Recognition of facial features on FairFace Dataset

Arcjellemzők felismerése a FairFace adathalmazon

Rise of AI group, Members: Tugyi Beatrix (T63K63), El-Ali Maya (BHI5LF), Simkó Máté (O3BMRX)

Abstract— Face recognition is a computer vision task with a long research history and a wide variety of applications. As this machine learning task has many real-world use-cases that affect human beings it is imperative to train AI networks for the task on unbiased datasets. This is why the FairFace dataset was chosen for our face feature classification work. Our goal was to classify images into gender, race and age classes. In our research we compared two existing image classification networks, ResNet and VGG, using transfer learning. In addition to implementing different classification methods for tackling the multi-label dataset, a demo app has also been created that showcases the models' results. The app uses the webcam to recognise faces and classify them.

Kivonat— Az arcfelismerés egy olyan gépi látás feladat, amely hosszú kutatási múlttal és sokféle alkalmazási területtel rendelkezik. Mivel ennek a gépi tanulási feladatnak számos, embereket érintő valós felhasználási esete van, elengedhetetlen, hogy a feladatra a mesterséges intelligencia-hálózatokat igazságos eloszlású adathalmazokon tanítsuk. Ezért választottuk a FairFace adathalmazt az arcjellemzők osztályozására irányuló munkánkhoz. A célunk az volt, hogy a képeket nem, faj és életkor szerinti osztályokba soroljuk. Kutatásunkban két létező képosztályozó hálózatot, a ResNet-et és VGG-t, hasonlítottunk össze transzfer tanulás segítségével. A többcímkés adathalmaz kezelésére szolgáló különböző osztályozási módszerek megvalósítása mellett egy demóalkalmazást is készítettünk, amely bemutatja a modellek eredményeit. Az alkalmazás a webkamera segítségével felismeri az arcokat, majd osztályozza őket.

1 Introduction

One of the most widespread applications of Deep Learning is image processing, including image recognition. Many different models and techniques have been created over the years to solve this, and one can choose from a very wide range of methods to solve these tasks. This choice depends on the dataset, the type of task, the time, and other factors.

Our dataset is special because the model does not have to recognize completely different figures, because all of the images are faces. Our task is to teach the network to recognize people's age, gender and race based on the characteristics of these faces. To prepare this, we experimented with several methods.

2 Previous Research

The biggest benchmark of image classification is the ImageNet[7]. This is a large dataset with various classes. The first trials on this test dataset were with computer vison methods, but when convolutional networks appeared they achieved much better results. Throughout the years many new architectures were created, like AlexNet, ResNet, VGG, GoogLeNet, DenseNet. The current state of the art model is better then human recognition and does not leave much room for improvements.

On the FairFace dataset many models were used with different results. The accuracies achieved on this dataset are lower than ImageNet, as it is more difficult to group images only from faces, based on the given features. The results of the best models can be found in the dataset article[2], the models we implemented approach these values.

3 System Architecture

All models were created in Python using the Pytorch[5] class library. The training was carried out on the GPU of Google Collaboratory.

For image recognition, it is always worth using transfer learning[7], based on a pre-trained model. For this, we chose the ResNet and VGG models trained on ImageNet. We used the original weights of the base models because it was necessary for lower the training time. After these base models we add some linear layers and a Sigmoid or Softmax nonlinear function before the output.

4 IMPLEMENTATIONS

We created 6 different notebooks, one for each label- model pair and performed the following steps in them.

4.1. Collection and preparation of data

We used a famous dataset, called FairFace. It is a huge, race-balanced dataset, with 97698 images labelled with gender, age, and race.

First, we loaded these images and labels from its original repository, which contained the train and validation images separately. We shuffled the train images and split into to train and test sets. This way the ratio between the datasets: train: 78%, validation 12%, and test 10%. The dataset contains a large amount of data, so we don't need more test and validation data then that.

After loading, all images go through some transformation functions for pre-processing and augmentation. On the train dataset we crop the images into the size 245x245, perform a flip with 0.3 propapility and do a random rotation up to 20 degrees. On every dataset we transform the images to tensors and normalize them with mean: [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225]. These are the normalization numbers that had been used on the pretrained model. And in the end, we put all data on the GPU.

We pre-processed the labels too. We used one-hot encoding for labels with more then two classes (age, race) and categorical encoding for labels with two class (gender).
We provided a base class for the images and labels.

4.2. Designing a training and evaluation function

For training we use two different loss function, Binary Cross Entropy for gender and Cross Entropy for the race and age. We use a built-in scheduler, which monitors the performance and lowers the learning rate if it is needed. For better training we use weight decay and gradient clipping. These parameters can be passed to the training functions.

We save the validation accuracy and losses after each epoch and print them out.

For evaluation we defined a function, that uses the same losses as the train function. This function is only for examine the results, we do not change the gradients and the weights of the model here.

4.3. Hyperparameter optimization

In order to achieve the best results, we optimalized the parameters of the model and training functions. We tried out more combination and then designed an optimasation function, that run 3 epochs with all the given parameter combinations. The parameretrs were: The optimalization function (SGD[8] vs ADAM), the Learning rate (0.005 vs 0.001), the scale of the Weight decay (0.0005, 0,0001), the scale of the Gradient Clip (0.1, 0.2) the size of the batch (32, 16) and the size of the last hidden layer (128, 256).

The final best parameters can be seen in the notebooks.

4.4. Training

We trained all models for 10 epochs. The training time was 13 minute/epoch with the ResNet base and 22 minute/epoch with the VGG base. The models learned the most in the first few epochs, after that their progress declined, because they reach their maximum performance.

At the end of the training, we can see the visualizations of the training curves and the accuracy scores, and we can save the trained models for later useage.

The first diagram shows the changes in the validation and training losses, trained on the age labels and the second diagram shows the changes in the validational accuracy on the same task. The other diagrams can be seen in the notebooks.

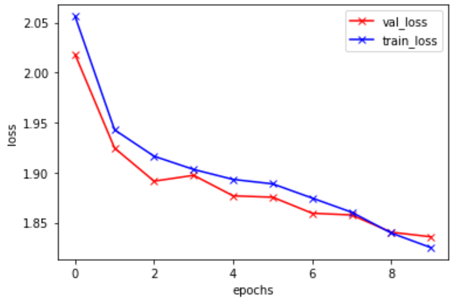


Figure 1 The loss scores in each epoch

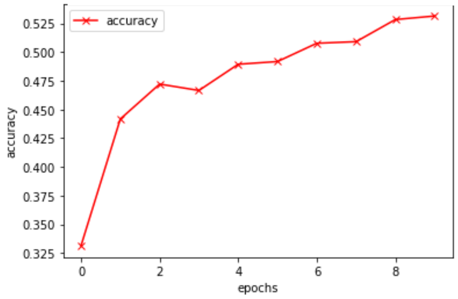
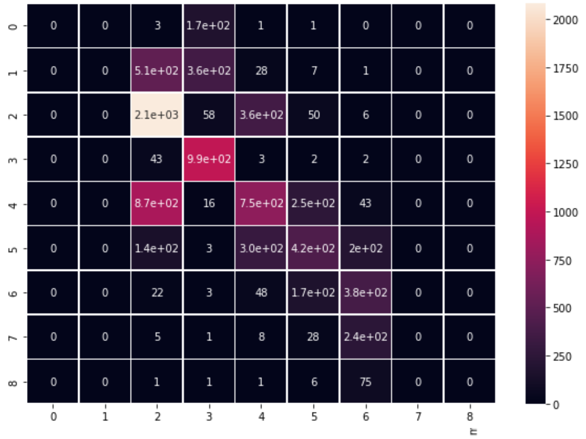


Figure 2The accuracy scores in each epoch

4.5. Evaluation method and results

We can evaluate the model on the unseen images of the test set and get a final performance score. After that we extract the predictions from the trained model and compare them with the true labels. We create a Confusion matrix with these values. As an example, this is the matrix obtained from the Age labels. The other matrixes can be seen in the notebooks.



	0	1	2	3	4	5	6	7	8
0	0	0	3	1.7e+02	1	1	0	0	0
1	0	0	5.1e+02	3.6e+02	28	7	1	0	0
2	0	0	2.1e+03	58	3.6e+02	50	6	0	0
3	0	0	43	9.9e+02	3	2	2	0	0
4	0	0	8.7e+02	16	7.5e+02	2.5e+02	43	0	0
5	0	0	1.4e+02	3	3.0e+02	4.2e+02	2e+02	0	0
6	0	0	22	3	48	1.7e+02	3.8e+02	0	0
7	0	0	5	1	8	28	2.4e+02	0	0
8	0	0	1	1	1	6	75	0	0

Figure 3Confusion matrix of the age labels

It can be seen from the diagram that some categories are unfortunately completely ignored by the model. The reason for this is that the original data set is only race balanced, which is why the age labels appeared very unevenly in the data set. Here is a diagram of the age labels distribution.

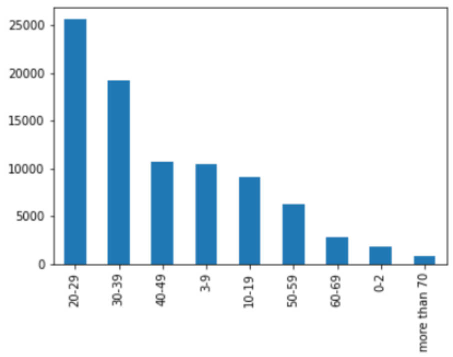


Figure 4 The distribution of the age labels

It can be seen how little training data was available from the omitted categories.
The confusion diagram also shows that, even if the model was incorrect, it didn't get too bad, it chose a close category.

5 Results

The final performance of the models can be seen in the following table.

Model	Accuracy
ResNet Gender	94 %
ResNet Age	53 %
ResNet Race	65 %
ResNet Multi	24 %
VGG Gender	93 %
VGG Age	
VGG Race	

6 CONCLUSIONS

All in all, it can be said that while image classification is a very well and successfully researched field, multi-target image classification still poses a significant challenge. We tackled this problem using two main approaches: by training 3 different models for each target and by training a universal model to predict all three targets.
When using three different models the ResNet based transfer learning proved much better, producing predictions with higher accuracy all well being less resource- and time-intensive. However, the VGG based models for race and gender did come close in performance.
Unfortunately training a multi-target model did not yield results. We tried two different arcitechtures: a model with a single dense network for predicting all outputs and a model with three different parallel dense networks for each output. Sadly, neither arcitechtures improved during the training, reinforcing that the best approach to deal with multi-target classification is to train different models for different tasks.

7 Application

Since the ResNet models showed better results, they were utilized in our demo app. This is a python application takes the stream input from the user's webcam and detects faces in it using the Haar cascade algorithm [12]. After this the image is converted and transformed into the appropriate format, after which the age, gender and race prediction models perform prediction on the image, the results of which appear on the screen.

8. Future Plans

Our further plans include improving the multi-target model by trying out different arcitechtures and loss functions. We want to try out more types of models too. We would like to implement a network capable of generating images on this dataset by transforming the used models. This could be achieved with two methods, Variational Autoencoder[9] and GANs[10][11].

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, *"Deep Residual Learning for Image Recognition"*, arXiv:1512.03385, *https://arxiv.org/pdf/1512.03385.pdf*

[2] Kimmo Kärkkäinen, Jungseock Joo, *"FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age"* arXiv:1908.04913, *https://arxiv.org/pdf/1908.04913.pdf*

[3] Karen Simonyan, Andrew Zisserman, *"VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION"*, arXiv:1409.1556, *https://arxiv.org/pdf/1409.1556.pdf*

[4] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", arXiv:1409.0575, *https://arxiv.org/pdf/1409.0575.pdf*

[5] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, Soumith Chintala, *"PyTorch: An Imperative Style, High-Performance Deep Learning Library"*, arXiv:1912.01703v1, *https://arxiv.org/pdf/1912.01703.pdf*

[6] Keiron O'Shea, Ryan Nash, "An Introduction to Convolutional Neural Networks", arXiv:1511.08458, *https://arxiv.org/pdf/1511.08458.pdf*

[7] Simon Kornblith, Jonathon Shlens, and Quoc V. Le Google Brain, *"Do Better ImageNet Models Transfer Better?"*, arXiv:1805.08974v3, *https://arxiv.org/pdf/1805.08974.pdf*

[8] Sebastian Ruder, "An overview of gradient descent optimization algorithms∗", arXiv:1609.04747v2, https://arxiv.org/pdf/1609.04747.pdf

[9] Adji B. Dieng, Yoon Kim, Alexander M. Rush, David M. Blei, *"Avoiding Latent Variable Collapse With Generative Skip Models"*, arXiv:1807.04863, https://arxiv.org/abs/1807.04863

[10] Meng Wang, Huafeng Li, Fang Li, *"Generative Adversarial Network based on Resnet for Conditional Image Restoration"*, arXiv:1707.04881, *https://arxiv.org/pdf/1707.04881v1.pdf*

[11] Tero Karras, Timo Aila, Samuli Laine Jaakko Lehtinen *"PROGRESSIVE GROWING OF GANS FOR IMPROVED QUALITY, STABILITY, AND VARIATION"* arXiv:1710.10196v3, *https://arxiv.org/pdf/1710.10196v3.pdf*

[12] Paul Viola, Mivheal Jones, "*Rapid Object Detection using a Boosted Cascade of Simple Features*" *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517, *https://ieeexplore.ieee.org/document/990517*