INHA UNIVERSITY TASHKENT

DEPARTMENT OF CSE

SPRING 2020

Multimedia Computing

PCL (Personal Character Learner)

Submitted by:

U1610145 Mirtolip Saitov CIE 16-3

Abstract

In this days artificial intelligence and neural networking systems are becomes more interested spheres in the world. To determine people and gathering their information plays very important role I the business. So here I decided to make face recognition system and gathering customers data. The platform about to learn people's character and recognize them. After configuring the program, the program automatically will find the person or people which were configured and show them by their names. Using the OpenCV we can analysis the face.

Introduction

Face detection program going to detect people's face. And helping to differentiate the people. The face detection program is used to analysis people to collect information. This program using Haar cascades is a machine learning based approach where a cascade function us trained with a set of input data. OpenVC already has many classifiers for detecting face, smiles, eyes, car numbers, etc,. In this project I used face detection classifier. In order to detect the face we need to transfer or convert the image from RGB (Red, Green, Blue) image (colorful image) to gray scale image. Then we need to Detect Multi Scale. Here we read the image and convert it to grayscale. Many operations in OpenCV are done in grayscale. After all analyzing performed we can draw the rectangle to people faces. Using the face recognition system we can store all faces to database system. When we want to search or find the person which we interested in we can easily can find them by their face images.

Actually system not just for business case, it is also can be used for general analyzing purposes. For example to put the camera for the markets or in the streets. This can detect face and can be reduced number of the crimes, some other vandalisms.

Methods

The function detects the actual face and is the key part of our code:

1. The detectMultiScale function is the general function that detects object. Since we are calling on the face cascade, that's what it detects.
2. The first option is the grayscale image.
3. The second is the scaleFactor. Since some faces may be closer to the camera, they would appear bigger than the faces in the back. The scale factor compensates for this.

4. The detection algorithm uses a moving window to detect objects. minNeighbors defines how many objects are detected near the current one before it declares the face found. minSize, meanwhile, gives the size of each window.

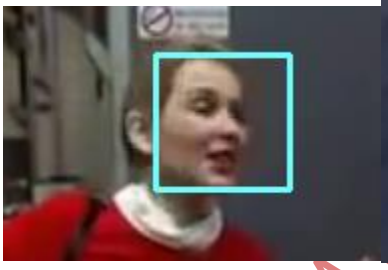
This function returns 4 values: the x and y location of the rectangle, and the rectangle's width and height (w , h).

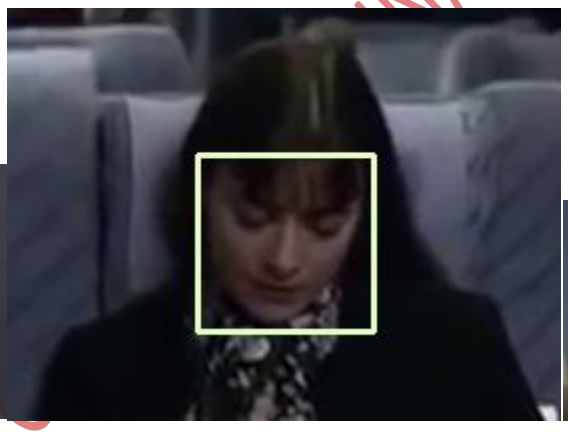
```
# Detect faces in the image
faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.1,
    minNeighbors=5,
    minSize=(30, 30)
)
```

```
# Draw a rectangle around the faces
for (x, y, w, h) in faces:
    cv2.rectangle(frame, (x, y), (x+w, y+h), (int(x), 255, int(y)), 2)
```

Results

The result are here as image as you can see. But original result output file as a video format. The camera catching the faces and generating the new video file with detected faces. Main problems of these system is that system can be catch other details also. It depends of the video format. Here I used little old video format. If the resolution of the video is high, then the performance also will increase.





References

https://docs.opencv.org/

https://github.com/opencv/opencv/tree/master/data/haarcascades