IN3310– Machine Learning for Image Analysis

Week 04

1 Coding

Learning objectives:

- custom data loaders
- prediction with a pretrained neural network from torchvision. No data augmentation here yet.

Another take away: prediction with a neural net runs fairly fast. It is not a must to use GPU for prediction.
Take the 2500 Imagenet val images, unpack them. The labels are in `ILSVRC2012_bbox_val_v3.tgz`.
`getimagenetclasses.py` has example routines to get the label for one image.

- write a dataset class for this dataset. `https://pytorch.org/tutorials/beginner/data_loading_tutorial.html`.
  - You need to implement __len__ and __getitem__
  - Suggestion: do not load all images in the __init__ method of the dataset class. That does not scale well if you have 500000 images :) . Load the filenames and the labels instead into a list or the like. load an image in `getitem` of your Dataset-derived class!
- rescale the images so that the smaller side is $s = 224$ pixels and perform a center crop of 224×224
- use on top a generic Dataloader class from PyTorch.
- neural network: Choose one from the torchvision model zoo. Initialize the weights of the chosen neural net so that you load weights from the so-called torchvision model zoo.

- compare performance to the case when you do not subtract the mean and normalize the subpixels. If too slow, use only the first 500 images, report the performance difference
- visualize an image and its top-5 predicted classes ... matplotlib can help.
- hint: use a net with little parameters, avoid VGG, Alexnet unless you got a GPU.