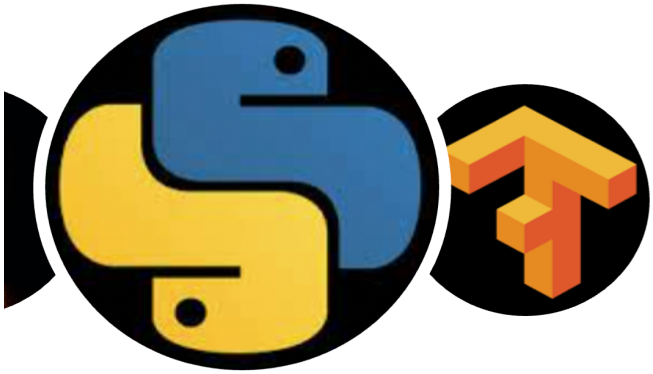
8/28/2021

Internship Report

Audience Demographics: Age and Gender Detection



Pooja Vijay Mahajan
IBC CUBE (INTELLIGENT BUSINESS COMPUTER SYSTEMS PVT. LTD.)

IBC CUBE INTERNSHIP REPORT

Use Cases:

- Age Detection
- Gender Detection

GitHub Project Link:
https://github.com/AngikarGhosal/IBC-Cube-ML-Internship-21/tree/main/Pooja

Objective:
To build a gender and age detection model that can approximately guess the gender and age of the person (face) using deep learning on the images/videos captured from IBC Cube Smart Cameras.

Audience Demographics
Motivation behind Demographics detection: Why do we need to predict demographic characteristics?

Predicting demographic data of people has many business applications. Today's marketing and decision-making process in organisations became increasingly data-driven, deploying data-driven analytics, marketing, and other solutions to their processes.

And demographic characteristics such as gender, age, ethnicity, or income level is one of the most crucial pieces of big data. Knowing this data empowers businesses and organisations in many ways. It helps knowing your customer better, it helps understanding your market, it enables more in-depth research and analysis for educational or scientific purposes. Some use cases include:

- **Demographic targeting** in advertising platforms like Google Ad Words or Facebook Ads
- **Segmenting customer and prospect lists** to deliver right offer to the right person. For example, if you're a clothing brand you want to offer dresses to women while ties to men
- **Demographics analytics** of your customers or prospects to understand your target audience and back your decisions with data

This includes:

- Age: 1-10, 10-20, 20-40, 40-50, 50-80
- Gender: Male, Female
- Nationality/Ethnicity: Middle Eastern, Latino Hispanic, East Asian, Black, Southeast Asian and Indian, Afro-American, Caucasian
- Apparel: Shirt-tie (indication corporate sector), causal clothing
- Income
- Education
- Socio-Economic Status
- Occupation
- Family life cycle

Project Overview:
In this python project, deep learning has been used to accurately identify the gender and age of a person from a single image of a face. The models used are trained on Adience dataset by Tal Hassner and Gil Levi. 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.

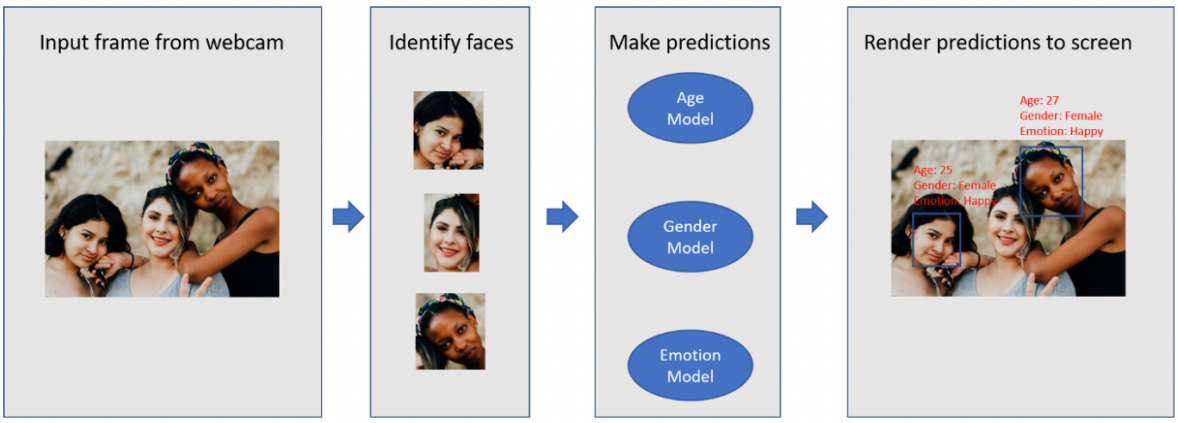
The CNN Architecture:
The convolutional neural network for this python project has 3 convolutional layers:

- 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, which 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. 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.

Model Overview:
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

1. 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 and age texts to the resulting image and display it with imshow().

Project Example:

Example 1:

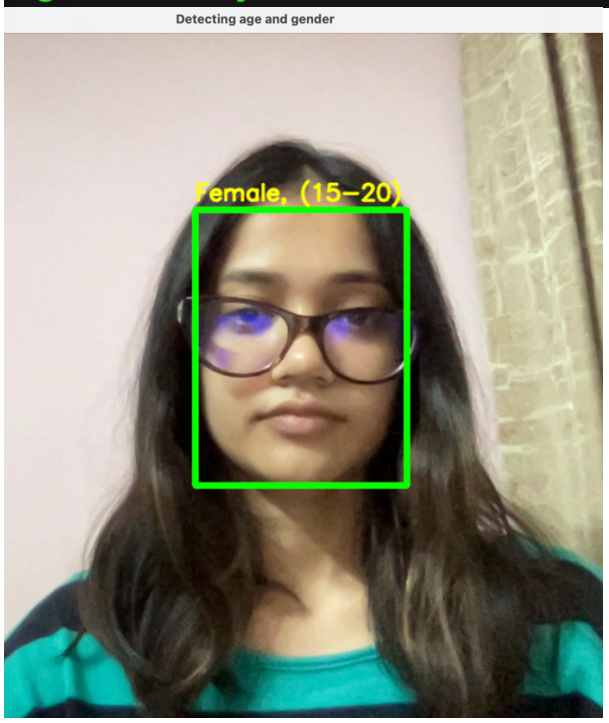
```
(base) poojamahajan@Poojas-MacBook-Air CNN % python gad.py --image 6.jpg
Gender: Male
Age: 4-6 years
```

Example 2:

```
(base) poojamahajan@Poojas-MacBook-Air CNN % python gad.py --image 3.jpg
Gender: Male
Age: 38-43 years
Gender: Male
Age: 8-12 years
```

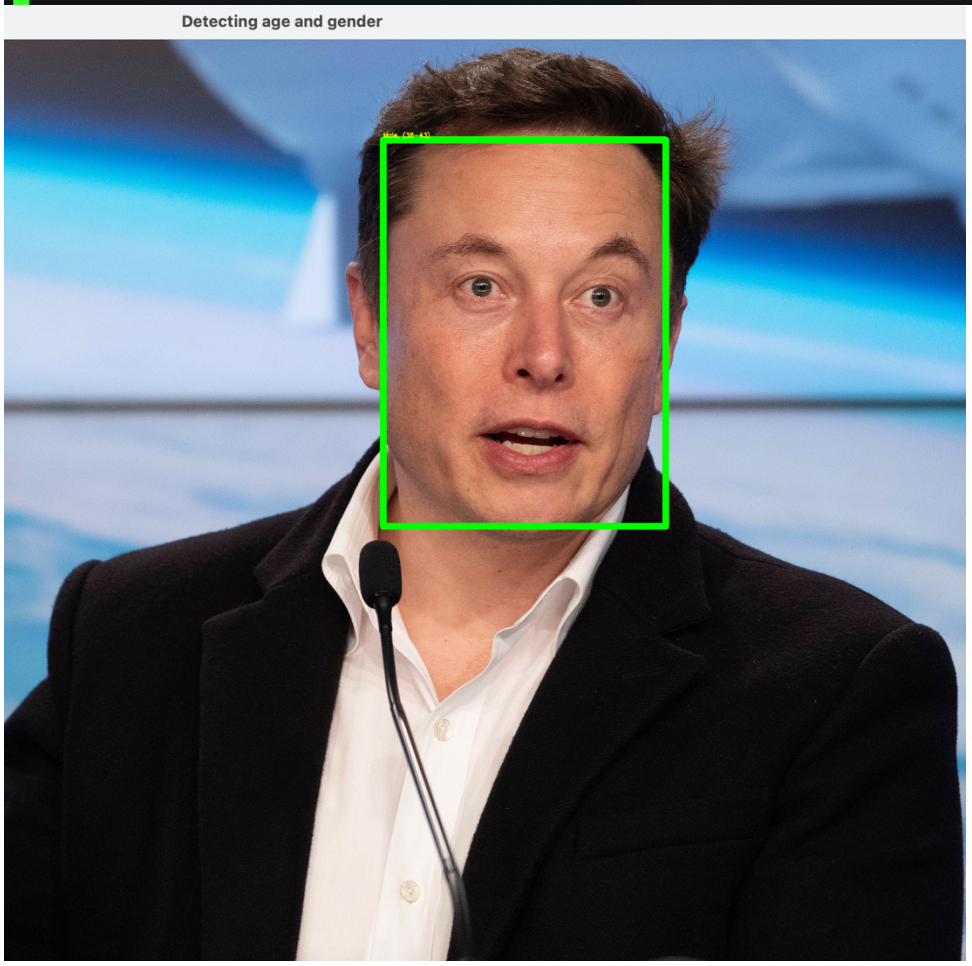
Example 3:

```
[(base) poojamahajan@Poojas-MacBook-Air CNN % python gad.py
No face detected
Gender: Female
Age: 15-20 years
Gender: Female
Age: 15-20 years
Gender: Female
Age: 25-32 years
```



Example 4:

```
(base) poojamahajan@Poojas-MacBook-Air CNN % python gad.py --image 2.jpeg
Gender: Male
Age: 38-43 years
```



References:

1. https://towardsdatascience.com/data-science-powered-segmentation-models-ae89f9bd405f

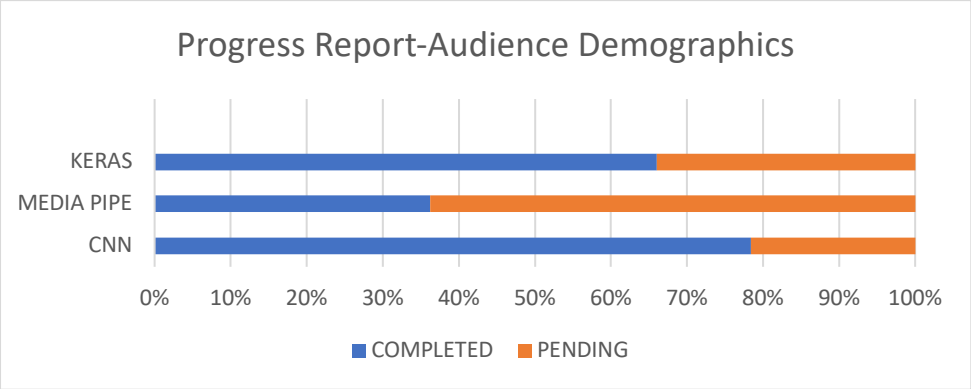
2. https://demografy.medium.com/artificial-intelligence-can-now-predict-demographic-characteristics-knowing-only-your-name-6749436a6bd3

3. https://www.tensorflow.org/

4. https://becominghuman.ai/detecting-age-and-gender-with-tf-lite-on-android-33997eed6c25

5. https://data-flair.training/blogs/python-project-gender-age-detection/

TO DO LIST:



WORKING ON 3 DIFFERENT MODELS:

KERAS MODEL (60% COMPLETED):

DATASET: UTKFACE DATASET

IMPLEMENT KERAS MODEL WITH CONV2D LAYERS

NEED TO TRAIN THE MODEL AND CONVERT IT INTO TFLITE

MEDIA PIPE (30% COMPLETED):

BASED ON BLAZEFACE, A LIGHTWEIGHT AND WELL-PERFORMING FACE DETECTOR TAILORED FOR MOBILE GPU INFERENCE.

NEED TO BUILD AGE AND GENDER DETECTION MODELS FOR FACES CROPPED AFTER MEDIA PIPE FACE DETECTION

CNN (80 % COMPLETED):

DATASET: ADIENCE

USED OPENCV TO DETECT FACES AND CNN TO RUN AGE AND GENDER DETECTION MODELS.

NEED TO CONVERT IT INTO TFLITE.

FINAL RESULT: COMPARE ALL 3 MODELS TRAINED ON DIFFERENT DATASETS AND FIND THE BEST MODEL WITH HIGHEST ACCURACY AND LEAST MISCLASSIFICATIONS.