

Aufgabe 2: Merkmalsextraktion II (Spektralanalyse)

In dieser Übung soll zunächst die Spektralanalyse der Merkmalsextraktion für den Sprach-
erkenner implementiert werden.

Implementierung der Spektralanalyse

Im ersten Teil der Merkmalsextraktion soll eine Spektralanalyse in Form einer Kurzzeit-
Fourier-Transformation (STFT) implementiert werden. Hierzu werden neben der Funktionen
aus Aufgabe 1 zwei weitere Funktionen benötigt, die in der Datei `feature_extraction.py`
der `recognizer` Bibliothek implementiert werden sollen.

2.1 Implementieren Sie eine Funktion

```
def compute_absolute_spectrum(frames):  
    ...
```

in der Datei `feature_extraction.py`, die aus einem Array `frames` mit Signalrahmen,
wie er von `make_frames()` zurückgegeben wird, den nicht-redundanten Teil des
Betragsspektrums als zweidimensionales Array im Datentyp `float` zurück gibt. Die
erste Arraydimension soll hierbei nicht verändert werden.

Hilfreiche Funktionen: `numpy.fft.rfft()`

2.2 Implementieren Sie eine Funktion

```
def compute_features(audio_file, window_size=25e-3, hop_size=10e-3,  
                    feature_type='STFT', n_filters=24, fbank_fmin=0,  
                    fbank_fmax=8000, num_ceps=13):  
    ...
```

in der Datei `feature_extraction.py`, die eine Audiodatei `audio_file` einliest, auf
Werte zwischen -1 und 1 normalisiert und die STFT berechnet und zurück gibt.
Nutzen Sie dazu die in den vorherigen Aufgabe implementierten Funktionen.

Die angegebenen Werte sind dabei die Default-Werte, die allerdings bei einem
Funktionsaufruf entsprechend überschrieben werden. Die Rahmenlänge `window_size`
und der Rahmenvorschub `hop_size` sind in Sekunden angegeben. Ignorieren Sie für
die aktuelle Übung zunächst die Variablen `feature_type`, `n_filters`, `fbank_fmin`,
`fbank_fmax` und `num_ceps`.

Signaldarstellung als Spektrogramm

- 2.3** Öffnen Sie die Datei `uebung2.py` und schreiben Sie hier ein Skript, welches mithilfe der zuvor implementierten Funktionen die Audiodatei `TEST-MAN-AH-3033951A.wav` im Unterordner `data` einliest und als Spektrogramm in dB darstellt. Die x -Achse sollte hierbei in Sekunden und die y -Achse in Hz angegeben werden. Fügen Sie auch eine entsprechende Achsenbeschriftung und `colorbar` hinzu. Das Ergebnis sollte in etwa so wie in Abbildung 2 aussehen. Verwenden Sie hierbei eine Rahmenlänge von 25ms und einen Rahmenvorschub von 10ms. Nutzen Sie für die Berechnung des Spektrogramms die zuvor implementierte Funktion `compute_features()`. Es ist üblich, das Spektrogramm logarithmisch darzustellen: $20 \log_{10}(\dots)$.

Hinweise: https://matplotlib.org/api/_as_gen/matplotlib.pyplot.imshow.html

Nutzen Sie `matplotlib.pyplot.show()`, damit der Plot angezeigt wird und `aspect='auto'`, innerhalb des Plot-Befehls, damit die Achsen an die veränderte Skalierung angepasst werden.

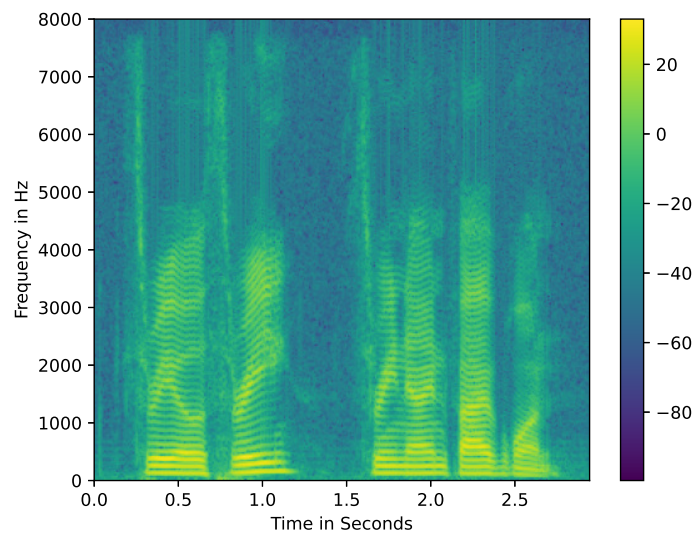


Figure 1: Spektrogramm für `TEST-MAN-AH-3033951A.wav`