P05 – Classification of dog breeds

In this assignment you will continue diving into deep learning concepts. To this extent, this lab description collects some tutorials for deep learning models optimization and fine-tuning, and for image classification, using pre-trained models.

Tasks 1.1-1.3 are to get acquainted with tutorials demonstrating the above, where you can change parameters and study the change in performance. Task 2, which will be the one graded, contains a classification problem, along with a dataset, testing your understanding of the concepts from the lectures.

1. Classification models using Pytorch

1.1 Get acquainted with classification with pretrained models by following the tutorial P05a_Inception_ResNet_DenseNet.ipynb. You will compare the performance of three different pretrained models and will see ways to visualize the learning performance of the models.

1.2 Run the tutorial P05b_FashionMNIST_Pytorch.ipynb. Notice, that the first way to build a model is using a pretrained one. The second way is to specify your own model. This could be useful for the graded exercise (Task 2). Can you increase the model performance in some ways?

1.3 Transformer models using Pytorch. Run the tutorial P05c_Transformers_and_MHAttention.ipynb. Note that the pretrained models and the datasets can be found here: https://drive.google.com/drive/folders/1DF7POc6j03pRiWQPWSl5QJX5iY-xK0sV?usp=sharing

2 Create your own classification for images of different dog breeds

Open the python notebook Breeds_classification_Assignment.ipynb. Make sure you have correctly adjusted for gpu or cpu (similar to the previous tutorials). Some data preprocessing is available, and you can start your own model.

You can train your own model, and/or use pretrained models.

Can you achieve accuracy > 90% on all classes?

Providing a report on the performance of the models can help you improving your classification, as you will have to analyze and report the effect on different hyperparameters. Think of different ways how to visualize and analyze the performance and the accuracy of your model. What is influencing them? How do you measure the performance of your model? Which hyperparameters are critical to improve the performance? How do the performance and accuracy change with the number of training epochs? Can you apply your model to random images from the internet?

Hint: When you fetch an image from the internet, you have to apply the same transforms to it as these required for training the model (see the beginning of the notebook: `# transforms for images`).

You can try it on several images from different classes (or outside of any class) that you found, outside of these in the providing dataset.