Assignment #1

Image Based Biometry 2022/23
Faculty of Computer and Information Science
University of Ljubljana

Basic Recognition

I. Introduction

Your task is to implement and evaluate a popular old-school algorithm Local Binary Patterns (LBP) [1]–[3] for the recognition task based on scientific papers. In order to see that your implementation works you will compare this to a plain pixel-wise comparison of images and optionally also to the OpenCV's implementation.

II. Materials

- **Literature** Search for Local Binary Pattern papers on Google Scholar. However, a good start is:
  - Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns [1] (https://tinyurl.com/3daky8a3)
  - RLBP: Robust Local Binary Pattern [4] (https://tinyurl.com/6wwbntwm)
  - A Completed Modeling of Local Binary Pattern Operator for Texture Classification [5] (https://tinyurl.com/3be6r6u6)
- **Code** Please follow the PEP8 guidelines (https://peps.python.org/pep-0008/) and comment a lot.
- **Data** Use data available here (1000 samples of 10 classes): https://tinyurl.com/3ucw29ar.
- **Report** Please, begin with the template available here: https://www.overleaf.com/read/ppbyvnbqfyrt and limit yourself to two pages.

III. Things You Need to Do

- A loop to read all the images in grayscale and in a fixed image size (e.g. 128×128) and then call feature extractor for each one.
- Plain pixel-by-pixel feature extractor (just transform two-dimensional image into one-dimensional vector) to have some baseline.
- LBP feature extractor with multiple stages of implementation and add-ons, e.g.: with or without histograms, uniform-version of LBP, different radius levels (during tutorials we used R=1), different code lengths (during tutorials we used L=8), possible local region/window overlaps and others.
- Compare all the computed feature vectors between themselves (you get a triangular matrix of distances) separately for each type of feature extractor (pixel-by-pixel, LBP, uLBP etc.) and compute rank-1 recognition rate. You do that by getting the closest vector to the currently observed one and check whether it corresponds to the same class (correct) or not (incorrect). When you go through all the vectors and sum up predictions you get the percentage of rank-1 recognition rate.
- Collect all the results, put them into the report and add the descriptions of what you have done.
- Submit the code (without the data) and the report in PDF.

IV. Questions

To help you with the report and programming goals, try answering the questions below in the report. Some of these are optional (extra) since the maximum amount of points you can get is 20.

- **2 pts** Report rank-1 classification accuracy when using pixel-wise image comparison.
- **8 pts** Report rank-1 classification accuracy with your own basic version of LBP.
- **2 pts for each** Report on improvement/deterioration for each LBP improvement, such as:
  - using histograms or not,
  - using different radii,
  - using different word lengths,
  - implementing uniform LBP,
  - using different levels of local region overlaps,
  - using different input image sizes,
  - etc.
- **2 pts** Report on improvement/deterioration when using different distance measures for feature vector comparison (e.g. euclidean vs cosine distance).
- **2 pts** Compare rank-1 of your implementation with the one from OpenCV (Scikit).

V. Submission

There are two deadlines:

- October 21: submission of only the first draft of your Python code on Eučilnica (we will not be checking this) on Eučilnica: https://tinyurl.com/55dvpy8t.
- October 28: final submission with the LBP implementation, evaluation code (Python files) and a two-page report (PDF) on Eučilnica: https://tinyurl.com/yckwf98d.

REFERENCES

[1] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[2] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *European conference on computer vision*. Springer, 2004, pp. 469–481.

[3] C. Silva, T. Bouwmans, and C. Frélicot, "An extended center-symmetric local binary pattern for background modeling and subtraction in videos," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP 2015*, 2015.

[4] J. Chen, V. Kellokumpu, G. Zhao, and M. Pietikäinen, "Rlbp: Robust local binary pattern." in *BMVC*, 2013.

[5] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE transactions on image processing*, vol. 19, no. 6, pp. 1657–1663, 2010.