Face & Gesture Analysis 2021

Face Recognition Challenge

Oriol Guardia[1], Johannes S. Fischer[2]

1) oriol.guardia01@estudiant.upf.edu

2) johannessimon.fischer01@estudiant.upf.edu

1. Method

1.1 Face Detection

We reused our approach of the first lab, as it already delivered a good performance, this approach being the *Viola-Jones* face detection algorithm. In order to implement it, we made use of the built in function Cascade object detector from the Computer Vision package. With this face detector we got a face detection F1-score of 83.16% for the training dataset given to us and also 85.19% for the teacher's test dataset, in the first lab of the course, with a merge threshold of 14.

1.2 Face recognition

Previous Approach

Our first approach for the face recognition task was to use the recommended HOG features and a Support Vector Machine (SVM) to classify the images. For that we trained 80 different SVMs on the whole given training data set and tested them on another, manually downloaded, validation data set. The accuracy on this validation dataset was 43.88%, but the accuracy on the final test executed by the professors only yielded an accuracy of 15.05%. This difference could partially be explained by a different ratio of images with identities and images without identities. In our own validation dataset we had 16% images with identities and the rest without identities. In the given training dataset, the proportion of images with identities was 40%. Assuming that the test dataset of the course had similar proportions, we achieved the high accuracy in our own validation set probably because of the high amount of refused images without faces. Furthermore, our model was overfitting, as the high training accuracy compared to the low test accuracy indicates.

Final Approach

For our final face recognition system we decided to use a Deep Neural Network approach as feature extraction is already incorporated and it also is the state-of-the-art in face recognition [1]. We loosely followed the approach of Gruber et al. [2], who used the ResNet architecture, which is a convolutional neural network that encounters the vanishing gradient problem by introducing so-called "shortcut-connections" that skip layers [2]. With this network we applied transfer learning by replacing the last learnable and the classification layer with new layers suiting our classification problem. We furthermore freezed some first layers and did not change their weights during training. The learning rate was 0.0003. We tried several models to find out the batch size yielding the highest accuracy, which is described in detail in the results section. To prevent overfitting we also used image augmentation as it can be seen in table 1.

Augmentation	**Description**
RandXReflection	random reflection in the left-right direction
RandXTranslation	horizontal translation in range [-30, 30]
RandYTranslation	vertical translation in range [-30, 30]
RandXScale	horizontal scaling in range [0.9, 1.1]
RandYScale	vertical scaling in range [0.9, 1.1]

Table 1. Image Augmentation methods to prevent overfitting. For further information see Matlab Documentation.

Once the model is trained and faces are detected, we classify the data. As stated before in the previous labs, when we have more than one face, we detect only the two biggest ones and classify each one with our model to see whether it belongs to an identity. We also have to take into account the probabilities given for each detection. For this reason we implemented a threshold, in order to avoid as many false positives as possible.

1.3 Data

As the training data we obtained was not sufficient to do transfer learning we added several other images by using two approaches.

Firstly, we did a google image search for the celebrities and used the google chrome extension "*download all images*" to download all the resulting images. After iterating through all the images and obtaining all the faces, we had to manually sort pictures out where either the person displayed was not the target identity or any other object was falsely detected and declared as face. This approach led of course to very noisy datasets in the beginning, that's why we had to sort out a lot of pictures.

Secondly, we programmed a web scraper that automatically searches for the name of the identity on a stock image supplier webpage (www.gettyimages.de). Therefore we extracted all the names of the identities, iterated through them and adjusted the search link accordingly. For example, if we had the name *Channing Tatum* we adjusted the link so it looked like this:

`https://www.gettyimages.de/fotos/channing-tatum?family=editorial&numberofpeople=one&phrase=Channing%20tatum&sort=mostpopular#license`

We restricted the search to only show one person, as this results in cleaner data. Hence, we had less to clean afterwards.

These two approaches combined led to a total image dataset size of 9,563 images, as it can be seen in table 3. The dataset we obtained was split into training, test, and validation sets, with 70%, 15%, 15% respectively, for the analysis of the best parameters.

2. Results

First, we wanted to compare different parameters for our model and used the settings displayed in table 2.

Settings	**Value**
Layers to freeze	16
Epochs	4
Learning rate	3e-4

Table 2. The model settings we used for our model comparison.

As a first step we compared different backpropagation/solver algorithms using the ResNet-18 architecture. With *Adam*, the accuracy increased faster and also, in the end, was higher. Therefore, we used *Adam* for all following comparisons.

Next, we compared the accuracy which we got from the different datasets we obtained. Firstly, the manually downloaded images (data-man), secondly, the data we obtained with the web scraper (data-scr), and lastly, the merged data (data-mer). We were able to get the results displayed in table 3.

	data-scr	**data-man**	**data-mer**
Number of images	3,772	5,791	9,563
Validation accuracy	85%	84.33%	90.43%
Test accuracy	72%	84.67%	90.14%
F1-score	61.72%	77.84%	79.66%

Table 3. Comparison of the different datasets we obtained.

More data enhances the model drastically as it can be seen in table 4. With 3.7k images we only achieved a test accuracy of 72% whereas with the 9.5k images we got 90.14%. Hence, the following comparisons were all done with the whole dataset.

Another setting to take into account is the batch size, which refers to the amount of data included in each sub-epoch weight change. Increasing the batch size can increment the performance of the model but can also increase the computational cost, as well as leading to overfitting. Results are displayed in table 4.

	16	32	64	128
Validation accuracy	89.97%	91.96%	92.17%	92.10%
Test accuracy	86.53%	88.65%	89.93%	88.79%
F1-score	79.10%	79.15%	80.34%	81.21%

Table 4. Batch size comparison.

Moreover, we compared the ResNet-18 architecture with batch size 64, 4 epochs, and the ResNet-50 architecture with batch size 32 (limited through memory restrictions), 32 freezed layers, and 3 epochs, which yielded 88.79% and 83.01% test accuracy, respectively. Looking at both models we decided to use ResNet-18, although ResNet-50 is supposed to improve with either loss value and accuracy rate, but in our case we could only run it with a maximum batch size of 32 because of hardware limitations. Consequently, we trained the final model with solver Adam, the merged dataset (80% training, 15% validation, 5% test set) including the given training images, batch size 64, 16 freezed layers, and the ResNet-18 architecture. We achieved a test accuracy of 87.85% and with a threshold of 0.6 we obtained a final F1 Score of 86.31% (86.67% with identities, 90.56% without identities). A sample prediction is displayed in figure 1, model size is 41.82 Megabyte.

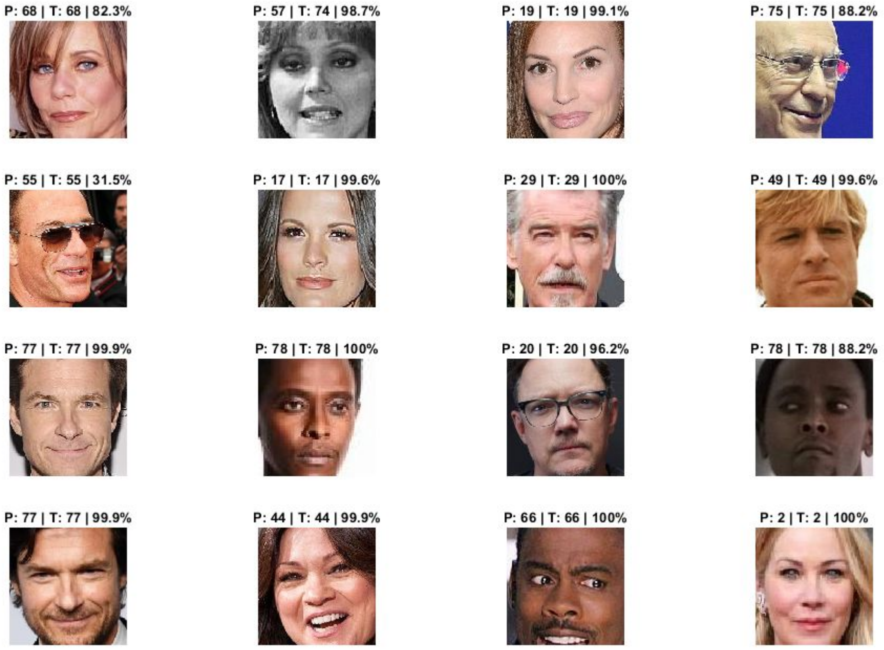

Figure 1. Predicted identity (P) and target identity (T),
as well as the prediction probabilities.

References

[1] Guo, G., & Zhang, N. (2019). A survey on deep learning based face recognition. *Computer vision and image understanding*, *189*, 102805.

[2] Gruber, I., Hlaváč, M., Železný, M., & Karpov, A. (2017, September). Facing face recognition with ResNet: Round one. In *International Conference on Interactive Collaborative Robotics* (pp. 67-74). Springer, Cham.