Govardhan Seshasayee

Face Detection

Face detection can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class. Examples include upper torsos, pedestrians, and cars. Face detection simply answers two question, 1. are there any human faces in the collected images or video? 2. where is the located?

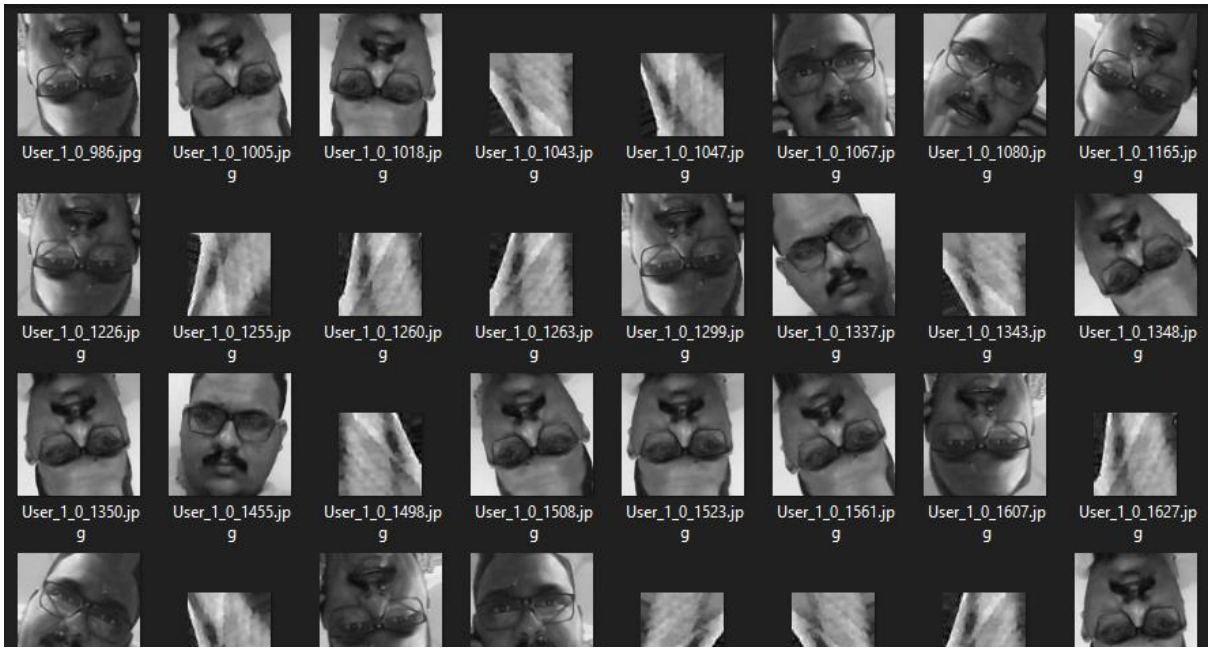
Face-detection algorithms focus on the detection of frontal human faces. It is analogous to image detection in which the image of a person is matched bit by bit. Image matches with the image stores in database. Any facial feature changes in the database will invalidate the matching process.

Question:

Implement a face tracking algorithm using haar cascade algorithm and opencv. Using haar cascade, first implement a face detection algorithm that counts the total number of faces present in any given frame. Write the total number of faces detected on the top left of the image. Further modify the code to track the face if only one face is detected. Make sure that you draw the bounding box corresponding to all video frames. Note: you may need to fine tune the parameters for Haar Cascade Classifier to get optimal results and remove false positives.

Implementation:

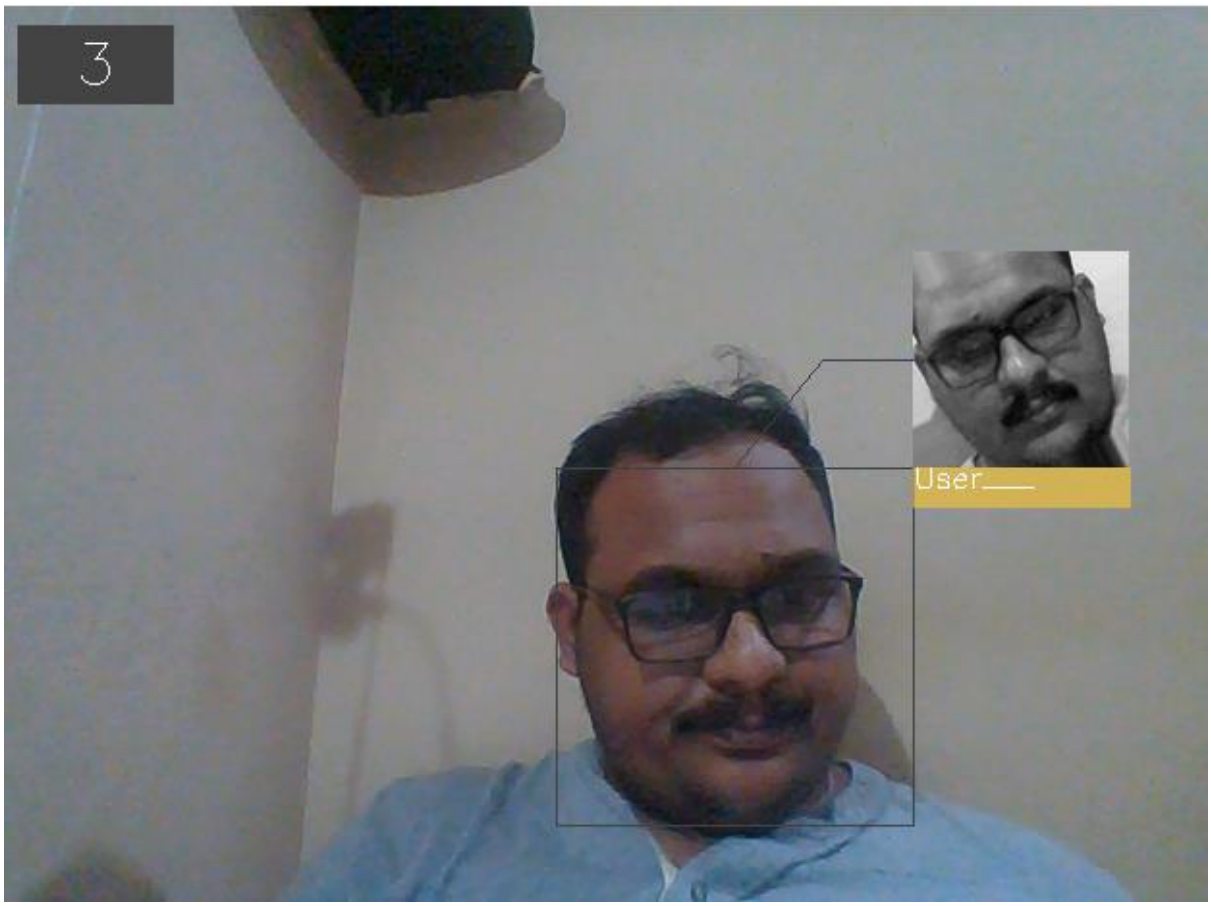
The necessary libraries are added and the live webcam is switched on first to detect the faces after which the images obtained are stored in grayscale in different rotated angles with which the images are recognised. The data of the images recorded from the camera are stored and then processed for detection later for which the deepface library is used.



```
import keras
import cv2
import os
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt

datagen = keras.preprocessing.image.ImageDataGenerator(
    rotation_range=30,
    width_shift_range=0.0,
    height_shift_range=0.0,
    horizontal_flip=True,
    vertical_flip=True,
    rescale=None,
    preprocessing_function=None
)

def mark_images(f):
    ex = -1
    img = cv2.resize(cv2.imread(f), (640, 480))
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    face_detector = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
    faces = face_detector.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
        plt.imshow(gray[y:y+h,x:x+w])
        plt.show()
        face_id = input('\n Assign an ID number and press enter ')
        im = np.expand_dims(np.expand_dims(gray[y:y+h,x:x+w], 0), 3)
        datagen.fit(im)
        for x, val in zip(datagen.flow(im, save_to_dir='C:\\Users\\Govardhan\\\\capstone\\face\\dataset', save_prefix="User_" +
            ex = 1
```

```
cam = cv2.VideoCapture(0)
cam.set(3, 640)
cam.set(4, 480)
face_detector = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

face_id = input('\n Assign an ID number and press enter ')
print("\n  Look the camera and wait ...")

ex = -1
while (True):
    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_detector.detectMultiScale(gray, 1.3, 5)

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

        cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
        cv2.imshow('image', img)
        im = np.expand_dims(np.expand_dims(gray[y:y + h, x:x + w], 0), 3)
        datagen.fit(im)

        for x, val in zip(datagen.flow(im, save_to_dir='C:\\Users\\Govardhan\\Capstone Projects\\Face detection',
                                       save_prefix="User_" + str(face_id), save_format='jpg'), range(100)):
            ex = 1

    k = cv2.waitKey(100) & 0xff
    if k == 27:
        break
    if ex == 1:
        break
print("\n Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()
```

```
from deepface import DeepFace

db_path = "C:\\Users\\Govardhan\\Capstone Projects\\Face detection" #you should store facial images in this directory

DeepFace.stream(db_path = db_path
   , model_name = 'VGG-Face' #candidates are VGG-Face, Facenet, OpenFace, DeepFace, DeepID and Dlib
   , enable_face_analysis = False)
k = cv2.waitKey(10) & 0xff # Press 'ESC' for exiting video
if k == 27:
    break
cam.release()
cv2.destroyAllWindows()
```


3
User