

A
Minor Project Report
On
Age & Gender Detection

In partial fulfillment of the requirement for the award of the degree of



SUBMITTED BY

Akash Parihar (0901IT203D01)

SUBMITTED TO

Prof. Vikas Sejwar

Dept. Of IT

Department of Information Technology

Madhav Institute of Technology and Science, Gwalior

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Session: July-Dec 2021

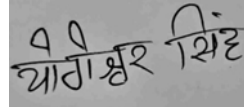
CERTIFICATE

This is to certify that Akash Parihar (0901IT203D01) minor project, " Age & Gender Detection" is a genuine record of a project completed under our supervision and guidance in partial fulfilment of the requirements for the award of a Bachelor of Technology in Information Technology in the Department of Information Technology, Madhav Institute of Technology and Science, Gwalior.



(Prof. Vikas Sejwar)

Mentor



(Prof. YOGESHWAR SINGH)

Mentor



Madhav Institute of Technology and Science, Gwalior (M.P.)

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CANDIDATE'S DECLARATION

I hereby declare that the Project entitled “**Age & Gender Detection**” which is being submitted in the partial fulfilment of the requirement for the award of **Bachelor of Technology in Information Technology**.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I have fully cited and referenced all material and results that are not original to this work.

To the best of my knowledge the material presented in this Project has not been submitted elsewhere for the award of any other degree/diploma.

Date: 02/12/2021

Place: Gwalior

Akash Parihar (0901IT203D01)

ACKNOWLEDGEMENT

First and foremost, we would like to express our highest appreciation to supportive academic professor. His supervision and support that gave me truly helps during the period of conducting our project. His never-ending supply of valuable advice and guidance has enlightens me and deeply engraved in our mind. Next, I would like to dedicate our thankfulness to him, for his enthusiastic support and supervision of the thesis revision. I'm also happy to present my gratefully acknowledge to IT Lab Technicians, who has been so warmth and kind to provide sincere assistance and good cooperation during this period. Their co-operation is much indeed appreciated. In addition, I would like to convey thanks to lecturers, for their assistance, which really spends their time to teach us a lots of knowledge regarding to the design development. Last but not least, I would like to state my appreciation to the staff – Faculty of Information Technology, our team and colleagues for supporting us and administration department for their help in the project.

Contents

Introduction

1. Age & Gender Prediction methods	
1.1 Face Prediction.....	2
1.2 Age Prediction.....	3
1.3 Gender Prediction	4
1.4 Model	4
1.4.1 Gender Net	
1.4.2 Age Net	
2. Problem Statement	
2.1 The Purpose	4
3. Implementation	
3.1 Required Modules	5
3.2 Code Design.....	6
4. Results	
4.1 Examples.....	9
5. Conclusion	
5.1 Possible Future Research.....	12

Bibliography

Introduction

The task of detecting age and gender, however, is an inherently difficult problem, more so than many other computer vision tasks. The main reason for this difficulty gap lies in the data required to train these types of systems.

While general object detection tasks can often have access to hundreds of thousands or even millions of images for training, datasets with age and/or gender labels are considerably smaller, usually in the thousands or, at best, tens of thousands.

The areas of classification by age and gender have been studied for decades. Various approaches have been taken over the years to tackle this problem, with varying levels of success. Now let's start with the task of detecting age and gender using the Python programming language.

I will present the problem of gender detection as a classification problem and the age detection problem as a regression problem. However, estimating age accurately using regression is difficult. Even humans cannot accurately predict an age by looking at a person. However, we do know if they are in their 30s or 40s. This is also what I'm going to follow using Python.

About the Project –

In this Python Project, we will use Deep Learning to accurately identify the gender and age of a person from a single image of a face. The predicted gender may be one of 'Male' and 'Female', and the predicted age may be one of the following ranges- (0–2), (4–6), (8–12), (15–20), (25–32), (38–43), (48–53), (60–100) . It is very difficult to accurately guess an exact age from a single image because of factors like makeup, lighting and facial expressions.

Objectives :-

- Detect faces
- Classify into Male/Female
- Classify into one of the 8 age ranges
- Put the results on the image and display it

1. Age & Gender Prediction Methods

Now let's get started with the task of Age and Gender detection using the Python programming language. I will first start with writing the code for detecting faces because without face detection we will not be able to move further with the task of age and gender prediction. You can download the necessary OpenCV pre-trained models that you will need in the task of age and gender detection. Now after importing the OpenCV module in your python file you can get started.

1.1 Face Prediction

Face detection -- also called facial detection -- is an artificial intelligence (AI) based computer technology used to find and identify human faces in digital images. Face detection technology can be applied to various fields -- including security, biometrics, law enforcement, entertainment and personal safety -- to provide surveillance and tracking of people in real time.

Face detection has progressed from rudimentary computer vision techniques to advances in machine learning (ML) to increasingly sophisticated artificial neural networks (ANN) and related technologies; the result has been continuous performance improvements. It now plays an important role as the first step in many key applications -- including face tracking, face analysis and facial recognition. Face detection has a significant effect on how sequential operations will perform in the application.

In face analysis, face detection helps identify which parts of an image or video should be focused on to determine age, gender and emotions using facial expressions. In a facial recognition system -- which maps an individual's facial features mathematically and stores the data as a faceprint -- face detection data is required for the algorithms that discern which parts of an image or video are needed to generate a faceprint. Once identified, the new faceprint can be compared with stored faceprints to determine if there is a match.

1.2 Age Prediction

Age detection is the process of automatically discerning the age of a person solely from a photo of their face.

Typically, you'll see age detection implemented as a two-stage process:

1. **Stage #1:** Detect faces in the input image/video stream
2. **Stage #2:** Extract the face Region of Interest (ROI), and apply the age detector algorithm to predict the age of the person

For Stage #1, any face detector capable of producing bounding boxes for faces in an image can be used, including but not limited to Haar cascades, HOG + Linear SVM, Single Shot Detectors (SSDs), etc.

Exactly *which* face detector you use depends on your project:

- Haar cascades will be very fast and capable of running in real-time on embedded devices — the *problem* is that they are less accurate and highly prone to false-positive detections
- HOG + Linear SVM models are more accurate than Haar cascades but are slower. They also aren't as tolerant with occlusion (i.e., not all of the face visible) or viewpoint changes (i.e., different views of the face)
- Deep learning-based face detectors are the most robust and will give you the best accuracy, but require even more computational resources than both Haar cascades and HOG + Linear SVMs

When choosing a face detector for your application, take the time to consider your project requirements — is speed or accuracy more important for your use case? I also recommend running a few experiments with each of the face detectors so you can let the empirical results guide your decisions.

Once your face detector has produced the bounding box coordinates of the face in the image/video stream, you can move on to Stage #2 — identifying the age of the person.

Given the bounding box (x , y)-coordinates of the face, you first extract the face ROI, ignoring the rest of the image/frame. Doing so allows the age detector to focus *solely* on the person's face and not any other irrelevant “noise” in the image.

The face ROI is then passed through the model, yielding the actual age prediction.

1.3 Gender Prediction

The problem of gender prediction and classification is closely related to face detection. One of the first method for gender classification was SEXNET introduced by Golomb et al. It classified images sampled at 30 x 30 pixels using a fully connected neural network with 3 layers of 900, 40 and 900 neurons. Another method introduced by Moghaddam and Yang [16] uses Support Vector Machines to classify gender. The method used subsampled images of 21 x 21 pixels and classified them using SVM with radial basis function kernel. More recently the gender classification problem has been treated as a subproblem of age prediction. Current approaches use the same neural network architecture for age as well as gender prediction. Gender can be easily modelled as a binary classification problem. If the gender is modelled using regression we can view the output number as confidence.

1.4 Models

1.4.1 GenderNet:

https://www.dropbox.com/s/iyv483wz7ztr9qh/gender_net.caffemodel?dl=0

1.4.2 Age Net:

https://www.dropbox.com/s/xfb20y596869vbb/age_net.caffemodel?dl=0

2. Problem Statement

Developing a mobile off-line application for the video-based age and gender recognition using OpenCV.

The Goal Set

- To conduct a review of the existing solutions for image recognition by age and gender.
- To describe the algorithm scheme of the proposed system.
- To develop the age/gender recognition architecture.
- To analyze the accuracy of decision making on the basis of each aggregation method by conducting experiments

2.1 The Purpose

Age and gender characteristics can be applied in retail for contextual advertising for particular group of customers



The reliability of the existing solutions remains insufficient for practical application



3. Implementation

3.1 Required Modules

1. OpenCV –

OpenCV is short for Open Source Computer Vision. Intuitively by the name, it is an open-source Computer Vision and Machine Learning library.

2. Time –

This module provides various time-related functions. For related functionality, see also the datetime and calendar modules

3. Math –

It will use for frame Width and Height.

4. Arg parse –

Recommended the python command line parsing module in python standard library means whatever the argument you pass in the arg parser will be added at the time of Output.

3.2 Code Design

Let's start by adding/importing some Libraries

1. The libraries that we need are **Cv2**, **math**, **time** and **argparse**.

```
2 import cv2 as cv
3 import math
4 import time
5 import argparse
6
```

For face detection, we have a .pb file- this is a protobuf file (protocol buffer); it holds the graph definition and the trained weights of the model. We can use this to run the trained model.

2. We use the argparse library to create an argument parser so we can get the image argument from the command prompt. We make it parse the argument holding the path to the image to classify gender and age.

```
27
28 parser = argparse.ArgumentParser(description='Use this script to run age and gender recognition using OpenCV.')
29 parser.add_argument('--input', help='Path to input image or video file. Skip this argument to capture frames from a camera.')
30
31 args = parser.parse_args()
32
33 faceProto = "opencv_face_detector.pbtxt"
34 faceModel = "opencv_face_detector_uint8.pb"
35
36 ageProto = "age_deploy.prototxt"
37 ageModel = "age_net.caffemodel"
38
39 genderProto = "gender_deploy.prototxt"
40 genderModel = "gender_net.caffemodel"
```

3. For face, age, and gender, initialize protocol buffer and model.

```
33 faceProto = "opencv_face_detector.pbtxt"
34 faceModel = "opencv_face_detector_uint8.pb"
35
36 ageProto = "age_deploy.prototxt"
37 ageModel = "age_net.caffemodel"
38
39 genderProto = "gender_deploy.prototxt"
40 genderModel = "gender_net.caffemodel"
41
42 MODEL_MEAN_VALUES = (78.4263377603, 87.7689143744, 114.895847746)
43 ageList = ['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)', '(38-43)', '(48-53)', '(60-100)']
44 genderList = ['Male', 'Female']
45
```

4. Initialize the mean values for the model and the lists of age ranges and genders to classify from.

```
41
42 MODEL_MEAN_VALUES = (78.4263377603, 87.7689143744, 114.895847746)
43 ageList = ['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)', '(38-43)', '(48-53)', '(60-100)']
44 genderList = ['Male', 'Female']
45
```

5. Now, use the readNet() method to load the networks.

```
47 ageNet = cv.dnn.readNet(ageModel, ageProto)
48 genderNet = cv.dnn.readNet(genderModel, genderProto)
49 faceNet = cv.dnn.readNet(faceModel, faceProto)
50
```

6. Let's capture video stream in case you'd like to classify on a webcam's stream. Now until any key is pressed, we read the stream and store the content into the names hasFrame and frame. If it isn't a video, it must wait, and so we call up waitKey() from cv2, then break.

```
52 cap = cv.VideoCapture(args.input if args.input else 0)
53 padding = 20
54 while cv.waitKey(1) < 0:
55     # Read frame
56     t = time.time()
57     hasFrame, frame = cap.read()
58     if not hasFrame:
59         cv.waitKey()
60         break
61
62     frameFace, bboxes = getFaceBox(faceNet, frame)
63     if not bboxes:
64         print("No face Detected, Checking next frame")
```