HW4 – submission 24.1.19 23:55

Guide lines

1. Include all your personal details including name, id, and **e-mail address.**
2. You should submit all function and script files written in MATLAB or Python. Your code should be well documented and clear. The code should run from **any** computer and include all path definitions (You should take care of this in the code).
3. Please divide the code by questions.
4. Final report – should include explanations on the implementation and the execution, answers to the questions, results, conclusions and visual results. Do elaborate on all parts of the algorithms/solution. **Please submit a PDF file and not a DOC file.**
5. Please post question regarding this HW on the facebook group: https://www.facebook.com/groups/294298727746314/
6. The grades are highly depended upon the analysis depth of the report.
7. HW can be submitted in pairs.
8. Eventually submit one compressed file including the code + images PDF.

Good luck!

Introduction to CNN

It might be that this is slightly easier to solve in MATLAB, yet 99% of deep learning researchers and developers are working in pytorch\tesorflow, thus it is worth the effort.

For this section you will need to use MATLAB Neural Network Toolbox if you are using MATLAB or pytorch\tesorflow if you are using python. You will not need a GPU (basic one should work) for implementing this section. You will probably need to install the tools above first.

*Keep in mind that some tools are harder to learn and get started with than others. This section was designed so it could be run in MATLAB without previous knowledge of the Neural Network Toolbox.

-Examples of MATLAB Neural Network Toolbox: www.mathworks.com/products/neural-network/videos.html
-For installing pytorch: http://pytorch.org/
-For installing tensorflow: www.tensorflow.org/install/

1. Download a pre-trained VGG16:
   For MATLAB you can use this example www.mathworks.com/matlabcentral/fileexchange/61733-neural-network-toolbox-model-for-vgg-16-network (try the code. Most likely you will get an error message. Try installing the network from the link you get in the error).
   For pyTorch you use this link: http://pytorch.org/docs/master/torchvision/models.html
   For tensorflow you can use the slim library (has pretrained VGG16):
   https://github.com/tensorflow/models/tree/master/research/slim
2. Load the images in the attached *birds* folder and display them. Transform the images so they would fit as input to the network and show the transformed images. Think about size, type, normalization and range. Use the images as input to the network. What are the outputs?
3. Find an image on the web, use it as input, and show the output of the network for it.
4. Use one geometric transformation, one color transformation and one filter on your image (the image from section 3). Show the resulting images. Then use them as input to the network and show the outputs. What has changed with respect to item 3?
5. For 2 filters in the first conv layer of VGG16 show the filters and show their response for the 3 images from section 4. Discuss the differences.
6. For every image in *dogs* and for every image in *cats* extract the feature vector of layer FC7. Think on a method to visualize the Vectors, if possible so you can see the differences between dogs and cats. You can be creative or use methods like PCA.
7. Take one image of a cat and one image of a dog from the internet and show them. Extract their layer FC7 feature vector and find the nearest neighbor [=closest vector] from the feature vectors of section 6 (all 20). You can select any distance measure, L1 for example, state it in your answer.
8. Take one image of a wolf and one image of a tiger from the internet and repeat section 7 (again compare to all 20 vectors) for these images (i.e., find the most similar cat or dog of each image).