

# ZPR - Dokumentacja Końcowa

Jan Górski, Konrad Pióro

20 stycznia 2025

## 1 Temat

Celem projektu jest zaimplementowanie prototypu biblioteki do analizy audio z API w Pythonie i niskopoziomowym backendem (np. Rust lub C++). Docelowo biblioteka ma mieć zbliżoną funkcjonalność do biblioteki librosa.

## 2 Architektura projektu

Projekt składa się z modułów, które są odpowiedzialne za odpowiednie grupy funkcjonalności:

- beat,
- convert,
- io,
- transform,
- mixer.

Chcieliśmy jak najbardziej uniezależnić moduły w obu językach w ten sposób, że jeśli ktoś chciałby wykorzystać biblioteki w środowisku C++, nie powinny się tu pojawić żadne przeszkody. Dlatego zamiast pisać moduły pybind wewnątrz plików bibliotecznych wydzieliliśmy je do oddzielnych plików i tworzyliśmy wrappery na te funkcje.

Tak jak w przypadku Librosy nie mamy żadnego podziału na klasy. Modularność w projekcie tworzą przestrzenie nazw. Mamy jedną globalną na cały projekt, i wewnątrz tego na mniejsze moduły, dość podobnie do samej librosy.

## 3 Zakres prac

W ramach prac nad projektem udało się zaimplementować funkcjonalności związane z:

- pracą na plikach
- obliczaniem parametrów sygnału
- konwersją wielkości opisujących sygnał
- obliczaniem spektrogramów.
- przekształcanie sygnału.

## 4 Nieosiągnięte cele

Nasz zespół zaimplementował wszystkie funkcjonalności przewidziane w pierwotnych koncepcjach projektu.

## 5 Zmiany w stosunku do pierwotnej wersji

- W finalnej wersji projektu nie znalazły się programy demonstracyjne w *C++*, która miała służyć demonstracji zaimplementowanych funkcjonalności. Zamiast tego postanowiliśmy podać przykłady generowania wykresów z użyciem naszej biblioteki w porównaniu do użycia *librosa*.
- Zmieniła się także konwencja testowania, poza testami jednostkowymi porównujemy także analogiczne funkcje z obu bibliotek.
- nie wykorzystaliśmy biblioteki *libmfcc*, algorytm udało się zaimplementować bez jej użycia

## 6 Mocne strony

Poszczególne funkcje zaimplementowane przez nasz zespół wykonują się szybciej niż ich analogiczne odpowiedniki z biblioteki *librosa*.

## 7 Napotkane problemy

- W październiku wyszła nowa wersja 3.13 Pythona. Co nasz zaskoczyło to częste komunikaty o tym, że jakaś biblioteka jest niekompatybilna z tą wersją Pythona i wskazanym jest używanie wersji 3.12. Z samą *librosa* były problemy aby ją zainstalować.

## 8 Co można poprawić?

- W projekcie operujemy na wielu zbiorach danych. Gdybyśmy mieli zacząć od nowa, inaczej byśmy operowali na tych zbiorach. Zamiast używać wszędzie kontenera w rodzaju wektora,

możnaby zamiast tego użyć stworzonych do tego narzędzi z różnych bibliotek matematycznych. Przyjęlibyśmy również konwencję o jak najmniejszym tworzeniu obiektów i bardziej działalibyśmy poprzez modyfikowanie już zainicjowanych zmiennych.