The signal processing program starts from the ’dataProcess.m’ Matlab script.

All the voice recording files were loaded into Matlab and some preprocessing and feature extraction were performed on each of them.

First, we gained 2.2s long stable data for the signal data processing. We needed to check if the audio file lasts 2.2 seconds at least, as we found some of them shorter.

A képen Betűtípus, szöveg, fehér, tervezés látható

Automatikusan generált leírás

The following recordings were shorter than 2.2 seconds.

EdemaReinke\_16\11\_\WV000073\_a.wav

Neurologica\_14\_Parte1\10-\a\_27\_11\_07 antes da deglutiçĂo\_2.wav

nodulos\_15\_Parte1\13\WV000297\_a.wav

The signal was normalized in order to each and every of them have the same scale. After that, we cut the signal into 100ms long pieces using Hamming window, which gave 22 segments.

For each segments we made signal decomposition with Discrete Wavelet Transform with Daubechies ’db1’ wavelet. In this step we decomposed the signal into 4 sub-bands to be able to extract significant information specific to human voice. These are the mel-frequency cepstral coefficients and six types of cepstral distances.

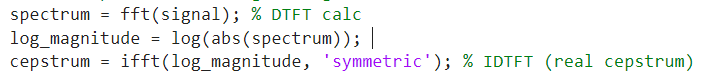
From each sub-band 13 mel-frequency cepstral coefficients were extracted by mfcc() matlab function using 20ms overlap, hamming window again and the default 40 filter-bank.

A képen szöveg, Betűtípus, képernyőkép, sor látható

Automatikusan generált leírás

At the end not only the average of these 13 mfcc features were counted, we also compressed this data again, so we got a mean value of mfcc, for every 100ms-long segments per audio file.

The cepstral coefficients were extracted for each sub-band based ont he equation from the research paper provided.



From these cepstras we computed 6 different types of cepstral distances, also based on the provided reasearch paper.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás

At the end we compressed this data by subbands, averaging by distance types. This way we received 132 mean distance values per audio file.

As the last step we computed cepstral peak-based features and computed their mean value.

A képen szöveg, Betűtípus, sor, képernyőkép látható

Automatikusan generált leírás

Finally we put together all the extracted features into one matrix. In order to map them to which audio file they belong to, we provided all rows with index numbers as the first column. Then we created a cell type of variable indicating which index corresponds to which audio file exactly.