

Aplikacja do analizy dźwięku w dziedzinie częstotliwości

Analiza i Przetwarzanie Dźwięku

Natalia Choszczyk

Politechnika Warszawska

Wydział Matematyki i Nauk Informatycznych

21 kwietnia 2025

Spis treści

1. Wprowadzenie	3
2. Interfejs i funkcjonalności aplikacji	4
3. Opis parametrów częstotliwościowych	9
3.1. Volume (Głośność)	9
3.2. Frequency Centroid (Środek widmowy)	9
3.3. Bandwidth (Szerokość pasma)	9
3.4. ERSB (Energy Ratio Subband)	10
3.5. Spectral Flatness (SFM)	10
3.6. Spectral Crest Factor (SCF)	10
4. Funkcje okienkowe i analiza widmowa	12
4.1. Transformata Fouriera (FFT)	12
4.2. Funkcje okienkowe dostępne w aplikacji	13
4.2.1. Okno prostokątne	13
4.2.2. Okno trójkątne	13
4.2.3. Okno Hamminga	13
4.2.4. Okno van Hanna	13
4.2.5. Okno Blackmana	14
4.2.6. Okno Gaussowskie	14
4.3. Podsumowanie wpływu okna na widmo	14
5. Spektrogram	15
6. Częstotliwość podstawowa F0 – metoda cepstralna	17
7. Struktura aplikacji i implementacja	19
7.1. Plik app_frequency.py	19
7.2. Plik frequency_features.py	19
7.3. Plik frequency_plots.py	19
8. Wnioski	21
9. Bibliografia	25

1. Wprowadzenie

Projekt polegał na stworzeniu aplikacji umożliwiającej analizę sygnału dźwiękowego w dziedzinie częstotliwości. Aplikacja została zaimplementowana w języku **Python**, z wykorzystaniem biblioteki **Streamlit** jako GUI. Projekt bazuje na poprzednim projekcie, zawiera na przykład wykres waveform, ale zawiera również nieco inne funkcjonalności:

- obliczanie i rysowanie wykresów parametrów częstotliwościowych w ramach,
- analizę wpływu funkcji okienkowych (windowing) na widmo sygnału oraz wykres czasowy,
- rysowanie widma sygnału z wykorzystaniem FFT,
- samodzielne generowanie spektrogramu,
- obliczanie częstotliwości podstawowej (F0) metodą cepstralną.

Wykorzystane biblioteki

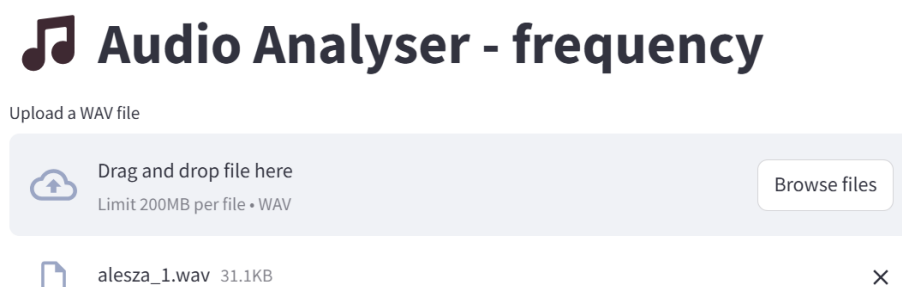
- **Streamlit** – budowa interfejsu graficznego użytkownika, strona internetowa aplikacji.
- **Librosa** – wczytywanie plików WAV, obliczanie STFT, częstotliwości widmowych, wektorów czasowych.
- **NumPy** – obliczenia macierzowe, implementacja algorytmów FFT, statystyki sygnału.
- **Plotly** – interaktywne wykresy (waveform, widmo, cechy częstotliwościowe, spektrogram, F0).

2. Interfejs i funkcjonalności aplikacji

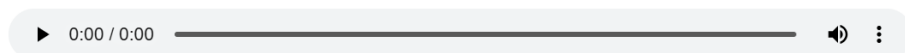
Po uruchomieniu aplikacji, użytkownik ma możliwość:

1. Załadowania pliku WAV - Rysunek 2.1.
2. Odtworzenia sygnału i sprawdzenia jego przebiegu czasowego - Rysunek 2.2.
3. Wyboru cechy częstotliwościowej i rysowania jej wykresu - Rysunek 2.3.
4. Analizy ramki sygnału z oraz bez zastosowania funkcji okienkowej - - Rysunek 2.4 oraz 2.5.
5. Wygenerowania spektrogramu z parametrami: długość ramki, nakładanie, typ okna - Rysunek 2.6.
6. Obliczenia częstotliwości podstawowej (F_0) za pomocą metody cepstralnej - Rysunek 2.7 i 2.8.

Całość opiera się na modularnej strukturze kodu z podziałem na plik główny (`app_frequency.py`) oraz dwa pliki pomocnicze: `frequency_features.py` i `frequency_plots.py` - odpowiednio dla wyliczania parametrów i dla rysowania wykresów.

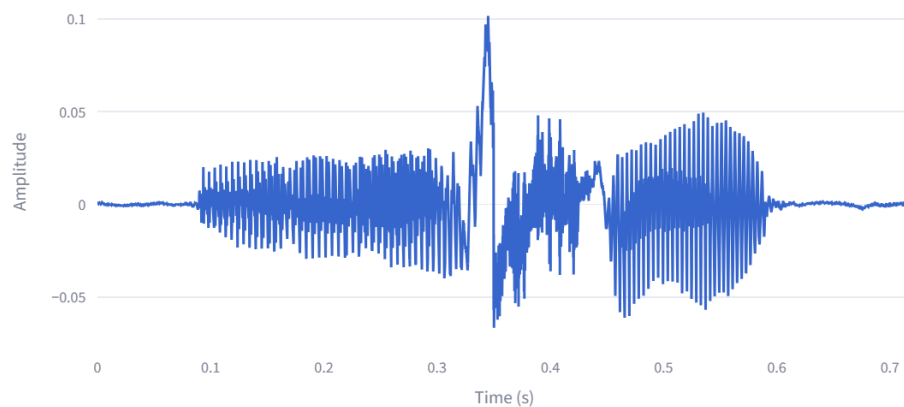


Rysunek 2.1: Załadowanie pliku WAV



Waveform Plot

Time Plot



Rysunek 2.2: Odtworzenie sygnału i sprawdzenie jego przebiegu czasowego

Frequency Features

Select frequency-domain feature to display

Volume

Volume

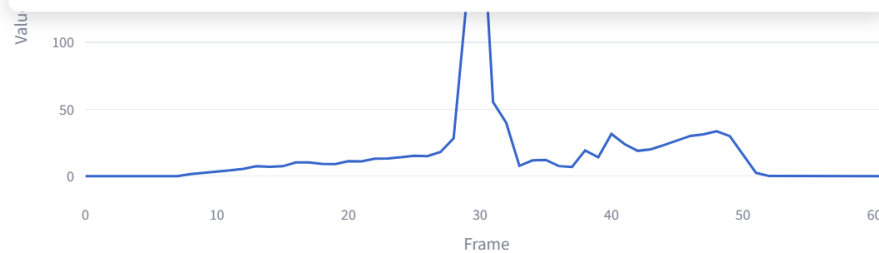
Frequency Centroid

Bandwidth

ERSB

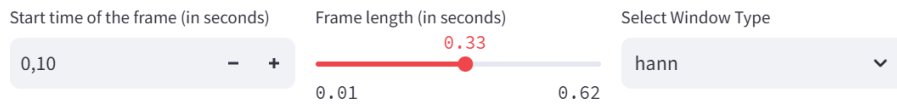
Spectral Flatness

Spectral Crest

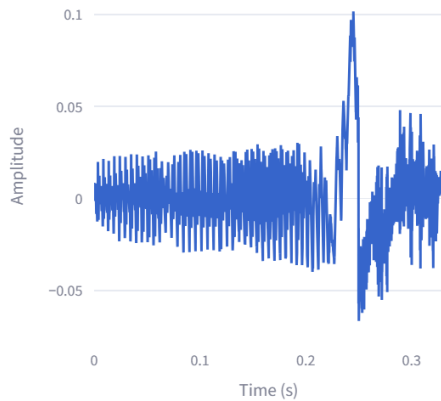


Rysunek 2.3: Rysowanie wykresu parametrów

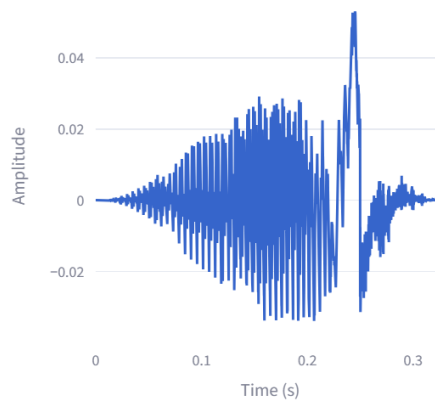
FFT and Windowing



Time (No Window)

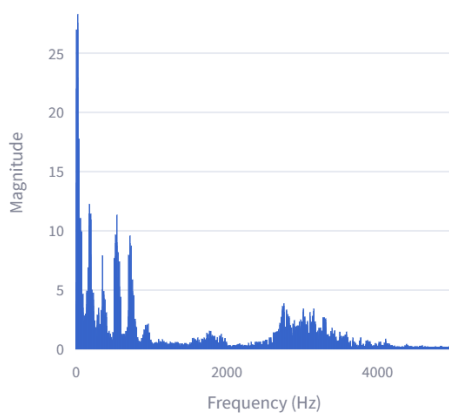


Time (With Window)

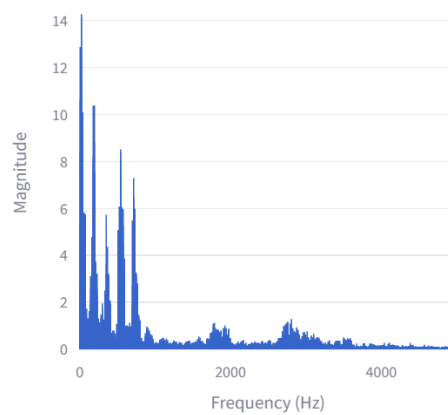


Rysunek 2.4: Ustawienie parametrów oraz wyświetlenie wykresów przebiegu czasowego audio z zastosowaniem funkcji okienkowej lub bez niej

FFT (No Window)

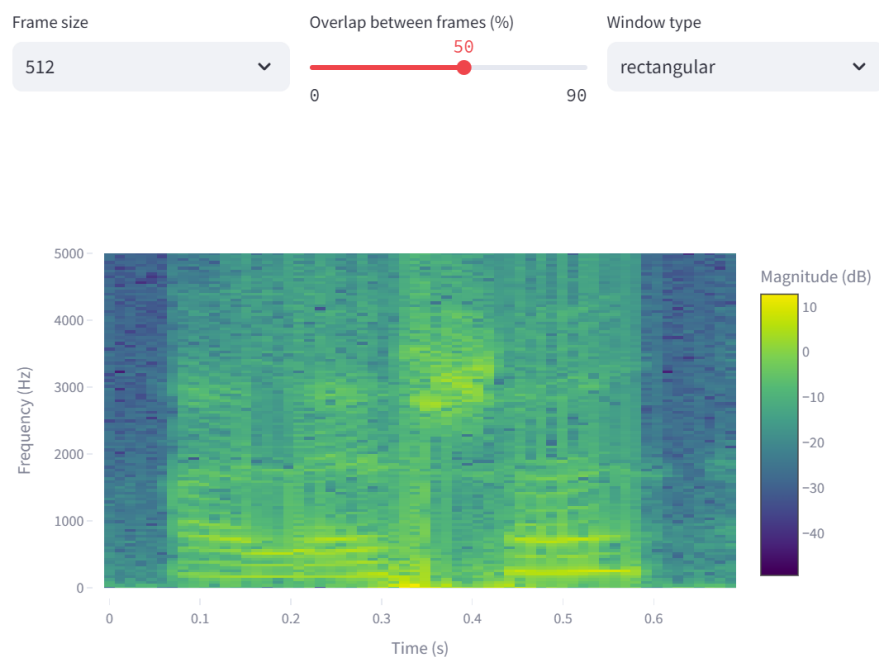


FFT (With Window)



Rysunek 2.5: Wyświetlenie wykresów FFT z zastosowaniem funkcji okienkowej lub bez niej

Spectrogram

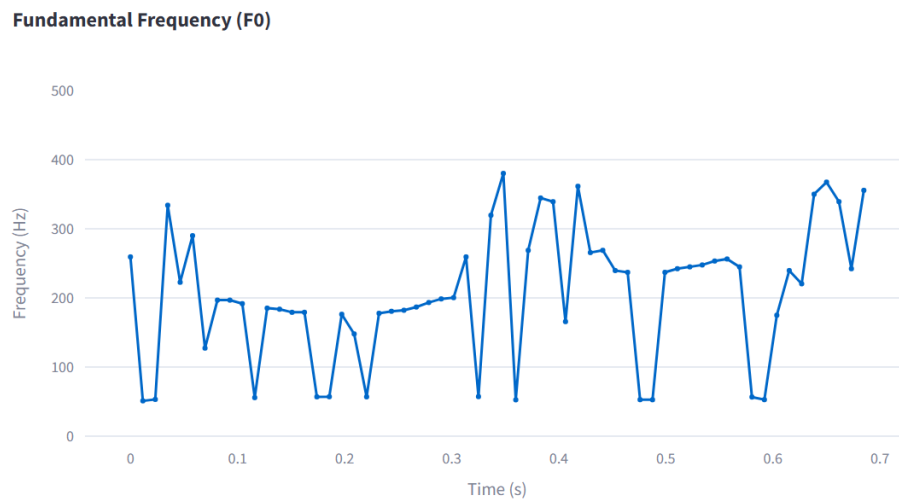


Rysunek 2.6: Wyświetlenie spektrogramu

Fundamental Frequency (Cepstrum Method)



Rysunek 2.7: Wyświetlenie cepstrum



Rysunek 2.8: Wyświetlenie F0 wyznaczonego na podstawie cepstrum

3. Opis parametrów częstotliwościowych

Wszystkie cechy są obliczane dla kolejnych ramek sygnału. Wykorzystywana jest transformata Fouriera (STFT), której wynik pozwala uzyskać widmo mocy i gęstość widmową sygnału. Cechy są następnie obliczane na podstawie wektorów amplitud i energii.

3.1. Volume (Głośność)

Głośność ramki obliczana jest jako suma kwadratów wartości amplitudowych widma:

$$Vol(n) = \frac{1}{N} \sum_{k=0}^{N-1} |S_n(k)|^2 \quad (3.1)$$

gdzie $S_n(k)$ to spektrum ramki n , a N to rozmiar ramki. Parametr ten reprezentuje energię sygnału.

3.2. Frequency Centroid (Środek widmowy)

Reprezentuje środek ciężkości rozkładu widmowego:

$$FC(n) = \frac{\sum_{\omega} \omega \cdot |S_n(\omega)|^2}{\sum_{\omega} |S_n(\omega)|^2} \quad (3.2)$$

Wartość ta koreluje z jasnością sygnału (brzmienie jasne vs. ciemne) oraz z cechą ZCR. W muzyce oraz mowie pomaga rozróżniać dźwięki dźwięczne i bezdźwięczne.

3.3. Bandwidth (Szerokość pasma)

Miara rozrzutu energii wokół środka ciężkości (frequency centroid):

$$BW(n) = \sqrt{\frac{\sum_{\omega} (\omega - FC(n))^2 \cdot |S_n(\omega)|^2}{\sum_{\omega} |S_n(\omega)|^2}} \quad (3.3)$$

3.4. ERSB (Energy Ratio Subband)

Podział spektrum na trzy przedziały częstotliwości:

- **ERSB1:** 0–630 Hz
- **ERSB2:** 630–1720 Hz
- **ERSB3:** 1720–4400 Hz

Dla każdego z przedziałów obliczane są energie pasmowe i wyznaczany jest ich stosunek do całkowitej energii:

$$ERSB_i(n) = \frac{\sum_{f_{low}}^{f_{high}} |S_n(f)|^2}{Vol(n)} \quad (3.4)$$

3.5. Spectral Flatness (SFM)

Miara "szumowości" sygnału. Obliczana jako stosunek średniej geometrycznej do średniej arytmetycznej wartości widmowych:

$$SFM(n) = \frac{\left(\prod_{k=0}^{N-1} |S_n(k)| \right)^{1/N}}{\frac{1}{N} \sum_{k=0}^{N-1} |S_n(k)|} \quad (3.5)$$

Wartość bliska 1 oznacza szum biały, wartość bliska 0 oznacza ton tonalny (np. sygnał sinusoidalny).

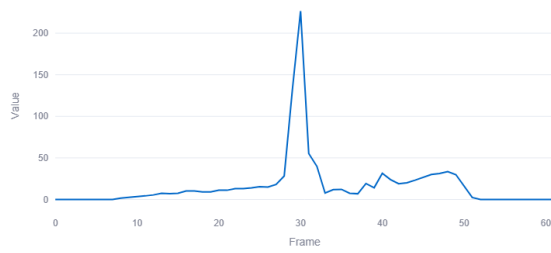
3.6. Spectral Crest Factor (SCF)

Miara stosunku wartości maksymalnej do średniej amplitudy:

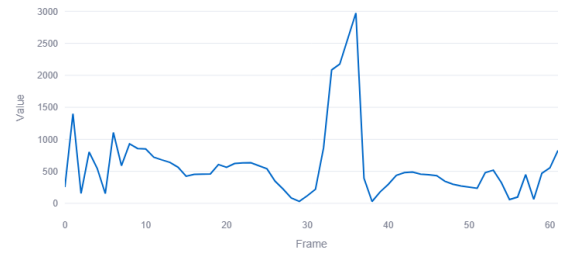
$$SCF(n) = \frac{\max_k |S_n(k)|^2}{\frac{1}{N} \sum_{k=0}^{N-1} |S_n(k)|^2} \quad (3.6)$$

Im większy SCF, tym bardziej pikowa natura widma — co może świadczyć o dźwiękach tonalnych.

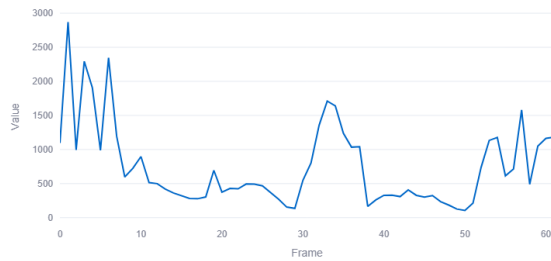
3.6. SPECTRAL CREST FACTOR (SCF)



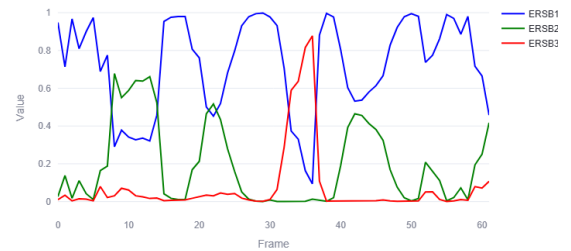
Rysunek 3.1: Przykładowy wykres Volume



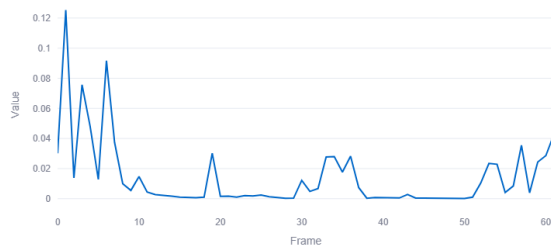
Rysunek 3.2: Przykładowy wykres FC



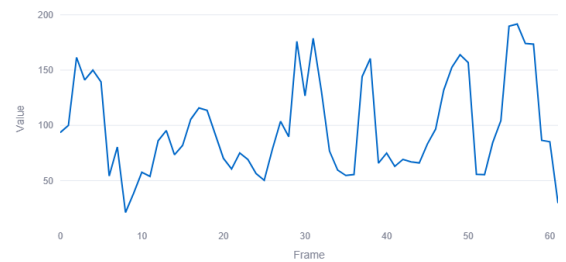
Rysunek 3.3: Przykładowy wykres Bandwidth



Rysunek 3.4: Przykładowy wykres ERSB



Rysunek 3.5: Przykładowy wykres SFM



Rysunek 3.6: Przykładowy wykres SCF

4. Funkcje okienkowe i analiza widmowa

W celu ograniczenia zjawiska wycieku widma (ang. *spectral leakage*) w analizie lokalnych fragmentów sygnału stosuje się funkcje okienkowe. Okno modyfikuje próbki sygnału przed wykonaniem transformacji Fouriera, wygładzając przejścia na początku i końcu ramki.

Użytkownik aplikacji może wybrać:

- moment rozpoczęcia analizowanej ramki (w sekundach),
- długość tej ramki,
- typ zastosowanej funkcji okienkowej.

Po dokonaniu wyboru generowane są cztery wykresy:

1. sygnał w dziedzinie czasu bez funkcji okna,
2. sygnał w dziedzinie czasu z zastosowanym oknem,
3. widmo częstotliwościowe (FFT) bez funkcji okna,
4. widmo częstotliwościowe (FFT) z funkcją okna.

4.1. Transformata Fouriera (FFT)

Transformata Fouriera pozwala przejść z dziedziny czasu do dziedziny częstotliwości. Służy do rozkładu sygnału na jego składowe harmoniczne – sinusoidy o różnych częstotliwościach i amplitudach.

Nieciągły sygnał dyskretny $x[n]$ zamieniamy na widmo $X[k]$ za pomocą szybkiej transformaty Fouriera (ang. *Fast Fourier Transform, FFT*):

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1 \quad (4.1)$$

W aplikacji:

- użytkownik wybiera interesującą go ramkę (czas startu + długość),

4.2. FUNKCJE OKIENKOWE DOSTĘPNE W APLIKACJI

- dla tej ramki wykonywane są dwa warianty FFT: bez okna oraz z zastosowanym wybranym oknem,
- obliczana jest część dodatnia transformaty ($0 \div f_s/2$) – z użyciem `np.fft.rfft`,
- wyniki przedstawiane są jako wykres amplitudy widma w funkcji częstotliwości.

Widmo wyświetlane jest tylko do 5000 Hz, aby uwidocznienie najistotniejsze cechy sygnału i poprawić czytelność.

4.2. Funkcje okienkowe dostępne w aplikacji

4.2.1. Okno prostokątne

$$w(n) = 1, \quad 0 \leq n < N$$

Najprostsze – nie modyfikuje sygnału.

4.2.2. Okno trójkątne

$$w(n) = 1 - \left| \frac{n - (N - 1)/2}{(N + 1)/2} \right|$$

Zapewnia łagodne przejścia na początku i końcu ramki, przez co zmniejsza wyciek widma.

4.2.3. Okno Hamminga

$$w(n) = 0.54 - 0.46 \cos \left(\frac{2\pi n}{N - 1} \right)$$

Popularne w analizie mowy, zapewnia dobre tłumienie bocznych listków widma.

4.2.4. Okno van Hanna

$$w(n) = 0.5 \left(1 - \cos \left(\frac{2\pi n}{N - 1} \right) \right)$$

Zapewnia lepsze tłumienie głównego piksu niż okno Hamminga, ale gorsze bocznych listków.

4.2.5. Okno Blackmana

$$w(n) = 0.42 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right) + 0.08 \cos\left(\frac{4\pi n}{N-1}\right)$$

Bardzo dobre tłumienie bocznych pętli widma, ale powoduje spadek rozdzielczości widmowej (szeroki pik główny).

4.2.6. Okno Gaussowskie

$$w(n) = \exp\left(-0.5 \cdot \left(\frac{n - (N-1)/2}{\sigma(N-1)/2}\right)^2\right)$$

Kształt funkcji zależy od parametru σ . Dla dużych wartości zbliża się do okna prostokątnego, dla małych – silne tłumienie.

4.3. Podsumowanie wpływu okna na widmo

- Brak okna (prostokątne) – wąski pik główny, ale bardzo silne boczne listki.
- Okna wygładzające (Hann, Hamming, Blackman) – zmniejszają boczne listki, ale rozszerzają pik główny.
- Wybór okna jest kompromisem między rozdzielczością a tłumieniem zakłóceń.

5. Spektrogram

Spektrogram jest wizualną reprezentacją widma częstotliwościowego sygnału dźwiękowego, analizowanego w czasie. Jest to wykres trójwymiarowy, w którym:

- oś X reprezentuje czas,
- oś Y – częstotliwość,
- wartość (kolor) – amplitudę danej częstotliwości w danym czasie.

Spektrogram pokazuje, jak zmienia się widmo sygnału w czasie, dzięki czemu możliwa jest analiza zjawisk dynamicznych – np. artykulacji głosek, zmian barwy dźwięku, intonacji w mowie, elementów muzyki.

Obliczanie spektrogramu

Spektrogram wyznaczany jest na podstawie przesuwanych ramek sygnału $x(t)$, dla których wykonuje się transformatę Fouriera:

$$Spec(f, t) = |\mathcal{F}\{x(t) \cdot w(t - t_0)\}|$$

Gdzie:

- $x(t)$ – analizowany sygnał,
- $w(t)$ – funkcja okienkowa,
- t_0 – środek aktualnej ramki,
- $\mathcal{F}\{\cdot\}$ – transformata Fouriera.

Każda ramka analizowana jest oddzielnie – wynikiem jest macierz o wymiarach (częstotliwości \times czas).

Wartości spektrogramu są następnie logarytmowane (ze względu na bardzo duży zakres amplitud):

$$z_{\text{dB}} = 10 \cdot \log_{10}(|\text{Spec}(f, t)|^2 + \epsilon)$$

Gdzie ϵ to bardzo mała liczba dodana w celu uniknięcia logarytmu z zera.

Możliwości użytkownika w aplikacji

Użytkownik może dostosować:

- **długość ramki** (np. 256, 512, 1024, 2048 próbek),
- **procent nakładania ramek** – overlap od 0% do 90%,
- **typ funkcji okna** – wszystkie wcześniej opisane.

Wizualizacja realizowana jest jako interaktywny wykres typu **heatmap** (kolorowa mapa amplitud). Zastosowanie różnych okien oraz długości ramek wpływa na rozdzielczość spektrogramu:

- krótsze ramki \rightarrow lepsza rozdzielczość czasowa, gorsza częstotliwościowa,
- dłuższe ramki \rightarrow odwrotnie.

6. Częstotliwość podstawowa F0 – metoda cepstralna

Częstotliwość podstawowa (F0) to najniższa składowa harmoniczna występująca w sygnale okresowym. W projekcie do estymacji F0 zastosowano metodę **cepstralną**, która opiera się na analizie struktury widma sygnału.

Metoda cepstralna wykorzystuje fakt, że sygnał zawierający okresowe harmoniczne (czyli dźwięczny) ma regularnie rozmieszczone piki w widmie logarytmicznym – co przekłada się na lokalne maksimum w tzw. quefreny, czyli pseudoczasie.

Analiza lokalna – wykres F0 w czasie

Aby śledzić zmiany częstotliwości podstawowej w czasie, sygnał dzielony jest na krótkie, częściowo nachodzące na siebie ramki. Dla każdej ramki obliczane jest cepstrum, a następnie wykrywane jest maksimum w zakresie quefreny odpowiadającym spodziewanym częstotliwościom F0 (np. 50–400 Hz). Na tej podstawie szacowana jest wartość tonu podstawowego dla danej chwili czasowej. Wynikiem jest wykres F0 w funkcji czasu.

Analiza globalna – cepstrum całego sygnału

Dodatkowo, aplikacja generuje wykres pełnego cepstrum dla całego sygnału (nie tylko jednej ramki). Wykres przedstawia zależność amplitudy od quefreny – czyli pseudoczasu wyrażonego w sekundach.

Pik cepstrum jest szukany w zakresie częstotliwości podanym przez użytkownika.

Parametry konfigurowane przez użytkownika

Użytkownik może ustawić następujące parametry:

- długość ramki do estymacji F0 (np. 256, 512, 1024 próbek),

- stopień nakładania ramek (overlap),
- minimalna i maksymalna wartość F0 (domyślnie 50–400 Hz),

7. Struktura aplikacji i implementacja

7.1. Plik `app_frequency.py`

Zawiera logikę interfejsu graficznego. Obsługuje:

- ładowanie pliku,
- prezentację przebiegu czasowego,
- wybór cechy częstotliwościowej,
- analizę ramki i widma FFT,
- generowanie spektrogramu,
- obliczanie cepstrum oraz F0.

7.2. Plik `frequency_features.py`

Zawiera:

- funkcje do obliczania cech: Volume, FC, BW, ERSB, SFM, SCF,
- własne funkcje okienkowe,
- implementację spektrogramu (FFT dla ramek),
- wyliczanie cepstrum oraz detekcję F0 metodą cepstralną.

7.3. Plik `frequency_plots.py`

Zawiera funkcje do tworzenia wykresów:

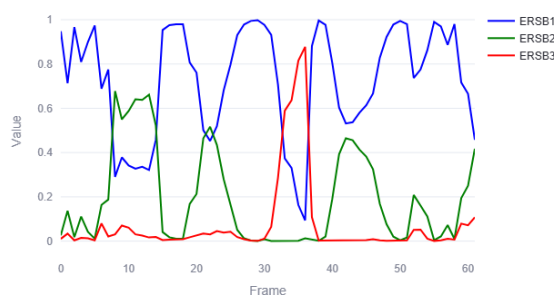
- waveform (wykres w dziedzinie czasu),

- widmo FFT,
- cechy częstotliwościowe (liniowe, wieloliniowe – ERSB),
- spektrogram (heatmap),
- cepstrum oraz F0 w czasie.

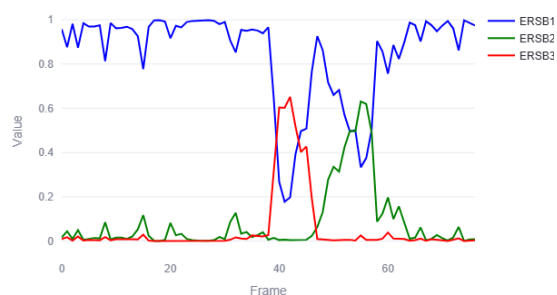
8. Wnioski

Porównanie głosu żeńskiego i męskiego

Przy porównaniu tego samego dźwięku, ale w przypadku głosu męskiego oraz żeńskiego, w większości parametrów nie widać znaczących, zauważalnych różnic. Jedynie w przypadku parametru ERSB widać dużą zależność. W przypadku głosu męskiego, ERSB3 jest często na wyższym poziomie, tak samo ERSB2. W przypadku głosu żeńskiego przeważa ERSB1.



Rysunek 8.1: Parametr ERSB dla głosu męskiego

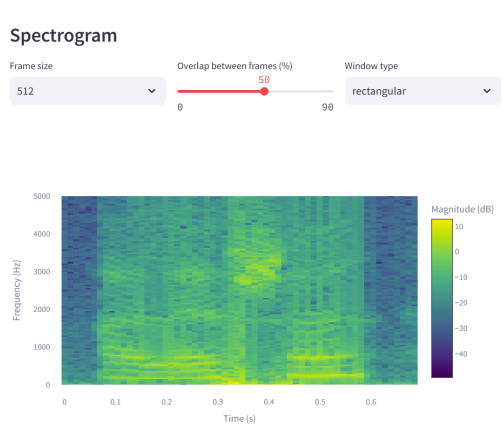


Rysunek 8.2: Parametr ERSB dla głosu żeńskiego

Porównanie parametrów dla spółgłosek i samogłosek

Porównałam parametry dla wyrazu "uwo", gdzie bardzo wyraźnie widać, które elementy na wykresie czasowego przebiegu audio odpowiadają poszczególnym literom, a więc zarówno spółgłoskom jak i samogłoskom.

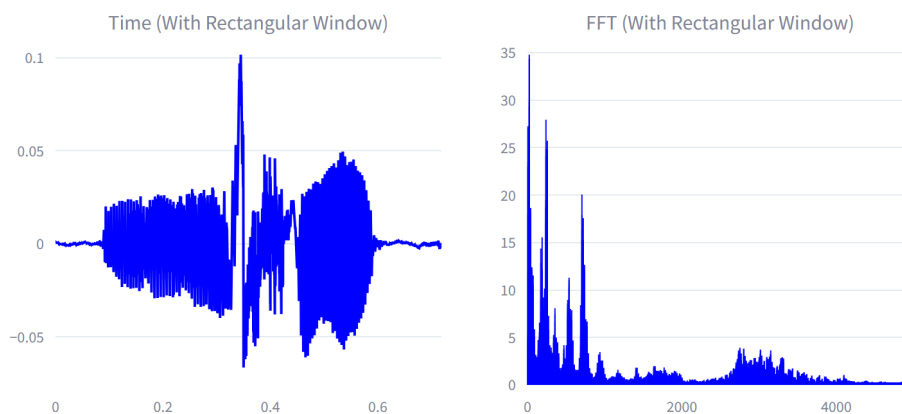
Na wykresie Frequency Centroid możemy wyraźnie zauważyć, że dla główki "w" wartość jest zdecydowanie najwyższa. Dla głoski "o" jest największe występowanie ERSB2, gdzie dla innych głosek znacząco przeważa ERSB1. Na spektrogramie jest zauważalny ciemniejszy pas dla głoski "w".



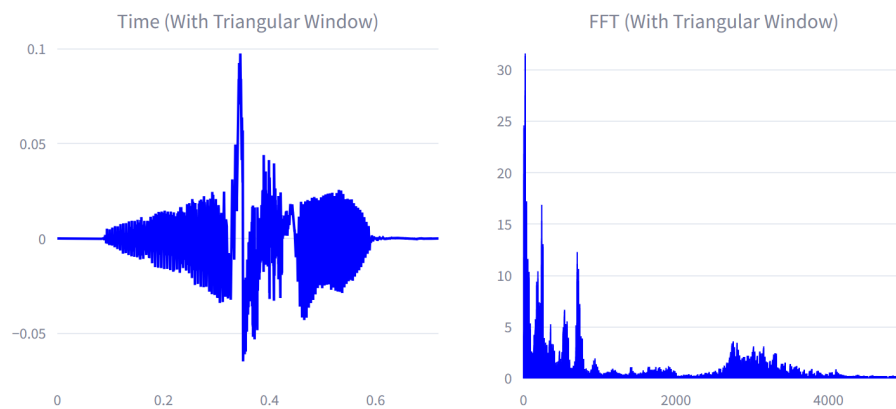
Rysunek 8.3: Wykres FC dla wyrazu "uwo"

Wpływ funkcji okienkowych

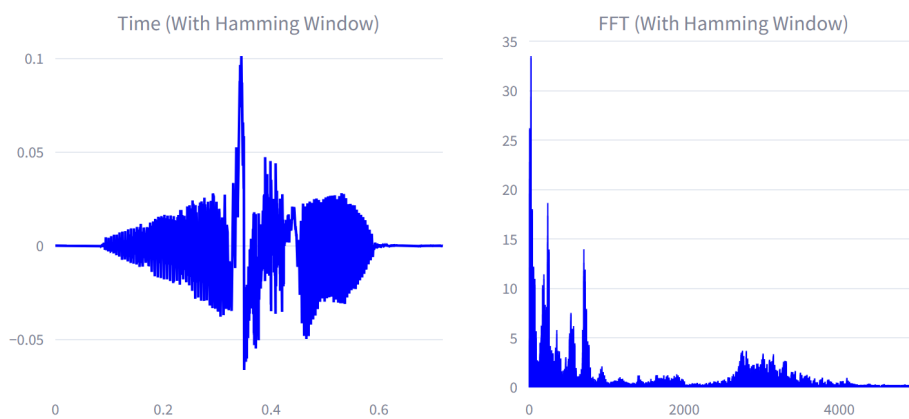
Działanie funkcji okienkowych można bardzo wyraźnie zaobserwować na wykresie przebiegu czasowego audio oraz wykresie FFT. Poniżej znajdują się wykresy dla kolejnych funkcji okienkowych. Wybrana funkcja okienkowa wpływa również na wygląd spektrogramu.



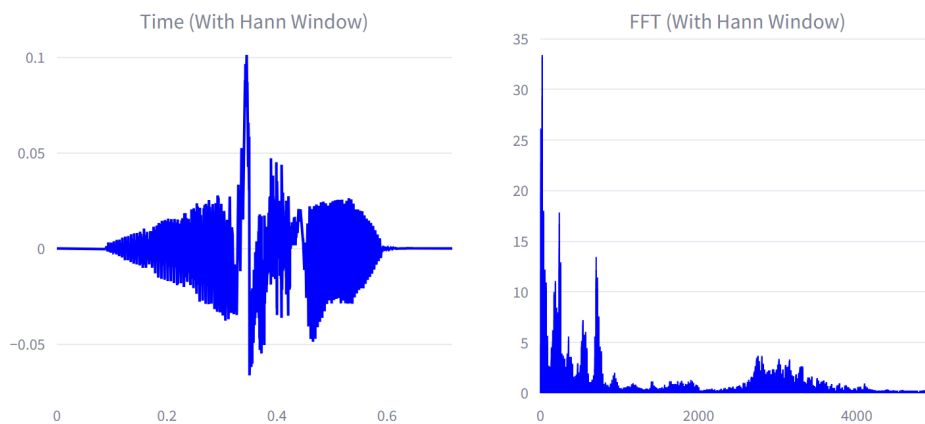
Rysunek 8.4: Okno prostokątne - bez zmian



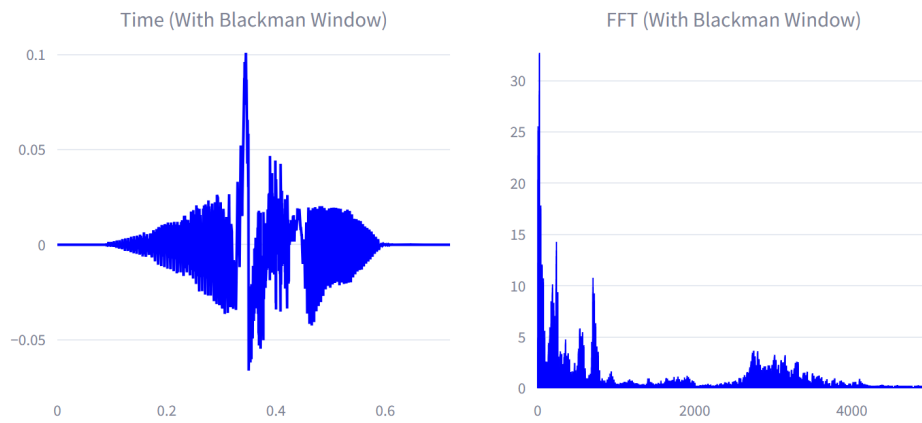
Rysunek 8.5: Okno trójkątne



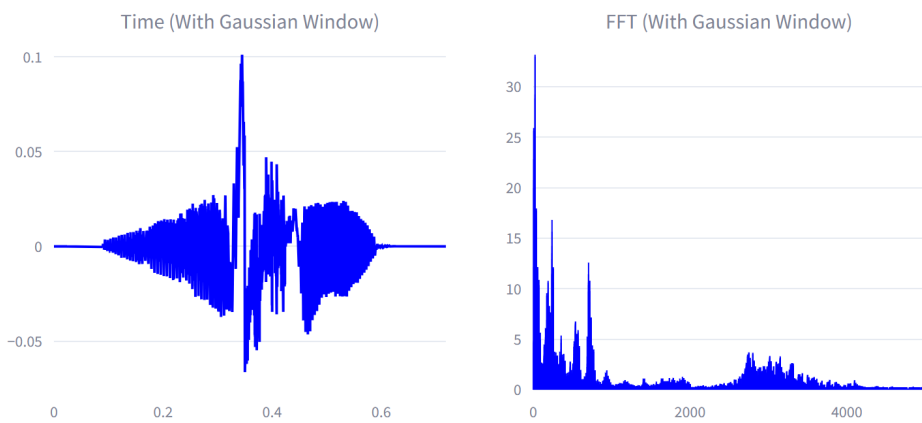
Rysunek 8.6: Okno Hamminga



Rysunek 8.7: Okno van Hanna



Rysunek 8.8: Okno Blackmana



Rysunek 8.9: Okno gaussowskie

9. Bibliografia

📖 Materiały z kursu „Analiza i Przetwarzanie Dźwięku”

📖 <https://librosa.org/>

📖 <https://docs.streamlit.io/>

📖 <https://plotly.com/python/>

📖 <https://en.wikipedia.org/wiki/Cepstrum>

📖 <https://en.wikipedia.org/wiki/Spectrogram>

📖 ChatGPT