Facial Gender Recognition

COMP 4601
Hector Zhang (1009479), Rui Li (100987686)

1.Abstract

Facial gender recognition has been relevant to a lot of applications. Since recent advances in convolutional neural network (CNN), a significant increase in performance can be obtained on this task. In this project, we will explore how to build a CNN for gender classification as well as evaluate the performance of our model. At the end, an android mobile app will be developed to demo our trained gender classification model.

2. Introduction

Extracting information from human facial image can be useful in various scenarios. For instance, a website can automate part of the new user signing up process by taking user's facial image using webcam. The traditional face recognition algorithms rely on facial feature models. Different model need to be built for different classification problems. The advantage of using CNN for image classification is that it uses relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. [1] Therefore, a lot of human efforts can be saved by using this approach.

In this project, we focus on building, training, and evaluating a CNN model for human facial gender estimation. An android application is developed to demo our model. Our CNN implementation is mainly based on the model proposed in [2]. The purpose of doing this project is to learn the convolutional neural network.

The rest of the report is structured as follow. Section 3 presents background information about the potential model which will be used to find the solution for the problem stated above. Section 4 describe the related work already been done by others. In section 5, we elaborate our solution to the problem in detail. Obtained result from experiment will also be analysed here. In section 6, we provide a critical review of what we achieved in this project. Finally, in section 7, we conclude this project and indicates possible future works that may improve this project.

3. Background

CNN is a class of deep, feed-forward artificial neural networks that has successfully been applied to analysing visual imagery [1]. A typical CNN consists of an input layer, multiple convolutional layers, multiple fully connected layers, a flatten layer, and an output layer. (Note the convolution process in CNN is actually a cross-correlation rather than a convolution) [1].

4. Related Work

The implementation of this project is based on the model proposed in [2]. A detailed evaluation of different gender classification methods can be found in [3]. Here we give a quick summary about the relevant methods mentioned in [3]: Moghaddam and Yang developed the first automatic system for combined face and gender classification. They found that Support Vector Machine (SVM) performed than other classifiers. The [3] discussed four gender classification methods: a multilayer neural network with pixel-based input, an SVM with pixel-based input, a discrete Adaboost with haar-like features, and an SVM with LBP features. The author of [3] concludes that SVM with 36*36 pixel face images as input achieved the best gender classification rate.

5. Methodology

5.1. Data Set

The data set used for training and validating our model is the Adience benchmark. This data set is published by Face Image Project from the Open University Israel (OUI). It consists of 26580 facial images (cropped) of 2284 subjects. The images are taken from Filckr so the images are less constrained [5]. This makes the images more challenging to be classified but can reflecting many real-world problems of classifying faces appearing in Internet. The author of [5] splits the images into 5 folds so that the same subject does not appear in different folds. If the images were simply randomly shuffled and divided, it will be possible that pictures of the same subjects appear in both training and testing set. This can affect the accuracy rate when testing the model and need to be avoid.

5.2. Data Pre-processing

Some of the images missing the gender label (Since this data set is not designed only for gender classification). Also, there exists images that have the gender of unisex because the subject is a baby (e.g. Age of 0-2). All those images are filtered out, so our data set is reduced to the following size as shown in table 1.

Fold	#Males	#Females	#Total
0	2047	1948	3995
1	1611	1998	3609
2	1363	1774	3137
3	1502	1804	3306
4	1597	1848	3445

Table 1 Gender Dataset

5.3. CNN Model Architecture

Our CNN implementation is mainly based on the model proposed in [2].

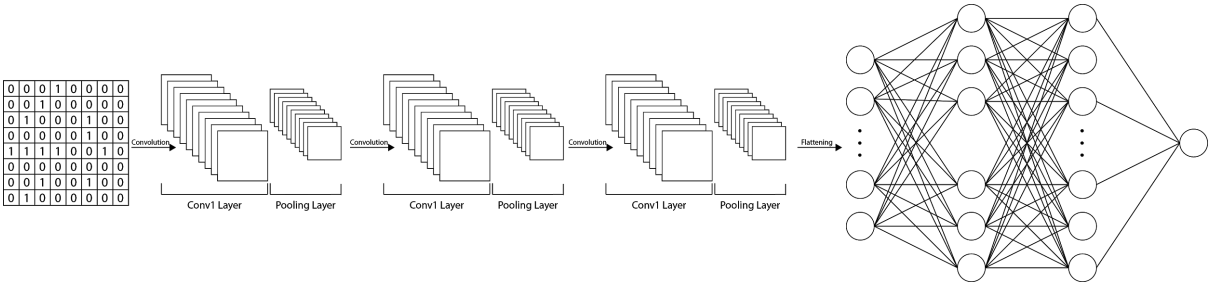

Figure 1 CNN Model

The hidden layers of this model include three convolutional layers and two fully connected layers. The depth of this network is relatively shallow since we only need to differentiate two gender classes. Another advantage of a shallow neural network is that the risk of overfitting is reduced. [2]

The details of the network are as following:

1.The input to the network is a 3*64*64 image.

2. Local response normalization (LRN) layer is not added after the first two max pools. The intent of LRN is to introduce inhibition schemes observed in the biological brain. However, those layers have minimal impact in practice. [6]

3. The first convolutional layer has 96 filters of size 3*7*7. The second convolutional layer then process 96 outputs from the previous layer with 256 filters of size 96*5*5. The third layer process 256 outputs from the previous layer with 384 filters of size 256*3*3. All those three conv layers are followed by ReLU activation and max pooling of 3*3 region.

4. There are two hidden fully connected layers which have 512 neurons and followed by ReLU and 0.5 drop rate [4].

5. The output layer has one neuron and followed by sigmoid function. Prediction is made by the probability output from the sigmoid function.

5.4. Training Model

Initialization: The weights of all layers are initialized with zero mean Gaussian random numbers with standard deviation of 0.01.

Image Augmentation: To further prevent them from overfitting, we do image augmentation each time before image passed in to our network. The augmentation parameters are as follow: rescale = 1/255, sheer_range = 0.2, zoom_range = 0.2, and horizontal_flip = True.

Network Training: We decide not to use 5-fold cross-validation since in practice, people prefer to avoid cross-validation since it's computational expensive. [7] So we choose fold 0 as our validation set and fold 1 to fold 4 as the training set. Training is performed using adam (an SGD optimization algorithm) having a batch size of 50.

5.5. Experiment

Our method is implemented using Keras with Tensorflow backend. Our model is trained by 20 epochs. The final accuracy rate we get is 80.08%. The following graph shows training history of our model.

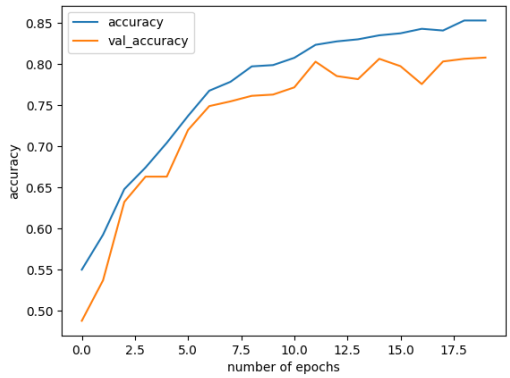

Figure 2 Accuracy

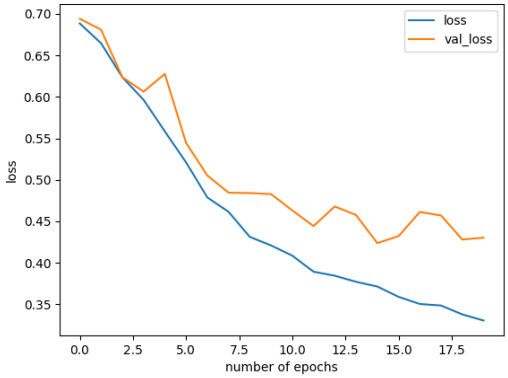

Figure 3 Loss

It can be seen from the above graph that the model start overfitting occur at epoch > 12.

5.6 Android Platform Implementation

Our Android implementation uses Client-Server Architecture in order to demonstrate the trained model. TCP protocol is used to communicate between the client and server. The server loads trained Keras model, receives the package from the client, converting the package to image, predict the image and send the result back to client. In client side, it takes picture, compresses the image and converts image to package. Afterward, it receives the result from the server then display it to the screen.

6. Discussion

In this project, we choose the data set that consists of images from real-world instead of constrained images taken in a lab. This setting can better reflect the problem of classifying social network images. We get relatively good result from our experiment. However, there are still some ways may improve the accuracy of the model. Increase the input image size may be the most apparent method to improve the accuracy. The input size proposed in [2] is 227*227. We use 64*64 as our input since limited computation resource. The other way to improve our model is tuning the model parameters. Different number of hidden layers and the number of neurons in each layer can be experimented on. From the loss function diagram, we can see that our model gets overfitted after epoch 13. Therefore, some other techniques may be used to further prevent the model from overfitting.

The android app serves as client to send images to the server (hosted on PC) then the server returns the prediction result to the app. A possible better way to implement the client is putting the trained model on the device and using android Tensorflow API to make prediction.

6. Conclusions and Future Work

By doing this project, we learn the architecture of CNN and how to implement CNN using Keras. By training the model using less constrained data set, 80% of accuracy is achieved. For further improvement, as mentioned in section 6, we can increase the input image size, fine tuning the parameters, and prevent the model from overfitting. The random erasing data augmentation described in [8] can be tried to prevent overfitting.

Reference

[1] https://en.wikipedia.org/wiki/Convolutional_neural_network
[2] *Gil Levi and Tal Hassner, Age and Gender Classification Using Convolutional Neural Networks, IEEE Workshop on Analysis and Modeling of Faces and Gestures (AMFG), at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, June 2015*
[3] E. Makinen and R. Raisamo. Evaluation of gender classification methods with automatically detected and aligned faces. Trans. Pattern Anal. Mach. Intell., 30(3):541–547, 2008.
[4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research 15 (2014) 1929-1958.
[5] https://www.openu.ac.il/home/hassner/Adience/data.html
[6] http://cs231n.github.io/convolutional-networks/#norm
[7] http://cs231n.github.io/classification/#val
[8] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random Erasing Data Augmentation," in arXiv:1708.04896, 2017.