**Detecting age and gender with Python**

**In this section, using machine learning in the Python programming language, we will teach you to write a code that detects the age and gender of people. Age and gender detection is one of the tasks of computer vision . So I will use the OpenCV library in Python .**

**Before we start coding for age and gender detection in Python, I will first introduce you to what this concept means and how to deal with potential problems in this area. It is very important to understand this concept; Because in this way you can do this in the future not only with Python, but also with any other programming language .**

**An introduction to age and gender diagnosis**

**Age and gender detection is one of the most difficult tasks that computer vision can do. The main reason for this lies in the difficulty of obtaining the data needed to train these types of systems .**

**Object recognition programs can easily access hundreds of thousands or even millions of images for training, but datasets with age/or gender labels are significantly smaller. (usually containing thousands or at best tens of thousands of data) .**

**The reason for this is that in order to have labels for such images, we need access to the personal information of the subjects in the images. That means we need their date of birth and gender, which is rarely made public. Therefore, we have to be content with the nature of the problem we are dealing with and adapt the network architecture and algorithmic events to deal with these limitations .**

**How to detect age and gender with Python**

**The fields of classification based on age and gender have been researched for decades. Over the years, various approaches have been taken to deal with this problem, with varying levels of success. Now let's start the age and gender detection using Python programming language .**

**will present the gender detection problem as a classification problem and the age detection problem as a regression problem. However, accurate age estimation using regression is difficult. Even humans cannot tell a person's age by looking at him. However, we usually find out what decade of their life they are in. This is what we will do in this free python project .**

**start of work :**

**The first thing we do is write face recognition code. Because without face recognition, we cannot move forward in age and gender prediction .**

**You can download the pre-trained OpenCV models you need for age and gender detection here. Now after importing the OpenCV module into your Python file, you can start with the following code .**

**Python code for face recognition :**

**def getFaceBox( net , frame , conf\_threshold = 0.7 ):**

**frameOpencvDnn = frame . copy ()**

**frameHeight = frameOpencvDnn . shape [ 0 ]**

**frameWidth = frameOpencvDnn . shape [ 1 ]**

**blob = cv . dnn \_ blobFromImage ( frameOpencvDnn , 1.0 , ( 300 , 300 \_ [ 104 , 117 , 123 ], true \_ false )**

**net . setInput ( blob )**

**detections = net . forward ()**

**bboxes = []**

**for i in range ( detections . shape [ 2 ]):**

**confidence = detections [ 0 , 0 , i , 2 ]**

**if confidence > conf\_threshold :**

**x1 = int ( detections [ 0 , 0 , i , 3 ] \* frameWidth )**

**y1 = int ( detections [ 0 , 0 , i , 4 ] \* frameHeight )**

**x2 = int ( detections [ 0 , 0 , i , 5 ] \* frameWidth )**

**y2 = int ( detections [ 0 , 0 , i , 6 ] \* frameHeight )**

**bboxes . append ([ x1 , y1 , x2 , y2 ])**

**cv . rectangle ( frameOpencvDnn , ( x1 , y1 ), ( x2 , y2 ), ( 0 , 255 , 0 ), int ( round ( frameHeight / 150 )), 8 )**

**return frameOpencvDnn , bboxes**

**Copy**

**Now we enter the next stage of recognizing the human gender in the image. Here, I load the gender network in memory and transfer the identified face to the entire network for gender recognition .**

**Python code to detect gender :**

**genderProto = "gender\_deploy.prototxt"**

**genderModel = "gender\_net.cafemodel"**

**ageNet = cv . dnn \_ readNet ( ageModel , ageProto )**

**genderList = [ 'Male' , 'Female' ]**

**blob = cv . dnn \_ blobFromImage ( face , 1 , ( 227 , 227 ), MODEL\_MEAN\_VALUES , swapRB = False )**

**genderNet . setInput ( blob )**

**genderPreds = genderNet . forward ()**

**gender = genderList [ genderPreds [ 0 ]. argmax ()]**

**print ( "Gender Output : {}" . format ( genderPreds ))**

**print ( "Gender : {}" . format ( gender ))**

**Copy**

**The next thing we need to do is predict the age of the person in the image. Here I load the aging network and use forward pass to get the output. Since the architecture of this network is similar to the architecture of the gender network , we can make maximum use of all outputs to obtain the desired age group for age detection .**

**Python code to detect age :**

**ageProto = "age\_deploy.prototxt"**

**ageModel = "age\_net.cafemodel"**

**ageNet = cv . dnn \_ readNet ( ageModel , ageProto )**

**ageList = [ '(0 - 2)' , '(4 - 6)' , '(8 - 12)' , '(15 - 20)' , '(25 - 32)' , '(38 - 43)' , '(48 - 53)' , '(60 - 100)' ]**

**ageNet . setInput ( blob )**

**agePreds = ageNet . forward ()**

**age = ageList [ agePreds [ 0 ]. argmax ()]**

**print ( "Gender Output : {}" . format ( agePreds ))**

**print ( "Gender : {}" . format ( age ))**

**Copy**

**The last code we have to write is to display the output :**

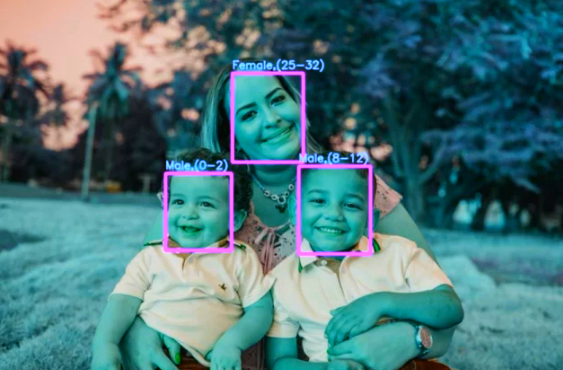
**label = "{}, {}" . format ( gender , age )**

**cv . putText ( frameFace , label , ( bbox [ 0 ], bbox [ 1 ] - 20 ), cv . FONT\_HERSHEY\_SIMPLEX , 0.8 , ( 255 , 0 , 0 ), 3 , cv . LINE\_AA )**

**cv . imshow ( "Age Gender Demo" , frameFace )**

**Copy**

**As you can see from the output, we can predict gender and age with high accuracy. I hope you like this part of the article about age and gender classification with Python programming language .**



Amirshayan Jalili

[Shayan138190@gmail.com](mailto:Shayan138190@gmail.com)

<https://github.com/Amirshayan2002>