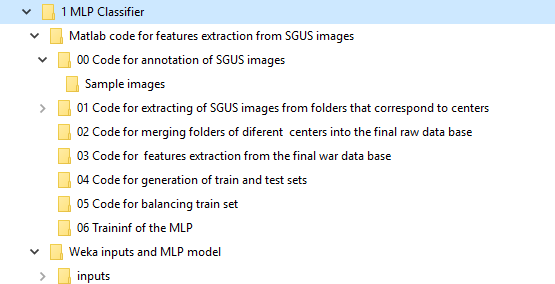
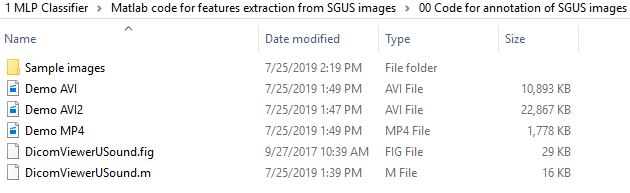
Structure of the folders



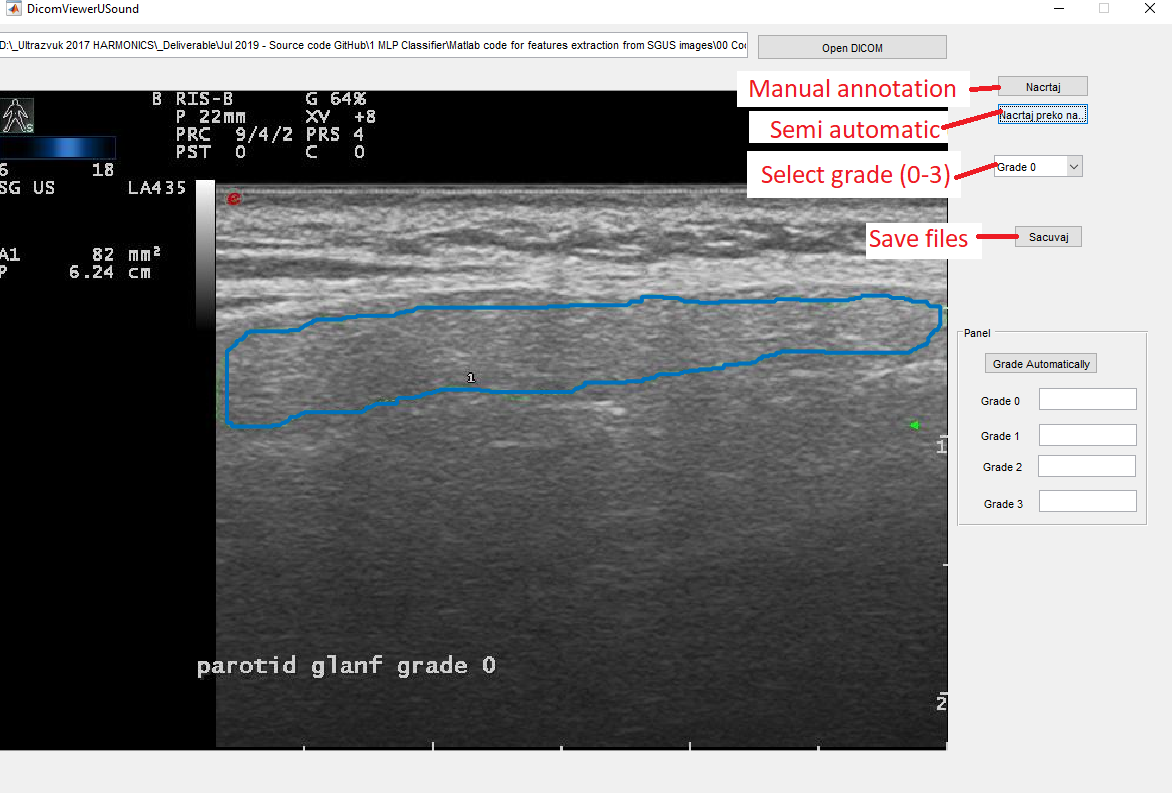
In the rest of this document key steps of each step are briefly explained.

**Step 0: Data annotation**

The video guide is provided in:

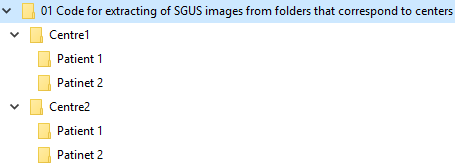


Run the “DicomViewerUSound.m” file and follow Demo video. The tool has two options: 1) Manual annotation and Semi-automatic segmentation (in this version, it is adopted to catch green and red lines present in images – they represent annotations made by clinicians); 2) Grading of SGUS images (on the scale 0-3) and 3) Saving data into local mat files for the further analysis (contour, grade and image itself)

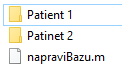
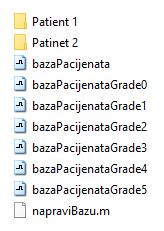


**Step 1: Extraction of SGUS images from a series of folders that correspond to various centers**

Structure of the demo folders are shown below. To centers with two patients are shown. In practice, it is important to have more data – the code assumes that there are at least one sample from each 0-3 classes. But for the demo purposes, two samples are enough assuming that we can’t share patients’ data publically.

****

For each centre, process each patient following the Step 00. After that, run the script “napraviBazu.m” within the root folder of a center.R

Runing \*.m file results with

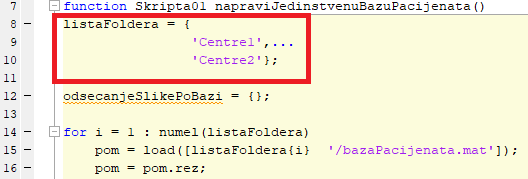
**Step 2: Merging folders of different centers into the final raw data base**

Open the folder @02 Code for merging folders of different centers into the final raw data base” and open the file “Skripta01\_napraviJedinstvenuBazuPacijenata.m”and setup names of your folders:

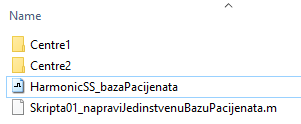
listaFoldera = {

'Centre1',...

'Centre2'};

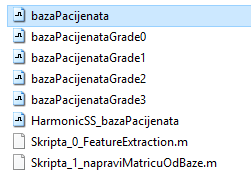


After running the script, it should create the \*.mat file “HarmonicSS\_bazaPacijenata.mat”.



**Step 3: Features extraction from the final raw data base**

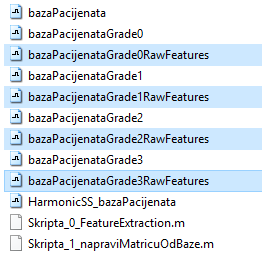
In this step we are extracting feature from previously merged database named “HarmonicSS\_bazaPacijenata”.



In the file “Skripta\_0\_FeatureExtraction.m” check that de dabase file name is correct:

nazivBaze = bazaPacijenata'; //your database name is 'HarmonicSS\_bazaPacijenata'

After running the script, the following files will be generated:



By running the script “Skripta\_1\_napraviMatricuOdBaze.m” these per-grade files will be merged into a single one called '\_baza1FeatureVsGradeZaAI.mat'.

The structure consists of the following fields:

sviAtriubti: [ number of samples × number of features double] // features matrix

SviAtributiNazivi: {1 × number of features cell} // names of each features in matrix

rezGradeLabel: [number of samples × 1 double] // pSS scores

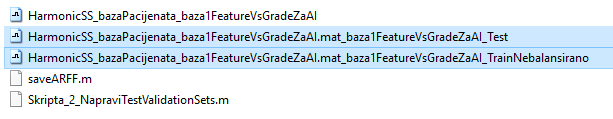
idPrimeraURawBazi: [1 × number of samples double] // Id of each sample in its centre data base

rezDataSetID: [number of samples × 1 double] // centre ID

rezDataSetName: {number of samples × 1 cell} // name of file that hold the centre data

**Step 4: Generation of randomized train and test sets**

Go to the folder “04 Code for generation of train and test sets”. Figure below shows files needed for generation of randomized train and test data sets. Open the file “Skripta\_2\_NapraviTestValidationSets.m” and run it. It uses the file “HarmonicSS\_bazaPacijenata\_baza1FeatureVsGradeZaAI” as input and generates highlighted files.

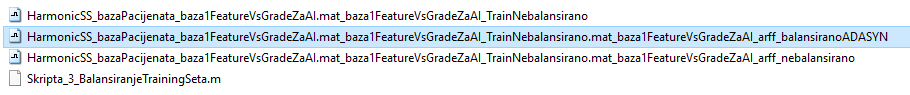
****

Train data set: “input file name”\_TrainNebalansirano (means unbalanced train set)

Test data set: “input file name”\_Test (means balanced test set)

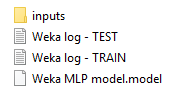
**Step 5: Balancing of the train set**

Go to the folder “05 Code for balancing train set”. Place files generated during the step 4 into this folder, to serve as inputs and run “Skripta\_3\_BalansiranjeTrainingSeta” file. It will generate highlighted file – briefly, it calls ADASYN algorithm for balancing train set.



**Step 6: 06 Traininf of the MLP by using the GA wrapper**

Navigate to the folder “06 Traininf of the MLP”:

****

Open the balanced data training data set in Weka software. Load train and test files from the “\inputs\2 Train and test set balanced using ADASYN” folder into the Weka. Use Genetic wrapper to develop your own classificatory or regression models (see this tutorial for help with Weka wrappers <https://www.youtube.com/watch?v=x5wa1w-BpRE>).