IN5520/IN9520 Mandatory term project 2020 – Part I

Segmentation of textured regions in an image

In this mandatory exercise you are going to describe textured regions in an image, compute and visualize GLCM's from each texture, extract GLCM feature images, and segment these images.

We recommend that you use Matlab and/or Python.

The next exercise (Part II) will be about classification, and will not necessarily be based on the same data set as Part I.

Time table

• Exercise Part I available: Wednesday 9th September 2020.
• Deadline Part I: Wednesday 30th September 2020 (3 weeks).

Submission:

Your solution must be submitted as a single PDF file containing the problem description, discussion, and the supporting source code. The files should be compressed (.zip or .tar) in a folder named YOURUIOUSERNAME_PARTI.zip/tar, and uploaded through the devilry system: http://devilry.ifi.uio.no before the deadline above. Questions about submission can be directed to the group teacher Edward Fabian Bull (*edwardfb@math.uio.no*).

Evaluation:

- The two mandatory exercises will be evaluated separately.
- Please note that in order to take the exam, both mandatory exercises must be passed.

Since image processing is a field where solutions often are found by experimenting with different methods, we would like to emphasis the following point:
You should analyze the problem and the input images so that you can select a suitable method and parameters. You should not start testing all available methods that you know of, even if that would be an impressive amount of work. Analysis and logical reasoning about alternatives, and then choosing and eventually comparing just a few approaches is usually a better approach.

How to work:

The exercise is an individual work, and each student should deliver a written report.
Your report should be genuine, in particular we will check that each report provides its own discussion of all method and parameter choices. Include references if you use external sources.

The report should contain a description of the problem, theory, chosen methods, results and algorithms used. You have to document all steps in the algorithms, and listings of your own code should be included as appendix.

The images:

The following two images should be used:
www.uio.no/studier/emner/matnat/ifi/IN5520/h20/undervisningsmateriale/mandatory1/mosaic1.png
www.uio.no/studier/emner/matnat/ifi/IN5520/h20/undervisningsmateriale/mandatory1/mosaic2.png

The four steps:

Part I can be divided into four steps, but must be seen together.

A. Analyzing the textures.

Describe the 8 different textures by words. What characterizes each texture? How do the textures differ? Keywords: texture direction, frequency, variance, homogeneity, texture element size.

B. Visualizing GLCM matrices

- For this step, create one subimage for each texture.
- Eventual histogram transforms must be discussed.
- **Instead of testing a massive list of GLCM parameter combinations, you will be rewarded for giving good reasons for a shorter list of parameters, so you should explain your choice of parameter values.** This should be related to your analysis in A.
- Requantize the image and compute the normalized, symmetric GLCM matrix for the chosen parameter values for each texture. If you use isotropic GLCM, you should give details on how it is computed. You should discuss whether directional or isotropic GLCM is beneficial for this specific problem. Again, your decision here should reflect what you observed during your analysis in A.
- We suggest that you try to find just a few values of the parameters (d, θ) that can be used for all features, knowing that the chosen parameters should help you differentiate between the textures in the mosaic.
- Include the GLCM matrices as images in the report, and include a color bar on the figures.
- From the GLCM matrices, discuss similarities and differences between the image textures. Select one of the GLCM features below, and discuss which parts of the matrices that seem to be useful for discriminating between the textures.

C. Computing GLCM feature images in local windows

Compute feature images from the GLCM features:

- GLCM homogeneity, also called Inverse Difference Moment (IDM)

$$IDM = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{1}{1+(i-j)^2} P(i,j)$$

- GLCM inertia (called Contrast in G&W)

$$INR = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{i-j\}^2 \times P(i,j)$$

- GLCM cluster shade

$$SHD = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{i+j-\mu_x-\mu_y\}^3 \times P(i,j)$$

To calculate a feature image, you let a local window glide over the entire image of the four textures and for each window position you get one GLCM matrix which – when multiplied with the proper feature weight function - will give you one scalar value that position for the chosen feature. Please give the reasons for your choice of window size.

Please include the GLCM feature images in your report. Make sure to scale them so they show proper contrast in your PDF file.

D. Segment the GLCM feature images and describe how they separate the textures.

Apply global thresholding to the GLCM feature images. You can experiment with the global threshold to manually find a threshold value for each feature image that gives the best separation of the textured regions.

Include the tresholded feature images in your report, and discuss which image textures that are best separated in each GLCM texture feature image, both in terms of correct boundaries between textures and errors within the textured regions.

You will see that no single thresholded feature image can distinguish between all textures. Discuss how segmentation results can be combined in order to solve the whole problem.

General remarks:

- You may use libraries and pre-programmed functions, as long as you cite your source(s).
- If you want to make absolutely sure that you understand the GLCM method, it may be wise to implement it.
- You should not test a massive list of GLCM feature/parameter combinations.
- You will be rewarded for giving a good discussion based on the image textures and the given GLCM features, leading up to a shorter list of features/parameters.
- You should try to find combinations of features/parameters that are not correlated, but rather complementary to each other.

Good luck!

Fritz Albregtsen and Anne S. Solberg