Medical Image Computing

(MT-M-3-ILV-IM2)

Exercise 03

Start Date: 22.12.2022

Due Date: 20.1.2023

Authors: Marco Augustin

---

0. General information

0.1. Report submission: Please add all authors of your group to the report. **Maximum number of authors in a group is three.** Each author has to upload the full submission (report+code). The report must be attached in PDF form.
Please note: Feel free to switch groups during the semester.

0.2. Code submission: Please add a single compressed package containing all relevant code and data.

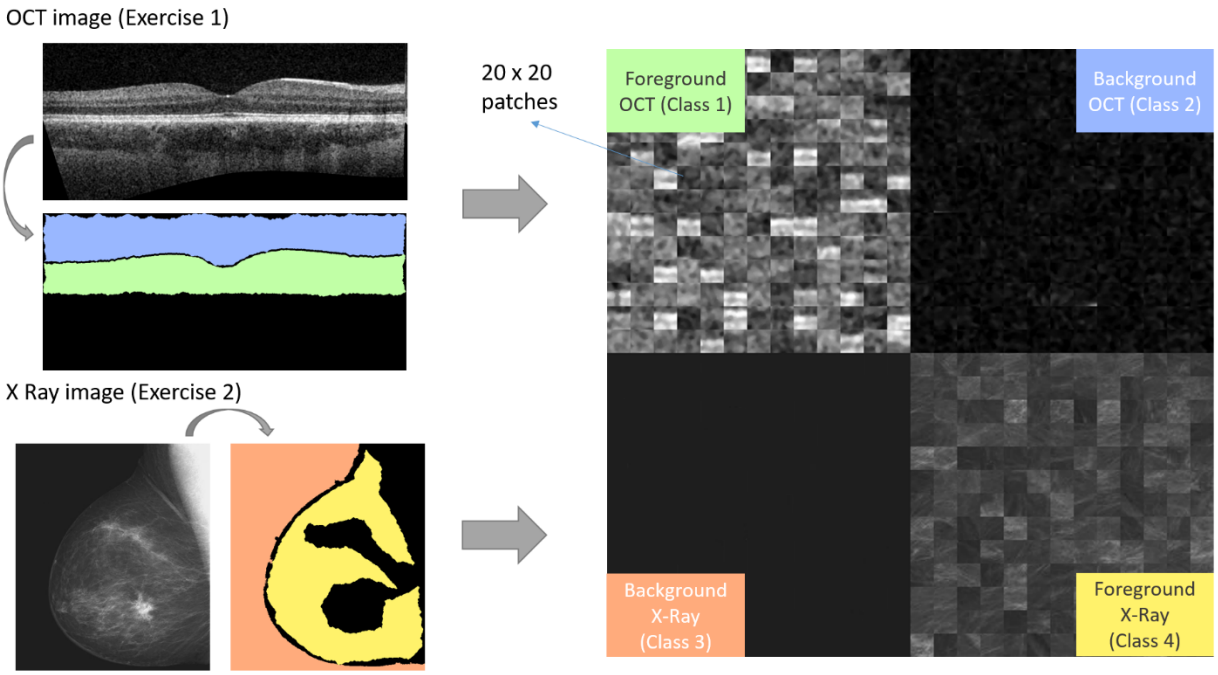
0.3. All relevant information can be found in the Sakai Course Site: https://sakai.mci4me.at/portal/site/Course-ID-SLVA-34453

0.4. In case of any questions please use the Sakai Forum, or e-mail to: marco.augustin@mci4me.at

0.5. Image data to use in the exercise is attached to the exercise specification in Sakai.

0.6. All built-in functions can be used from Matlab or Python libraries.

1. Problem description



The goal of this exercise is to train and evaluate two classifiers. The task for the classifiers is to distinguish between different regions (classes) in the images you already know from Exercise 1 and 2.

Patches (20 x 20 pixels) of the OCT fore- and background as well as the X- Ray fore- and background were extracted, see Figure above. The textural features you know from Exercise 2 were extracted for each patch and are stored in the attached csv file.

Two files are attached to the exercise:

- "XL_2022.csv": The design matrix was constructed based on the textural features of Exercise 2 and can be found in this file. In addition, the last column of this data contains the class labels (1 = OCT foreground, 2 = OCT background, 3 = X-Ray background, 4 = X-Ray foreground)
- "patches.tif": The image containing all patches. **Only needed for the Bonus work!**

2. Feature selection (3 points)

2.1. Load the data of the "XL_2022.csv" file. While the last column contains the class labels the rest of the array contains the design matrix X. How many examples does the data include? How many features?

2.2. Extract the design matrix from the dataframe by dropping the last column ("Label") and split the data into a training and a test set. The test set should contain the first and the last 52 examples (rows) of the data. All other examples should be part of the training set. How big is your test and training dataset, how many samples from each class do they contain?

2.3. Calculate and plot the covariance matrix of the features and visualize the result from your training dataset. Based on your visualization, select a reduced set of features you think might be useful for further classification and generate a reduced design matrix X1. Give a rationale for your choice of features.

3. Classification (3 points)

3.1. Use a kNN classifier to classify the data into 4 groups using 5 nearest neighbors. Analyze the performance of the classifier using the test dataset by calculating the classification error as the number of false classifications divided by the total number of samples. Compare the performance of the classifier when applied to the data containing all features X and to your reduced design matrix X1.

3.2. Train a random forest on the training data based on the design matrix X. Analyze and interpret the influence of the number of trees using the out-of-bag classification error (oob_score). Control the random number generation to ease interpretation (Matlab function rng; sklearn random forest random_state=0). Evaluate the performance of the random forest on the test dataset using two different numbers of decision trees in your ensemble. Again, use the predict function to get the label predictions.

3.3. Analyze and interpret the importance of the various features using rf.O*OBPermutedPredictorDeltaError* (Matlab) / RandomForestClassifier.feature_importances_ (Python sklearn). Which features were most valuable for the random forest?

4. Bonus work (2 points)

4.1. The design matrix used in the previous sections is based on the image “patches.tif”. Use the image attached and write a code to extract two more features of your choice for the 20 x 20 pixel sized patches. Re-use the functions you have used in exercise 2.

4.2. Include the features in the design matrix and re-evaluate the performance of the random forest using your new design matrix. Discuss the influence of your added features.