Age, Gender and Race Prediction

Kor, nem és rassz predikció

Team Artificial Incoherence
Bálint Gergő, Medgyes Csaba, Virsinger Dominika

December 11, 2022

Abstract

The aim of this project was to train a model for predicting age, gender and race of a face image. We implemented three different models for this task, and compared the results by evaluating the predictions. To improve our model, we also used an MTCNN model for face detection, and made a pipeline so that we first get the face from a picture and then we predict the target variables from only the face. Hyperparameter optimization was also part of our work. For illustration of our model we then created a mobile application. The user can either select a photo from gallery or take a photo with the camera and then our model makes the prediction.

Kivonat

Projektünk célja egy jól teljesítő modell tanítása volt kor, nem és rassz predikciójára arcképeken. Három különböző modellt implementáltunk és összehasonlítottuk a teljesítményeiket a predikciók kiértékelését használva. A modell további fejlesztése érdekében egy MTCNN modellt is felhasználtunk arc detekcióra. Ennek működését úgy ágyaztuk be a projektbe, hogy elsőként az MTCNN modellel kinyertük az arcot a képekről, majd a másik modellnek csupán az arcot bemenetként megadva prediktáltuk a kort, nemet és rasszt. A modell tanításnál hiperparaméter optimalizációt is alkalmaztunk. A predikciós folyamat szemléltetésére létrehoztunk egy mobil applikációt. A felhasználó kiválaszthat egy képet a galériából, vagy akár készíthet is egyet a kamera segítségével, és ezután a modellünk prediktálja a három célváltozó értékét.

1 Introduction

Deep Learning is gaining popularity as it has wide applications in many fields such as computer vision, speech recognition, natural language processing, text processing, etc. One of the biggest challenges of Deep Learning is that large-scale training datasets are very difficult to build. In this project we used a publicly available dataset from Kaggle, and with the help of face detection we want to predict age, gender and race, which is a highly researched task in Deep Learning. There are many influential papers about this topic, according to our knowledge the first big breakthrough was in 2013 in (1). Following this the year 2014 and 2016 also produced highly influential papers in (2), (3), (4). With the increased computational power in the later years there were many work which would deserve a reference, although the work in (5), (6) were quite exceptional, gaining a lot of references in a short amount of time.

2 The Network of this Paper

We created three different neural network models. Our first attempt was to make a CNN based model, that is defined by us. This model has multiple Conv2D, MaxPooling2D, Dense and Dropout layers and three output layers (for the three target variables). The second and the third models are defined by using transfer learning and training some added layers. The second model uses the pretrained Xception model (7) with ImageNet weights (we call this the imagenet model). We added some Dense, Dropout layers and then the output layers. It is important to mention, that we only trained the last few layers which we added (and are not part of the Xception model). The third model uses the pretrained InceptionV3 model (8), and has the same few layers after the given model as the second model (we call this the inceptionv3 model). At this model, we did train a part of the given InceptionV3 model. The first 52 layers were not trained, but all of the layers past that and also the ones that we added were trained.

At the transfer learning models we simply used a GlobalAveragePooling2D layer for the output of the pretrained model. Then we made a branching in order to get three different outputs. These branches have a Dense layer, a Dropout layer

and then the final Dense output layer. At the age and gender branches, the output layers have one neuron, while at the race branch it has five neurons. The activations of the outputs are the following: for age it is relu, for gender it is sigmoid, and for race it is softmax.

3 Execution

3.1 Data processing

We imported the UTKface dataset including more than 20000 images with labels from Kaggle. First we had to do some data cleaning, for example the removal of data with missing labels. After we cleaned our dataset, we made an exploratory data analysis. This includes some statistics about the age variable, some plots with the distribution of number of people per age, number of people per race and the number of people per gender. We split the data into train and test sets, with the ratio $80\% - 20\%$. The picture itself (with the RGB values of the pixels) is our predictor variable, and the age, gender and race are the target variables. Age is a continuous value, gender is a binary variable and race is a multiclass variable.

It is a usual method, to normalize the data, so we did this as well. We had to face an important decision, because our train data was too big to fit in the memory. We decided to solve this problem by cutting the train data into four parts, and deal with them separately.

3.2 Modeling

When fitting the models, we used a 0.1 ratio of the train data to use as validation data. The aim of the training was to minimize the validation loss. This loss combines the mean squared error of the age attribute, the binary crossentropy of the gender attribute and the categorical crossentropy of the race attribute. We also defined some callbacks like the reduction of learning rate when the metric stops improving, early stopping when validation loss does not get lower for 5 epochs, and lastly we only saved the best model from training. For each model we used the 'adam' optimizer (9).

As mentioned before, our train data was split into four parts. We did the training in four parts, we defined the model, used the first part of the training data to train the model, then saved the model. Then we could remove the first part of the training data from memory, so we had enough space to load the second training data. So this way, we loaded the new data, and trained the model further. And the same way with the third and fourth part of the training data. Later we realized, that maybe this would have been nicer with generators, but this method can be quite useful if we get new data every time, and want to improve an existing model.

After training these models for predicting the three target variables, we also implemented an MTCNN model (10). This is used for face detection. Our aim is to predict the given values not just on the test set (with pictures of faces) but also on any other picture. It is possible that an image does not contain only a face, but the whole body, or even a person with something in the background. In this case our models would not predict well, because it was taught on a dataset including images of only faces. So we used this MTCNN model to first detect the face from the given picture, and then we give only the detected face to our model to predict age, gender and race.

Regarding the race attribute, it is an unbalanced classification problem, because the different races do not have the same amount of training data. So in order to deal with this, we also tried creating a weighted model with only one output (race). This gives a balanced weight for each class, so the model is trained with balanced data. Another attempt for this was the SMOTE method (11), that we implemented.

After making the models with parameters we thought were good, we also did a hyperparameter optimization for all the three types of models.

3.3 Evaluation of the Best Model

As there are three outputs, and all of them have different types, we have to evaluate them separately. Age is a continuous value, so we used mean squared error and root mean squared error to evaluate it. Gender is a binary value, here we used accuracy, precision, recall, f1 score, confusion matrix, AUC, and ROC curve. Race is a categorical variable, that can be evaluated with multiclass methods (12). We used accuracy and macro f1 score.

After evaluating our models, we were satisfied with the results. Our best performing model was the InceptionV3 based model with the results in Table 1.

Table 1: Results of InceptionV3 based model

Variable	Accuracy	F1	Precision	Recall	AUC	RMSE
Age	-	-	-	-	-	8.70
Gender	0.9239	0.9191	0.9272	0.9112	0.9783	-
Race	0.8307	0.7110	-	-	-	-

After hyperparameter optimization, all three models were trained with optimal parameters, and evaluating those, we saw very similar results as our first models. Some values got a bit better, others a bit worse. This might be because we minimized the validation loss, that is calculated from the three variable losses, and this does not directly mean that our evaluation values improve. Or another reason could be that we initialized our first models with almost optimal parameters. So altogether our best model remained the InceptionV3 model with the original parameters.

4 Testing and Application

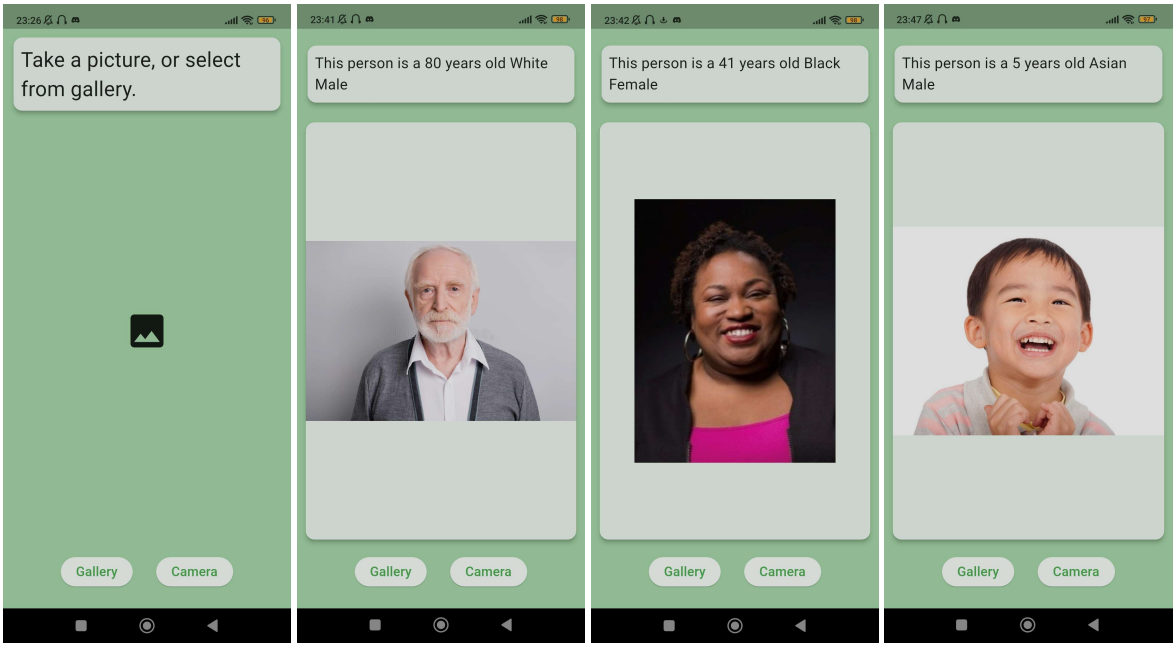

Figure 1: Screenshots from the mobile app

We created a simple server-client architecture application to test and verify the model. The server side contains a Flask web application that communicates with rest api. It gives two endpoints to the user, one to test the availability of the api and one to upload an image, which is answered with the prediction of the model in json format. The resulting web application is rendered as a Docker image so that it can be run as a container. We chose Amazon Web Services as host, as the image is relatively large due to tensorflow and other python dependencies. The service is available on port 5000 which is forwarded to standard http port 80. The incoming image is passed through a pipeline on the server, where we first delineate the face in the photo using an MTCNN model, and then resize the result to get our own model, which predicts age, race, and gender. Of course this has its time cost, but we have found that we still get a response within a reasonable time.

To access the service, we created a mobile app using the Flutter framework for the client side. Flutter is a good choice because it's cross-platform technology and it's quick to create small apps like this one. The user can take a photo or select an image from the gallery and send it to the server to get our model's prediction of the person in the image.

5 Summary

In this project we explored the problem of predicting age, race, gender from face images. We tried several models for this task, made hyperparameter optimization and eventually got an InceptionV3 based model with acceptable results. We made a mobile application, that illustrates the predictions on any chosen or taken photograph.

References

[1] Xiaolong Wang, Vincent Ly, Guoyu Lu, Chandra Kambhamettu. *Can we minimize the influence due to gender and race in age estimation?* 2013 12th International Conference on Machine Learning and Applications.

[2] Guodong Gou, Guowang Mu. *A framework for joint estimation of age, gender and ethnicity on a large database.* Image and Vision Computing. 2014.

[3] Hachim El Khiyari, Harry Wechsler. *Face verification subject to varying (age, ethnicity, and gender) demographics using deep learning.* Journal of Biometrics and Biostatistics. 2016.

[4] Neeru Narang, Thirimachos Bourlai. *Gender and ethnicity classification using deep learning in heterogeneous face recognition.* 2016 International Conference on biometrics (ICB). 2016.

[5] Abhijit Das, Antitza Dantcheva, Francois Bremond. *Mitigating bias in gender, age and ethnicity classification: a multi-task convolution neural network approach.* Proceedings of the european conference on computer vision (eccv) workshops. 2018.

[6] Anoop Krishnan, Ali Almadan, Ajita Rattani. *Understanding fairness of gender classification algorithms across gender-race groups.* 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). 2020.

[7] Francois Chollet. *Xception: Deep Learning with Depthwise Separable Convolutions.* April 2017.

[8] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna. *Rethinking the Inception Architecture for Computer Vision.* December 2015.

[9] Diederik P. Kingma, Jimmy Lei Ba. *Adam: A Method For Stochastic Optimization.* 2015.

[10] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, Yu Qiao. *Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks.* April 2016.

[11] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer. *SMOTE: Synthetic Minority Over-sampling Technique.* Journal of Artificial Intelligence Research 16, 321–357. 2002.

[12] Margherita Grandini, Enrico Bagli, Giorgio Visani. *Metrics for Multi-Class Classification: An Overview.* Department of Computer Science, University of Bologna, August 2020.