Lecture ”Digital Signal Processing”

Prof. Dr. Dietrich Klakow, Summer Term 2021

Assignment 8

Submission deadline: 14 June 2021, 23:59

Submission Instructions:

You have one week to solve the assignments.
The code should be well structured and commented. Do not use any Matlab-Toolbox or Python external libraries if it is not mentioned that you can use them.

- You are required to hand in your solutions in a group of two students.
- There are two parts in this assignment: theoretical part and practical part.
- The practical part that is solved with Python should be submitted as an ipynb file (Jupyter Notebook or Google Colab), where every function is written in a separate block.
- The theoretical part can either be written by hand and scanned, or typed with LaTeX in the text / markdown area of ipynb file.
- Submission of both parts should be done via Microsoft Teams by one of the group members.
- Submission should be named as: Ex08_matriculationnumber1_matriculationnumber2.zip

The submission should contain the following files:

- file “README” that contains an information on all team members:
  name, matriculation number, and email address.
- code files
- file “answers.pdf” which contains answers to the questions appearing in the exercise sheet. *Note: If you use ipynb file, you don’t have to submit “answers.pdf”. You can embed your scanned copy or write your answers in the text / markdown area.*

1 MFCC Feature Extraction

In this exercise, you will learn and implement a simple feature extraction.
Allowed Python libraries: numpy, math, dct (e.g. from scipy.fftpack), matplotlib.pyplot, and soundfile to read the audio file. Matlab libraries would be the equivalent ones.

1.1 (1P) Pre-emphasis

1. Implement pre-emphasis function using the following definition:

$$f_n^{'} = f_n - \alpha f_{n-1}, \quad \text{with} \ \ \alpha = 0.95 \tag{1}$$

2. Apply the function to the audiostream *point8.au* attached in this exercise.

1.2 (1P) Framing and Windowing

1. Implement a function that does framing and windowing. The function should take as arguments: the input signal s of length k (i.e. the audiostream signal), the frame shift, and the frame width. And it should then return the n^{th} frame $M \in \mathbf{R}^{\text{width}}$. You can use Hamming window (i.e. you can use the function *hamming*).

2. How many frames can be produced from k samples?

1.3 (3P) Mel-filterbank

1. Implement a function that produces Mel-scale filterbank M with L filters (m = 1, 2, ..., L), where filter m is triangular filter given by $H_m[k]$ of equation (6.140) in the Reference attached in this exercise. (The Reference is taken from *"Spoken Language Processing - A Guide to Theory, Algorithm, and System Development"* by Huang, Acero, and Hon). *Hint: the function should take as arguments:*

    - *f_l and f_h, which are the lower and upper filter frequency, respectively,*
    - *Nfft, which is the FFT size that must be powers of 2,*
    - *Fs, which is the sampling frequency, and*
    - *L, which is the number of filters.*

    The function should return M, which is the triangular filterbank, containing L filters.

2. Use the function above to plot the filterbank with the following parameters:

    - f_l = 133 Hz
    - f_h = 6855 Hz
    - f_s = 16000 Hz
    - *fft_size* = 1024
    - L = 20

1.4 (3P) MFCC Implementation

1. Use the audiostream produced in task 1.1, the framing and windowing function of task 1.2, and the Mel-filterbank of task 1.3 to compute a MFCC-stream. You can use the function *DCT*. And for each frame, keep only DCT coefficient 1 - 12.
2. Show the original audiostream and the MFCC-stream spectrogram. You can use *imagesc* in Matlab or *imshow* in Python to show your spectrogram-matrix.
3. Why can we discard the DCT coefficient 0 in this implementation?

1.5 (2P) Dynamic Features

1. Implement a function that computes the first derivative based on the Regression Formula given in the DSP Chapter 8, slide 34.
2. Add the dynamic features of the first and the second derivatives of the cepstral coefficients onto the MFCC-stream that you have produced in task 1.4.
3. Show the overall features spectrogram.