Gender and Age Detection Python Project- Objective

To build a gender and age detector that can approximately guess the gender and age of the person (face) in a picture using [***Deep Learning***](https://data-flair.training/blogs/deep-learning/) on the Adience dataset.

Gender and Age Detection – 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. We will use the models trained by [Tal Hassner and Gil Levi](https://talhassner.github.io/home/projects/Adience/Adience-data.html). 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) (8 nodes in the final softmax layer). It is very difficult to accurately guess an exact age from a single image because of factors like makeup, lighting, obstructions, and facial expressions. And so, we make this a classification problem instead of making it one of regression.

The CNN Architecture

The convolutional neural network for this python project has 3 convolutional layers:

* Convolutional layer; 96 nodes, kernel size 7
* Convolutional layer; 256 nodes, kernel size 5
* Convolutional layer; 384 nodes, kernel size 3

It has 2 fully connected layers, each with 512 nodes, and a final output layer of softmax type.

To go about the python project, we’ll:

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

The Dataset

For this python project, we’ll use the Adience dataset; the dataset is available in the public domain and you can find it [***here***](https://www.kaggle.com/ttungl/adience-benchmark-gender-and-age-classification). This dataset serves as a benchmark for face photos and is inclusive of various real-world imaging conditions like noise, lighting, pose, and appearance. The images have been collected from Flickr albums and distributed under the Creative Commons (CC) license. It has a total of 26,580 photos of 2,284 subjects in eight age ranges (as mentioned above) and is about 1GB in size. The models we will use have been trained on this dataset.

Prerequisites

You’ll need to install OpenCV (cv2) to be able to run this project. You can do this with pip-

1. pip install opencv-python

Other packages you’ll be needing are math and argparse, but those come as part of the standard Python library.

Steps for practicing gender and age detection python project

1. [**Download this zip**](https://drive.google.com/file/d/1yy_poZSFAPKi0y2e2yj9XDe1N8xXYuKB/view). Unzip it and put its contents in a directory you’ll call gad.

The contents of this zip are:

* opencv\_face\_detector.pbtxt
* opencv\_face\_detector\_uint8.pb
* age\_deploy.prototxt
* age\_net.caffemodel
* gender\_deploy.prototxt
* gender\_net.caffemodel
* a few pictures to try the project on

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. And while a .pb file holds the protobuf in binary format, one with the .pbtxt extension holds it in text format. These are TensorFlow files. For age and gender, the .prototxt files describe the network configuration and the .caffemodel file defines the internal states of the parameters of the layers.

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 for.

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

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

5. Now, use the readNet() method to load the networks. The first parameter holds trained weights and the second carries network configuration.

6. Let’s capture video stream in case you’d like to classify on a webcam’s stream. Set padding to 20.

7. 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.

8. Let’s make a call to the highlightFace() function with the faceNet and frame parameters, and what this returns, we will store in the names resultImg and faceBoxes. And if we got 0 faceBoxes, it means there was no face to detect.  
Here, net is faceNet- this model is the DNN Face Detector and holds only about 2.7MB on disk.

* Create a shallow copy of frame and get its height and width.
* Create a blob from the shallow copy.
* Set the input and make a forward pass to the network.
* faceBoxes is an empty list now. for each value in 0 to 127, define the confidence (between 0 and 1). Wherever we find the confidence greater than the confidence threshold, which is 0.7, we get the x1, y1, x2, and y2 coordinates and append a list of those to faceBoxes.
* Then, we put up rectangles on the image for each such list of coordinates and return two things: the shallow copy and the list of faceBoxes.

9. But if there are indeed faceBoxes, for each of those, we define the face, create a 4-dimensional blob from the image. In doing this, we scale it, resize it, and pass in the mean values.

10. We feed the input and give the network a forward pass to get the confidence of the two class. Whichever is higher, that is the gender of the person in the picture.

11. Then, we do the same thing for age.

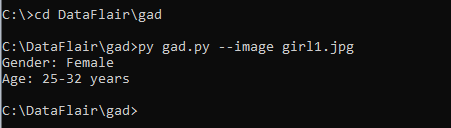
12. We’ll add the gender

Python Project Examples for Gender and Age Detection

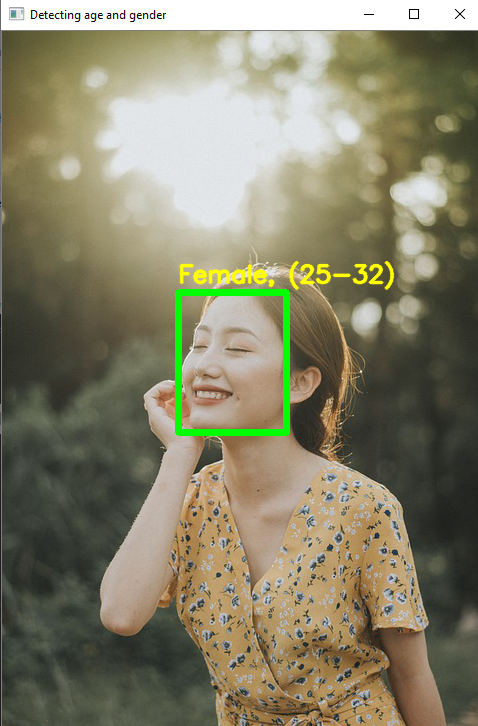
Let’s try this gender and age classifier out on some of our own images now.

We’ll get to the command prompt, run our script with the image option and specify an image to classify:

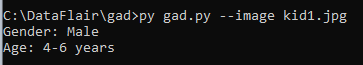
**Python Project Example 1**

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2019/09/interesting-python-project-example.png)

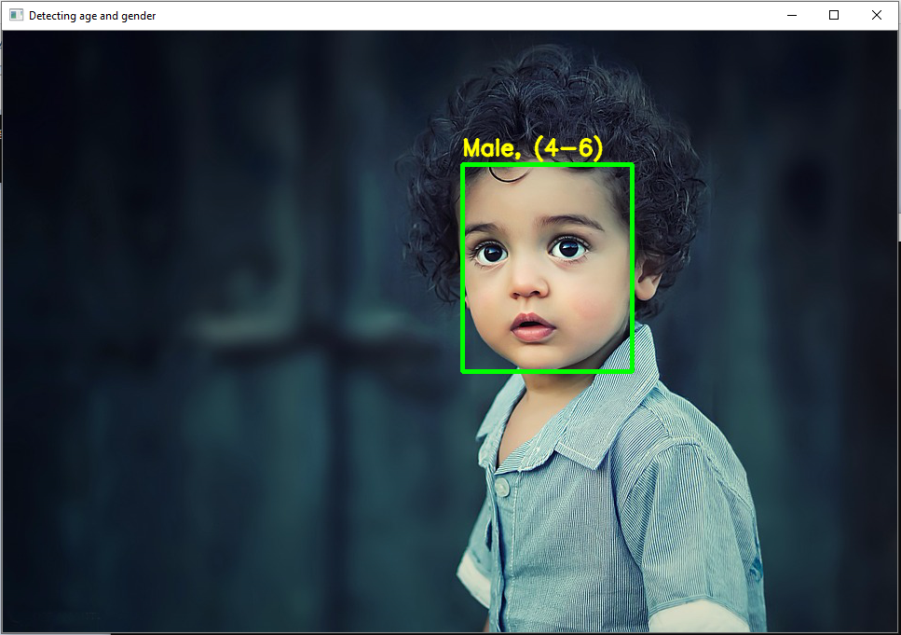
**Output:**

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2019/09/python-project-example-output-1.png)

**Python Project Example 2**

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2019/09/intermediate-python-project-example.png)

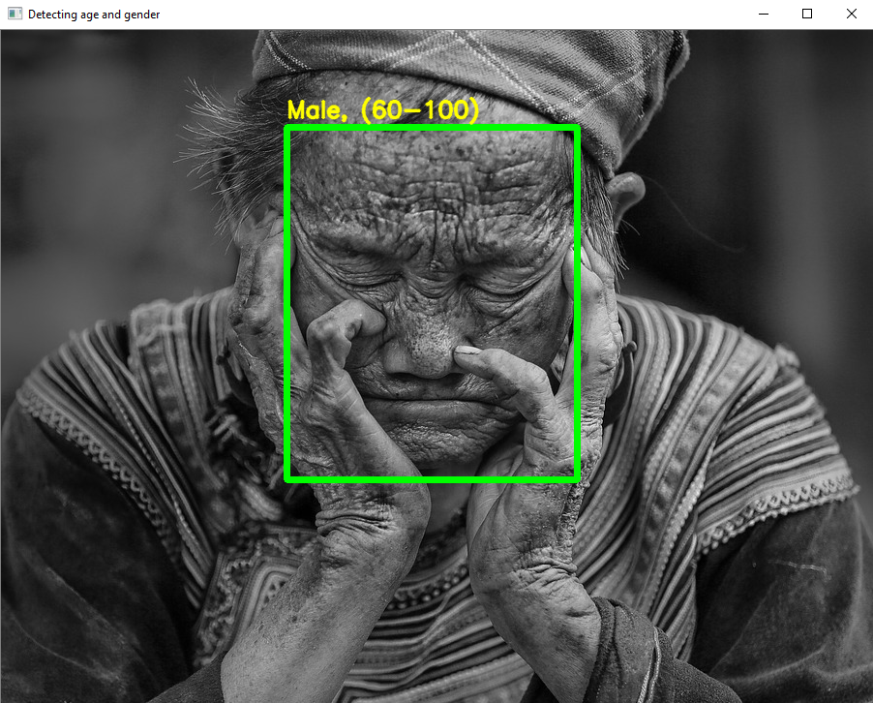
**Output:**

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2019/09/advanced-python-project-example.png)

**Python Project Example 3**

[python open source project](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2019/09/python-project-example.png)

**Output:**

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2019/09/python-project-for-practice.png)

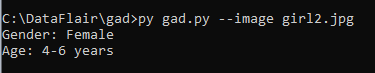
**Python Project Example 4**

[python projects for practice](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2019/09/python-projects-for-practice-woman.png)

**Output:**

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2019/09/interesting-python-project-3.png)

**Python Project Example 5**

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2019/09/learning-python-projects-1.png)

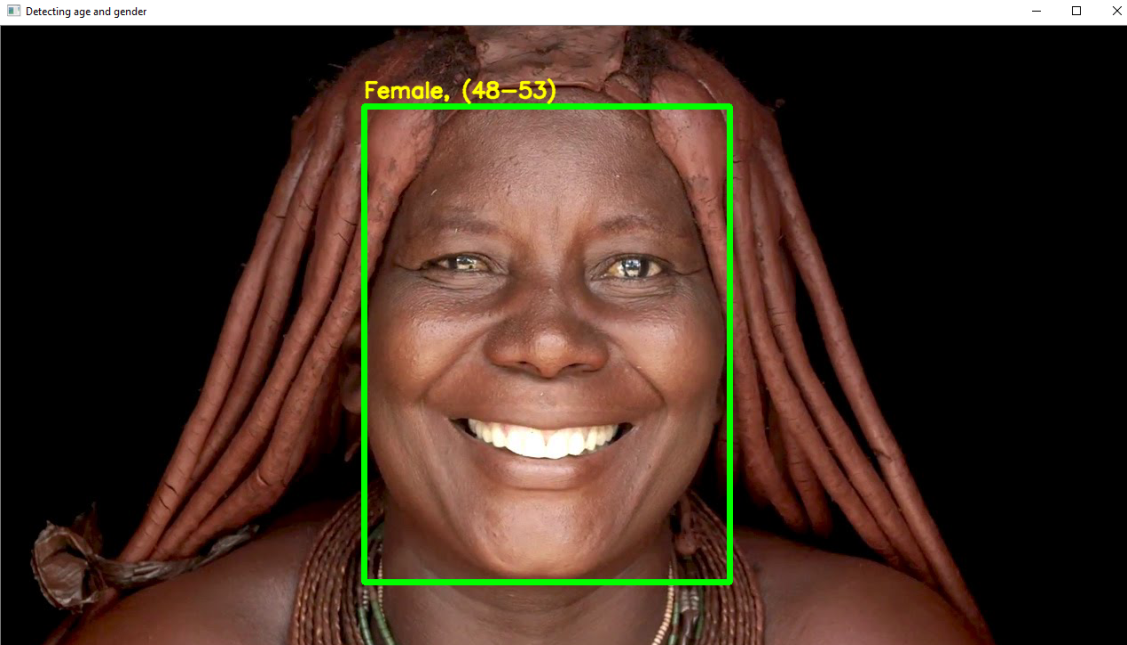
**Output:**

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2019/09/python-open-source-projects.png)

**Python project Example 6**

[python project age detection](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2019/09/Python-project-ideas.png)

**Output:**

[](https://d2h0cx97tjks2p.cloudfront.net/blogs/wp-content/uploads/sites/2/2019/09/advanced-python-project-age-detection.png)

Summary

In this python project, we implemented a CNN to detect gender and age from a single picture of a face. Did you finish the project with us? Try this on your own pictures. Check more***cool projects in python with source code*** published by DataFlair.