Deep Learning Project:

Analysis of Sample Size based on Dog Breed Classification

Module "Deep Learning"

Authors

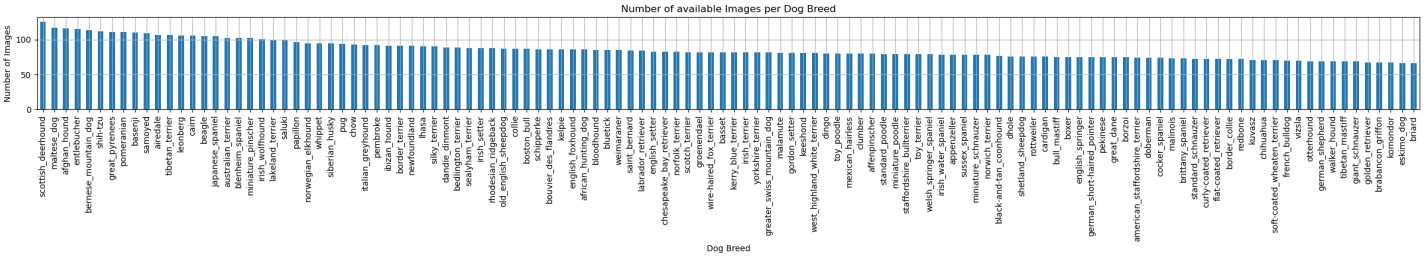
Mathieu Huber, Vito Semeraro

ZHAW - Zürcher Hochschule für Angewandte Wissenschaften, CAS Machine Intelligence,

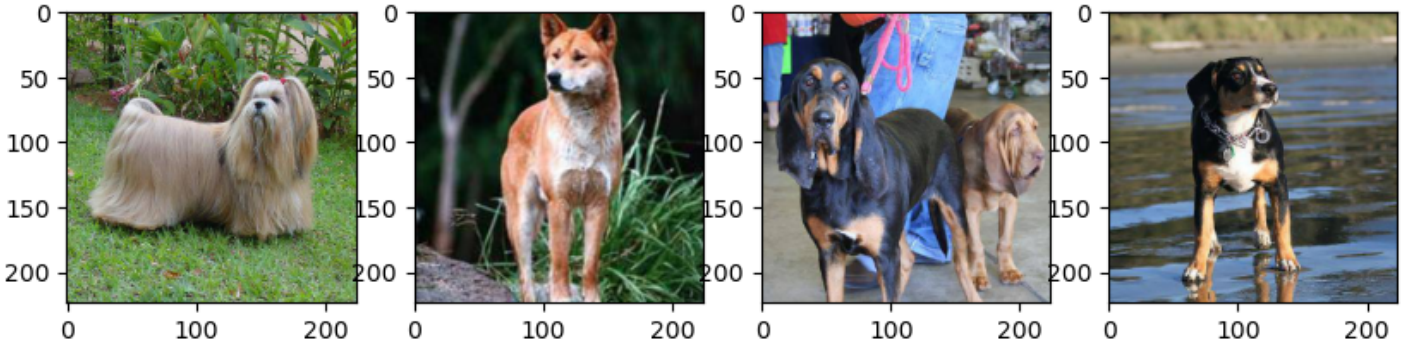
Teachers: Beate Sick, Oliver Dürr, Pascal Bühler

Dataset

For this project we used the Kaggle dataset for dog breeds classification. The dataset is published on the following site https://www.kaggle.com/competitions/dog-breed-identification/data The dataset consists of classified images of 120 dogbreeds. The total number of available images is about 10000. The sample images are of different size. Also the number of available samples per dogbreed differs. It ranges from 66 to 126 samples.




```
<matplotlib.image.AxesImage at 0x152b650d0>
```



Project Scope

In this project we will focus on the different number of training samples per dogbreed and evaluate their impact. For this purpose we will create 3 different training datasets and use these to fit the same deep learning model. Finally, we evaluate these models based on the test set accuracy metric. The following three training dataset configuration will be used:

- *Training set 1*: Image samples "as-is"
- *Training set 2*: Same image sample size for all dogbreeds based on the minimial count over all dogbreed classes
- *Training set 3*: Same image sample size for all dobbreeds based on the maximal count over all dogbreed classes. The outnumbered classes will be augmented using a image data augmentation process using bilinear interpolation

For each modeling process we have a separate training, validation and test dataset. We also fixed the size of the images such that alle have the same dimension, namely 224 x 224 x 3.

Architecture and Modeling

As mentioned above we will fit 3 different models based on different training dataset configurations. In the beginning of this project we planned to use an architecture based on the `AlexNet` by adding extra batch normalization and further drop out steps. Since the fitting process was very time consuming on our machine and the test accuracy was very low, namely close to the baseline accuracy we sticked to a pretrained model, namely the `MobileNetV2` and added some average pooling plus one dense layer and the final decision layer with node equal the number of classes. We used `5` epochs for the model estimation.

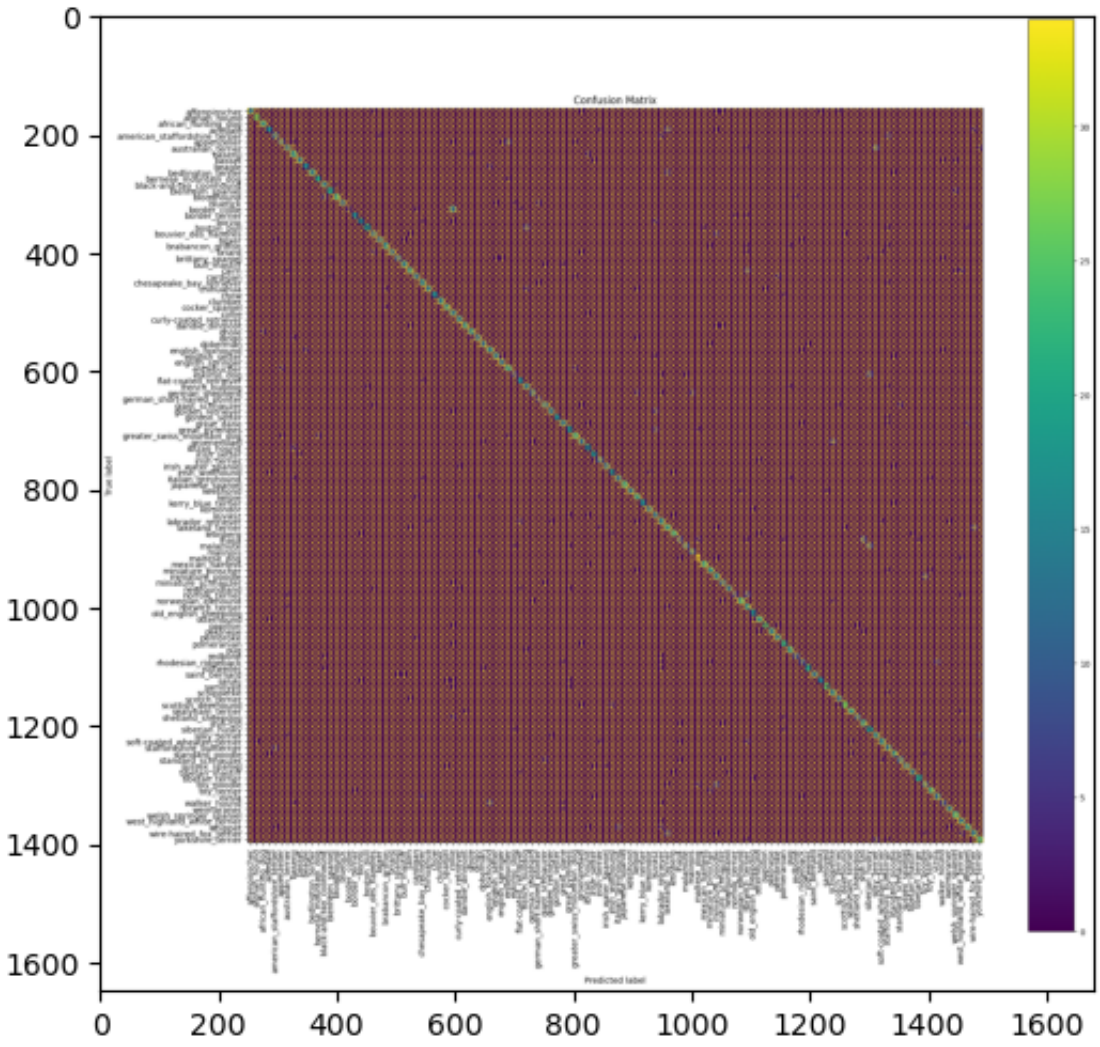
Evaluation: Accuracy metric and confusion matrix

In this section we will give the model evaluation result and report the accuracy metrics. The following accuracies were achieved by the different models:

- *Model 1*: 79.85%
- *Model 2*: 79.17%
- *Model 3*: 75.84%

The baseline accuracy in this classification task is: **0.83%**. We can see that all of the model variants are clearly better than the baseline accuracy. The winner is model 1 which is trained with the "as-is" dataset. The difference between the first and the second is not that striking such that they can be regarded as equally good. Model 3 performs slightly worser. The reason might be that the data augmentation introduced more uncertainty for classes that are strongly outnumbered or introduced more similarities to different breeds.

```
<matplotlib.image.AxesImage at 0x1538d7590>
```



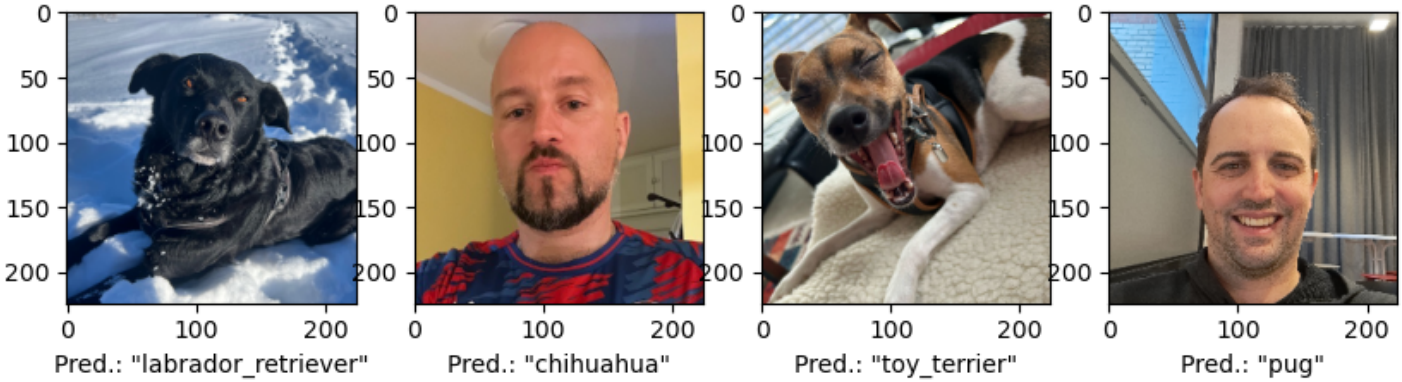
Conclusion

- Implementation of an own architecture requires a lot of time and can completely fail
- Using pretrained networks are performance accelerators
- Image data augmentation is not always benefitial
- Missclassification mainly explainable due to similarities in dogbreed appearance

Fun Factor: Which Dog fits me best?

Since the authors have a strong relationshop with dogs, we will evaluate if we have the right dog for us by asking the model.

```
Text(0.5, 0, 'Pred.: "pug"')
```




```
<matplotlib.image.AxesImage at 0x153c8a190>
```