Analiza i Przetwarzanie Dźwięku Projekt 1 - Cechy sygnału audio w dziedzinie czasu

Karolina Dunal

29 marca 2025

Spis treści

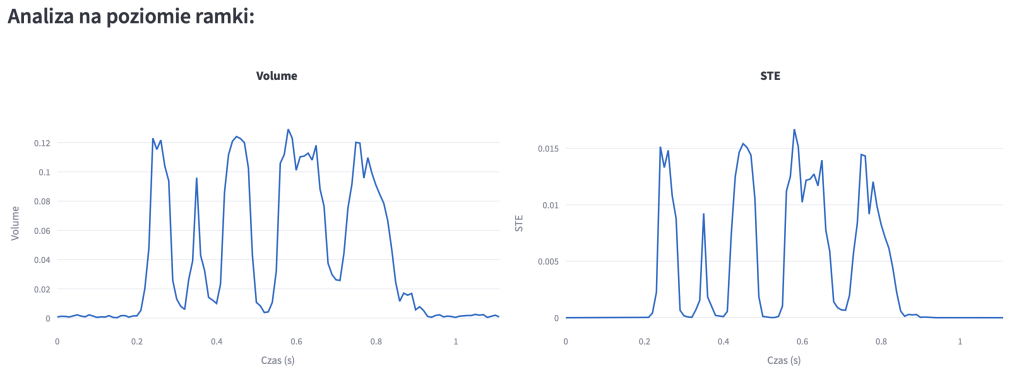
1 Opis Zadania

Celem projektu było stworzenie aplikacji do analizy plików audio, która umożliwia użytkownikowi wczytywanie, przetwarzanie oraz wizualizowanie wybranych parametrów plików audio w formacie WAV. Aplikacja ma na celu dostarczenie narzędzi do analizy dźwięku w dziedzinie czasu, zarówno na poziomie ramki jak i klipu, co pozwala na dokładną charakterystykę zmieniającego się sygnału audio.

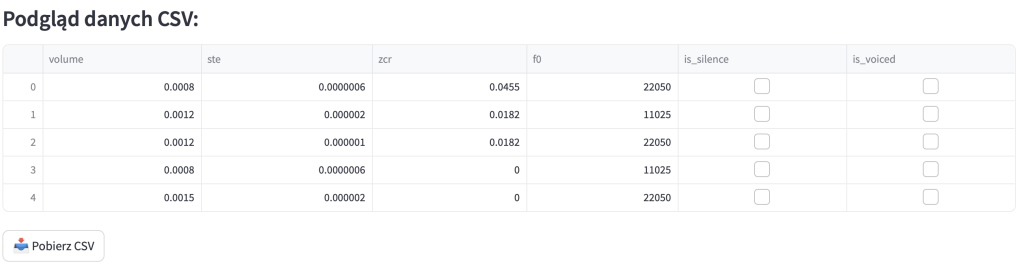
2 Opis Aplikacji

Aplikacja służy do analizy plików audio w formacie WAV, umożliwiając obliczanie i wizualizację kluczowych parametrów akustycznych. Oferuje funkcje takie jak detekcja ciszy, klasyfikacja fragmentów audio na dźwięczne i bezdźwięczne oraz rozróżnianie pomiędzy nagraniami mowy i muzyki. Dodatkowo, umożliwia zapis wyników analizy do pliku CSV, co pozwala na ich dalszą obróbkę i dokumentację. Aplikacja posiada przyjazny i prosty w obsłudze interfejs. Użytkownicy mogą łatwo załadować pliki audio z własnego urządzenia, a także wybierać, czy chcą analizować dane na poziomie ramki czy całego klipu oraz dla jakiej długości ramki chcą przeprowadzić analizę (Rysunek 1). W zależności od wyboru, wyświetlane są odpowiednie wykresy i opcje detekcji (Rysunek 2). Na dole aplikacji znajduje się podgląd pierwszych kilku wierszy pliku CSV, z możliwością zapisania wyników analizy do pliku (Rysunek 3).

Rysunek 1: Zrzut ekranu przedstawiający górną część interfejsu aplikacji. Na zdjęciu widoczne są przyciski umożliwiające wybór różnych funkcji.



Rysunek 2: Zrzut ekranu przedstawiający część przykładowych wykresów dla analizy na poziomie ramki widocznych w interfejsie aplikacji.



Rysunek 3: Zrzut ekranu przedstawiający część interfejsu aplikacji z podglądem wyników analizy w formacie CSV oraz opcją zapisu do pliku.

3 Metody Użyte w Aplikacji

W aplikacji zastosowano różne metody analizy audio na dwóch poziomach: ramki i klipu. Poniżej przedstawiono opis zaimplementowanych metod wraz z odpowiednimi wzorami matematycznymi. W implementacji wykorzystano biblioteki takie jak `streamlit`, `librosa`, `io`, `pandas`, `numpy` oraz `plotly.graph_objects` do przetwarzania dźwięku, tworzenia interaktywnych aplikacji webowych oraz wizualizacji danych.

3.1 Analiza na Poziomie Ramki

3.1.1 Głośność (Volume)

Głośność obliczana jest jako pierwiastek średniej wartości kwadratów próbek sygnału. Jest to miara ogólnej intensywności sygnału w danej ramce.

$$\text{Volume} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$$

gdzie x_i to próbki sygnału w ramce, a N to liczba próbek w ramce.
Metoda została zaimplementowana w funkcji `compute_volume`.

3.1.2 Short Time Energy (STE)

Short Time Energy (STE) jest miarą energii sygnału w ramce i jest obliczana jako kwadrat głośności. Wyższe wartości STE wskazują na większą intensywność dźwięku.

$$\text{STE} = \text{Volume}^2$$

Metoda została zaimplementowana w funkcji `compute_ste`.

3.1.3 Zero-Crossing Rate (ZCR)

Zero-Crossing Rate (ZCR) jest miarą liczby przejść sygnału przez zero w ramce.

$$\text{ZCR} = \frac{1}{N} \sum_{i=1}^{N-1} |\text{sign}(x_i) - \text{sign}(x_{i+1})|$$

gdzie N to liczba próbek w ramce, a x_{i+1} oznacza amplitudę i-tej próbki w danej ramce.
Metoda została zaimplementowana w funkcji `compute_zcr`.

3.1.4 Average Magnitude Difference Function (AMDF)

AMDF jest funkcją, która pozwala obliczyć różnice między próbkami sygnału w określonych opóźnieniach (lag). Jest wykorzystywana do wykrywania częstotliwości podstawowej w sygnale.

$$\text{AMDF(lag)} = \frac{1}{N - \text{lag}} \sum_{i=1}^{N-\text{lag}} |x_i - x_{i+\text{lag}}|$$

gdzie x_{i+1} oznacza amplitudę i-tej próbki w danej ramce, a lag to opóźnienie.
Metoda została zaimplementowana w funkcji `compute_amdf`.

3.1.5 Autokorelacja

Autokorelacja mierzy, jak bardzo sygnał jest podobny do swojej przesuniętej w czasie wersji. Jest używana do wykrywania okresowości oraz określania częstotliwości podstawowej.

$$\text{Autocorr(lag)} = \frac{1}{N - \text{lag}} \sum_{i=1}^{N-\text{lag}} x_i x_{i+\text{lag}}$$

gdzie x_{i+1} oznacza amplitudę i-tej próbki w danej ramce, a lag to opóźnienie.
Metoda została zaimplementowana w funkcji `compute_autocorrelation`.

3.1.6 Częstotliwość Podstawowa (F0)

Częstotliwość podstawowa (f_0) jest obliczana jako odwrotność opóźnienia (lag) w funkcji AMDF lub autokorelacji. Określa ona główną częstotliwość w sygnale, np. ton w przypadku mowy.

$$f_0 = \frac{sr}{\text{lag}}$$

gdzie sr to częstotliwość próbkowania, a lag to opóźnienie, dla którego minimalizowana jest funkcja AMDF lub autokorelacji.
Wartość f_0 w obecnej implementacji jest wyznaczana jedynie dla dźwięcznych fragmentów mowy, dlatego jeśli dana ramka zostanie sklasyfikowana jako bezdźwięczna, f_0 zostaje ustawione na 0.
Metoda została zaimplementowana w funkcji `compute_f0`.

3.1.7 Detekcja ciszy

Detekcja fragmentów ciszy opiera się na analizie ZCR oraz głośności sygnału (volume). Cisza jest wykrywana, gdy wartość ZCR przekracza próg `zcr_threshold=0.01`, a głośność spada poniżej `volume_threshold=0.005`. Domyślne progi zostały wyznaczone eksperymentalnie na podstawie badanych próbek audio. Wartości progowe można regulować w aplikacji za pomocą suwaków dostępnych po wyborze danej metody.
Metoda została zaimplementowana w funkcji `detect_silence`.

3.1.8 Klasyfikacja fragmentów dźwięcznych i bezdźwięcznych

Klasyfikacja fragmentów sygnału jako dźwięczne lub bezdźwięczne bazuje na analizie STE oraz głośności. Fragment jest uznawany za dźwięczny, jeśli wartość STE przekracza `threshold_ste=0.0005`, a głośność jest większa niż `threshold_volume=0.005`. Domyślne progi zostały wyznaczone eksperymentalnie na podstawie badanych próbek audio. Wartości progowe można regulować w aplikacji za pomocą suwaków dostępnych po wyborze danej metody.
Metoda została zaimplementowana w funkcji `classify_voicing`.

3.2 Analiza na Poziomie Klipu

3.2.1 Odchylenie Standardowe Głośności (VSTD)

Odchylenie standardowe głośności w klipie mierzy rozproszenie wartości głośności w czasie. Jest ono obliczane jako stosunek odchylenia standardowego do maksymalnej wartości głośności w klipie.

$$\text{VSTD} = \frac{\sigma(\text{volumes})}{\max(\text{volumes})}$$

gdzie $\sigma(\text{volumes})$ to odchylenie standardowe głośności w klipie.
Metoda została zaimplementowana w funkcji `compute_vstd`.

3.2.2 Dynamiczny Zakres Głośności (VDR)

Dynamiczny zakres głośności (VDR) mierzy różnicę między najwyższą a najniższą wartością głośności w klipie, w odniesieniu do maksymalnej wartości głośności.

$$\text{VDR} = \frac{\max(\text{volumes}) - \min(\text{volumes})}{\max(\text{volumes})}$$

Metoda została zaimplementowana w funkcji `compute_vdr`.

3.2.3 Różnice Głośności (VU)

Różnica głośności (VU) jest obliczana jako suma wartości różnic między kolejnymi próbkami głośności w klipie.

$$\text{VU} = \sum_{i=1}^{N-1} |\text{volume}_i - \text{volume}_{i+1}|$$

gdzie volume_i to wartość głośności w i-tej ramce, a N to liczba ramek w klipie.
Metoda została zaimplementowana w funkcji `compute_vu`.

3.2.4 Low Short-Time Energy Ratio (LSTER)

LSTER (Low Short-Time Energy Ratio) to miara określająca stosunek liczby ramek o niskiej energii krótkoczasowej (niskie STE) do całkowitej liczby ramek w klipie.

$$\text{LSTER} = \frac{\sum_{i=1}^{N} \mathbf{1}(\text{STE}_i < 0.5 \times \text{avg\_STE})}{N}$$

gdzie $\mathbf{1}(\cdot)$ to funkcja indykatorowa, STE_i to energia krótkoczasowa w i-tej ramce, avg_STE to średnia energia krótkoczasowa, a N to liczba ramek w klipie.
Metoda została zaimplementowana w funkcji `compute_lster`.

3.2.5 Entropia Energii

Entropia energii pozwala mierzyć zróżnicowanie energii w różnych częściach klipu.

$$\text{Energy Entropy} = -\sum_{i=1}^{J} p_i \log_2 p_i$$

gdzie p_i to znormalizowana energia i-tego segmentu, a J to liczba segmentów w klipie (w implementacji przyjęto 10 domyślnie 10 segmentów).
Metoda została zaimplementowana w funkcji `compute_energy_entropy`.

3.2.6 Odchylenie Standardowe ZCR (ZSTD)

Odchylenie standardowe ZCR (ZSTD) mierzy zmienność liczby przejść przez zero w klipie.

$$\text{ZSTD} = \sigma(\text{ZCR})$$

gdzie $\sigma(\text{ZCR})$ to odchylenie standardowe ZCR dla ramek w danym klipie.
Metoda została zaimplementowana w funkcji `compute_zstd`.

3.2.7 High Zero-Crossing Rate Ratio (HZCRR)

Wskaźnik HZCRR mierzy proporcję prób z wysoką wartością ZCR w klipie.

$$\text{HZCRR} = \frac{\sum_{i=1}^{N} \mathbf{1}(\text{ZCR}_i > \text{avg\_ZCR})}{N}$$

gdzie ZCR_i to wartość ZCR dla i-tej ramki, avg_ZCR to średnia wartość ZCR w klipie, N to liczba próbek ZCR w klipie, a $\mathbf{1}(\cdot)$ to funkcja indykatorowa.
Metoda została zaimplementowana w funkcji `compute_hzcrr`.

3.2.8 Detekcja Muzyki

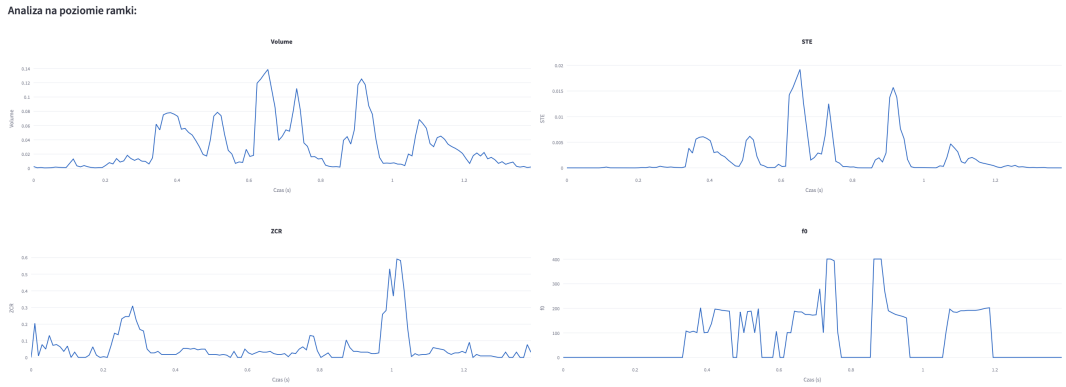
Detekcja fragmentów muzycznych opiera się na analizie wskaźnika LSTER (Low Short-Time Energy Ratio) oraz odchylenia standardowego ZSTD. Fragment jest klasyfikowany jako muzyka, jeśli wartość LSTER jest mniejsza od `lster_threshold=0.25`, a ZSTD przekracza `zstd_threshold=0.3`. W przeciwnym przypadku fragment klasyfikowany jest jako mowa. Domyślne progi zostały wyznaczone eksperymentalnie na podstawie badanych próbek audio. Wartości progowe można regulować w aplikacji za pomocą suwaków dostępnych po wyborze danej metody.
Metoda została zaimplementowana w funkcji `detect_music`.

4 Porównanie Wyników

W tej sekcji porównane zostaną wyniki detekcji dla różnych typów nagrań audio oraz zaprezentowane przykładowe wizualizacje opisanych wcześniej parametrów sygnału audio, które są dostępne w aplikacji.

4.1 Wizualizacje parametrów sygnału audio

Poniżej przedstawione zostały przykładowe wizualizacje oferowane przez aplikację, pokazujące analizę nagrania słowa *szymankowszczyzna* na poziomie ramki oraz fragmentu nagrania muzyki z intro serialu Totally Spies na poziomie klipu.

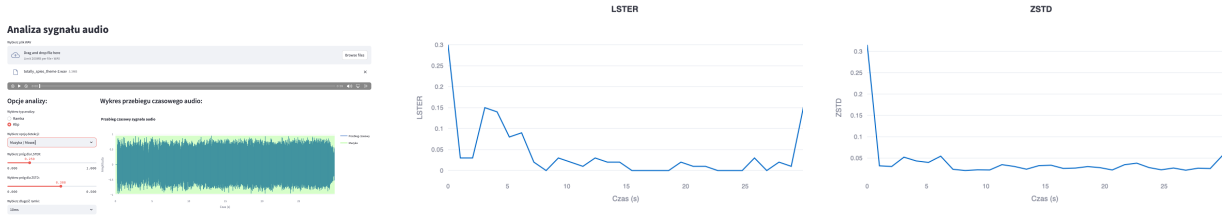


Rysunek 4: Wizualizacje dla słowa *szymankowszczyzna*

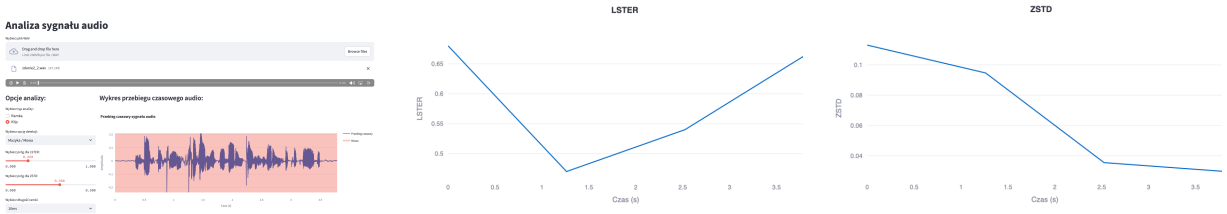
Rysunek 5: Wizualizacje dla muzyki z intro serialu *Totally Spies*

4.2 Detekcja muzyki i mowy

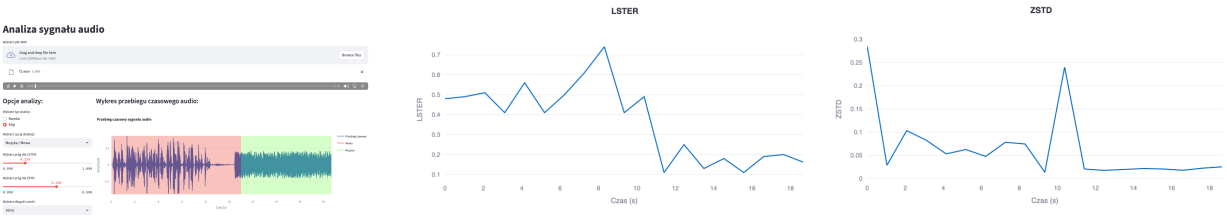
Poniżej przedstawiono porównanie detekcji dla nagrań mowy, muzyki oraz mowy połączonej z muzyką, wraz z analizą wykresów parametrów, na podstawie których przeprowadzane jest rozróżnianie tych kategorii.



Analiza parametrów dla nagrania muzyki



Analiza parametrów dla nagrania mowy



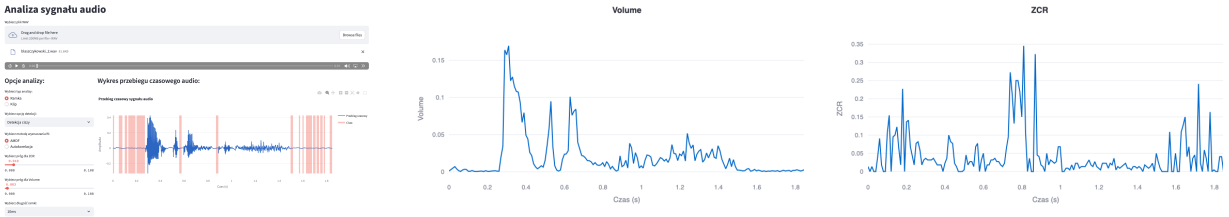
Analiza parametrów dla nagrania zawierającego mowę i muzykę

Rysunek 6: Porównanie wykresów parametrów dla nagrań muzyki, mowy oraz mowy połączonej z muzyką.

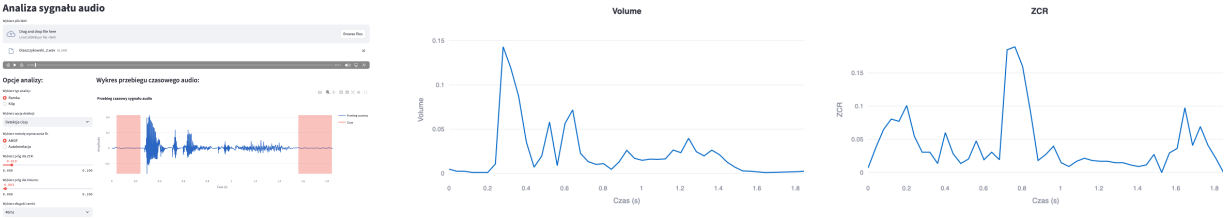
Jak widać, przy domyślnych wartościach progowych detekcja skutecznie rozróżnia mowę i muzykę, nawet w przypadku nagrań mieszanych. Wartość LSTER dla muzyki jest wyraźnie niższa niż dla mowy, co wskazuje na bardziej równomierny rozkład energii w czasie. Z kolei ZSTD dla mowy jest nieco wyższe niż dla muzyki, choć różnica nie jest znacząca.

4.3 Detekcja ciszy

Poniżej przedstawiono porównanie detekcji ciszy dla nagrań mowy dla dwóch rozmiarów ramek: 10 ms oraz 40 ms.



Analiza parametrów dla ramki długości 10 ms



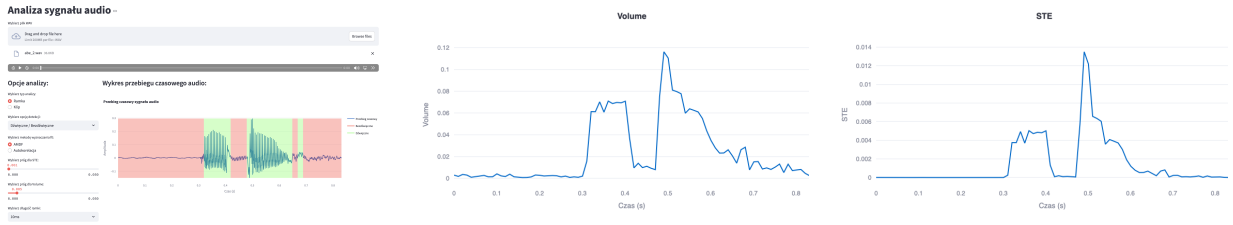
Analiza parametrów dla ramki długości 40 ms

Rysunek 7: Porównanie wykresów parametrów dla detekcji ciszy.

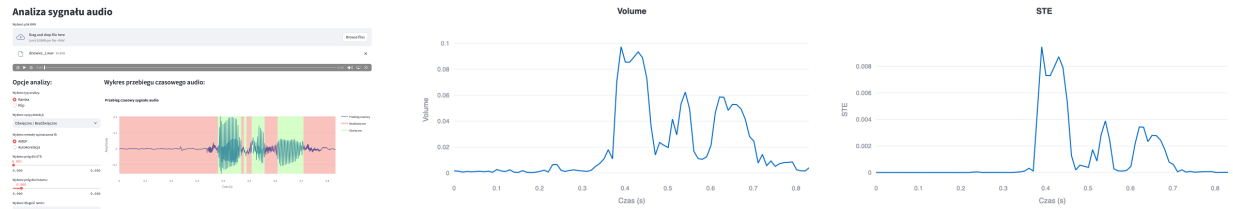
Detekcja ciszy dla wyznaczonych progów działa w miarę poprawnie, choć w niektórych przypadkach mogą występować krótkie przerwy. Wynika to z faktu, że szum tła często utrzymuje się nieznacznie powyżej zera, przez co próg ZCR nie zawsze jest skuteczny. Mimo to metoda stosunkowo dobrze identyfikuje fragmenty ciszy, zwłaszcza przy większych rozmiarach ramek, które zwiększają stabilność detekcji i redukują liczbę błędnych przerw. Zgodnie z intuicją, w przypadku ciszy głośność jest niska, a ZCR powinno być stosunkowo wysokie, ponieważ szum generuje liczne przejścia przez zero. W praktyce jednak to założenie nie zawsze się sprawdza – jeśli cisza to w rzeczywistości niskopoziomowy szum, sygnał może nieznacznie oscylować wokół zera niekoniecznie przez nie przechodząc (utrzymywanie się tuż nad bądź pod 0), co wpływa na wartość ZCR i może prowadzić do drobnych nieścisłości w detekcji.

4.4 Klasyfikacja fragmentów na dźwięczne i bezdźwięczne

Poniżej przedstawiono wyniki klasyfikacji na fragmenty dźwięczne i bezdźwięczne dla słów *abe* oraz *dziewięć*.



Analiza parametrów dla słowa *abe*



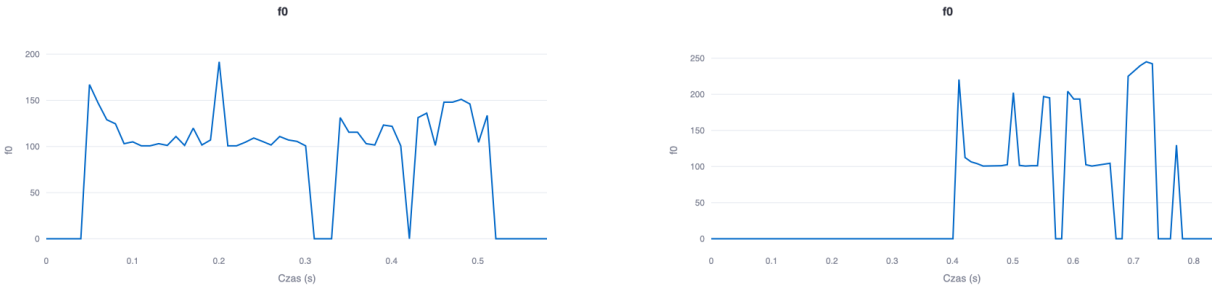
Analiza parametrów dla słowa *dziewięć*

Rysunek 8: Porównanie wykresów parametrów dla detekcji ciszy.

Detekcja dźwięcznych i bezdźwięcznych fragmentów mowy działa stosunkowo dobrze, choć występują krótkie przerwy w klasyfikacji. Mimo to, ogólna skuteczność metody jest zadowalająca.
Dla głosek dźwięcznych można zauważyć, że wartości *volume* oraz *STE* są wyraźnie wyższe w porównaniu do fragmentów bezdźwięcznych.

4.5 Mowa męska i żeńska

Poniżej przedstawiono porównanie wartości f_0 dla głosu męskiego oraz żeńskiego dla słowa *dziewięć*.



(a) Wykres f_0 dla głosu męskiego

(b) Wykres f_0 dla głosu żeńskiego

Rysunek 9: Porównanie wartości f_0 dla głosu męskiego oraz żeńskiego dla słowa *dziewięć*

Wartości częstotliwości podstawowej f_0 dla mowy żeńskiej są przeważnie wyższe niż dla mowy męskiej, jednak różnice te nie zawsze są wyraźnie zauważalne. Średnio f_0 dla głosów żeńskich mieści się w zakresie około 165–255 Hz, podczas gdy dla głosów męskich wynosi około 85–180 Hz. Należy jednak pamiętać, że rzeczywista wartość f_0 może się znacznie różnić w zależności od indywidualnych cech głosu, sposobu mówienia, a także emocji i intonacji danej osoby.

5 Wnioski i Podsumowanie

Przeprowadzona analiza wskazuje, że aplikacja działa poprawnie i skutecznie dokonuje detekcji oraz klasyfikacji sygnałów dźwiękowych. Jednak jej efektywność w dużej mierze zależy od jakości nagrania oraz jego charakterystyki, w tym poziomu szumu tła i dynamiki sygnału. Dodatkowo, wartości progowe używane w detekcji nie są uniwersalne dla każdej próbki – w zależności od nagrania mogą wymagać dostosowania. Optymalne progi mogą różnić się w zależności od warunków akustycznych, rodzaju dźwięku oraz indywidualnych cech mowy danej osoby. W związku z tym możliwość ich regulacji pozwala na lepsze dopasowanie działania aplikacji do konkretnego przypadku.

6 Źródła

- dr inż. J. Rafałko, *Wykłady i materiały z Analizy i Przetwarzania dźwięku*, Politechnika Warszawska, 2025
- OpenAI, ChatGPT (GPT-4), https://openai.com/chatgpt
- H. Jiang, M. K. T. Tye, and E. Gudes, "A Robust Audio Classification and Segmentation Method"ACM Multimedia, 2001.