Assignment 2

Deadline: Friday, 29th May (23:59)

15th May, 2020

Question 1 - Neural Codes for Image Retrieval (10pt):

Use the representations learned by a convolutional neural network (ConvNet) for image retrieval, as proposed in

> A. Babenko, A. Slesarev, A. Chigorin, V. Lempitsky, *Neural Codes for Image Retrieval*, ECCV, 2014 (https://arxiv.org/abs/1404.1777).

You can use the Jupyter notebook `2IMM10_Assignment_2_1.ipynb`, which already downloads, loads and pre-processes the data, and provides some helper functions. Write your code between all two consecutive occurrences of "`# ...`". See the text cells in the notebook for additional information.

Tiny Imagenet Use the training set of *Tiny Imagenet*,[1] consisting of 200 classes, with 500 images per class (total $100,000$ images), each image being of dimensions 64×64 RGB. The Jupyter notebook already gets the data for you, and also performs the following steps.

- Split the 200 classes into two sets, one containing 190 classes and the other containing the remaining 10 classes.
- Shuffle the set with 190 classes and divide it into *training*, *validation*, and *test* sets, according to the proportions 80/10/10.
- The set with the remaining 10 classes serves as *out-of-domain* (ood) data, used for image retrieval.
- Normalize pixel values to $[0, 1]$.

Train ConvNet Reproduce the ConvNet architecture from Babenko et al., with two exceptions: For `Layer 1`, use kernel size 4×4 (instead of 11×11) and stride 1 (instead of 4). For the hidden fully connected layers, `Layer 6` and `Layer 7`, use 2048 units (instead of 4096).

- Implement the model in Keras.
- Train it by optimizing *cross-entropy* with the *Adam* optimizer, using a learning rate of 0.0001 and a *batch size* of 100. Set the flag *amsgrad* to *True*.
- Evaluate and report the *train*, *validation* and *test* performance, in terms of *cross-entropy*, *classification accuracy* and *top-5 classification accuracy*.
- Name two techniques which would likely improve the test accuracy.

[1] https://tiny-imagenet.herokuapp.com/

Image Retrieval Use the trained ConvNet to perform image retrieval on the ood data. When using a certain image as query image, the remaining 4,999 should serve as a retrieval date base.

- Obtain neural codes for each image in the ood data. Use the same 3 layers for neural codes which were used in the paper by Babenko et al.[2]
- Normalize the codes to have unit length.
- For the first 10 images in the ood set, find the respectively 5 closest[3] images in the data base. Plot the query image next to the 5 retrieved images (sorted from most to least similar) and mark the images which have the same class as the query image (see Fig. 2 and 3 in the paper).
- What are the qualitative differences between the different layers for neural codes?
- Compute and report the *mean average precision* (mAP) over the whole ood set, for each of the 3 layers.
- Do the observed mAP values (roughly) confirm the observations by Babenko et al.?

Question 2 - Peer review (0pt):

Finally, each group member must write a single paragraph outlining their opinion on the work distribution within the group. Did every group member contribute equally? Did you split up tasks in a fair manner, or jointly worked through the exercises? Do you think that some members of your group deserve a different grade from others?

[2] The Jupyter notebook already provides functions to get these codes.

[3] Hint: You might want to exploit the relation between inner products and Euclidean distances.