AGE INVARIANT FACIAL RECOGNITION SYSTEM

PART 1: Facial Detection

By

Noxolo Maluleke : University of Limpopo
Florah Chuene : University of Limpopo
Mbuso Makitla : University of the Witwatersrand
Cecil Setati : University of Johannesburg

Due: 25th January 2020

Table of Content

1. Introduction

As the name of the project states. Our mission was to develop an age invariant facial recognition system given a dataset of images. Due to lack of GPU we couldn't reach the final stage of our project. We then had to restructure our project into facial recognition without considering age.

Our project structured is two parts. Part 1 being Facial Detection and Part 2 being Facial Recognition. This paper explains only part one of the project that is **Facial Detection**.

2. Image Capturing

```
import cv2
```

```
camera = cv2.VideoCapture(0)
i = 0
while True:
    i+=1
    return_value, image = camera.read()
    cv2.imshow("frame",image)
    key = cv2.waitKey(1)
    if key == ord('q'):
        cv2.imwrite('opencv'+str(i)+'.png', image)

#showPic = cv2.imwrite("filename.jpg",frame)
#print(showPic)
# 8. shutdown the camera
del(camera)
cv2.destroyAllWindows
```

Figure 1. Webcam and key capturing code

The code in figure 1 was used for capturing images through a webcam to create a dataset of images. Cv2 is a library from OpenCV for reading and writing images.

- "***camera***" in the code is used as a variable to store images being read,
- "***imshow***" - is used to show an image being read,
- "***waitKey***" - is used to pause the camera,
- "***imwrite***" - is for converting the image taken into an image format
- "***del***" - to exit the camera

3. Face Detection

```
import cv2
import os
```

```
haar_file = 'haarcascade_frontalface_default.xml'

# All the faces data will be
#  present this folder
datasets = 'florah'


# for my faces I've used my name you can
# change the label here
sub_data = 'Cecil'

path = os.path.join(datasets, sub_data)
if not os.path.isdir(path):
    os.mkdir(path)


(width, height) = (130, 100)

#'0' is used for my webcam,
# if you've any other camera
#  attached use '1' like this
face_cascade = cv2.CascadeClassifier(haar_file)
camera = cv2.VideoCapture(0)

# The program loops until it has 30 images of the face.
count = 1
while True:
    #count+=1
    return_value, image = camera.read()
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 4)
    for (x, y, w, h) in faces:
        cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 2)
        face = gray[y:y + h, x:x + w]
        face_resize = cv2.resize(face, (width, height))
        cv2.imwrite('% s/% s.png' % (path, count), face_resize)
    count += 1

    cv2.imshow('image', image)
    key = cv2.waitKey(1)
    if key == ord('q'):
        cv2.imwrite('opencv'+str(count)+'.png', )

del(camera)
cv2.destroyAllWindows
```

```

```

Figure 2. Facial detection code with Haar Cascade

The code in Figure 2 was used to detect images using Haar Cascade method. Haar Cascade is pre-trained to detect faces of humans only. Anything that wasn't a human being wasn't detected.

"**Rectangle**" in the code is used to draw a box around the face when the algorithm detects a face. Haar Cascade method detects a face only when the algorithm can detect both eyes.

The "os" module allows the use of the functionalities that are dependent on the operating system.