Recognition of Facial Features with Deep Learning

Deep Learning a gyakorlatban Python és LUA alapon - BMEVITMAV45 2021/22/1

Gergő Bolla
András Schifferer

Budapest University of Technology and Economics

January 11, 2022

Overview

Problem statement

- Recognizing of facial features such as race, gender and age is a very easy, almost automatic process for humans.
- However extracting this information using computers is more challenging.
- Human faces have a high variability and the combination of these factors results in a big number of individual categories.
- For optimal performance of the computer this also has to be present in the dataset.

Data acquisition

- In classification it is important to have a diverse and balanced dataset.
- Unfortunately in most datasets certain races and age groups are over-represented, due to the biases stemming from web scraping.
- This is why we chose a more racially balanced dataset called Fair Face (see Karkkainen and Joo [1]).

Dataset

- The Fair Face dataset is a balanced well prepared dataset (see Karkkainen and Joo [1]).
- Since using the images itself would be a very GPU heavy task we opted to use embedding using FaceNet (see Schroff [2]).
- FaceNet embeddings creates a mapping from face images to compact Euclidean space where face recognition and feature selection is much easier.

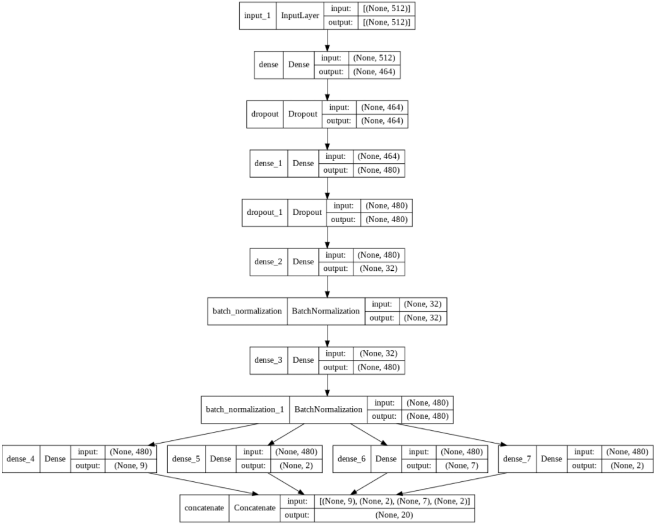
Difficulties

- Regular images and the use of CNN-s require substantial GPU power.
- To make our work stand out and to solve this problem we wanted to make the model so it doesn't require that much GPU power and that's why we used embeddings. This embedding consists of two steps:
  1. algorithm based face alignment (based on features e. g. eyes, using scaling and cropping),
  2. CNN network for a lower dimensional (512) embedding.
- We had to create our own loss function since the others didn't prove to be useful and with this all feature blocks are minimized.

Optimization and training

- To build our model we used the `Keras` framework.
- We used the `KerasTuner` for hyperparameter optimization to create the best model which can be seen in the slide below
- After the best model was established we further trained it and evaluated the results.

Architecture



Custom Loss Function

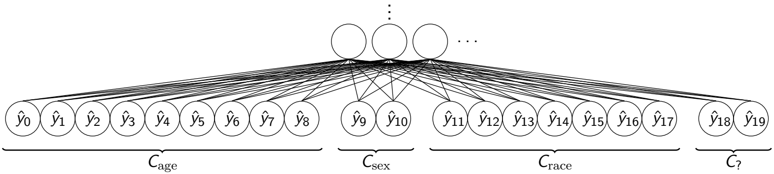


Figure: Loss function - categories

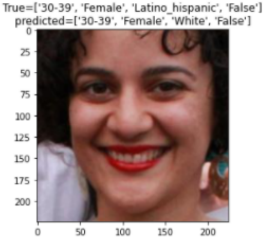
- The last layer splits the output into 4 categories then applies `softmax` activation to each separately.
- The loss function is calculated using categorical crossentropy for each category (age, sex, race) respectively.

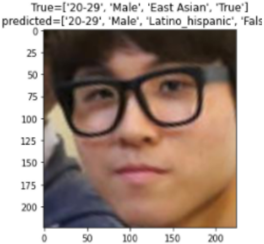
$$\texttt{CustomLoss} = C_{\mathrm{age}} + C_{\mathrm{sex}} + C_{\mathrm{race}} \tag{1}$$

- The last category is ignored, since it has no relevant meaning.

Results

- The model’s binary accuracy was around 80% which would not be bad but since we have more features not just one it is considered inaccurate.
- You can see the evaluation on the pictures below.



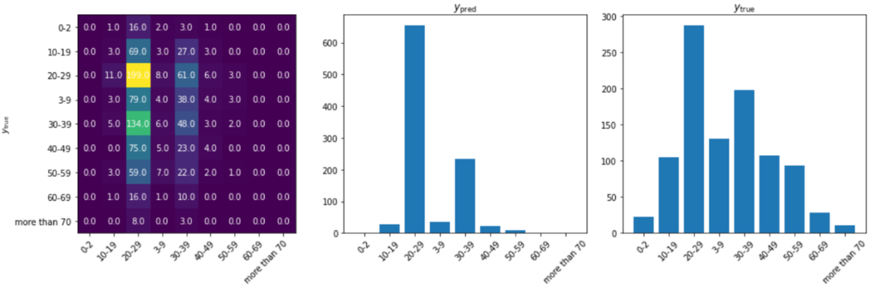


- However the model can be accurate sometimes we also created a confusion matrix that shows the problems of the model.

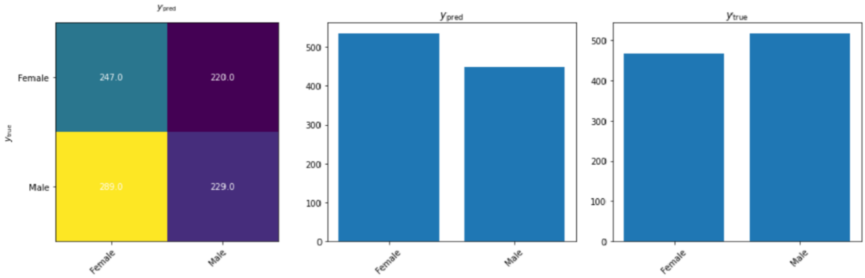
Confusion matrix

- We created a confusion matrix for the model to show how accurate it is in different fields which you can see in the slide below.
- The matrix shows the flaws of our model.
- This also show some flaws in the dataset itself especially in the age category.

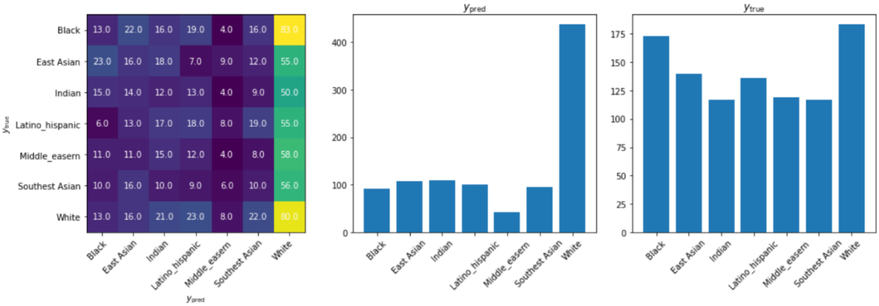
Confusion matrix age



Confusion matrix gender



Confusion matrix race



Further steps

A few steps for future improvement have been identified as possible courses of action to improve the results.

- Weighting in the individual categorical crossentropies so that the imbalanced nature of the dataset is not so pronounced. It could be done with

$$w_i \sim \frac{1}{n_i} \tag{2}$$

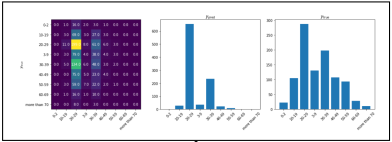
  where is the weight w_i and n_i is the number of elements belonging to the ith category.

- Using a bigger slice of the training data with improved hardware and more training time would have doubtless improved the results as well.

Summary

Our project consisted of

1. data acquisition,
2. training and validation data embedding consisting of
   1. alignment,
   2. pretrained CNN-network,
3. hyperparameter optimization,
4. optimal modeltype training,
5. testing and evaluation.



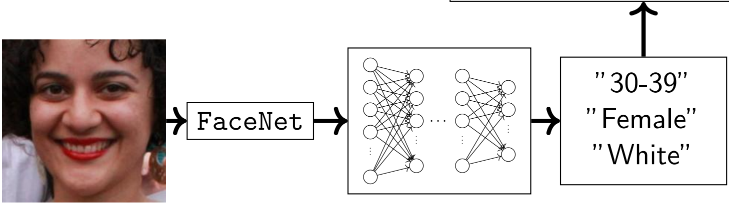


Figure: Process pipeline

End

Thank you for your attention!

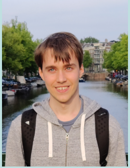
FacExcercisebook

Recognition of Facial Features with Deep Learning

Gergő Bolla

BME Biomedical Engineering MSc student

András Schifferer



BME Mechatronics Engineering BSc student

References

List of used literature.

[1] Kimmo Karkkainen and Jungseock 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] Florian Schroff. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.