INF 4300 Mandatory term project 2015 – Part II

Feature evaluation and classification

In this mandatory exercise you are going to implement a multivariate Gaussian classifier and use it to classifiy images with 4 different texture classes. You must implement your own classifier, but you are allowed to use library functions to invert matrixes, compute the determinant, and compute the mean vector and the covariance matrix for each class. You choose if you implement this in Matlab, C, IDL, Python, or other programming environments. However, it is recommended that you use an environment where you have access to a numerically stable library function for matrix inversion (for example Matlab).

The exercise combines feature evaluation, classifier implementation, training a classifier, testing different feature combinations and evaluating classifier performance.

Time table:

- Exercise and images available: Thursday October 22, 2015
- Deadline for Part II: Wednesday November 11, 2015

Submission:

Your solution should be submitted through Devilry (devilry.ifi.uio.no), and should contain a report (.pdf) containing the problem description and discussion. The supporting source code should also be provided, and be included in you report as a formatted appendix. Any questions regarding this exercise can be asked to the group teacher Ole Marius Hoel Rindal omrindal@ifi.uio.no.

Evaluation:

Since image processing is a field where solutions often are found by experimenting with different methods, we would like to emphasize the following point: You will be credited for analyzing the problem and the input images so you can select suitable methods and parameters. You will not be credited for testing all available methods/features, even if it is a huge amount of work. Analysis and discussion, of both input and output, are very important. Please note that both mandatory exercises must be passed in order to take the exam.

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 the description of the problem, theory, chosen methods, results and algorithms used. You have to document all steps in the algorithms, and listings of our own code should be included as appendix. **The code for your classification algorithm should be listed in your report.**

The image data set

You have available a separate training data set and a test data set of three original images and a set of precomputed GLCM matrices computed from the training image. The images can be found under …/undervisningsmateriale/mandatory2 .

Training data set:
mosaic1_train.mat
training_mask.mat
texture1_glcmdx0dy-1.mat
texture1_glcmdx+1dy0.mat
texture1_glcmdx+1dy-1.mat
texture1_glcmdx-1dy-1.mat
texture2_glcmdx0dy-1.mat
texture2_glcmdx+1dy0.mat
texture2_glcmdx+1dy-1.mat
texture2_glcmdx-1dy-1.mat
texture3_glcm0dx0dy-1.mat
texture3_glcmdx+1dy0.mat
texture3_glcmdx+1dy-1.mat
texture3_glcmdx-1dy-1.mat
texture4_glcmdx0dy-1.mat
texture4_glcmdx+1dy0.mat
texture4_glcmdx+1dy-1.mat
texture4_glcmdx-1dy-1.mat

Test data set:
mosaic2_test.mat
mosaic3_test.mat

The task has the following steps

1. Choosing glcm images to work with.
2. Discussing new features by subdividing the GLCM matrices
3. Selecting and implementing the best features from the GLCM matrices
4. Implementing a Gaussian classifier
5. Training the classifier on the chosen features
6. Classifying the test images. Compute the classification accuracy and confusion matrices and discussing the performance of the classification

1. Choosing GLCM images to work with

mosaic1_train.mat contains 4 different textures (texture1,…texture4). A subimage of each texture was used to precompute GLCM matrices as you did in Part I. GLCM matrices with distances ($\Delta x=1, \Delta y=0$), ($\Delta x=0, \Delta y=-1$), ($\Delta x=1, \Delta y=-1$) and ($\Delta x=-1, \Delta y=-1$) for each texture, and G=16. These GLCM images are given above.
Analyze the GLCM matrices given and select maximum two directions that you expect to be useful for discriminating the textures.

2. Discussing new features by subdividing the GLCM matrices

In this exercise you should NOT use any of the GLCM features from the lectures, but implement your own features that are based on only parts of the GLCM matrices, not the entire matrix. Divide the 16x16 GLCM matrix into four quadrants Q1, Q2, Q3 and Q4 of the same size. Create new features by summing the amount of energy/percentage of gray level transitions found in each quadrant, e.g.

FeatureQ1: $$Q1 = \frac{\sum_{i=1}^{8}\sum_{j=1}^{8} P(i,j)}{\sum_{i=1}^{G}\sum_{j=1}^{G} P(i,j)}$$

Feature Q2: $$Q2 = \frac{\sum_{i=1}^{8}\sum_{j=9}^{G} P(i,j)}{\sum_{i=1}^{G}\sum_{j=1}^{G} P(i,j)}$$

Features Q3 and Q4 should be computed correspondingly.

Based on just visual inspection of the selected GLCM matrices, discuss if you think all four textures can be separated with these features. If you do not think the textures can be separated using a subdivision into 4 quadrants, you can subdivide ONE of the quadrants into 4 smaller quadrants of equal size. If you choose to do this, discuss which quadrant you should subdivide.

How many quadrants do you need?

3. Selecting and implementing a subset of these features.

Implement the features chosen in step 2 using sliding windows of size 31x31, G=16, and the direction(s) you chose. Consider all your features but discuss if you will need all of them in the classification. Select some of the features and include the corresponding feature images computed from mosaic1_train.mat in your report.

4. Implement a multivariate Gaussian classifier.

The classifier can and should use library functions for matrix inversion and computing the determinant. If you want, you can use library functions for estimating the mean vector and the covariance matrix. However, you must implement the computation of the posterior probability using Bayes rule yourself.

5. Training the classifier based on the feature subset from point 3.

Compute the overall classification accuracy and the full confusion matrix based on the training data. Discuss the performance, what does the confusion matrix show? (Hint: if you run into a singular covariance matrix using a certain combination of features, there might be a reason for that. Try to understand why. If you don't run into singular matrices you feature selection has avoided this little "trap".)

6. Evaluation of classification performance on the test data set using the set of features selected in point 3.

Now compute the overall classification accuracy and the confusion matrix on the two test images. These images are slightly different from the training image.

Good luck!
Fritz Albregtsen and Anne Solberg