Identifying Facial Features using Deep Neural Networks

Domonkos Debreczeni, *Student, BME-VIK,* Benedek Juhász, *Student, BME-VIK,* and Nándor Szécsényi, *Student, BME-VIK*

***Abstract*—Facial feature recognition is a very popular research topic, although its roots go back many years. To this day facial recognition still has a lot of unsolved problems, some are quite easy for a human, some can be challenging too (for instance, accurately guessing someone's age). An old and universal difficulty, not just in facial recognition, but in all machine learning branches, is not enough data or imbalance in the dataset. An imbalanced training dataset can result in very biased models, incapable of recognizing general features, rendering them practically useless in a production environment. In our project we worked with a heavily imbalanced dataset (the FairFace dataset), and we propose a solution using a pretrained VGG-16 model and a custom loss function. The purpose of the latter is to counter the effects of the imbalanced training data. During our project, we experimented with a number of other models, configurations and transfer learning techniques resulting in little bit surprising, but very interesting conclusions. About our final model, which by far surpassed all the others, we can proudly say, that according to its evaluation output, most of its mistakes are very similar to human errors (such as taking a 20-29 year old for 30-39 year old). Furthermore, we didn't just stop after the final training, we even took the time and studied the final models latent space. During this we developed an Autoencoder model too and tried identifying the more important latent space features, both of them yielding very interesting results. We firmly believe that our experiments and studies are a useful contribution to the ever going deep learning communities work.**

***Index Terms*—Deep Learning, Facial classification, Latent Space.**

I. Introduction

OUR task during this semester was to make a Deep Neural Network that can label a picture of a face using 3 different categories: age, race and gender. For this, we used the FairFace dataset[1], which contains several good quality face pictures with corresponding labels. As an extra, we also decided to explore the latent space created by our networks using several methods, so we could better understand its strengths and weaknesses.

II. Dataset and preprocessing

AS recommended, we used the FairFace dataset[1], which comprises nearly 100 thousand labeled images. The labels are gender (male/female), age (9 unique values) and race (7 unique values). The dataset has two versions: one with images of size 224x224 and one with 442x442, we used the former as it needed no further cropping. During the EDA (Exploratory Data Analysis) we quickly realized that this is a heavily imbalanced dataset, for instance in the age label there were thousands of pictures labeled 20-29, but not even one thousand images with the 0-2 year old label. Despite than the imbalanced nature of the dataset, we found it very well prepared and easy-to-use. However, there's no data which doesn't need preprocessing. According to our experiences, we used and tried a number of image augmentations techniques including more basic ones, such as rotation, zoom and blurring that were already implemented in Keras, but for the more advanced ones we needed an external library. The imgaug[2] library has everything but the kitchen sink, containing dozens of augmenting functions. Of course we never applied too many augmenting functions, as every little detail of the face can carry a lot of information, so we only used two for the final training: JPEG compression and CoarseDropout. Lossy JPEG compression is very useful, as it forces the network to learn more abstract features and the latter is literally dropping out random pixels of the image. Both of these techniques proved themselves useful in the past and in our present project too.

III. Constructing the Network Architecture

FIRST of all, our task was to make a classification model structure that can assign the labels to the faces with a reasonable accuracy. From the beginning, we had already become aware that the key to achieving a good result is using Transfer Learning and correct preprocessing (for example see [3] and [4] for reference in the field of facial expression recognition), since the dataset's images are large and the classification task itself can be considered really hard even for the naked eye. We'd also like to note that despite the final architecture and training techniques, we were very open minded and tried a number of different configurations.

A. Using the pretrained VGG-19 structure

Our first experimental model was the VGG-19 architechture[5] as it is widely used for other image classification problems and through Keras we could access a pretrained (on the ImageNet dataset) model. After the main structure, we included a GlobalAveragePooling and several Dense layers before producing the output with a Fully Connected classifier layer. Because this was our first attempt, we only chose one category at a time, to make us better understand the dataset's characteristics and difficulty and to get a feel for the complexity we need for better results. Some of the hyperparameters we tried out and their selected results can be found in Table I below.

From these results it was clear that we need to give the network more capacity to make its performance better and it

TABLE I
TABLE SHOWING THE RESULTS OF OUR EXPERIMENTS

Was VGG-19 trained	Dense layer count (100 neurons each)	Resulting Accuracy
No	1	Gender: 70.4%
No	2	Gender: 72.8%
No	1	Age: 35.7%
Yes	1	Age: 43.3%

seemed that the general ImageNet weights from the pretrained model also needed further training.

B. Using the pretrained VGG-Face model and Processing the Dataset

From the previous trials we concluded that we needed a network which was already pretrained on faces so its convolutional blocks have all the necessary features learned thus providing better results from the start. For this we used the VGG-Face network (which is a result from research done in [6] and [7]) as it was trained on only images of faces and it shares the same architecture with the original VGG-16 model. From this saved model we only used the convolutional blocks then added our own Dense layers and classification output. With this model structure we were able to achieve much better results, so we tried its performance on all 3 categories at the same time, using 3 classification outputs (one for each): for the age and race categories we used a Dense output layer with softmax activation and a custom weighted categorical-crossentropy loss function, while the gender category's output layer contained a sigmoid activation with a binary-crossentropy loss function.

As we mentioned earlier, the dataset was quite imbalanced (especially in the terms of age labels). To solve this, we used a custom loss function that implements Categorical Crossentropy with the use of class weights that gives more priority to low sample sized classes to counterbalance the others.

Quite to our surprise, we managed to find the best model configuration in the end by freezing all the convolutional layers and training only the dense layers. To read more about our other approaches, please see the next subsection. After finding the right approach we performed hyperparameter optimalization on the Dense layers we put after the VGG-Face's Convolutional blocks. The resulting network structure can be seen below in Table II.

With this network architecture we were able to get reasonably better results than with our first experiment. We evaluated our model's performance using several metrics (precision, recall, F1) including a confusion matrix for each category. These confusion matrices can be seen in Figures 1, 2 and 3, while the classification reports can be seen in Tables III, IV and V.

C. Other transfer learning approaches

In the previous subsection we mentioned that we stuck to our first approach, by training only the dense layers. We made this decision based on a number of different trials, which all yielded worse results. As instructed, we kept our open mindset and went through all the possible configurations. First of all, we tried different models: SqueezeNet[8], Resnet[9] and of course the original VGG-19, all pretrained on the ImageNet dataset. After trying these models, returning back to the VGG-Face, we continued with different training strategies, for example training the final 5 convolutional layers or all of them. Actually, initializing the entire network with pretrained

TABLE II
TABLE SHOWING THE RESULTING MODEL STRUCTURE AFTER HYPERPARAMETER OPTIMALIZATION

Layer name	Layer Parameters
Dense	640 neurons with Leaky-ReLU activation
Dropout	20% chance
Dense	896 neurons with ReLU activation
Dropout	30% chance
Dense	2176 neurons with Tanh activation
Dropout	40% chance
Dense	640 neurons with Leaky ReLU activation
Dropout	60% chance
Dense	1664 neurons with Leaky ReLU activation
Dropout	20% chance

TABLE III
TABLE SHOWING THE CLASSIFICATION REPORT OF THE AGE CATEGORY

Label name	Precision	Recall	F1-score	Support
0-2	0.59	0.65	0.62	371
10-19	0.40	0.40	0.40	1836
20-29	0.49	0.85	0.62	5220
3-9	0.79	0.48	0.60	2084
30-39	0.59	0.03	0.05	3827
40-49	0.35	0.15	0.21	2077
50-59	0.26	0.64	0.37	1232
60-69	0.37	0.23	0.28	553
more than 70	0.25	0.68	0.37	149

TABLE IV
TABLE SHOWING THE CLASSIFICATION REPORT OF THE RACE CATEGORY

Label name	Precision	Recall	F1-score	Support
Black	0.79	0.69	0.74	2373
East Asian	0.69	0.59	0.64	2499
Indian	0.59	0.52	0.55	2456
Latino Hispanic	0.39	0.35	0.37	2709
Middle Eastern	0.48	0.21	0.30	1816
Southeast Asian	0.44	0.60	0.51	2186
White	0.52	0.76	0.62	3310

TABLE V
TABLE SHOWING THE CLASSIFICATION REPORT OF THE GENDER CATEGORY

Label name	Precision	Recall	F1-score	Support
Female	0.86	0.83	0.84	8163
Male	0.85	0.88	0.86	9186

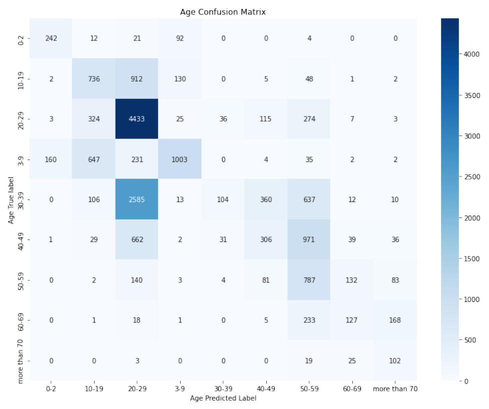

Fig. 1. Age Category Confusion Matrix

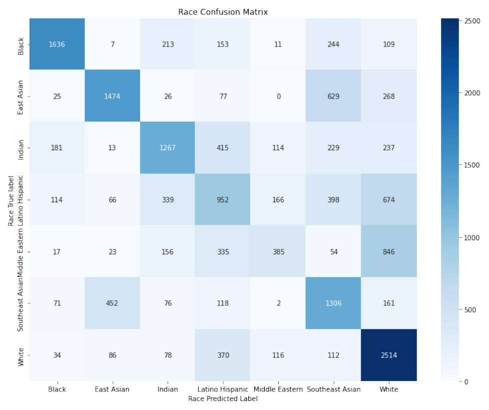

Fig. 2. Race Category Confusion Matrix

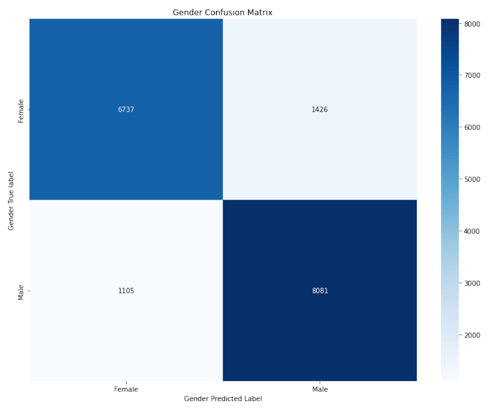

Fig. 3. Gender Category Confusion Matrix

weights (using it as a random state) and then handing it all of our data proved to be quite effective in the past, as it results is faster learning. The latter is due to the fact that the lower convolutional layers learn very simple and universal patterns. A detailed chart of some of the trials can be seen on VI: There were a number of other experiments, but the pattern can be seen clearly on these trials, they all yielded roughly similar numerical results.

IV. Exploring the Latent Space

TABLE VI
Transfer learning experiments

Trained layers	Additional methods	Accuracy age, race, gender
Last 7 conv	-	$37.4, 43.1, 78.9\%$
All	Dropout after each layer	$23.8, 20.2, 52.6\%$
All	BatchNorm after each layer	$22.5, 24.6, 51.7\%$
All	More image augmentations	$27.0, 31.5, 62.3\%$

After we have optimized and evaluated our model's performance, we wanted to know how exactly does it work, how does its Convolutional Layers learn and represent the facial features of an input sample. To visualize this we needed to analyze the output of the last Convolution (after a GlobalAveragePooling layer), which results in an embedding vector with given size. In this vector –presuming the model correctly learned some facial features– if we compare the same sequence of values for different input samples, we should see some patterns unfold: for example which position(s) in the vector corresponds to hair color. In this way, we could further analyse the performance based on which feature is more dominant and which is less and modify the model parameters to make it more balanced.

Another great tool for exploring a networks latent space is the AutoEncoder architecture. In this task, we used a modified version of the original U-Net model[10], which consists of a symmetric encoder and decoder module. The model's output is the embedding vector we have discussed above, but with the help of the decoder model we can also input our own code to generate custom images and to better see which value corresponds to a particular feature. During training, we only train the decoder modul while freezing the pretrained classifier in order to preserve its feature recognition capacity.

In our experiment, we made a linear interpolation between two input samples in the network's Latent Space and saved the output for each step along the process. This resulted in an animation which showcases the transformation from each input image to another. For this, it was not necessary for the AutoEncoder to produce very good reconstructions as it was obvious that the face's definitive features can be seen regardless.

Our other experiment's goal was to identify redundant features(to be precise, redundant as in, not carrying any information related to the classification) in the output of the last convolutional layer (same layer as before). To be specific, we searched for features, which represented the background of the image(everything but the face). Usually in other works [11], they use attention to get rid of the background's influence. In this experiment we created two modified images for each original image. One of them is the face segmented from the background (see 4), and other is the latter's inverse - only the background is visible.

First to identify if the background really has an affect on the prediction of the model, we fed the network the original inputs and images with removed backgrounds. In 66 percent of the cases, this resulted in a major increase in the models

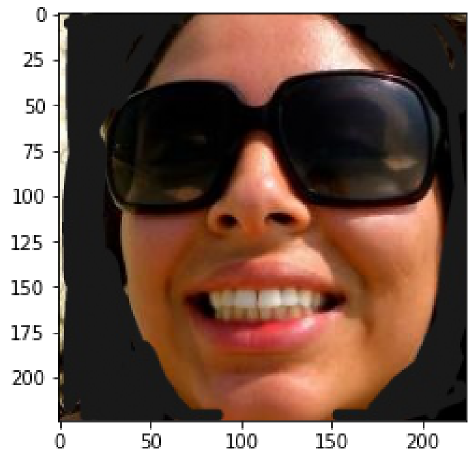

Fig. 4. Image with removed background

confidence (about 15-20 percent), meanwhile in the remaining cases we experienced a slight decrease (2-3 percent). Either way, the test was successful: the background had an affect. Interestingly enough, the predictions didn't change, only its confidence was boosted.

In the next phase we tried to identify the background's features by checking each element of the vector if it's close to the corresponding element of the original inputs embedded vector. Then we looked for the common elements in all the embedded vectors and in the end, creating a sort of "mask" (for each element of the vector it was known if it represented the background or not). To test this mask, we applied to the predictions of our testing subset (first calculating the embedded vectors of the convolutional layers, applying the mask then feeding to the dense layers). Unfortunately, as the results showed, this indentification of the features, brought in a bias. A bias which stemed from the test images: the images were mostly of women in their 20's and that left it's mark on all of the feature maps. All in all, on the downside, it turned out to be a lot more complicated to get a universal pattern of the background. On the upside however, we now know how to purposely manipulate and mask the predictions, using only a handful of feature maps.

V. Conclusion

We believe that during our project we did as thorough work as possible and keeping the mindset which is recommended in this field. The latter being that open minded state, open to any new ideas and solutions which can bring us closer to success, no matter how it may sound first.

In this project we tried and experimented with countless model configurations, always trying new ideas to increase the accuracy of our model. Many experimental results couldn't make it into this paper, simply because there were way too many, and we had to focus on the more significant ones. We hope that our results are useful to the deep learning community.

Our latent space experiments and research also gave us a lot of extra knowledge and insight into the working of the CNN classifiers' world.

To sum up, we are proud of our experiments, research and accomplishments, and we all gained much experience during this project.

References

[1] K. Karkkainen and J. Joo, "Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1548–1558.

[2] A. B. Jung, K. Wada, J. Crall, S. Tanaka, J. Graving, C. Reinders, S. Yadav, J. Banerjee, G. Vecsei, A. Kraft, Z. Rui, J. Borovec, C. Vallentin, S. Zhydenko, K. Pfeiffer, B. Cook, I. Fernández, F.-M. De Rainville, C.-H. Weng, A. Ayala-Acevedo, R. Meudec, M. Laporte *et al.*, "imgaug," https://github.com/aleju/imgaug, 2020, online; accessed 01-Feb-2020.

[3] V. Mavani, S. Raman, and K. P. Miyapuram, "Facial expression recognition using visual saliency and deep learning," *CoRR*, vol. abs/1708.08016, 2017. [Online]. Available: http://arxiv.org/abs/1708.08016

[4] A. Gudi, "Recognizing semantic features in faces using deep learning," *CoRR*, vol. abs/1512.00743, 2015. [Online]. Available: http://arxiv.org/abs/1512.00743

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[6] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proceedings of the British Machine Vision Conference (BMVC)*, M. W. J. Xianghua Xie and G. K. L. Tam, Eds. BMVA Press, September 2015, pp. 41.1–41.12. [Online]. Available: https://dx.doi.org/10.5244/C.29.41

[7] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*. Los Alamitos, CA, USA: IEEE Computer Society, may 2018, pp. 67–74. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/FG.2018.00020

[8] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and $<$1mb model size," *CoRR*, vol. abs/1602.07360, 2016. [Online]. Available: http://arxiv.org/abs/1602.07360

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[11] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," 2018.