Projekt zaliczeniowy

Przetwarzanie sygnałów

Izolowanie głosu ludzkiego od muzyki w nagraniu dźwiękowym

Celem projektu było stworzenie programu, który izoluje głos ludzki od muzyki w wybranej piosence. Jako przykład użyta została piosenka ‘Perfect’ Ed’a Sheeran’a. Można go użyć również do innych nagrań dźwiękowych, pod warunkiem, że posiadają rozszerzenie WAV. Przekonwertowanie piosenki na rozszerzenie WAV odbyło się na pomocą strony internetowej: https://loader.to/pl52/youtube-wav-converter.html

Program generuje spektrogramy całej piosenki, głosu i muzyki oraz trzy pliki dźwiękowe (głos, muzyka oraz ich połączenie).

Na początku programu zaimportowane zostały biblioteki (librosa,numpy, ect.) niezbędne do jego działania. W wierszu poleceń zainstalowana została librosa – biblioteka do przetwarzania dźwięku za pomocą komendy ‘conda install -c conda-forge librosa’.

```
from __future__ import print_function
import numpy as np
import matplotlib.pyplot as plt
import librosa
import librosa.display
import soundfile as sf
```

Do programu wczytana została piosenka z uwzględnieniem jej długości. W przypadku przykładowego uruchomienia programu fragment piosenki trwający minutę, został ucięty do 20 sekund. Ten parametr można modyfikować w zależności od potrzeby. Za pomocą funkcji librosa.magphase i librosa.stft wygenerowany został sygnał w domenie czasowo-częstotliwościowej przez obliczenie dyskretnych transformacji Fouriera.

```
y, sr = librosa.load('piosenka.wav', duration=20)

S_full, phase = librosa.magphase(librosa.stft(y))

idx = slice(*librosa.time_to_frames([1, 20], sr=sr))
```

Utworzony został filtr za pomocą funkcji librosa.decompose do odszumiania spektrogramu. Dla ostatecznego przekonwertowania całości użyte zostały softmaski. Klatki zostały porównane przy użyciu podobieństwa cosinusów oraz połączone biorąc pod uwagę ich częstotliwość.

```
S_filter = librosa.decompose.nn_filter(S_full,aggregate=np.median,metric='cosine',width=int(librosa.time_to_frames(2, sr=sr)))

S_filter = np.minimum(S_full, S_filter)

margin_i, margin_v = 5, 20
power = 5
mask_m = librosa.util.softmask(S_filter,margin_i * (S_full - S_filter),power=power)
mask_g = librosa.util.softmask(S_full - S_filter,margin_v * S_filter,power=power)
```

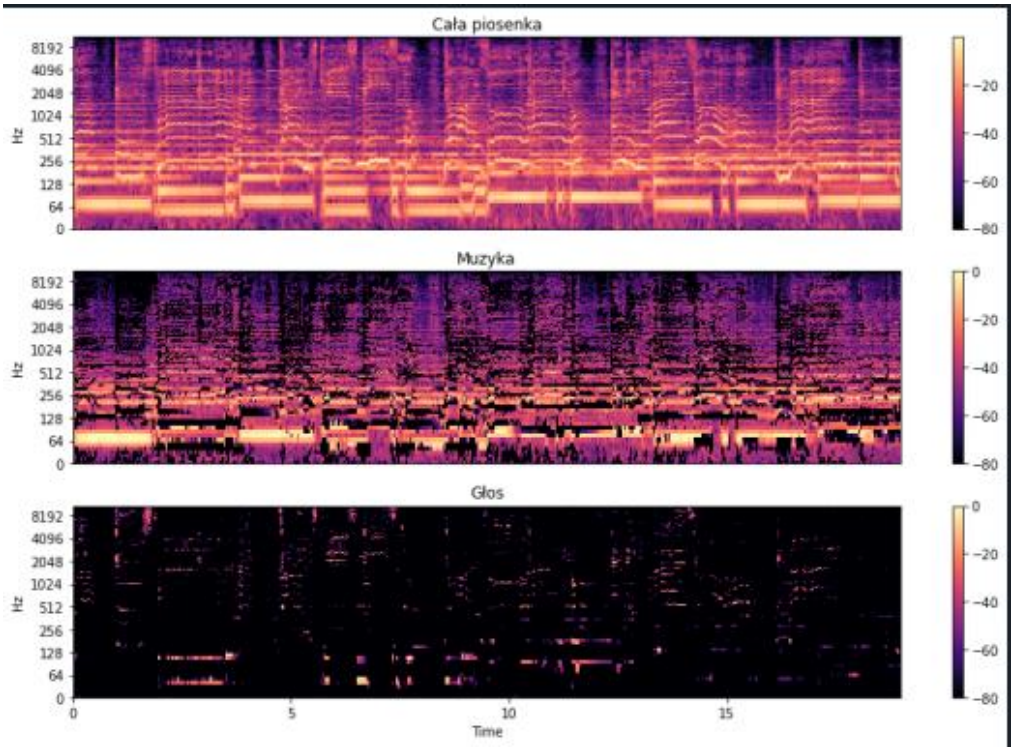
Głos został utworzony za pomocą połączenia maski z wygenerowanym sygnałem

```
glos = mask_g * S_full
muzyka = mask_m * S_full
```

W tym fragmencie wygenerowane zostały trzy spektrogramy

```
plt.figure(figsize=(12, 8))
plt.subplot(3, 1, 1)
librosa.display.specshow(librosa.amplitude_to_db(S_full[:, idx], ref=np.max),y_axis='log', sr=sr)
plt.title('Cała piosenka')
plt.colorbar()
plt.subplot(3, 1, 2)
librosa.display.specshow(librosa.amplitude_to_db(muzyka[:, idx], ref=np.max),y_axis='log', sr=sr)
plt.title('Muzyka')
plt.colorbar()
plt.subplot(3, 1, 3)
librosa.display.specshow(librosa.amplitude_to_db(glos[:, idx], ref=np.max),y_axis='log', x_axis='time', sr=sr)
plt.title('Głos')
plt.colorbar()
plt.tight_layout()
plt.show()
```

Po uruchomieniu programu wykreślone zostały 3 wykresy widma amplitudowego sygnału dla każdej chwili t.



Następnie za pomocą funkcji librosa.istft spektrogram o wartościach zespolonych został skonwertowany na szereg czasowy. Stworzone zostały 3 pliki dźwiękowe - głos, muzyka oraz ich połączenie.

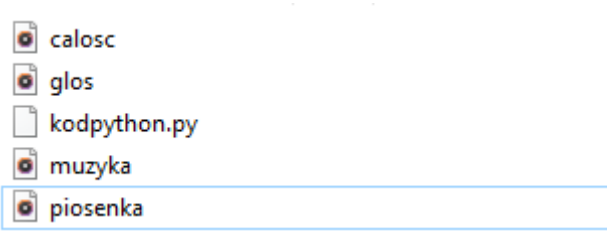
```
y_glos = librosa.istft(glos)
sf.write("glos.wav", y_glos, samplerate=sr, subtype='PCM_24')

y_muzyka = librosa.istft(muzyka)
sf.write("muzyka.wav", y_muzyka, samplerate=sr, subtype='PCM_24')

calosc = muzyka + glos
y_calosc = librosa.istft(calosc)
sf.write("calosc.wav", y_calosc, samplerate=sr, subtype='PCM_24')
```

Wnioski i ogólne uwagi:

Dla ułatwienia użytkowania programu należy utworzyć folder na pulpicie komputera oraz umieścić w nim kod programu z rozszerzeniem PY oraz piosenkę z rozszerzeniem WAV. Dzięki temu wygenerowane pliki dźwiękowe również będą zapisywać się w tym miejscu.



Analizując spektrogramy można wywnioskować, że gdyby nałożyć na siebie wykres głosu i muzyki to powstałby wykres całej piosenki.

Przy odsłuchiwaniu powstałych trzech plików dźwiękowych usłyszymy drobne niedoskonałości. Może to wynikać z braku dokładności programu, błędnemu ustaleniu parametrów oraz braku dodatkowych funkcji poprawiających jakość.

Kod programu w języku Python:

```
from __future__ import print_function

import numpy as np

import matplotlib.pyplot as plt

import librosa

import librosa.display

import soundfile as sf


y, sr = librosa.load('piosenka.wav', duration=20)
```

```
S_full, phase = librosa.magphase(librosa.stft(y))

idx = slice(*librosa.time_to_frames([1, 20], sr=sr))

S_filter = librosa.decompose.nn_filter(S_full,aggregate=np.median,metric='cosine',width=int(librosa.time_to_frames(2, sr=sr)))

S_filter = np.minimum(S_full, S_filter)

margin_i, margin_v = 5, 20
power = 5
mask_m = librosa.util.softmask(S_filter,margin_i * (S_full - S_filter),power=power)
mask_g = librosa.util.softmask(S_full - S_filter,margin_v * S_filter,power=power)

glos = mask_g * S_full
muzyka = mask_m * S_full

plt.figure(figsize=(12, 8))
plt.subplot(3, 1, 1)
librosa.display.specshow(librosa.amplitude_to_db(S_full[:, idx], ref=np.max),y_axis='log', sr=sr)
plt.title('Cała piosenka')
plt.colorbar()
plt.subplot(3, 1, 2)
librosa.display.specshow(librosa.amplitude_to_db(muzyka[:, idx], ref=np.max),y_axis='log', sr=sr)
plt.title('Muzyka')
plt.colorbar()
plt.subplot(3, 1, 3)
```

```
librosa.display.specshow(librosa.amplitude_to_db(glos[:, idx], ref=np.max),y_axis='log', x_axis='time', sr=sr)

plt.title('Głos')

plt.colorbar()

plt.tight_layout()

plt.show()



y_glos = librosa.istft(glos)

sf.write("glos.wav", y_glos, samplerate=sr, subtype='PCM_24')


y_muzyka = librosa.istft(muzyka)

sf.write("muzyka.wav", y_muzyka, samplerate=sr, subtype='PCM_24')


calosc = muzyka + glos

y_calosc = librosa.istft(calosc)

sf.write("calosc.wav", y_calosc, samplerate=sr, subtype='PCM_24')
```

Źródła:

http://librosa.org/doc/main/generated/librosa.magphase.html

https://librosa-org.translate.goog/doc/main/generated/librosa.istft.html?_x_tr_sl=en&_x_tr_tl=pl&_x_tr_hl=pl&_x_tr_pto=sc

https://librosa.org/librosa_gallery/auto_examples/plot_vocal_separation.html

https://librosa.org/doc/main/generated/librosa.griffinlim_cqt.html

https://www.programiz.com/python-programming/methods/built-in/slice

https://ichi.pro/pl/klasyfikacja-gatunkow-muzycznych-w-jezyku-python-218892206272728