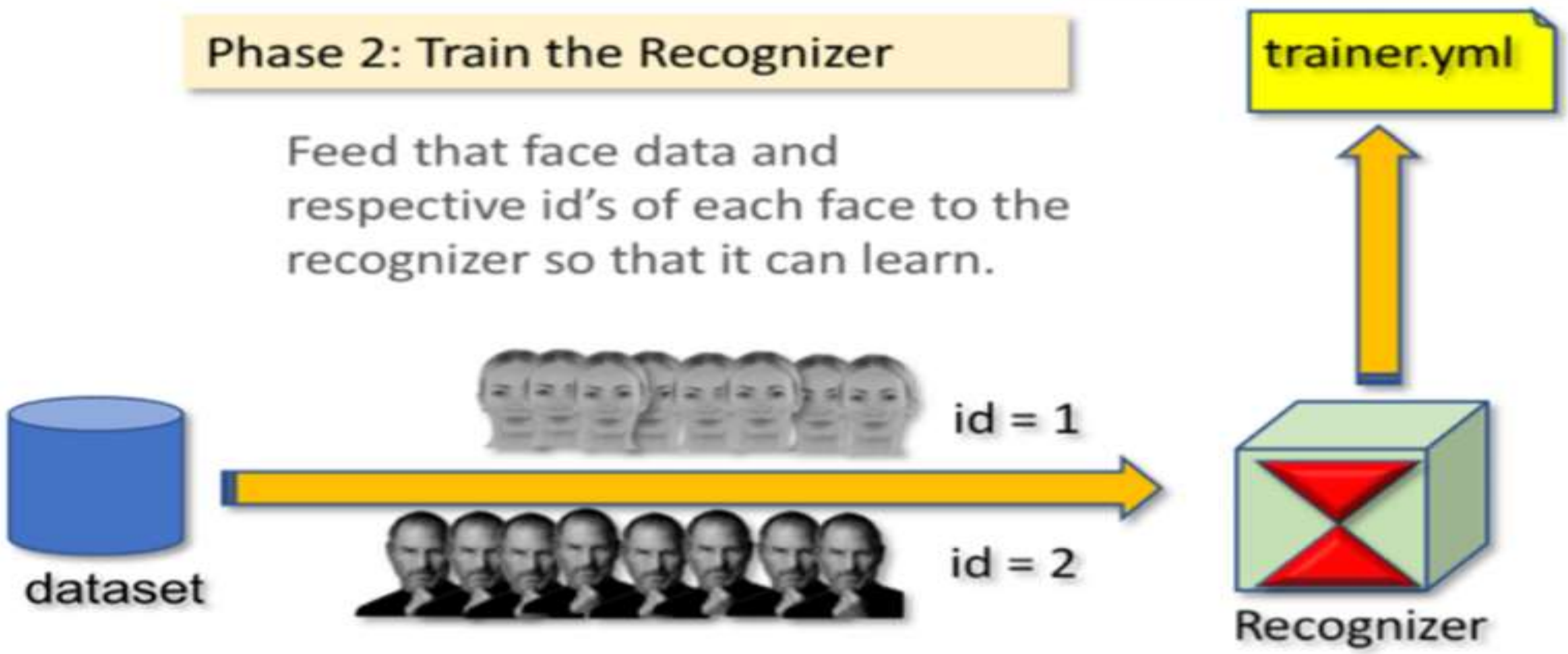
Taking Input: Acquiring the face image from the Camera. Before anything, you must “capture” a face (Phase1)in order to recognize it, when compared with a new face captured on future (Phase 3).

The most common way to detect a face (or any objects), is using the “Haar Cascade classifier” - effective object detection method.

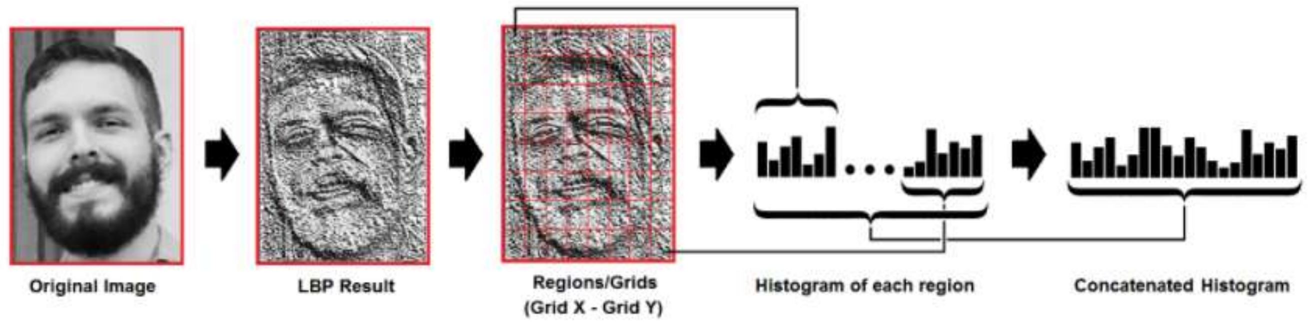


The algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it.

Data Gathering : After Face detection, multiple Captured face images are stored in a dataset folder with a face id.


Phase 2: Train the Recognizer
trainer.yml
Feed that face data and respective id's of each face to the recognizer so that it can learn.
id = 1
id = 2
dataset
Recognizer

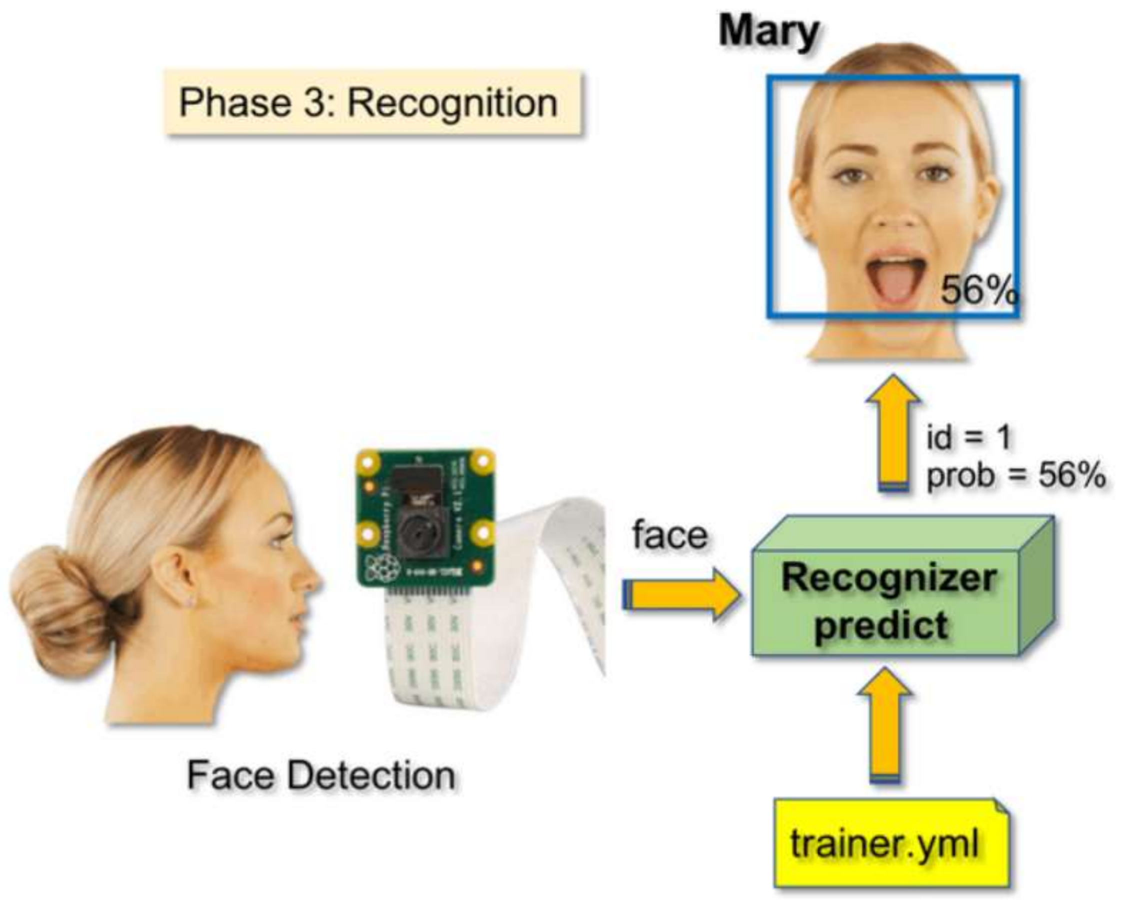
Face data(Grayscale image) with the respective id's will be trained with help of LBPH (LOCAL BINARY PATTERNS HISTOGRAMS) Face Recognizer, which converts the datasets(face, ids) into ".yml" file (store content network connections as a database file)



Phase 3 is a testing phase to check and test for identification process

we will capture a fresh face on our camera and if this person had his face captured and trained before, our recognizer will make a “prediction” returning its id and an index, shown how confident the recognizer is with this match.

Finally faces are detected and recognized by comparing the database for Identification



Implementation

- Download and Install latest Python version in PC

  (You can download it for free from the following website: https://www.python.org/)
- To check if you have python installed on a PC, Open the Terminal and type:

  python –version
- Install the required libraries, Open terminal and type:

  pip install opencv-python

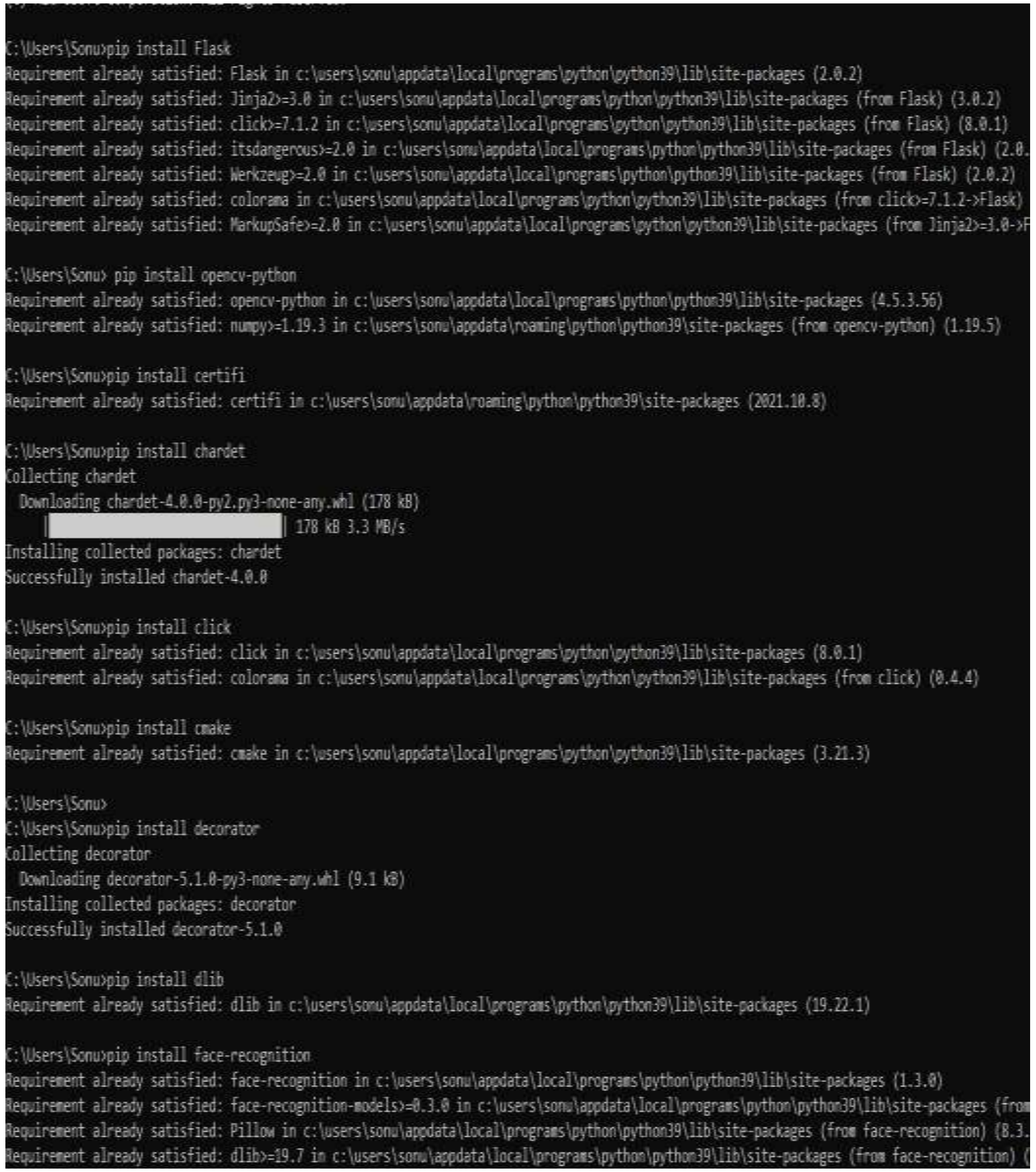
  pip install face-recognition

  pip install cmake

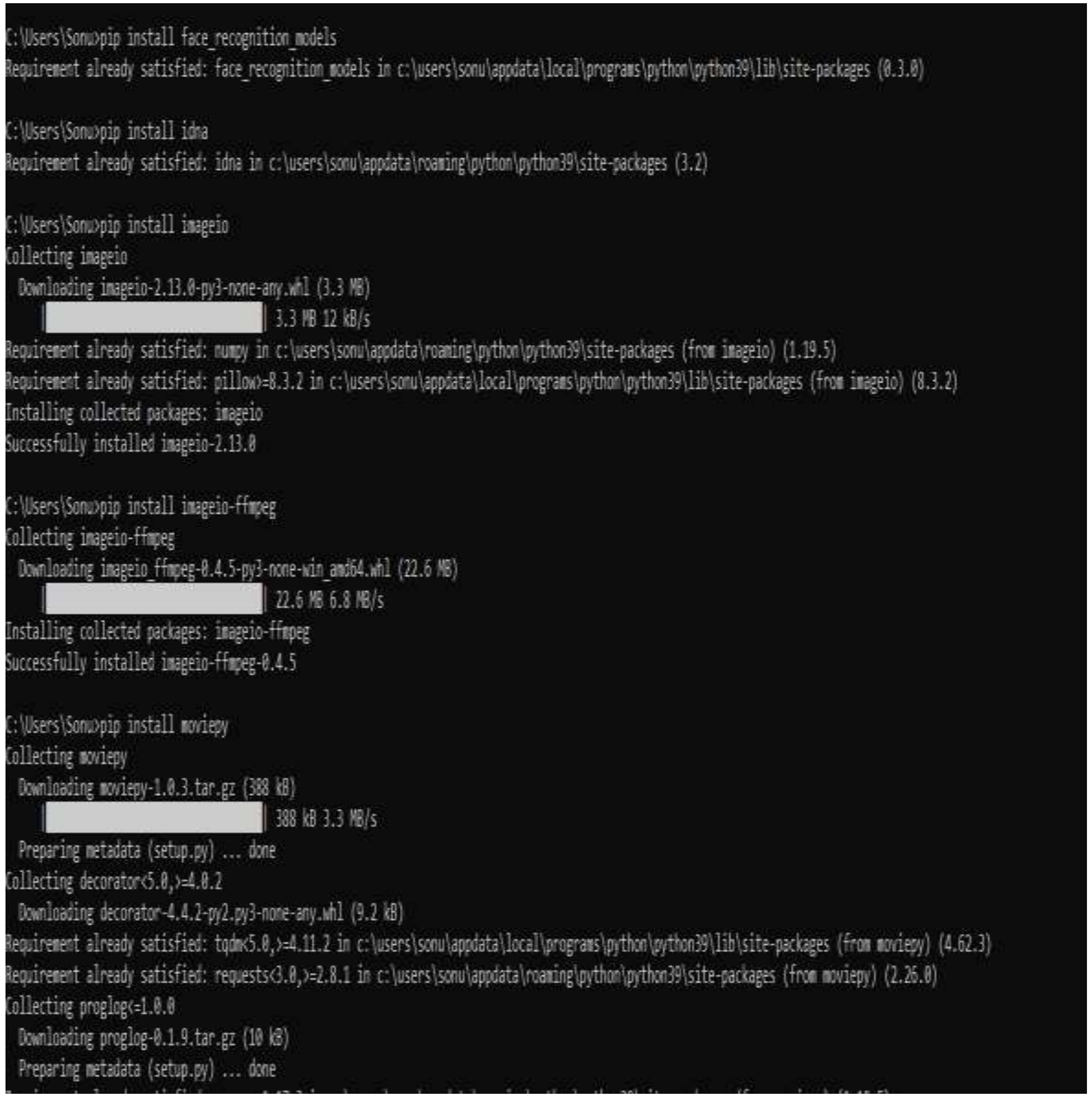
  pip install numpy ….,

  haarcascade_frontalface_default.xml file
- Code to implement face detection and taking the sample images into the dataset folder
- Code to implement to train the face recognition model for the datasets
- Code to identify faces by comparing the dataset and images

Snapshots of libraries:



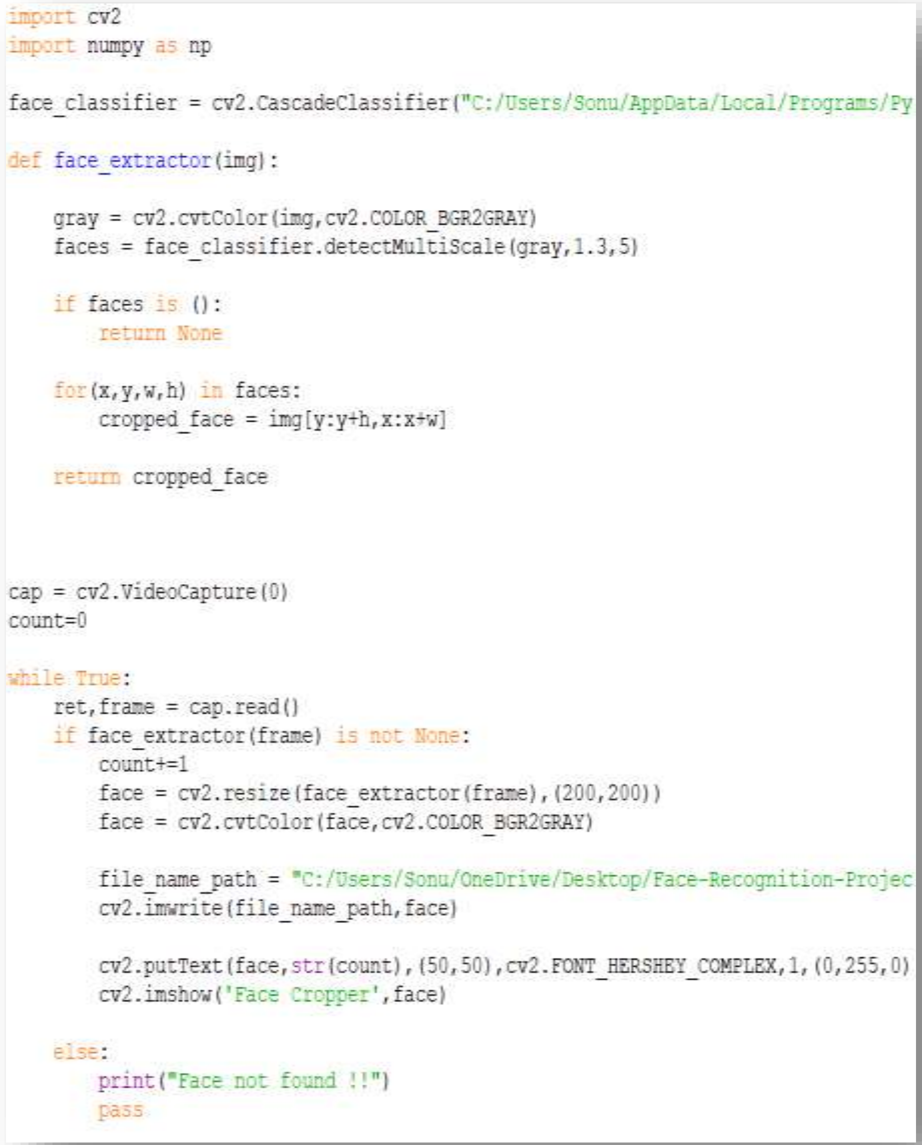
```
C:\Users\Sonu>pip install Flask
Requirement already satisfied: Flask in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (2.0.2)
Requirement already satisfied: Jinja2>=3.0 in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (from Flask) (3.0.2)
Requirement already satisfied: click>=7.1.2 in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (from Flask) (8.0.1)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (from Flask) (2.0.
Requirement already satisfied: Werkzeug>=2.0 in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (from Flask) (2.0.2)
Requirement already satisfied: colorama in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (from click>=7.1.2->Flask)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (from Jinja2>=3.0->F

C:\Users\Sonu> pip install opencv-python
Requirement already satisfied: opencv-python in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (4.5.3.56)
Requirement already satisfied: numpy>=1.19.3 in c:\users\sonu\appdata\roaming\python\python39\site-packages (from opencv-python) (1.19.5)

C:\Users\Sonu>pip install certifi
Requirement already satisfied: certifi in c:\users\sonu\appdata\roaming\python\python39\site-packages (2021.10.8)

C:\Users\Sonu>pip install chardet
Collecting chardet
  Downloading chardet-4.0.0-py2.py3-none-any.whl (178 kB)
     |████████████████████████████████| 178 kB 3.3 MB/s
Installing collected packages: chardet
Successfully installed chardet-4.0.0

C:\Users\Sonu>pip install click
Requirement already satisfied: click in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (8.0.1)
Requirement already satisfied: colorama in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (from click) (0.4.4)

C:\Users\Sonu>pip install cmake
Requirement already satisfied: cmake in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (3.21.3)

C:\Users\Sonu>
C:\Users\Sonu>pip install decorator
Collecting decorator
  Downloading decorator-5.1.0-py3-none-any.whl (9.1 kB)
Installing collected packages: decorator
Successfully installed decorator-5.1.0

C:\Users\Sonu>pip install dlib
Requirement already satisfied: dlib in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (19.22.1)

C:\Users\Sonu>pip install face-recognition
Requirement already satisfied: face-recognition in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (1.3.0)
Requirement already satisfied: face-recognition-models>=0.3.0 in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (from
Requirement already satisfied: Pillow in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (from face-recognition) (8.3.
Requirement already satisfied: dlib>=19.7 in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (from face-recognition) (
```



```
C:\Users\Sonu>pip install face_recognition_models
Requirement already satisfied: face_recognition_models in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (0.3.0)

C:\Users\Sonu>pip install idna
Requirement already satisfied: idna in c:\users\sonu\appdata\roaming\python\python39\site-packages (3.2)

C:\Users\Sonu>pip install imageio
Collecting imageio
  Downloading imageio-2.13.0-py3-none-any.whl (3.3 MB)
     |████████████████████████████████| 3.3 MB 12 kB/s
Requirement already satisfied: numpy in c:\users\sonu\appdata\roaming\python\python39\site-packages (from imageio) (1.19.5)
Requirement already satisfied: pillow>=8.3.2 in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (from imageio) (8.3.2)
Installing collected packages: imageio
Successfully installed imageio-2.13.0

C:\Users\Sonu>pip install imageio-ffmpeg
Collecting imageio-ffmpeg
  Downloading imageio_ffmpeg-0.4.5-py3-none-win_amd64.whl (22.6 MB)
     |████████████████████████████████| 22.6 MB 6.8 MB/s
Installing collected packages: imageio-ffmpeg
Successfully installed imageio-ffmpeg-0.4.5

C:\Users\Sonu>pip install moviepy
Collecting moviepy
  Downloading moviepy-1.0.3.tar.gz (388 kB)
     |████████████████████████████████| 388 kB 3.3 MB/s
  Preparing metadata (setup.py) ... done
Collecting decorator<5.0,>=4.0.2
  Downloading decorator-4.4.2-py2.py3-none-any.whl (9.2 kB)
Requirement already satisfied: tqdm<5.0,>=4.11.2 in c:\users\sonu\appdata\local\programs\python\python39\lib\site-packages (from moviepy) (4.62.3)
Requirement already satisfied: requests<3.0,>=2.8.1 in c:\users\sonu\appdata\roaming\python\python39\site-packages (from moviepy) (2.26.0)
Collecting proglog<=1.0.0
  Downloading proglog-0.1.9.tar.gz (10 kB)
  Preparing metadata (setup.py) ... done
```

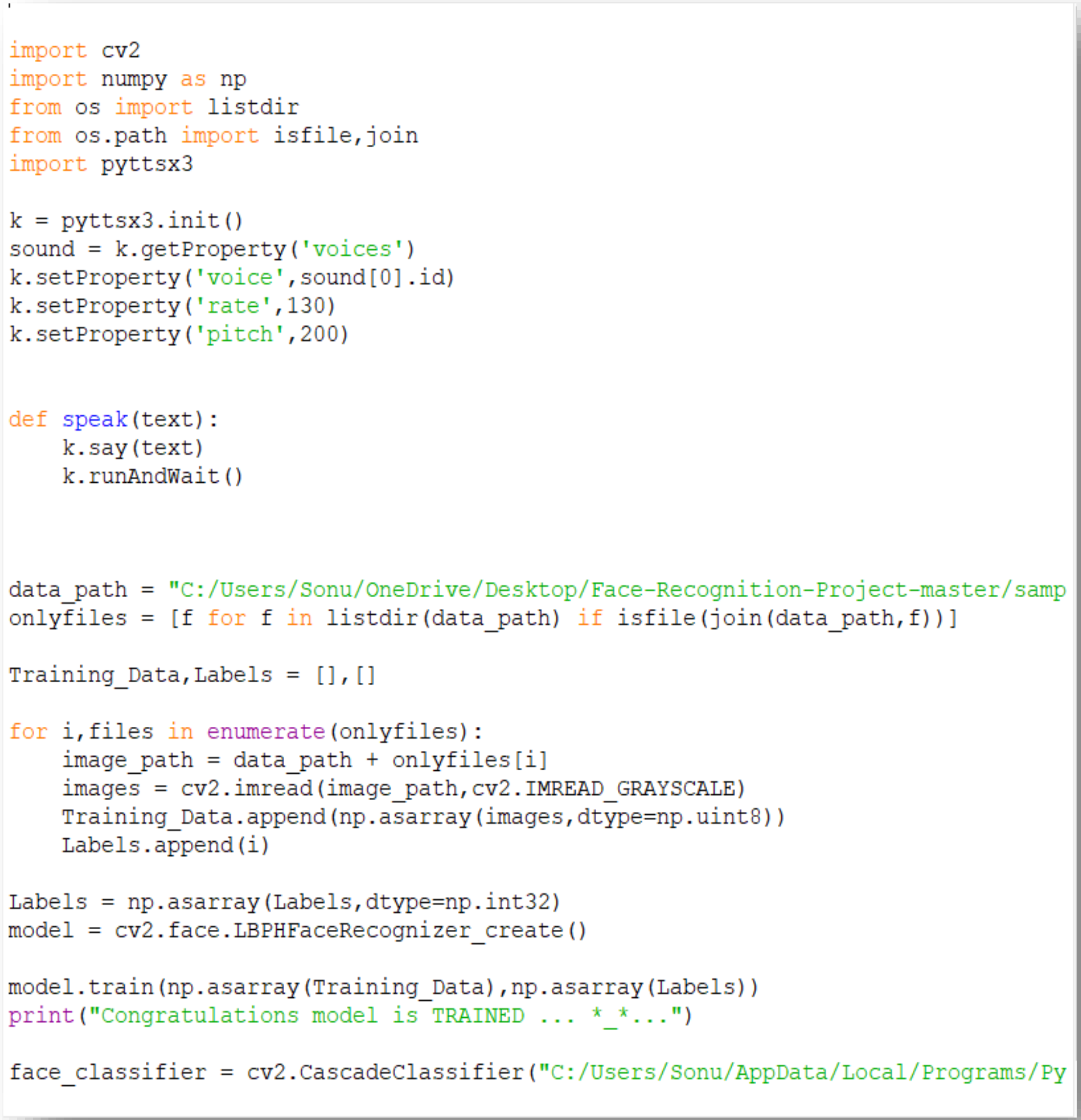
Snapshots of coding Part :

1. Face Detection and Data Gathering



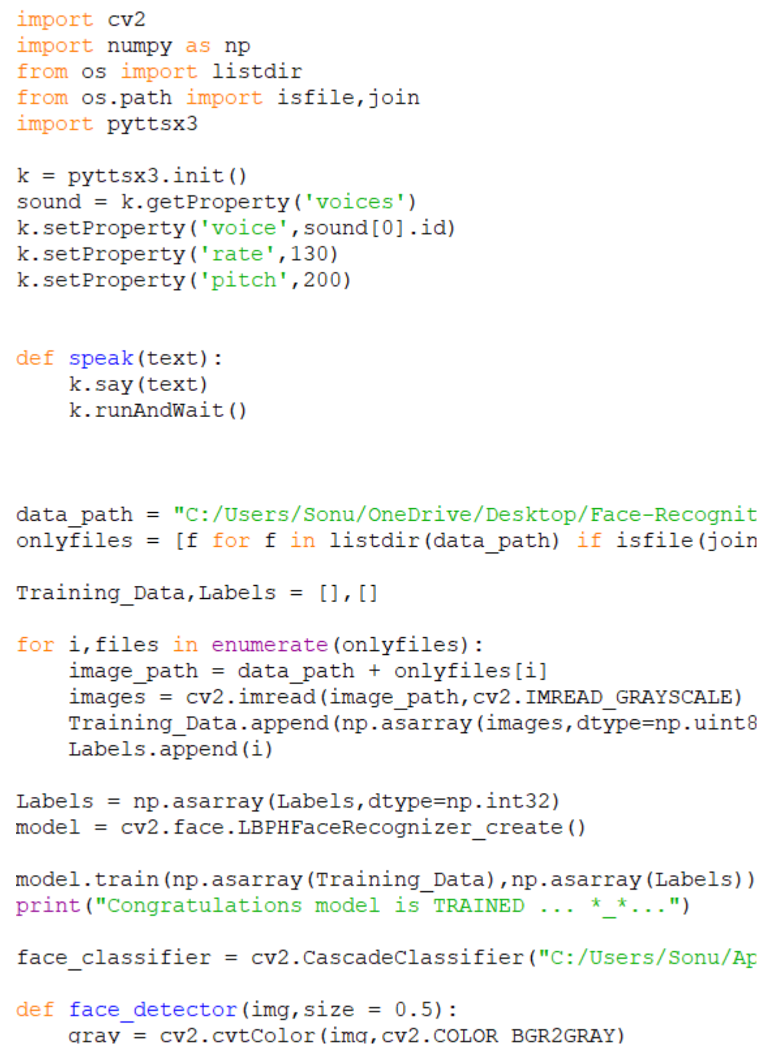
```
import cv2
import numpy as np

face_classifier = cv2.CascadeClassifier("C:/Users/Sonu/AppData/Local/Programs/Py

def face_extractor(img):

    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = face_classifier.detectMultiScale(gray,1.3,5)

    if faces is ():
        return None

    for(x,y,w,h) in faces:
        cropped_face = img[y:y+h,x:x+w]

    return cropped_face



cap = cv2.VideoCapture(0)
count=0

while True:
    ret,frame = cap.read()
    if face_extractor(frame) is not None:
        count+=1
        face = cv2.resize(face_extractor(frame),(200,200))
        face = cv2.cvtColor(face,cv2.COLOR_BGR2GRAY)

        file_name_path = "C:/Users/Sonu/OneDrive/Desktop/Face-Recognition-Projec
        cv2.imwrite(file_name_path,face)

        cv2.putText(face,str(count),(50,50),cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0)
        cv2.imshow('Face Cropper',face)

    else:
        print("Face not found !!")
        pass
```

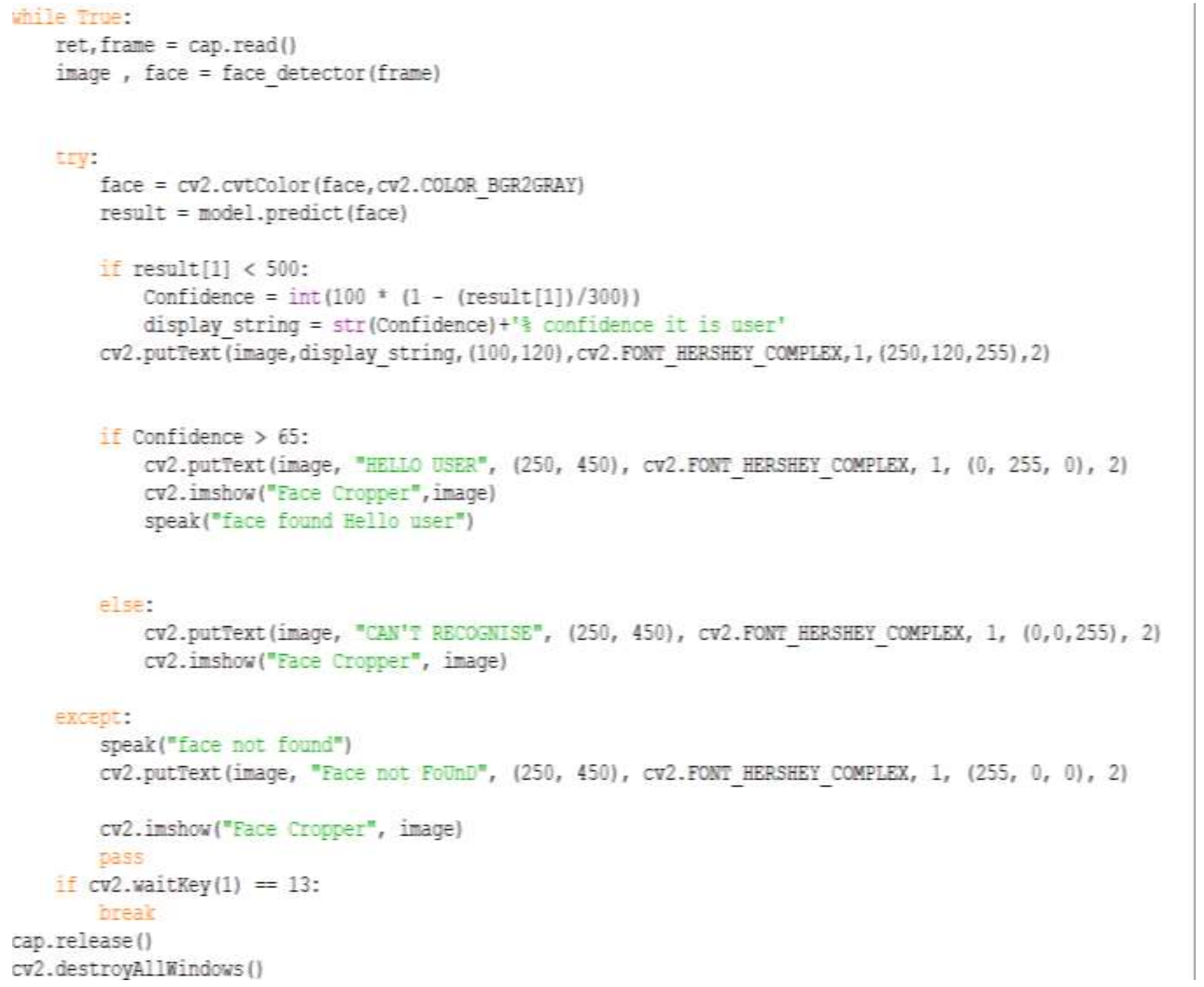
2. Training Model

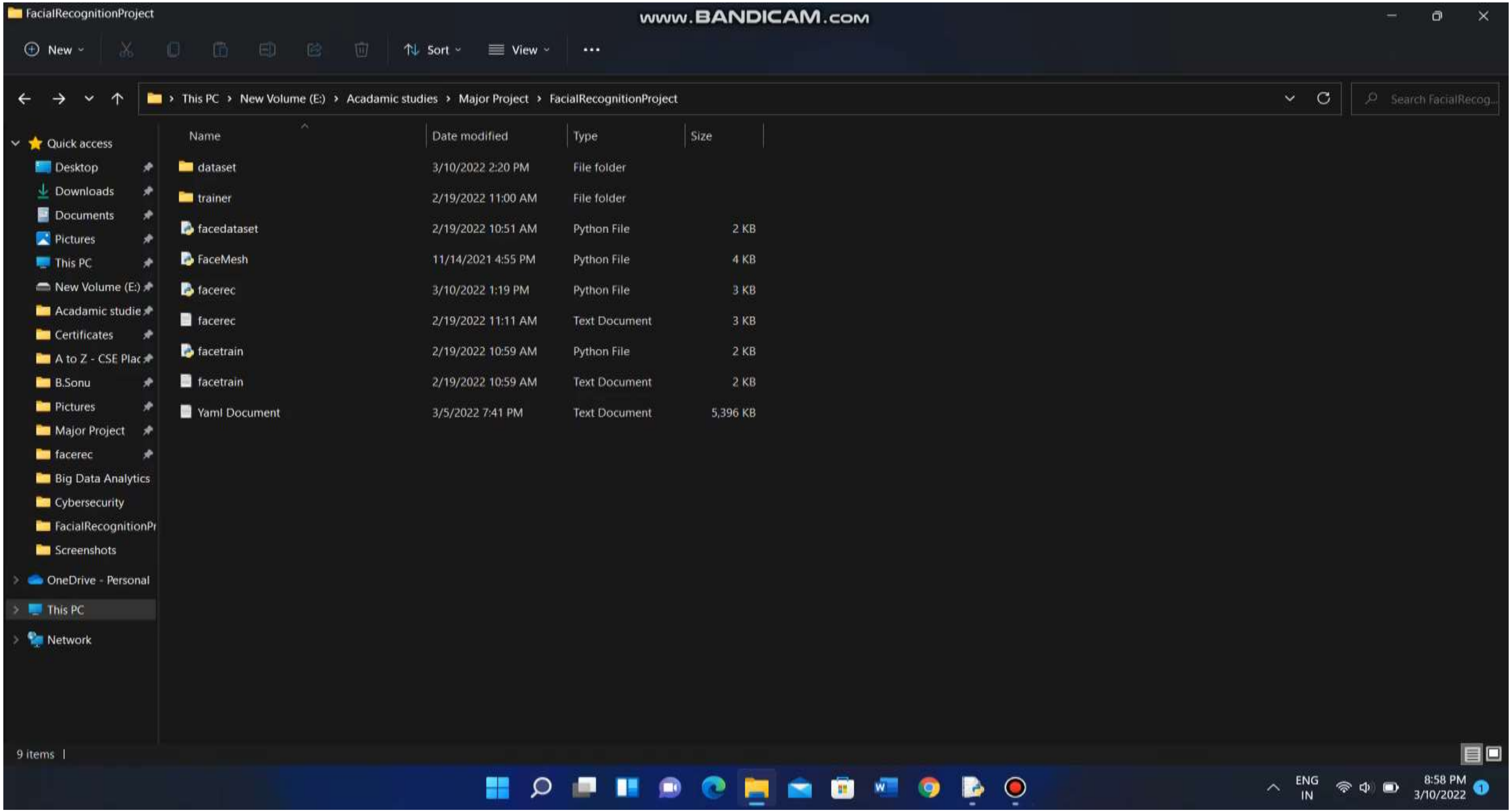


```
import cv2
import numpy as np
from os import listdir
from os.path import isfile,join
import pyttsx3

k = pyttsx3.init()
sound = k.getProperty('voices')
k.setProperty('voice',sound[0].id)
k.setProperty('rate',130)
k.setProperty('pitch',200)


def speak(text):
    k.say(text)
    k.runAndWait()



data_path = "C:/Users/Sonu/OneDrive/Desktop/Face-Recognition-Project-master/samp
onlyfiles = [f for f in listdir(data_path) if isfile(join(data_path,f))]

Training_Data,Labels = [],[]

for i,files in enumerate(onlyfiles):
    image_path = data_path + onlyfiles[i]
    images = cv2.imread(image_path,cv2.IMREAD_GRAYSCALE)
    Training_Data.append(np.asarray(images,dtype=np.uint8))
    Labels.append(i)

Labels = np.asarray(Labels,dtype=np.int32)
model = cv2.face.LBPHFaceRecognizer_create()

model.train(np.asarray(Training_Data),np.asarray(Labels))
print("Congratulations model is TRAINED ... *_*...")

face_classifier = cv2.CascadeClassifier("C:/Users/Sonu/AppData/Local/Programs/Py
```

3. Recognize Model


```
import cv2
import numpy as np
from os import listdir
from os.path import isfile,join
import pyttsx3

k = pyttsx3.init()
sound = k.getProperty('voices')
k.setProperty('voice',sound[0].id)
k.setProperty('rate',130)
k.setProperty('pitch',200)


def speak(text):
    k.say(text)
    k.runAndWait()



data_path = "C:/Users/Sonu/OneDrive/Desktop/Face-Recognit
onlyfiles = [f for f in listdir(data_path) if isfile(join

Training_Data,Labels = [],[]

for i,files in enumerate(onlyfiles):
    image_path = data_path + onlyfiles[i]
    images = cv2.imread(image_path,cv2.IMREAD_GRAYSCALE)
    Training_Data.append(np.asarray(images,dtype=np.uint8
    Labels.append(i)

Labels = np.asarray(Labels,dtype=np.int32)
model = cv2.face.LBPHFaceRecognizer_create()

model.train(np.asarray(Training_Data),np.asarray(Labels))
print("Congratulations model is TRAINED ... *_*...")

face_classifier = cv2.CascadeClassifier("C:/Users/Sonu/Ap

def face_detector(img,size = 0.5):
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
```


```
while True:
    ret,frame = cap.read()
    image , face = face_detector(frame)


    try:
        face = cv2.cvtColor(face,cv2.COLOR_BGR2GRAY)
        result = model.predict(face)

        if result[1] < 500:
            Confidence = int(100 * (1 - (result[1])/300))
            display_string = str(Confidence)+'% confidence it is user'
        cv2.putText(image,display_string,(100,120),cv2.FONT_HERSHEY_COMPLEX,1,(250,120,255),2)


        if Confidence > 65:
            cv2.putText(image, "HELLO USER", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
            cv2.imshow("Face Cropper",image)
            speak("face found Hello user")


        else:
            cv2.putText(image, "CAN'T RECOGNISE", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1, (0,0,255), 2)
            cv2.imshow("Face Cropper", image)

    except:
        speak("face not found")
        cv2.putText(image, "Face not FoUnD", (250, 450), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 0, 0), 2)

        cv2.imshow("Face Cropper", image)
        pass
    if cv2.waitKey(1) == 13:
        break
cap.release()
cv2.destroyAllWindows()
```


FacialRecognitionProject

This PC > New Volume (E:) > Acadamic studies > Major Project > FacialRecognitionProject

Name	Date modified	Type	Size
dataset	3/10/2022 2:20 PM	File folder	
trainer	2/19/2022 11:00 AM	File folder	
facedataset	2/19/2022 10:51 AM	Python File	2 KB
FaceMesh	11/14/2021 4:55 PM	Python File	4 KB
facerec	3/10/2022 1:19 PM	Python File	3 KB
facerec	2/19/2022 11:11 AM	Text Document	3 KB
facetrain	2/19/2022 10:59 AM	Python File	2 KB
facetrain	2/19/2022 10:59 AM	Text Document	2 KB
Yaml Document	3/5/2022 7:41 PM	Text Document	5,396 KB

9 items

References

- TowardsDataScience (link)
- SuperdataScience(link)
- Kimmel, Ron : Face Recognition Blog
- Github (Adam Geitgey, Sefik Ilkin Serengil)
- PyimageSearch
- Youtube channels :

  Pysource, Clever Programmer, Krish Naik

THANK YOU