Journal of Engineering Sciences
Vol 11, Issue 03, March/2020
ISSN NO:0377-9254

A MACHINE LEARNING BASED APPLICATION TO IDENTIFY GENDER IN IMAGES

Kommu Gangadhara Rao
Assistant Professor
Kgangadhar_it@cbit.ac.in
Department of Information Technology
Chaitanya Bharathi institute of technology
Hyderabad, India

ABSTRACT: In this Work, we have used Python, Deep Learning on Adience data set to accurately identify the gender and age of a person from a single image of one or more faces. We implemented a CNN to detect gender and age from a single picture of a face. The predicted gender may be one of 'Male' or 'Female', and the predicted age maybe one of the following ranges- (0 – 2), (4 – 6), (8 – 12), (15 – 20), (25 – 32), (38 – 43), (48 – 53), (60 – 100). Thus, the age prediction network has 8 nodes in the final softmax layer indicating the mentioned age ranges. 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. This work can be helpful in many real time applications like smarter advertising, to prevent crime, to find missing persons, and also in beauty industry. This Work is based on convolution neural network and this Work is executed by using OpenCV.

Keywords: *Deep Learning, CNN, Python, softmax, OpenCV*

I. Introduction

1.1 Motivation

Age and Gender detection is a python Work in which we have used deep learning on Adience dataset. We have framed Gender Prediction as a classification problem. The output layer in the gender prediction network is of type softmax with 2 nodes indicating the two classes "Male" and "Female". Ideally, Age Prediction should be approached as a Regression problem since we are expecting a real number as the output. However, estimating age accurately using regression is challenging. Even humans cannot accurately predict the age based on looking at a person. However, we have an idea of whether they are in their 20s or in their 30s. Because of this reason, it is wise to frame this problem as a classification problem where we try to estimate the age group the person. For example, age in the range of 0-2 is a single class; 4-6 is another class and so on. The Adience dataset has 8 classes divided into the following age groups [(0 – 2), (4 – 6), (8 – 12), (15 – 20), (25 – 32), (38 – 43), (48 – 53), (60 – 100)]. Thus, the age prediction network has 8 nodes in the final softmax layer indicating the mentioned age ranges.

1.2 Objective of the Work

This study intends to develop a model for age and gender detection. The objectives of the Work are mainly to detect faces, classify into male/female, classify into one of the 8 age ranges then put the results in image and then display it.

1.3 Observations

Even though the gender prediction network performed well, the age prediction network fell short of our expectation.[1] We tried to find the answer in the paper and found the following confusion matrix for the age prediction model.

	0-2	4-6	8-13	15-20	25-32	38-43	48-53	60-
0-2	**0.699**	0.147	0.028	0.006	0.005	0.008	0.007	0.009
4-6	0.256	**0.573**	0.166	0.023	0.010	0.011	0.010	0.005
8-13	0.027	0.223	**0.552**	0.150	0.091	0.068	0.055	0.061
15-20	0.003	0.019	0.081	**0.239**	0.106	0.055	0.049	0.028
25-32	0.006	0.029	0.138	0.510	**0.613**	0.461	0.260	0.108
38-43	0.004	0.007	0.023	0.058	0.149	**0.293**	0.339	0.268
48-53	0.002	0.001	0.004	0.007	0.017	0.055	**0.146**	0.165
60-	0.001	0.001	0.008	0.007	0.009	0.050	0.134	**0.357**

Figure 1.3: Age estimation confusion matrix on the Adience benchmark.

The following observations can be made from the above table

- The age groups 0-2, 4-6, 8-13 and 25-32 are predicted with relatively high accuracy. (see the diagonal elements)

- The output is heavily biased towards the age group 25-32 (see the row belonging to the age group 25-32). This means that it is very easy for the network to get confused between the ages 15 to 43. So, even if the actual age is between 15-20 or 38-43, there is a high chance that the predicted age will be 25-32. This is also evident from the Results section.

Apart from this, we observed that the accuracy of the models improved if we use padding around the detected face. This may be due to the fact that the input while training were standard face images and not closely cropped faces that we get after face detection[2].

We also analyzed the use of face alignment before making predictions and found that the predictions improved for some examples but at the same time, it became worse for some. It may be a good idea to use alignment if you are mostly working with non-frontal faces.

1.4 Adience Dataset

For this python Work, we have used the Adience dataset; the dataset is available in the public domain. 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 [3] . 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.

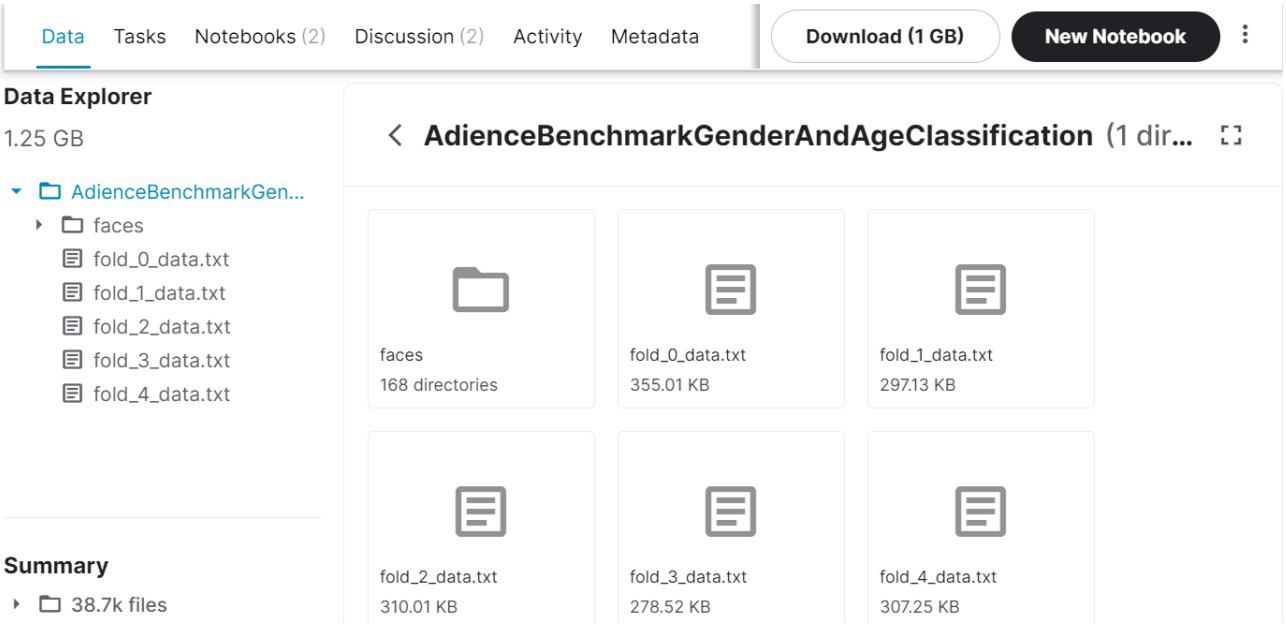


Figure 2.1: Adience dataset.
Source: Kaggle.com

III. Proposed System

3.1 Problem Statement

The goal of this Work is to create a model that will be able to detect and determine the age and gender from image by using the concept of classification, Though the goal is to create a model which can detect the age and gender , it can be extended to many other things in real time world.

3.2 Methodology

The following methodology applied while developing this model.

1. Programming language python is used.
2. Adience dataset is used.
3. Need to install OpenCV (cv2), Keras libraries .
4. Other packages you'll be needing are math and argparse, but those come as part of the standard Python library.
5. The images are tested and trained with the dataset.
6. The output will be displayed using OpenCV through command prompt.

Steps for practicing gender and age detection python Work :

1. Download the zip file. 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 Work on.

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, initializes protocol buffer and model.
4. Initialize the mean values for the model and the lists of age ranges and genders to classify form.
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[5].
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 result Img 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 and age texts to the resulting image and display it with imshow().

3.3 Proposed Architecture

The convolutional neural network for this python Work has 3 convolution layers:

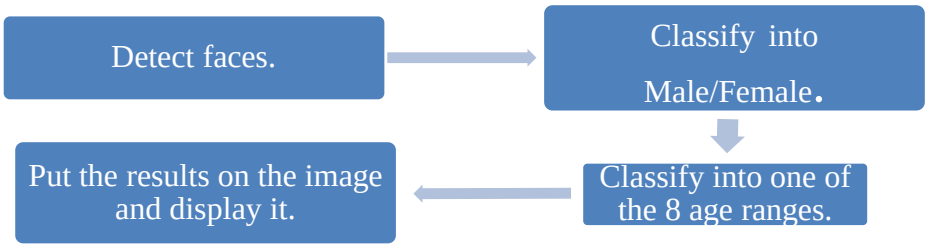
- Convolutional layer; 96 nodes, kernel size 7
- Convolutional layer; 256 nodes, kernel size 5
- Convolutional layer; 384 nodes, kernel size 3

It has 2 fully connected layers, each with 512 nodes, and a final output layer of softmax type. To go about the python Work, 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 process of the execution is presented the form of flow chart in the Fig 3.3.



IV. RESULTS

The result or output of this program is very simple. Once you open command prompt and specify the directory in which the pictures are and type "py gad.py –image imagename.jpg" and following outputs are displayed according to the pictures.

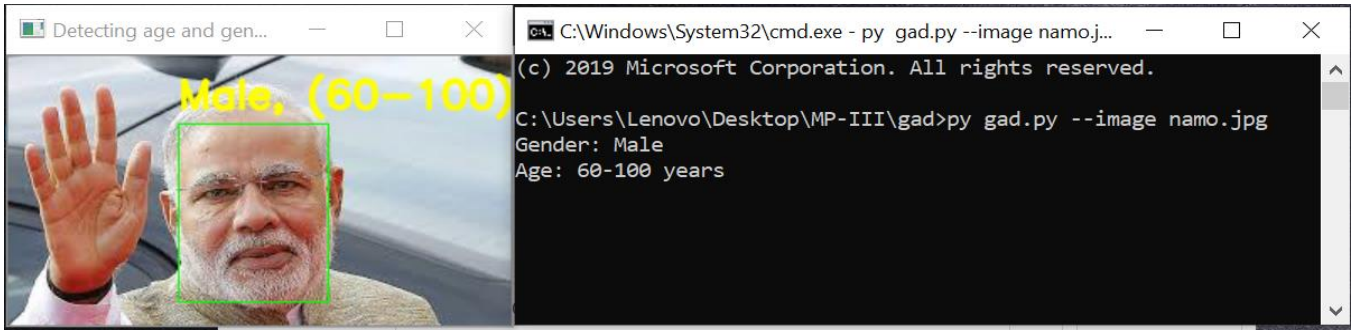


Figure 5.1: Age of male (60-100)

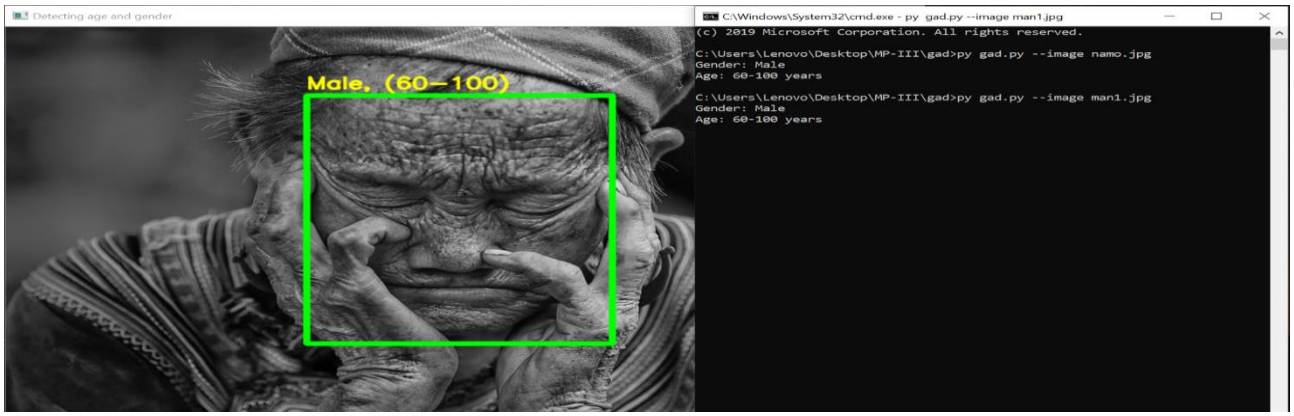


Figure 5.2: Age of Female (60-100)

Figure 5.3: Age of Female (38-43)

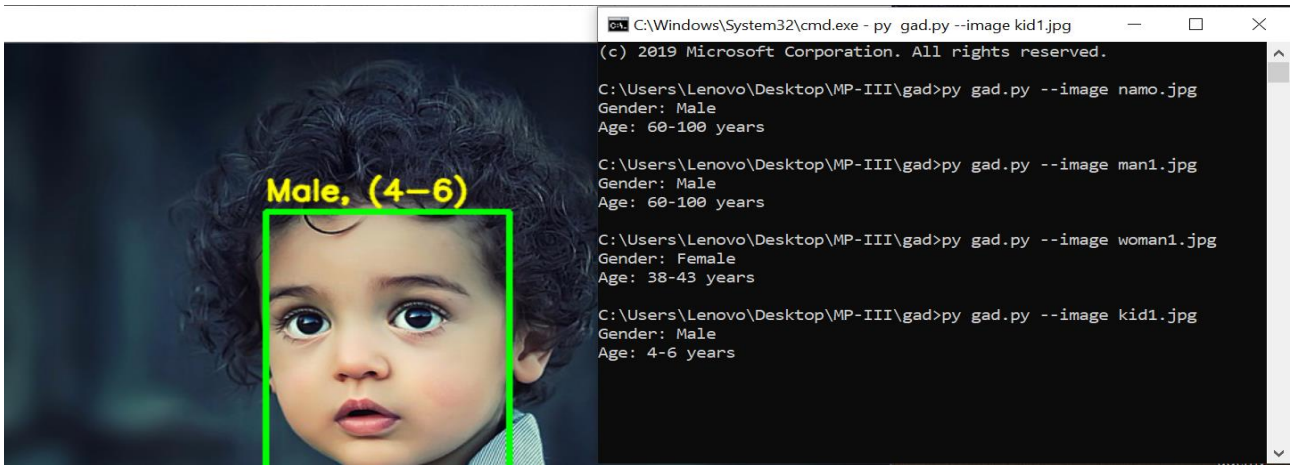


Figure 5.4: Age of Male (4-6)

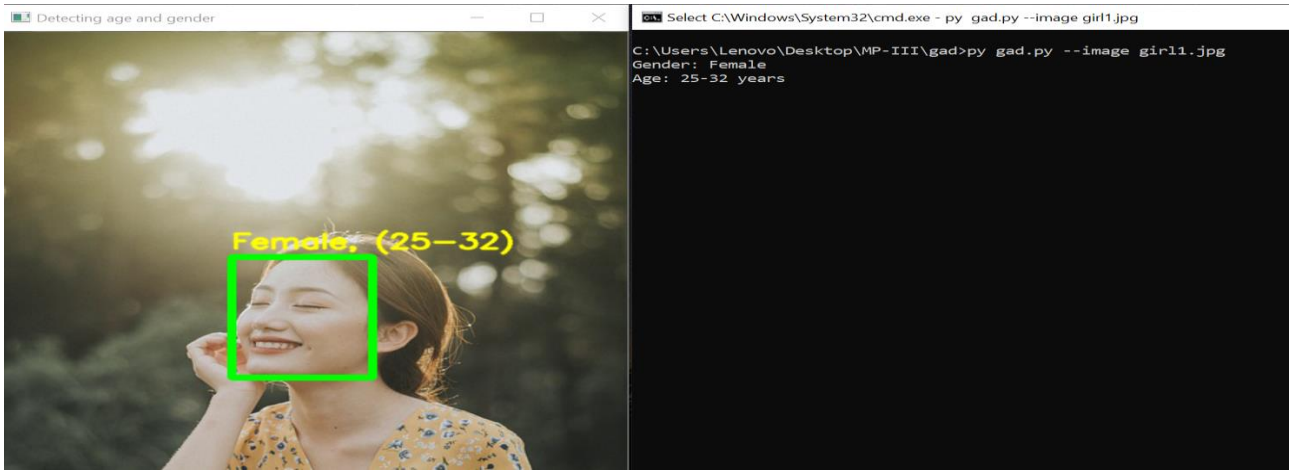


Figure 5.5: Age of Female (25-32)

V. CONCLUSIONS AND FUTURE SCOPE

In this Work we have used adience data set which has been trained by Tal Hassner and Gil Levi. Overall, we think the accuracy of the models is decent but can be improved further by using more data, data augmentation and better network architectures. One can also try to use a regression model instead of classification for Age Prediction if enough data is available. This Work is useful in detecting faces and their age gaps in cameras. This application is also useful in detecting faces through a CCTV camera. This Work is very feasible and affordable. It can easily be implemented and it is the best use of data analytics. For the Work, one needs to possess a good knowledge of python languages and data manipulation. The objective of the Work has been achieved. A decent model of age and gender detection has been built.

The Gender Classification and Age Detection algorithms can be implemented with an increased number of facial image data set. This will increase the accuracy level of the output.Moreover; there are several other applications where gender recognition plays a crucial role which includes biometric authentication, high technology surveillance and security systems, image retrieval, and passive demographical data collections. We are expecting that the future scope work in this is to involve using face age, human expression classification to aid face recognition, facial disease detection, improve experiences with images, pics of social media, and much more than this.

References:

[1] A. Saxena, P. Singh and S. Narayan Singh, "Gender and Age detection using Deep Learning," *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2021, pp. 719-724, doi: 10.1109/Confluence51648.2021.9377041.
[2] X. Wang, A. Mohd Ali and P. Angelov, "Gender and Age Classification of Human Faces for Automatic Detection of Anomalous Human Behaviour," *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*, 2017, pp. 1-6, doi: 10.1109/CYBConf.2017.7985780
[3] B. Delgado, K. Tahboub and E. J. Delp, "Automatic Detection of Abnormal Human Events on Train Platforms", *IEEE National Aerospace and Electronics Conference*, no. 2009, pp. 169-173, 2014.
[4] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks", *European Conference on ComputerVision*, pp. 818-833, 2014.
[5] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks", *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 11-18, pp. 4489-4497, 2016.
[6] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", *Conference on ComputerVision and Pattern Recognition*, pp. 1-5, 2001.