Computer vision project

Team name: Faces2
Team members:

- Jonas Brandenburg, if20b243, if20b243@technikum-wien.at
- Violeta Garcia Espin, if20b091, if20b091@technikum-wien.at
- Rawan Mousa, if20b129, if20b129@technikum-wien.at

Date: 09.11.2022

Repository: https://github.com/variableVG/CV_project.git

Summary

This project aims to determine the age of people in a series of photographs using convolutional neural networks. The development of an efficient and reliable model to predict the age of people can help to understand how to detect other physical characteristics of pictures. This concrete project could have commercial applications, for example for marketing companies looking for models in certain age ranges. It could also be used for solving crimes, by narrowing down the age of the suspects in a picture.

Task and data description

Our goal in this project was to strengthen the knowledge about convolutional neural networks and to interact with the PyTorch libraries as a part of the subject of Computer Vision.

The data was provided by the lector and can be found at https://susanqq.github.io/UTKFace/. The data consists of 24,106 pictures. The labels of each picture are embedded in the file name in the format: age_gender_race_data&time.jpg.

The age varies from 0 to 116 showing a disproportionate amount of people between the age of 26 and 0. The gender is indicated by 0 if male and 1 if female. Ethnicity is encoded by a number where 0 denotes White, 1 Black, 2 Asian, 3 Indian, and 4 others like Hispanic, Latino, or Middle Eastern.

We decided to address the regression problem of predicting people's ages based on their aspect in a picture. For this, we used convolutional neural networks in PyTorch. The images were resized to have a standard size of 128x128 pixels. Normalization and data Augmentation has also been applied before the convolutional network.

For the architecture of the convolutional neural network, we researched several web pages, choosing the architecture found at https://medium.com/thecyphy/train-cnn-model-with-pytorch-21dafb918f48. The model's architecture was then adapted to our data to make the model work. We have defined it as a loss function L1 (absolute mean error), and we used the Adam optimizer.

Methods

Jupyter Notebooks were used to develop code and documentation. Our main interest was to get in touch and learn skills in the PyTorch framework, so we have tried to implement these libraries whenever possible. For teamwork, we have used GitHub. Due to the limitations of our personal computers, we have used Google Collaboratory which allows us to use GPUs through CUDA.

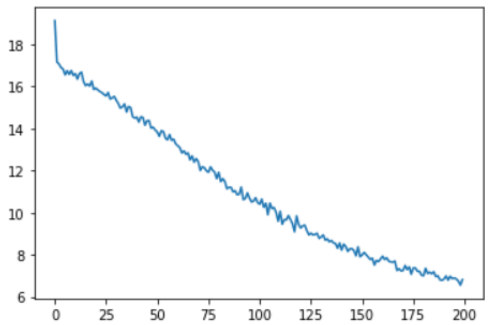
Outcome and conclusions

The biggest challenges were processing the images in an acceptable time and managing RAM. To speed up the processing time we have used GPUs in Google Collaboratory through the CUDA framework. Due to limitations in RAM, we have been forced to drastically reduce our dataset. Pictures of white people in targeted age groups was randomly removed to reduce data imbalances as well. Finally, we have also created a class called TensorDataset that loads the pictures on demand in order not to overload the RAM.

The model can predict the age with a Mean Absolute Error (MAE) of 15 years. It is not a very accurate prediction but considering the limitations regarding time and the size of the dataset, it is a very promising algorithm. It is also an improvement over the first architecture we generated, which threw a 30-year-old error. Due to the time limitations of the authors, different architectures, optimizers, and loss functions were not systematically tested. Therefore, we consider that with more resources in terms of GPUs and RAM, the training data set could be increased, and different architectures, optimizers, and loss functions could be tested to optimize the model.

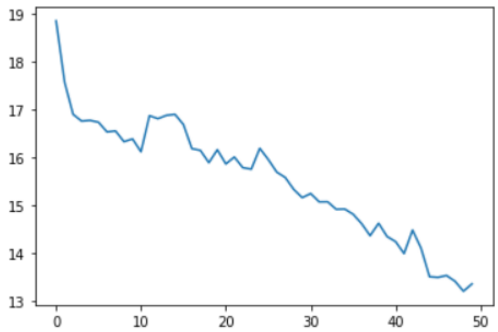
An interesting observation is perhaps the effect the number of epochs has on the MAE in the training data, as reflected in the graphs below. Increasing the number of epochs reduces the error in the training set but not necessarily in the testing set, which is a sign of overfitting. From this observation, we could deduce, that the increase in epochs in this particular case does not necessarily improve the prediction in testing data.

Epochs vs. MAE



MAE for training set: 6.59 years

MAE for testing set: 15.30 years



MAE for training set: 12.58

MAE for testing set: 15.39

Statistics on the data: Distributions