Project Task 2

Applied Medical Image Analysis

(M2.03100.20.030)

10.06.2024

# Submission guidelines

* + One submission per team uploaded to Moodle
  + Deadline: July 15th, 3:00 pm.
  + A script RUN.m, which executes, computes all results, and plots them without user interaction. Supporting functions can be in separate MATLAB files.
  + The code has to be commented to briefly and clearly describe what the code does.
  + A PDF report that contains an explanation and interpretation of all exercises (approximately 2 - 4 pages).
  + A front page of the report contains the group number and team member names.
  + The report has to be short and precise, do not include code.
  + You can register for max. 2 project supervision sessions to answer any questions regarding the exercise, and provide feedback on your code or parts of your report. No support outside these sessions will be provided.

# Assignment

The goal of Project Task 2 is to classify different cell types on microscopy images using a supervised random forests classifier. This includes the steps of an image segmentation of a large number of images, selection of informative features and their extraction, training and application of the classifier, and evaluation of the classification accuracy on previously unseen data.

**Data**

The dataset consists of 2000 microscopy images, 500 of each on four different cell types – basophils, eosinophils, erythroblasts, and immunoglobulins. The images are separated into folders named respectively.

**Useful Matlab functions**

* rng, randperm, TreeBagger, predict, oobError, confusionchart, confusionmat

Documentation and example usages of built-in Matlab functions can be accessed with the command “doc function”.

## Tasks and grading

Maximum number of points per subtask is denoted in brackets (20 in total).

A script RUN.m should run through without any error, compute all the results, and display them without user interaction.

The code should be clean, organized, and commented to briefly and clearly describe what the code does **(2 points)**.

1. **Cells and nuclei segmentation (4 points)**

Load the images of cells from different categories and crop them to the same size. Use morphological operations, filtering, and thresholding to segment both, the whole cells and the nuclei only. The same segmentation steps should be applied to all images (there should be no adjustment to different types). Conversion from RGB to HSV might improve the segmentation.

1. **Feature Extraction (5 points)**

Extract a large number (15+) of different features to classify the different types of cells. Preview to see that the chosen features are reliable and discriminating. If not and your classification doesn’t perform well, you might need to extract more informative features. If your features are not independent (you suspect or see high level or correlation), use PCA to reduce the redundancy.

1. **Classification and feature selection (7 points)**

* Using the extracted features, train a Random Forest classifier implemented in the Matlab class TreeBagger with a flag OOBPredictorImportance='on'. To use only a subset of all datasets to train each individual tree, you can use the flag InBagFraction='your number'.
* Apply the trained classifier to test data and assess how well it performs (described in step 4). Your classifier should achieve at least 98% accuracy on test data.
* Display and compare the structure of at least three trees.
* Evaluate and interpret the importance of different extracted features using *plot(rf.OOBPermutedVarDeltaError).*
* Evaluate and interpret the impact of the number of trees using *oobError*.
* Display the out-of-bag classification error of each individual tree using the same function.
* Retrain the classifier using only the most informative features and a reduced number of trees and compare the time needed for training+prediction and the classification accuracy compared to the original classifier.

*Hint: To achieve reproducible results when running the script repeatedly, don’t forget to set the random number generator to default wherever required.*

1. **Assess the classifier performance (2 points)**

Display the confusion matrix and assess the sensitivity, specificity, and accuracy of your trained classifier with respect to correctly classifing the eosinophils.