FFT Workshop 2024
October 3rd 2024

Due date: Monday October 14th 2024, 23.59h.

Deliverables:
Submit your answers in one single ***zip-file*** to Brightspace before Monday October 14th 2024, 23.59h. The ***zip-file*** should contain:

- a pdf-file ***answers.pdf*** with title ***FFT Workshop 2024***, your ***name*** and ***student number*** and all the answers of this workshop,
- a text-file ***piano_energies.txt*** calculated for Assignment 2a of this workshop.
- a directory ***code*** with your python or jupyter code-files.

Grading: 0-10 (10% of your Final API Grade).

Assignment 1: Getting started

This is just an example using *Python 3.8*, *numpy* and *librosa* that calculates the Fourier transform of an audio file and displays its spectrogram. (Note, You may want to use a more recent version of Python.) This and many more examples can be found on https://librosa.org .

1. Make a virtualenv:

```
virtualenv fft --python=python3.8
source ./fft/bin/activate
```

2. Install some packages in this virtual environment:

```
python3.8 -m pip install --upgrade pip
python3.8 -m pip install jupyter
python3.8 -m pip install matplotlib
python3.8 -m pip install librosa
```

3. Start a Jupyter notebook.

```
jupyter notebook
```

4. Enter and run the following code in the notebook to calculate the Fourier transform of an audio file and display a spectrogram:

```
import numpy as np
import matplotlib.pyplot as plt
import librosa
import librosa.display
```

```
y, sr = librosa.load(librosa.ex('trumpet'))
D = librosa.stft(y)  # STFT of y
S_db = librosa.amplitude_to_db(np.abs(D), ref=np.max)
plt.figure()
librosa.display.specshow(S_db)
plt.colorbar()
```

Now compute the Fourier transform and depict a spectrogram of the file ***piano.wav*** which can be found in the ***audio_data.zip*** file. Add the image of the spectrogram to your ***results.pdf***. Describe what is depicted, i.e., what do the color-scheme, x-axis and y-axis mean.

Assignment 2: Feature Vectors

a) Under *'Getting started'* we showed how we can compute the Fourier transform of a sound signal. Implement a procedure that takes as input a *wav-file* (8bits resolution, 16kHz sample rate) and gives as output for every part (window) of 512 samples of the *wav-file* the energy of N=8 frequency bands (for example the following 8 frequency bands: [0Hz,1kHz), [1kHz, 2kHz), … , [7kHz, 8kHz)) using the Fourier transformed signal.
   Calculate these features for the ***piano.wav*** file which can be found in the ***audio_data.zip*** file, i.e., a list of 8-dimensional real-valued vectors results. List these vectors in the file ***piano_energies.txt*** and add them to your ***zip-file***.

b) Such features can be used to calculate consecutively for each window of samples of the ***piano.wav*** audio the following code (C_i), where:

   - C_i = **U (Up)**, if the max-energy band of the current input-window is of a higher average frequency than the average frequency of the previous input-window's max-energy band
   - C_i = **D (Down)**, if the max-energy band of the current input-window is of a lower average frequency than the average frequency of the previous window's max-energy band
   - C_i = **Rx (Repeat-x)**, otherwise (where **x** is the number of consecutive **repeats**)

   This encodes the signal as a sequence of pitch tendencies, which can be used to recognize melodies. In some respect it resembles the so-called Parsons code.
   NB Select the window size (number of bits per window) and N the number of frequency bands and the specific frequency ranges of the N bands in such a way that the features are effective for processing music played by a piano.
   Describe, justify and explain your choices in your ***answers.pdf*** file. Also add comments to your code such that it is clear how, and where in your code you implemented these choices.