BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

SAKSHAM CHOWDHARY

11802120

Roll No:-32

SECTION : K18KK

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

Submitted To: Mr. Sagar Pande

1.Abstract

At one of the most successful application of images analysis and understanding, face recognition has recently received significant attention, especially during the past few years, Face recognition technology(FRT) has emerged as an attractive solution to address many contemporary needs for identification and the verification of identity claims. It brings together the promise of other biometric system, which attempt to tie identity to individually distinctive features of the body, and the more familiar functionality of visual surveillance systems.

2. Introduction

2.1 Aim:

The aim of this project is to identify people in images or videos using pattern recognition techniques.

2.2 Motivation:

This project has drawn its motivation from the facial recognition techniques used virtually everywhere in the world for recognition (identification) purposes.

2.3 Objective:

The objective was to design and implement a face detector in python that will detect human faces in an image similar to the training images.

2.4 Scope:

- The facial recognition is use for ID verification services. Many companies and others are working in the market now to provide these services to banks, ICOs, and other e-businesses.
- Social Media platforms have adopted facial recognition capabilities to diversify their functionalities in order to attract a wider user base amidst stiff competition from different applications.

- A variety of phones including the latest iPhone are now using face recognition to unlock phones. This technology is a powerful way to protect personal data and ensure that, if a phone is stolen, sensitive data remains inaccessible by the perpetrator.

3. Implementation

3.1 Code:- This code will create the dataset and makes an entry into the sqllite3 database.

```
import cv2 # importing opencv library
import sqlite3 #import sqlite3 to connect with
database
camera = cv2.VideoCapture(0) # capturing images
from device camera
detector =
cv2.CascadeClassifier('haarcascade_frontalface_d
efault.xml')

def insertOrUpdate(Id,Name,Profession):
    conn = sqlite3.connect("FaceBase.db") #
Connect with database
    cmd = "SELECT * FROM People WHERE
ID="+str(Id) # command to check whether id
already exists
```

```
    cursor = conn.execute(cmd)
    isRecordExist=0
    for row in cursor:
        isRecordExist = 1
    if(isRecordExist == 1): #check if the
entered id already exists then update the
database
        cmd="UPDATE People SET
Name='"+str(Name)+"',Profession='"+str(Professio
n)+"' WHERE ID="+str(Id)
    else:
        cmd="INSERT  INTO
People(ID,Name,Profession)
Values("+str(Id)+",'"+str(Name)+"','"+str(Profes
sion)+"')" # If new id is entered then enter the
whole information into database
    conn.execute(cmd)
    conn.commit()
    conn.close() # close the connection with
database
Id = input('enter your id:') # Take id as input
from user it will act as primary key in database
name = input("Enter your name:") # Take name
from user
profession = input("Enter your Profession:") #
Take user's profession as input
insertOrUpdate(Id,name,profession )# calling the
function to check if the id already exist then
update the database else enter it into the
database
sampleimage = 0 # Initially number of images in
the dataset for a particular id is zero
while (True):
    ret, image = camera.read()
    gray = cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY) # converting the images into
grayscale
    faces = detector.detectMultiScale(gray, 1.3,
```

```
5)
    for (x, y, w, h) in faces: #To find faces in
image and draw rectangle on faces
        cv2.rectangle(image, (x, y), (x + w, y +
h), (255, 0, 0), 2)

        # incrementing number of images taken as
sample
        sampleimage = sampleimage + 1
        # save the sample images into dataset
folder
        cv2.imwrite("dataSet/User." + Id + '.' +
str(sampleimage) + ".jpg", gray[y:y + h, x:x +
w])

        cv2.imshow('frame', image)
    # wait for 100 miliseconds
    if cv2.waitKey(100) & 0xFF == ord('q'):
        break
    # if the sample image count is more than 120
then break
    elif sampleimage > 120:
        break
camera.release() # release the camera
cv2.destroyAllWindows()
```

3.2 Code :- This code will generate a .yml file from the images of dataset so that the dataset can be used for face recognition.

```
#importing required libraries
import cv2,os
import numpy as np
from PIL import Image
```

```
recognizer =
cv2.face.LBPHFaceRecognizer_create()
detector=
cv2.CascadeClassifier("haarcascade_frontalfa
ce_default.xml") #XML file to detect faces

path="dataSet"
def getImagesAndLabels(path):
    #getting the path of the folder  which
contain all the sample images
    imagePaths=[os.path.join(path,f) for f
in os.listdir(path)]
    #create empth face list
    faceSamples=[]
    #creating a list of empty ids
    Ids=[]
    #now iterarting through the the image
paths and loading the Ids and the images
    for imagePath in imagePaths:
        #Loading the images and converting
it into gray scale

pilImage=Image.open(imagePath).convert('L')
        #Converting the PIL images into
numpy array
        imageNp=np.array(pilImage,'uint8')
        #getting the Id associated with
image
        Id=int(os.path.split(imagePath)[-
1].split(".")[1])
        # extracting the face from the
sample images

faces=detector.detectMultiScale(imageNp)
        #If a face is there then append that
```

```
in the list as well as Id of it
        for (x,y,w,h) in faces:

faceSamples.append(imageNp[y:y+h,x:x+w])
            Ids.append(Id)
    return faceSamples,Ids


faces,Ids = getImagesAndLabels('dataSet')
recognizer.train(faces, np.array(Ids))
recognizer.save('trainner/trainner.yml')
```

3.3 Code:- This code will detect the faces which match the faces in the dataset and will display their corresponding data.

```
# importing required libraries
import cv2
import os
import numpy as np
from PIL import Image
import pickle
import sqlite3

recognizer =
cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainner/trainner.yml')
cascadePath =
"haarcascade_frontalface_default.xml"
faceCascade =
cv2.CascadeClassifier(cascadePath) #
```

```
Loading the xml file to detect faces
path="dataSet"

def getProfile(Id):
    conn = sqlite3.connect("FaceBase.db")
# connecting with the database
    cmd = "SELECT *  FROM People WHERE
ID="+str(Id) # fetching the row which
matches the id from the database
    cursor = conn.execute(cmd)
    fetch_data = None
    for row in cursor:
        fetch_data = row
    conn.close() # closing the connection
with the database
    return fetch_data


camera = cv2.VideoCapture(0) # turning on
the device camera
fontface = cv2.FONT_HERSHEY_SIMPLEX
fontscale = 1.2
#fontcolor = (0, 255, 255)
while True:
    ret, image =camera.read() # capturing
images to detect

gray=cv2.cvtColor(image,cv2.COLOR_BGR2GRA
Y) # converting the images into grayscale

faces=faceCascade.detectMultiScale(gray,
1.2,5)
```

```
    for(x,y,w,h) in faces:

cv2.rectangle(image,(x,y),(x+w,y+h),(225,
0,0),2) # to draw rectangle on faces
        Id, conf =
recognizer.predict(gray[y:y+h,x:x+w])
        fetch_data = getProfile(Id) #
calling the function with id as parameter
to get the row which matches the id from
the database
        if(fetch_data != None): # if the
id matches with yhe id in database
            if(conf<80):
                cv2.putText(image,
str(fetch_data[0]), (x, y + h+30),
fontface, fontscale,
(125,255,255),2,cv2.LINE_AA)
                cv2.putText(image,
str(fetch_data[1]), (x, y + h+60),
fontface, fontscale,
(255,0,255),2,cv2.LINE_AA)
                cv2.putText(image,
str(fetch_data[2]), (x, y + h+90),
fontface, fontscale,
(0,0,255),2,cv2.LINE_AA)
            else: # if id does not
matches then do this
                cv2.putText(image, "New
Face", (x, y + h + 60), fontface,
fontscale, (0, 0, 0), 2, cv2.LINE_AA)
    cv2.imshow('image',image)
    cv2.waitKey(100) # wait for 100
```

```
milliseconds
camera.release()
cv2.destroyAllWindows()
```

4. Screenshots

4.1 This is the dataset

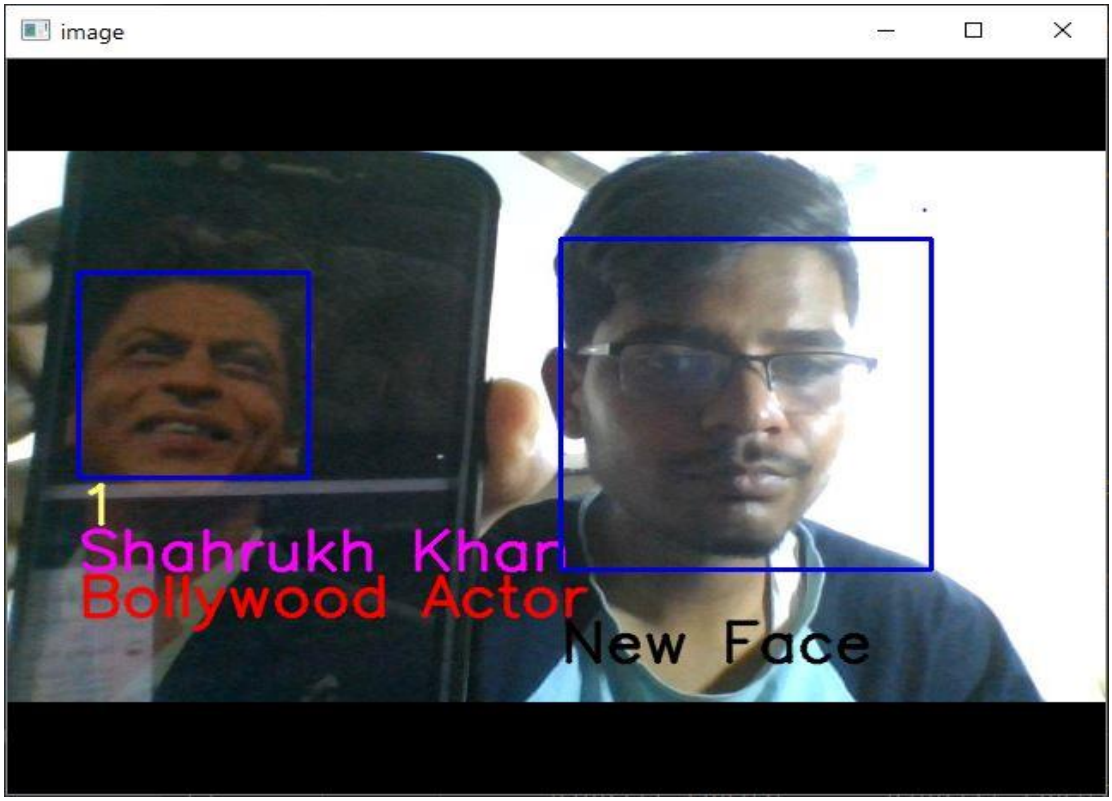
image
1
Shahrukh Khan
Bollywood Actor
New Face

4.2

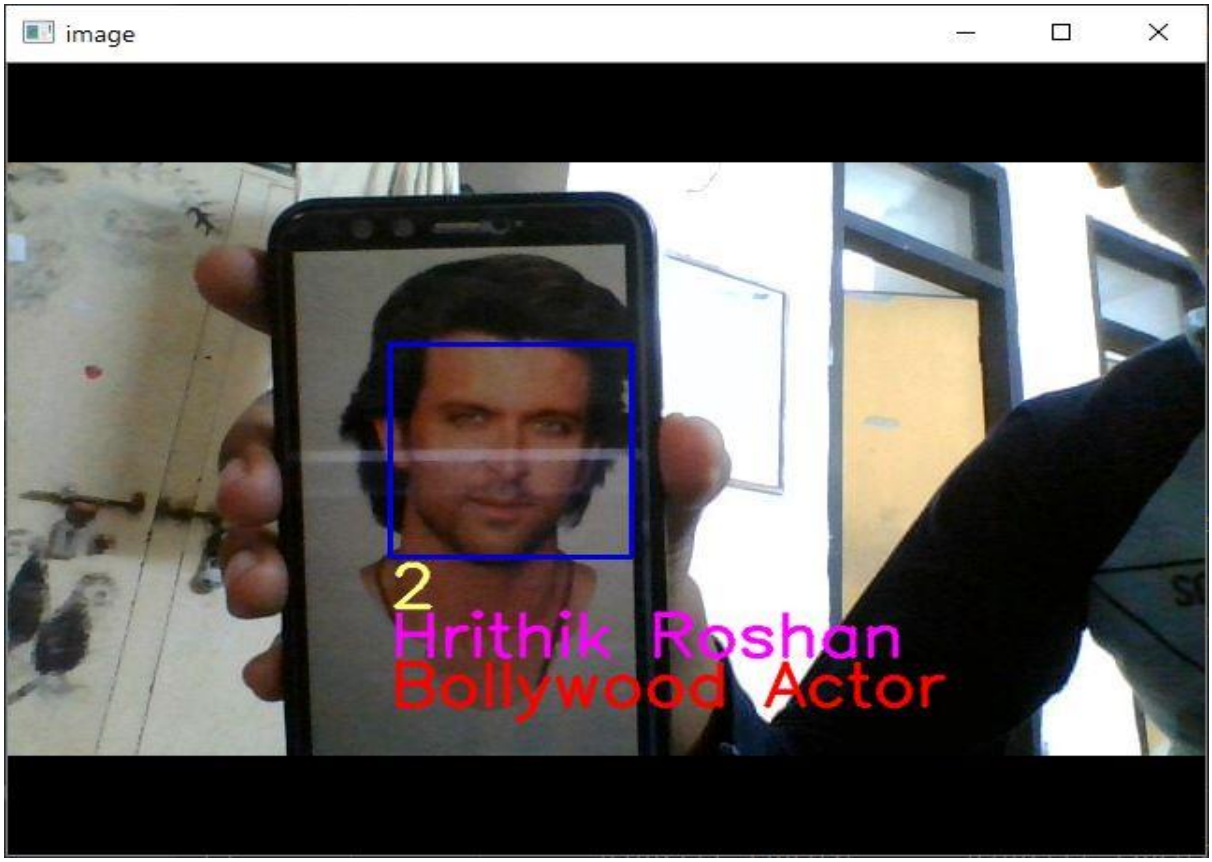
image
2
Hrithik Roshan
Bollywood Actor

4.3

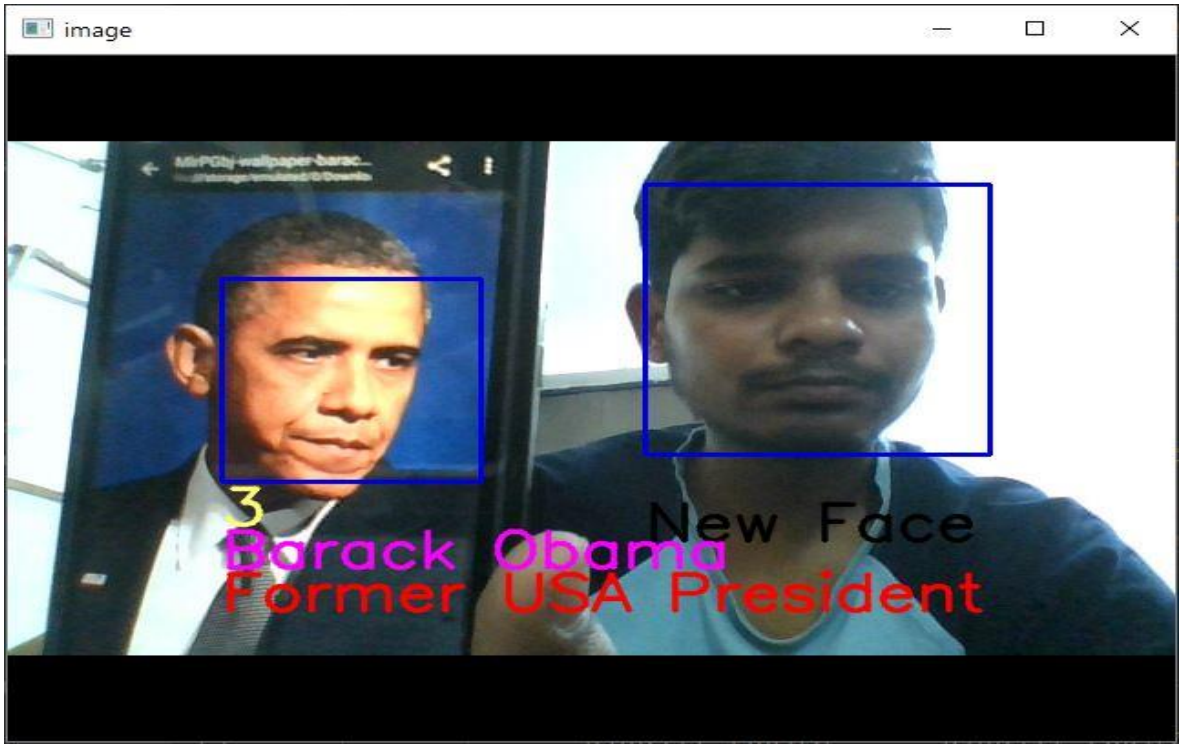
image
3
Barack Obama
Former USA President
New Face

4.4


image
4
Saksham Chowdh
Student
3
Barack Obama
Former USA President

5. Libraries Used:

Open CV: This library uses machine learning algorithms to search for faces within a picture. Because faces are so complicated, there isn't one simple test that will tell you if it found a face or not. Instead, there are thousands of small patterns and features that must be matched. The algorithms break the task of identifying the face into thousands of smaller, bite-sized tasks, each of which is easy to solve.

numpy: The library is used to search for the row and column value of the face using numpy ndarray(the face rectangle coordinates) .

sqlite3 :This library is used to connect the sql database with the dataset created.

Pillow: This library is used for opening, manipulating, and saving many different image file formats.

Pickle: This library is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk.

os: This library is used as it provides functions for interacting with the operating system. OS, comes under Python's standard utility modules. This module provides a portable way of using operating system dependent functionality.

6. References

https://en.wikipedia.org/wiki/Facial_recognition_system

https://medium.com/better-programming/step-by-step-face-recognition-in-images-ad0ad302058a

http://www.researchpublish.com/download.php?file=Scope%20of%20Face%20Recognition%20Techniques-1815.pdf&act=book

https://www.edureka.co/blog/python-opencv-tutorial/