Title: Gender Classification using Convolutional Neural Networks

Introduction

Automatic gender and age classification has become very popular in many areas, especially in all kinds of social media and websites. This surging amount of user-generated data online has pushed the development of sophisticated algorithms for face detection as well as age and gender classification. Gender and age classification can also be used in commercial, research, military, finance and security fields.

Problem Formulation

As tremendous progress has been made by using deep convolutional neural networks (CNN) on face recognition techniques, a CNN model was trained from scratch to perform gender estimation on images of human faces. Audience benchmark database of unfiltered face images were used. Comparison was made between my trained model with a fine-tuned version of Google's inception model V3 that is available online.

Model

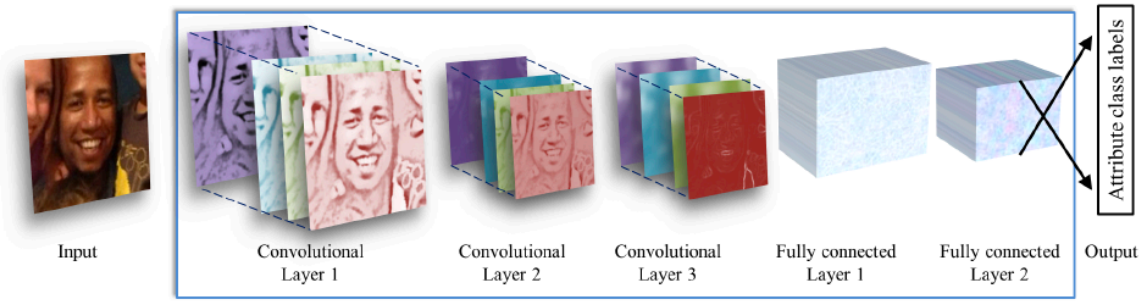
Data: total number of images from the dataset is 26,580 and total number of subjects is 2,284.

Preprocessing: The dataset was processed by first running the Viola and Jones face detector, aligned to a reference coordinate frame and then discarded if faces angels were at a greater than $\pm$ 45° yaw angle from 0 (forward facing). Finally, all images were manually labeled for age, gender, which left us with 19539 images altogether.

Network architecture:

Target for simplicity: 3 convolutional layers and 2 fully connected layers to lower the complexity since gender classification is a simple binary classification problem in contrast

to more complicated vision recognition problems that requires hundreds of labels as in other networks.



This network contains three convolutional layers, each followed by a rectified linear operation and a pooling layer. The first two layers were followed by normalization using local response normalization. The first convolutional layer contains 96 filters of 7×7 pixels, the second convolutional layer contains 256 filters with the size of 5×5, the third and final convolutional layer contains 384 filters of 3×3 pixels. Eventually, two fully connected layers were added, each containing 512 neurons followed by a RELU and a dropout layer. A third, fully connected layer was used to map to the final classes for gender. Finally, the output of the last fully connected layer was fed to a soft-max layer that assigned a probability of each class. The prediction was made by taking the class with the biggest probability.

Training and Testing:

Aside from use of network architecture, the dropout ratio is 0.5 for the last two fully connected layers. Weights initialization are random values from a 0 mean Gaussian with standard deviation of 0.01. Data was also augmented by extracting five 227 ×227 pixel crop regions, four from the corners of the 256 ×256 face image and one region from the center of the face. All 3 RGB channels were processed.

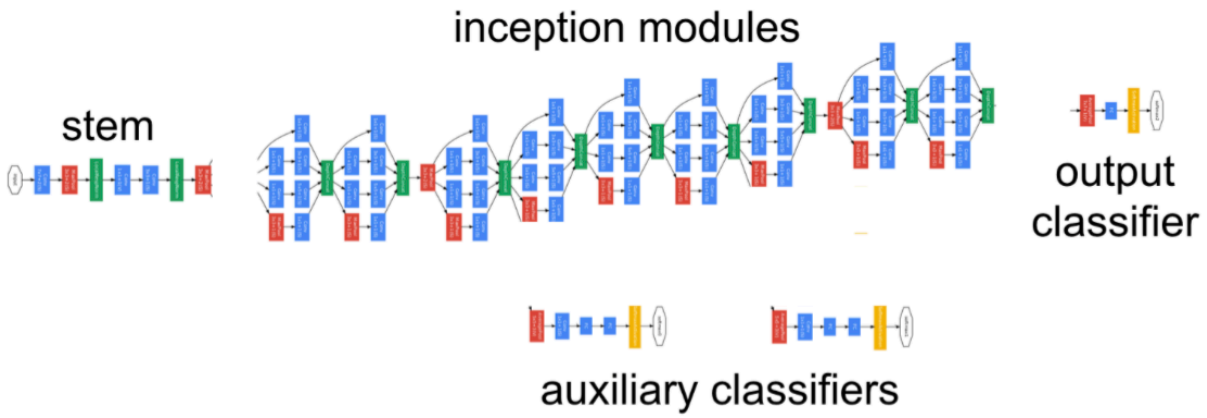
Training was performed using stochastic gradient decent batch size of 128 images.

Testing for gender classification is performed using a ***five-fold cross validation*** protocol.

Result

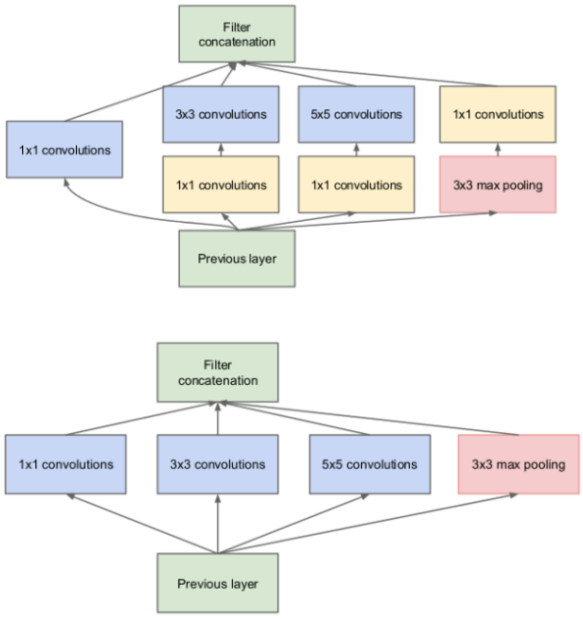
Class (label)	Folder_0	Folder_1	Folder_2	Folder_3	Folder_4
Female(@1)	0.897 (1263/1408)	0.910 (1281/1408)	0.906 (1391/1536)	0.904 (1388/1536)	0.911 (1399/1536)
Male(@2)	1.000 (1408/1408)	1.000 (1408/1408)	1.000 (1536/1536)	1.000 (1536/1536)	1.000 (1536/1536)

Comparison with Google inception:



GoogleNet starts with a sequential chain of convolution, pooling and local response normalization operations, in a similar fashion to previous convolutional neural models, such as AlexNet. The stem segments are referred to as ‘inception’ modules. The inspiration comes from the idea that instead of trying to make a decision as what type of convolution you want to make at each layer: such as 3×3 or 5×5, why not use all of them and let the model decide? The way to do this by doing each convolution in parallel and concatenating the resulting feature maps before going to the next layer.

Architecture



Upper figure: inception module from “going deeper with convolutions” lower: naïve inception model. The difference between these two modules is the lack of 1 $\times 1$ convolutional layer before the large convolutions. This architecture with 1×1 covolutions, dimensions are reduced. This also results in a high performing model with drastically fewer parameters. And the name it is called inception is because the module represents a network within a network.

The above diagram shows this inception model contains 9 of these modules, sequentially stacked, with two max pooling layers along the way to reduce the spatial dimensions. Finally, the output classifier, which performs an average pooling operation followed by a softmax activation on a fully connected layer.

Conclusion and discussion

In conclusion, we were able to use CNN to classify gender from facial images with very high accuracy. We also compared our model with GooleNet which is a much more complex image recognition system.

Even with high accuracy by using our pre-processed dataset to test, the images downloaded randomly from the internet may not give us satisfying result especially the ones without frontal view images, this could simply because our training dataset deliberately discarded angled faces whose facial features are too hard to extract. Also from the paper, a lot of gender misclassifications were made on babies which we think is quite understanding since babies gender features are not distinct anyways.

The GoogleNet apparently is a more complex image recognition model and were trained on a very large dataset containing all categories of images. By fine tuning the parameters on our dataset whatever mistakes our own model made did not show in their model. One important conclusion is the simplicity of our CNN model implies more elaborate systems using more training data may well be capable of substantially improving results.