Documentation of the face recognition neural network

- Soroncz-Szabó, Flórián (J0QWCB)
- Schäffer, Tamás (CO2WYC)
- Frecska, Hajnalka (C1MTMR)

The plan of the work

Our objective is to compare 2 types of neural networks, the first one is a sequential model, the second one is a transfer learning model which is trained with the pre-trained sequential model. To save time and energy, we were looking for some models on github.com, which are already trained for recognize human faces. The one we chose already has the required features like the labels.

We are aiming on testing the validation accuracy and loss of the networks with different parameter settings. The benefits of activation functions, effects of dropouts and freezing, and testing with images of other datasets.

Database source

For the training we found 2 database sources, these have the names UTKFace and FairFace Dataset. Both of them have 3 same labels but with different format. The similar labels are the age, gender and race. The difference in the format is that for the first one the age is given as a simple integer, but for the second one it is given as an age period. Gender is very simple to understand, it is either 0 or 1, it symbolizes male or female gender. The UTKFace dataset has a label for showing the date and time of every single picture. To make our work easier, we use only one, the FairFace dataset.

Preparation

The model is written in tensorflow framework. At the beginning of the model we install all the required dependencies. After importing the needed libraries, we download the dataset from its Google Drive folder. After the download the inner folders are unzipped we fetch the CSV file that contains the label for the images and read it into the memory. The next step is to split the dataset into 3 train, validation and test subsets, shape them and convert them to the output of the model for One-Hot encoding.

Model

The model is built as a sequential model with maxpoolings at the end of every 2 or 3 convolutional layers. These are responsible for gaining features from the images. At the end of the model the convolutional layers are flattened and next there is a fully connected network with Dense layers and Dropouts. On the output for the One-hot encoding there is a softmax activation.

The transfer learning model is based on the weights of the first model, because it is connected to the base model. This is a fully connected network with dropouts and also a classification with softmax activation for the one-hot encoding.

Training and results

Unfortunately, the model is finished but does not learn.

The aim of the project is mentioned on the top, but we could not test and compare them because of the model.