Age and Gender Prediction

Nihal Syed
101107401

Abstract

With advancements in computer hardware, the field of computer vision has taken leaps in all aspects from graphics to having computers understand the world around us. Computers can be able to identify the world around us and extract meaningful data, in specific feature extraction and detection can allow us to even understand who a person is based on their face. The most popular tool for pattern recognition that has taken over computer vision, is the Convolutional Neural Network (CNN). With a CNN we can train a neural network to extract information and recognize patterns in images, to be able to determine what the image is. This project is focused on using and training a dataset with a CNN to be able to determine the age and sex of a person in the image. Through this project, I trained a CNN by running a dataset through it and reached 89% accuracy in determining sex and a 50% accuracy in detecting age. We then use a trained model to detect a face in a video stream and prints out the predicted age and sex of the individual.

Introduction

Face detectors have evolved to be able to identify specific characteristics of a person, similar to how we are able to identify who somebody is based on their appearance. With the use of CNNs and a large dataset, we can be able to add additional pattern recognition for a varied array of uses. The project of building a CNN that is able to predict characteristics like age and gender can be helpful in many cases such as security uses, targeted marketing, and identifying

store demographics. Understanding how computers are able to recognize patterns brings on more ideas of general use cases as prediction models like these can have a vast array of use cases. I chose this project because it is a good introduction to multi-task learning and Neural Networks. The main challenges encountered during this project were understanding the frameworks necessary to implement and train this network, such as TensorFlow and Keras, as well as the process of cleaning and processing a dataset so it is clean to train with. Luckily, there are many age and sex-labeled datasets available for research that only requires minor cleaning to be able to train and test.

Background

This project is based on the paper published by Gil Levi and Tal Hassner from the University of Israel. Their approach uses a CNN with multiple layers parameterized for image classification. This approach far exceeds any approaches used without a neural network in terms of accuracy, the only loss is the time necessary to train a model for specific classification as well as needing a large enough dataset. However, Levi and Hassner's proposed method produces relatively accurate results even with limited datasets. In their paper, they use the Adience dataset to train their algorithm. This project follows the same approach to create and train a model to be able to accurately determine age and sex, as well as using a model to predict age and sex in real time with a webcam stream.

Methodology

The first process of this project was getting the dataset and preparing it for training and testing. This means going through all images and labels in the dataset and removing any null entries or labels. The original dataset consists of 26580 images and text files for validation on all images. Additionally in preprocessing we must make sure the images are equally distributed

between males and females, which can be seen in figure 3. After processing the images to only take necessary data and filtering null values we have a cleaned dataset with 19345 entries that are trainable. We must then go through the cleaned dataset and split it with the sklearn framework, this splits our data into a test and train split, we then have to take this data and resize all images to (227,227) and convert them to NumPy arrays so that we can add them through the model. The model architecture is explained in the Architecture section of this paper. The main process of experimenting with the model I had done was to change hyperparameters such as batch size and epochs. The batch size defines the number of samples to go through before updating internal model parameters, while epochs define the number of times that the model will work through the entire training dataset. However, with respect to the time that training a model can take, I limited the epochs to test on. With the best prediction accuracy in my tests were models trained with a batch size of 32 over 25 epochs. These age and gender models are created and trained separately, however, this can also be done with multi-task learning having our neural network output two final predictions. Once a model is trained it can be saved with all the weights and parameters saved to be able to use in other aspects. To test the use and speed of such a model I read a webcam frame by frame to detect a face in the image with a faceCascadeClassfier and cropped the detected face from the webcam output and passed that image through the model to get a prediction on sex and age, this is then printed on the screen displaying the webcam footage.

Architecture

The model architecture is built off of 3 convolutional layers and two fully-connected layers as can be seen in Figure 1. The CNN follows these steps by first processing all three color channels in an image and is rescaled to (256, 256) and cropped to (227,227). These images are then passed through the convolutional layers. With the first convolutional taking input and

running it through the layer having parameters of 96 filters with a size (3,7,7), followed by a rectified linear operator, or ReLU function, and end the convolutional layer with a max-pooling layer and a local response normalization layer. The same architecture is used with the other two convolutional layers other than that the second convolutional layer contains 256 filters of size (96,5,5) , and the third convolutional layer takes a (256,14,14) blob and applies 384 filters of size (256,3,3). These convolutional layers are followed by three fully connected layers where: the first layer contains 512 neurons and a ReLU and dropout layer, the second layer receives a 512-dimensional input and contains 512 neurons with a ReLU and dropout layer, the third layer maps to the final classes for sex and age. This output is then given to a soft-max layer and assigns a probability for the prediction. This is the CNN model specified in Levi and Hassner's paper to get the best prediction results.

Results

After training our model with a batch size of 32 over 25 epochs, after training our accuracy for sex detection is 60.05%, while our age detection accuracy is 52%. As you can see these are not optimal results that are necessary to accurately determine age and sex. These low accuracy results can be a result of a low epoch and batch size number which made training the model faster, if I were to retrain the model I would increase the number of epochs greatly. Additionally, the first training split for age had a very low accuracy score of 0.4% because my model was trying to predict exact age. To improve this problem I had to divide ages into a range of ages, as predicting age can be hard due to many different problems such as people aging differently, makeup, or lighting and angle differences. In terms of angles, the dataset contains thousands of images for angled faces to train over as well. With these low accuracy results I had originally used the same model in my webcam capture program to use that prediction, however,

the accuracy was very off and would need more iterations over the dataset to improve accuracy, for this reason in the webcam detection script I use the Caffe model that has been trained by Levi and Hassner. The results can be seen in the image below, with the program drawing a rectangle over a detected face and cropping the image to input into the model. The model outputs the prediction and is then displayed on the screen.

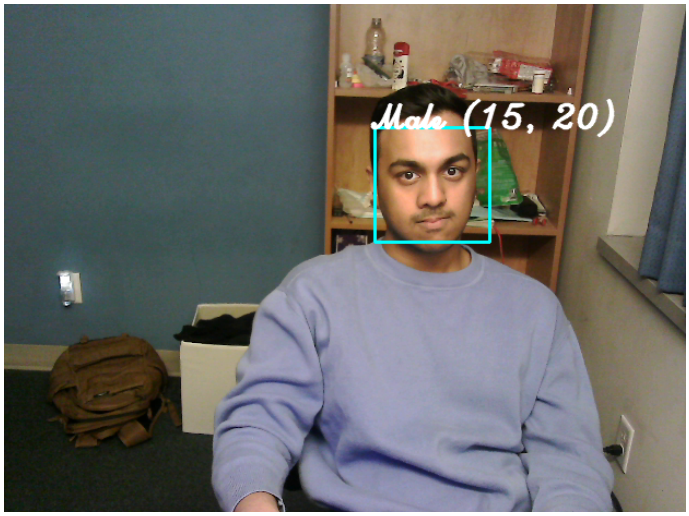


Image 1. Webcam output predicting age and gender

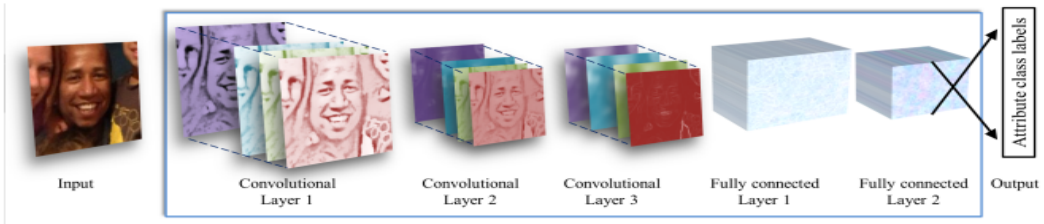
Discussion

The main limitation of my finding was the time necessary to perform training of a model over a very large set of epochs. However, with that can come the issue of overfitting where the model becomes too accurate on only the dataset images. This project is relevant in today's world with the impact and data social media can gather. Being able to identify features of somebody based on what the computer sees can be used for a broad range of uses in terms of marketing to understand demographics better and pragmatically. One feature I wanted to implement was a text-to-speech compliment based on detected age and sex, although this is a minor feature. In terms of future work that can build off of this project is making identifier-specific processes

happen. The main limitation I found was the time necessary to train models as well as further knowledge to change the internal parameters of the model to produce better accuracy.

Figures and Charts

Figure 1. Illustration of CNN Architecture(source)




```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)            (None, 56, 56, 96)        14208

 max_pooling2d_3 (MaxPooling  (None, 28, 28, 96)        0
 2D)

 layer_normalization_3 (Laye  (None, 28, 28, 96)        192
 rNormalization)

 conv2d_4 (Conv2D)            (None, 28, 28, 256)       614656

 max_pooling2d_4 (MaxPooling  (None, 14, 14, 256)       0
 2D)

 layer_normalization_4 (Laye  (None, 14, 14, 256)       512
 rNormalization)

 conv2d_5 (Conv2D)            (None, 14, 14, 256)       590080

 max_pooling2d_5 (MaxPooling  (None, 7, 7, 256)         0
 2D)

 layer_normalization_5 (Laye  (None, 7, 7, 256)         512
 rNormalization)

 flatten_1 (Flatten)          (None, 12544)             0

 dense_3 (Dense)              (None, 512)               6423040

 dropout_2 (Dropout)          (None, 512)               0

 dense_4 (Dense)              (None, 512)               262656

 dropout_3 (Dropout)          (None, 512)               0

 dense_5 (Dense)              (None, 2)                 1026

=================================================================
Total params: 7,906,882
Trainable params: 7,906,882
Non-trainable params: 0
```

Figure 2. My CNN architecture summary, age, and gender models follow the same architecture

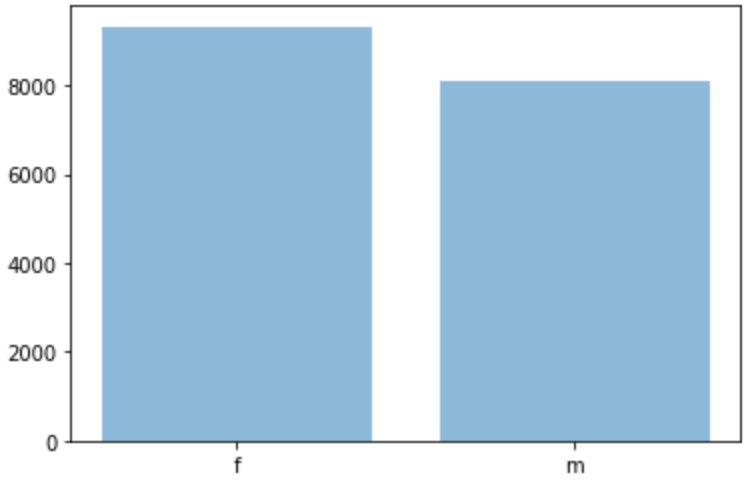


Figure 3. Gender Dataset Distribution Graph

References

Levi, G., & Hassncer, T. (2015). Age and gender classification using Convolutional Neural Networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. https://doi.org/10.1109/cvprw.2015.7301352