ARG (Age Race Gender) Detection Using Transfer learning based on FaceNet Pretrained Model

Abstract

Recently, by the advent of deep networks, detecting proxy features from input face images has attracted much attention; however, the promising results show to lack sufficient accuracy due to the elaborated network architecture and complexity of time regarding the weight sub-optimal solution. Meanwhile, in this project, we explore the problem of predicting three attributes related to human face (Age, race and gender) using pre-trained FaceNet model using transfer learning. Leveraging pre-trained model, it is rather easy to get satisfying performance with deep learning solutions in Tensorflow framework. We achieved gender, race and age recognition accuracy of 93%, 84%, and 73%, respectively. Finally, we developed an online prediction module that automatically detects a face in an image using YOLO and passes the image to the network to generate final predictions, real-time. Tensorboard histogram is rendered to showcase the results and compare the outcomes.

1 Introduction

Face recognition and classification have emerged as a promising yet challenging topic in deep learning area [1]. There are many prior researches on gender recognition like [4]. Yet, ethnicity and age recognition are not as popular and considerably more challenging. The other challenge is that labeling an exact age number to an image is difficult and results in many mismatches between the ground-truth and predicted labels. While we eventually built a multi-task learning model, there is insignificant improvement for gender and race recognition and some improvement (proved by t-test) for age recognition compared to baseline model. In addition, we employed binning classification to deduce the residual in network. FaceNet is a model that directly captures a mapping from face images to an euclidean space on UTKFace dataset [2]. In other words, it is a CNN-based model that learns its representation directly from the pixels of the face. Inspired by recent advances in transfer learning for face recognition, we explore the problem of building a deep learning model to predict ARG of a face image.

Based on the steps and requirement mentioned above, we split the whole project into two sections. The first part would be deploying a network that is initialized by FaceNet model, in order to learn how to map ARGs labels to each face, and the second part is to implement a real-time platform which finds the regions that would potentially contain faces in a live video and then label ARGs to that region of interest.

2 Main Objectives

Our source codes are developed using TensorFlow framework and available online in ***Github*** repository. Our objectives can be summarized as:

1. Classification of face images in three categories namely, Gender: male, female, Race: White, Black, Asian, Indian, Other (Hispanic, Latino, Middle Eastern), Age: (1,5), (6,12), (13,19), (20,26), (27,34), (35,42), (43,49), (50,61), 62+.
2. Leveraging pre-trained model in the context of transfer learning to enhance the accuracy.
3. Using a multi-task approach to generate the results of classification all at once.
4. Online prediction of age, race, and gender using a webcam.

3 Preliminaries

3.1 Pretrained network (FaceNet)

As we mentioned earlier above, FaceNet is a CNN-based model for face recognition, but the crucial difference with other face recognition architectures is that it uses a deep convolutional network trained to directly optimize the embedding itself, rather than an intermediate bottleneck layer. It is based on inception-resnet-v1 to extract features. On the widely used Labeled Faces in the Wild (LFW) dataset, the system achieves a new record accuracy of 99.63%. On YouTube Faces DB it achieves 95.12%. FaceNet is ideal for our project as there are pre-trained models on CASIA-WebFace and MS-Celeb-1M faces.

3.2 Transfer Learning using FaceNet

In [5], the author used the pre-trained FaceNet and setup a multi-task learning model. It first detects and align faces in the picture and then uses a deep CNN to estimate age and gender. IMDB or Wiki [6] dataset is employed for training.

3.3 YOLO

YOLO reframes object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities [3]. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection. First, YOLO is extremely fast. Second, YOLO reasons globally about the image when making predictions.Third, YOLO learns generalizable representations of objects.

4 Methodology

4.1 Data-set preparation

The images dataset with gender, race, and age labels which was used in this project was UTKFace. It contains 23,708 faces with even distribution of gender, but uneven distribution of race and age. The files contain a single face in an image with its corresponding labels. In addition, we divided the dataset into these ratios: 80% training data, 10% validation, 10% for testing.

4.1.1 Data-set Augmentation

The dataset initially given has an uneven distribution relative to the classes of race and age. This has potential to adversely affect learning and skew the model to favor the classes with more data points. To alleviate this issue, we performed data augmentation on the 4 other categories of race (excluding class: white), and 20 of the categories of age for un-binned age prediction. Binned age prediction did not require data augmentation as the binning classes were chosen to fit a uniform distribution. The augmentation was done by choosing 1 of 4 random adjustments: rotating the image, flipping vertically, adding Gaussian noise, or adding Gaussian noise and flipping vertically. After augmenting the dataset, it has approximately 40,000 images.

4.1.2 Image alignment

The alignment module uses dlib to detect a face and create landmarks; the landmarks are then adjusted to create uniformity in the faces that are passed in. Relatively however the dataset that was used (UTKFace) already had most of the faces cropped and adjusted to have the facial features centred to the picture. However in the case of using a raw image or during a live session, the images captured needed to be cropped and aligned for accurate predictions.

4.1.3 Extracting labels

Execution of the network was done through TensorFlow-gpu 1.12; a base code from [7] was acquired and modified. Initially for label extraction each of the images had embedded the label within the name of the image. Age, Race, and Gender were extracted from the image name and formulated into TensorFlow records along with the image following data augmentation. TensorFlow records were used for seamless data pipeline. Modifications to the TensorFlow graph were made by adding the age prediction to the base code.

4.2 Overall structure

For this project we chose to use the weights from FaceNet trained on VGGFace2 dataset, as a baseline for transfer learning. Following this we use a multitask-learning approach, The network has some layers of shared variables (facenet in our project) to identify common features, then proceeds to 3 separate auxiliary layers for age, race, and gender detection respectively. All three output layers use softmax cross entropy as a loss function. The shared layers' weights are optimized by minimizing the weighted sum of all three losses. The auxiliary layers in our network have 128 neurons with linear activation.

4.3 Multi-task learning and variations of the model

We used the network in [7] which was previously designed to predict race and gender, and modified the structure in order to add another task (age detection) to the network. Our data set had an age range of 0 to 116. Since the network was designed for classification, we needed to define our age classes. We applied 2 different variations. First we considered 1 class for each age up to the age of 62 and 1 class for the age of

63 and older. We trained the network for 150 epochs ($\approx$ 7 hours) and generated the results (which will be explained in more detail in the result section). In this process we assumed a learning rate close to the one chosen for gender. It is true that the learning rate must be modified as is obvious from the results, because race and gender converged much faster than age. But this modification required long hours of training and retraining which was impossible due to our limited time frame. Also it was not guaranteed that learning rate modification would make the results rise up to our standards. That is because the difference between the number of classes in each task of the multi-task model also played a role in decreasing the quality of performance. So, we used a binning method for age classification. That is, we considered age classes according to what was previously mentioned in objective section. Having an overall number of 9 classes for age, we retrained the network. The second approach showed much better performance after one third of the number of training epochs relative to the last approach, i.e. 50 epochs (≈ 2.5 hours).

Training of both variations of the network were done on an RTX 2080Ti. During training, the variables for tracking training progress were recorded using TF.summary and saved which allowed for visualizing training progress through Tensorboard. The final results of the trained networks can be seen in the results section. The charts were produced by Tensorboard after training was complete. The trained model was later used for the purpose of online prediction.

5 Results

The final results of our project is shown here. First the training results are presented in Sec 5.1. Then, online prediction outputs are presented for some example cases.

5.1 Training model

In this section, we will illustrate the training result of age, gender and race accuracy in addition to the total loss of the network over 240k iterations. According to the multi-task learning, in the first run, we set the learning rate as γ_0 for the ad-joint output layers of age, $1/10 \times \gamma_0$ for race and gender output layers and $1/100 \times \gamma_0$ for the FaceNet back-end layer. The training was carried out in two phases. In the first round we extracted the result, then we changed the learning rate and generated the second round.

The comparison of the utilized method are depicted in Table 1.

Table 1: Test Result

	Exact Age Prediction	Binning
Age	0.014	0.73
Race	0.84	0.82
Gender	0.93	0.92

5.2 Demo Setup

For this part, we utilize a webcam and Opencv to access the images online. YOLO is utilized to recognize the faces and draw a box around each face. Then the picture inside the box will be alignment and saved as an input image. During the prediction time ($\approx$ 20 secs) several images are captured and stored as a nominative of the person. Each image is passed to the tensorflow session through distinguished threads. By using the concept of asynchronous parallelism, as each session obtained its response, it will recorded in a pool of nominative response. The final result will shown up as an image tag, at the top of the figure. This result will evolve over time as more information is observed by different threads, while considering the previous observations of that specific person. If the person moves out of the frame, the output pool would

(a) Age training accuracy

(b) Gender training accuracy

(c) Race training accuracy

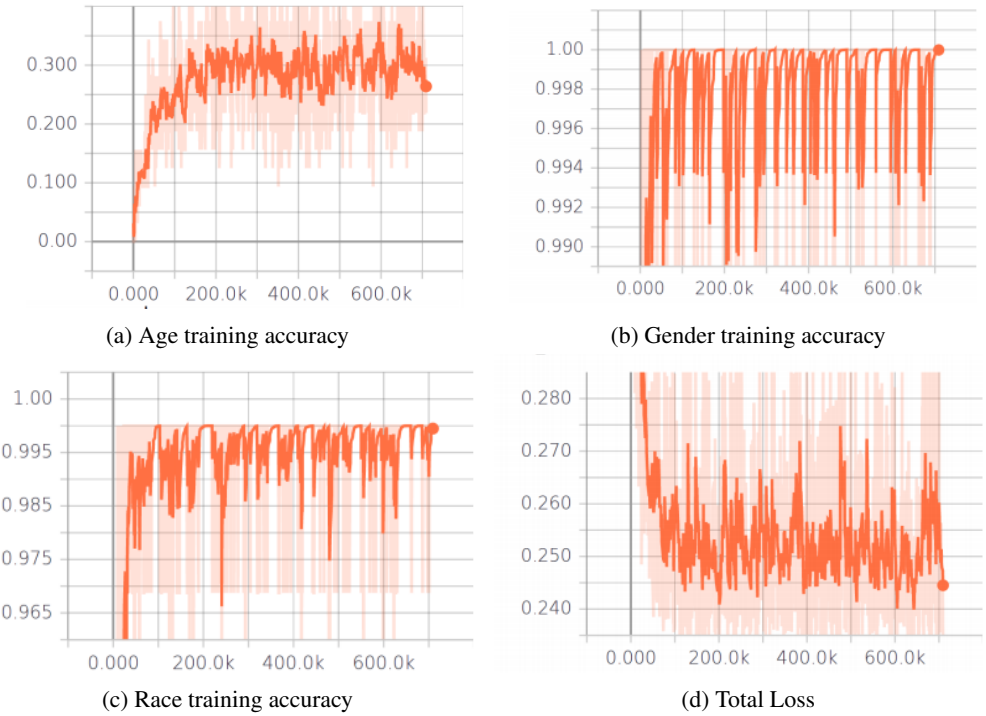
(d) Total Loss

Figure 1: Training accuracy with binned age classes

be reinitialized. The online prediction of 6 people is shown in Fig. 3. The results admit efficiency and accuracy of the proposed method. Table 1 presents the training results for the case of ARG prediction using a binning method and 9 classes for age. The figures on the right, show the case of full classes. The network trained on 9 classes shows a much better performance in training and evaluation. This result shows one of the disadvantages of conventional multi-task learning methods. Larger variations in the number of classes for each task will deteriorate the performance in multitask model.

Contribution

The following are the contributes of each member to make this project done in the limited time.



(a) Age training accuracy

(b) Gender training accuracy

(c) Race training accuracy

(d) Total Loss

Figure 2: Training accuracy with 63 individual age classes

Acknowledgments

The authors would like to thank the constructive help of the CMPT726 TA group in Fall 2019 namely Akash Abdu Jyothi, Lei Chen, Ruizhi Deng, Sha Hu, and Mengyao Zhai for the delightful insights during the projects. Finally, none of this would have been accomplished if not for the competitive and inspirational environment created by Prof. Greg Mori and fellow classmates.

References

[1] Schroff, Florian & Kaleniochenko, Dmitry & Philbin, James *aceNet: A Unified Embedding for Face Recognition and Clustering.*. CVPR, 2015.

[2] Yang Song and Zhifei Zhang *UTKFace: Large Scale Face Dataset* . `https://susanqq.github.io/UTKFace/`

[3] Redmon, Joseph & Farhadi, Ali *YOLO9000: Better, Faster, Stronger*. ICCV, 2017.

[4] Dong, Yuan & Liu, Yinan & Lian, Shiguo *Automatic Age Estimation Based on Deep Learning Algorithm*. Neurocomputing, 2015.

[5] Boyuan Jiang *Face age and gender estimate using TensorFlow* . `https://github.com/BoyuanJiang/Age-Gender-Estimate-TF`

[6] Rasmus Rothe, Radu Timofte, Luc Van Gool *IMDB-WIKI – 500k+ face images with age and gender labels* . `https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/`

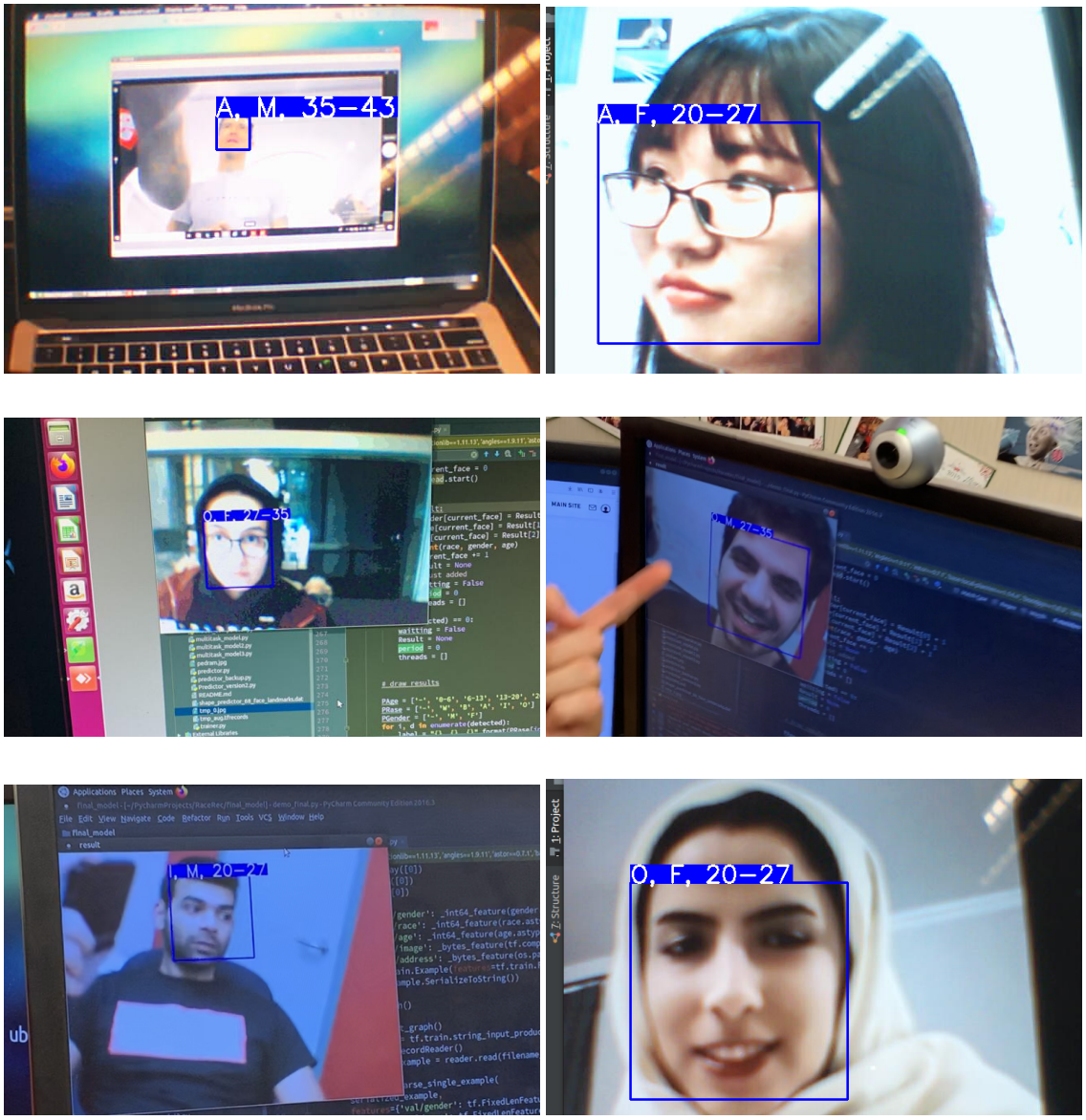


Figure 3: Online prediction using demo

[7] Zhu Yan *Tensorflow implementation; pre-trained FaceNet,* . `https://github.com/zZyan/race_gender_recognition`