Fall 2023: ELG7186
Assignment 2

Due: 23:00 on October 12th, 2023

University of Ottawa - Université d'Ottawa

Jochen Lang

1 Texture Image Comparison

This assignment explores image processing and the multilayer perceptron with texture data. The idea here is that we don't see all texture classes during training. Instead we want to find out if two texture images show the same texture or not. We will be using the Kylberg Texture Dataset v. 1.0 by Dr. Gustaf Kylberg [1] at the Centre for Image Analysis, Uppsala University, Sweden. The original database contains 28 texture classes and is available at `https://www.cb.uu.se/~gustaf/texture/`. However, for this assignment, we are only using the small subset of 6 classes with 40 images each. We will be using two subsets of images: 180 images for training and validation and 60 images for testing. Please note that these image sets are also available on BrightSpace as attachment to this assignment. You are not allowed to use the testset for anything else than for your final assessment of the approaches in Section 1.5.

1.1 Getting Started

You will need to download the two subsets from BrightSpace or from Uppsala University. Unpack the images in a directory relative to your jupyter notebook called `textures`. We will be marking your notebook with the data installed in `textures/training` and `textures/testing`, and your notebook will have to work with the images at these locations in the corresponding six subdirectories named for the texture (canvas1, cushion1, linsseeds1, sand1, seat2 and stone1). The training and validation data are the images numbered 001 to 030, while the testing images have the numbers 031 to 040. Do not rename images, directories or reorganize the data. You will loose marks if your notebook does not work with images at the expected locations.

1.2 Image Preprocessing [1.0]

You need to write a python function that loads images and preprocesses them as descibed below. The images are of size 576×576. Build an image pyramid for each image with 3 levels of downsampling by a factor of 2. Utilize histogram equalization and Gaussian smoothing before the downsampling. The final image pyramid size should be $[576 \times 576, 228 \times 228, 144 \times 144]$. Visualize one example in the notebook.

Utilize the Sobel edge filter on each level of the image pyramid, then summarize the filtered texture image by a histogram with a fixed bin size of 256. The final feature shape, i.e., the complete histogram over all levels for each image should be $[3 \times 256]$. Use this feature space for the following questions. Keep in mind the normalization as you answer the questions.

1.3 Learning-Free Classification [2.0]

Use the histogram for the first pyramid level (with shape $[1 \times 256]$) for this question. Generate an overall histogram for each texture category by fusing the histograms from all the textures of the same category in the training set. This can be done by averaging the histograms in each category. Visualize the six histograms for each category in one chart to observe the difference of the distribution. Ideally, you can get a chart similar to Fig 1 (only an example, not for the exact data) which shows that the histograms for most of the classes are well separated.

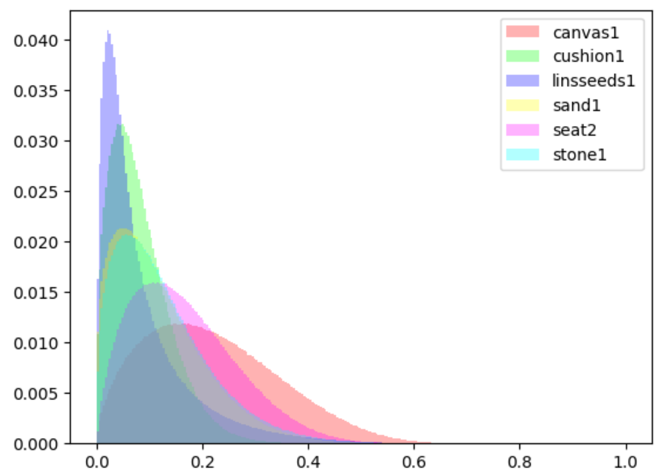

Figure 1: Example chart for class histogram visualization.

Define a function to measure the distance between a given image histogram and the histogram for each class. Then use this function to classify unseen images. For example, in Fig 1, the width of the histogram, the peak and the shape are all different. Your function should not contain any training step for this question.

Evaluate your method on both training set and validation set by analyzing the accuracy, recall and precision. Visualize one of the miss-classified sample and briefly discuss your observation.

1.4 Learning-Based Classification [2.0]

Build a multilayer perceptron model to classify an image by using the image feature pyramid from Task 1.2. The simplest approach is to flatten the input feature from shape $[3 \times 256]$ to $[1 \times 768]$ For this part of the assignment, you must build and train the Multi-layer perceptron model with scikit-learn, or alternatively with the Keras API of tensorflow. Use the validation set to monitor the training of your classifier. Evaluate your method on both training set and validation set by analyzing the accuracy, recall and precision.

1.5 Classification Comparison [1.0]

Compare the classifier of Sections 1.3 and 1.4 on the test data subset. Consider classifier performance but also other criteria, e.g., training effort, prediction speed, generalization and robustness. Your brief discussion based on quantifiable criteria need to be contained in your Jupyter notebook.

1.6 Improved Classification [1.0]

Show histograms for the second and third levels of the pyramid in the same way as in Section 1.3. Discuss if other levels of the pyramid could potentially lead to an improvement of your method in Section 1.5. Support your discussion with diagrams and/or numbers.

2 Submission

You will need to submit your solution in a Jupyter file, do *not* submit the image data. Make sure you have run all the cells. All text must be embedded in the Jupyter file, I will not look at separately submitted text files. If your Jupyter file needs a local python file to run, please submit it as well. Assignment submission is only though Virtual Campus by the deadline. No late submissions are allowed, you can submit multiple times but only your last submission is kept and marked.

References

[1] G. Kylberg, *Kylberg texture dataset v. 1.0.* Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, Sweden, 2011.