

ОСОБЛИВОСТІ АНАЛІЗУ БАГАТОКАНАЛЬНИХ АУДІОСИГНАЛІВ

Вінницький національний технічний університет;

Анотація

У роботі описано розширені можливості програмного середовища мови Python при аналізі багатоканальних аудіосигналів, та їх класифікації.

Ключові слова: аналіз аудіосигналу, обробка звуку, Python, програмні засоби, класифікація, багатоканальний аудіосигнал

Abstract

The paper describes the advanced capabilities of the Python software environment in the analysis of multi-channel audio signals, and their classification.

Keywords: audio analysis, sound processing, python, program facilities, classification, multichannel audio signal

Вступ

На сьогоднішній день мова програмування *Python* надає великі можливості при розробці програмного забезпечення для великого кола напрямків та платформ. Наявність великої кількості бібліотек дозволяє програмно отримати дані будь-якого типу та формату. Апаратні та програмні засоби і системи можуть бути налаштовані за допомогою спеціально розроблених додатків [1]. Існуючі багатоканальні системи запису та відтворення аналогових та цифрових аудіосигналів потребують точного контролю, аналізу та відповідної параметризації. Більшість подібних систем не мають надійного програмно-апаратного забезпечення для контролю якості їх роботи [2].

Результати дослідження

Для роботи із середовищем мови програмування *python*, в першу чергу імпортуємо необхідні бібліотеки [3]:

```
from pyAudioAnalysis import audioBasicIO
from pyAudioAnalysis import audioFeatureExtraction
import matplotlib.pyplot as plt
[Fs, x] = audioBasicIO.readAudioFile("data/doremi.wav");
F = audioFeatureExtraction.stFeatureExtraction(x, Fs, 0.050*Fs, 0.025*Fs);
plt.subplot(2,1,1); plt.plot(F[0,:]); plt.xlabel('Frame no'); plt.ylabel('ZCR');
plt.subplot(2,1,2); plt.plot(F[1,:]); plt.xlabel('Frame no'); plt.ylabel('Energy'); plt.show()
```

Для прикладу на рисунку 1 наведено вигляд сегментації аудіофайлу для подальшої його обробки. На рисунку 2 представлено графічний вигляд сегментації аудіосигналу [3].

Даний приклад наведений при роботі з одноканальним сигналом. Також бібліотеки, такі як *Librosa* надають схожі можливості, проте не дають змоги одразу працювати із сигналом, який має більше одного потоку.

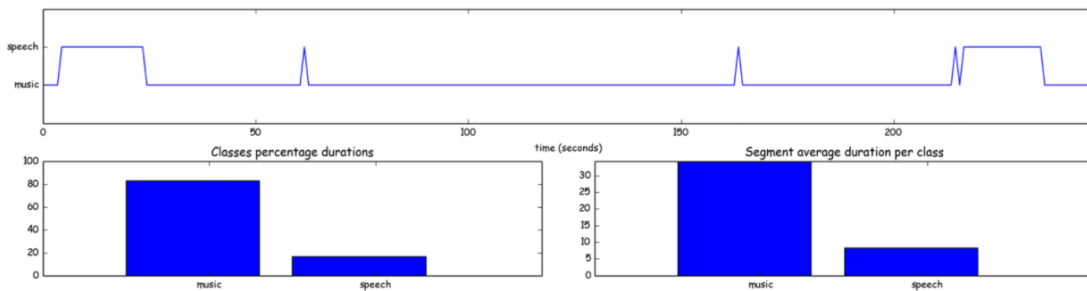


Рисунок 1– Сегментація аудіофайлу

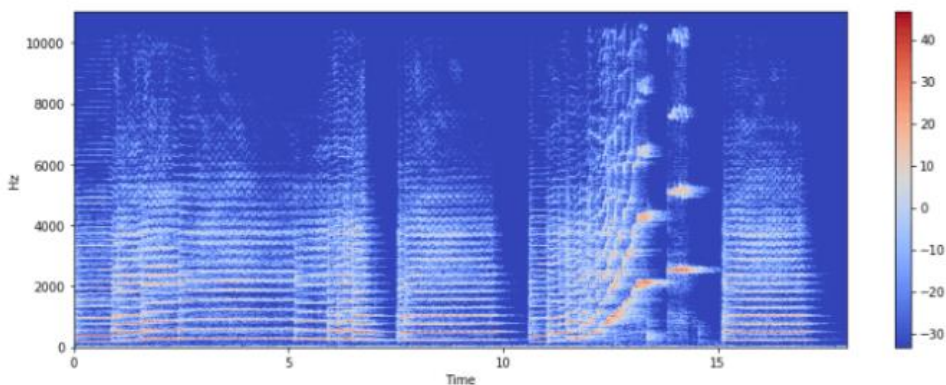


Рисунок 2 – Спектограма звукового сигналу

Багатоканальна реалізація стереосигналів виконується через 2D-масиви[6]. У наступному прикладі масив `data_wav` має два стовпці, по одному для кожного каналу. За замовчуванням, лівий канал завжди є першим, а другий - правим каналом. Аналогічно можна розробити рішення для більшої кількості каналів.

Фрагмент коду [4]:

```
# Handling stereo signals
fs_wav, data_wav = wavfile.read("data/stereo_example_small_8k.wav")
time_wav = np.arange(0, len(data_wav)) / fs_wav
plotly.offline.iplot({ "data": [go.Scatter(x=time_wav,
                                           y=data_wav[:, 0],
                                           name='left channel'),
                               go.Scatter(x=time_wav,
                                           y=data_wav[:, 1],
                                           name='right channel')])})
```

Висновки

Отже, розглянувши існуючі програмні засоби аналізу одно- канального звукового сигналу, стало можливим розробити новий підхід, який можна застосувати під час роботи із багатоканальними системами.

Враховуючи принципи роботи із одноканальним сигналом можна удосконалити вже наявні можливості, зробивши певні налаштування у розробленій програмі. Перш за все є необхідність у встановленні базових бібліотек для роботи програмного середовища, і вже потім підбираються програмні модулі для роботи саме із тим типом сигналу, який аналізується.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Python for Signal Processing: Featuring IPython Notebooks / José Unpingco / Springer; 2014 edition (October 5, 2013) 128p
2. Digital Modulations using Python / Mathuranathan Viswanathan / Independently published (December 2, 2019)
3. A to D and D to A Converter ICs [Електронний ресурс] /edited by Laroy Davis. – 2010. – Режим доступу: http://www.interfacebus.com/Analog_DAC.html.
4. Walt Kester Which ADC architecture is right for your application / Walt Kester // Analog dialogue. – 2005. – Vol. 39, № 2. – P. 11–18.

Крупельницький Леонід Віталійович - канд. техн. наук, доцент, заступник завідувача кафедри обчислювальної техніки, Вінницький національний технічний університет, Вінниця

Зайцев Микола Олександрович - аспірант, інститут інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний, Вінниця

Leonid Vitaliyovich Krupelnitskyi - PhD, assistant professor, deputy head of the department of Computer Technology, Vinnytsia National Technical University, Vinnytsia.

Mykola Oleksandrovych Zaitsev - graduate student, department of ITKI, Vinnytsia National Technical University, Vinnytsia,