# Projekt 1 - Analiza i Przetwarzanie Dźwięku

Zofia Kamińska

27 marca 2025

## Spis treści

# 1 Opis aplikacji

Aplikacja **Audio Analyzer** to narzędzie umożliwiające wczytywanie, analizowanie i wizualizowanie sygnałów audio zapisanych w plikach `.wav`. Została zaprojektowana jako aplikacja okienkowa z intuicyjnym interfejsem użytkownika, pozwalająca na interaktywną analizę parametrów dźwięku w dziedzinie czasu. Aby włączyć aplikację należy uruchomić plik app.py.

## 1.1 Główne funkcje aplikacji

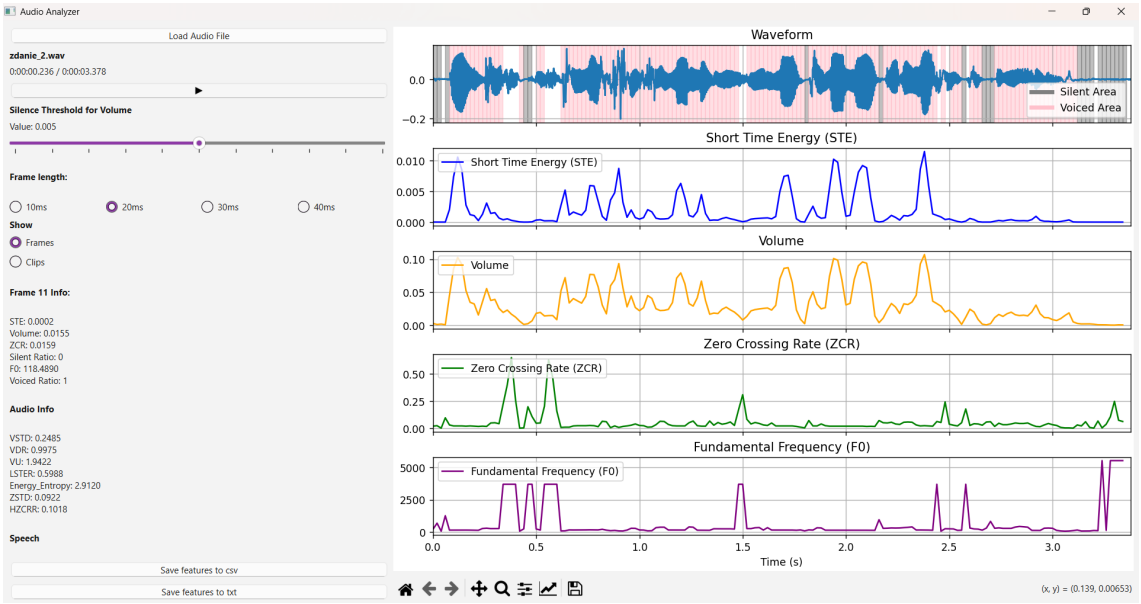
- **Wczytywanie plików audio** – umożliwia załadowanie pliku dźwiękowego w formacie `.wav`.
- **Odtwarzacz audio** – aplikacja umożliwia odtwarzanie wczytanego pliku wraz z wyświetlaniem aktualnej pozycji odtwarzania.
- **Regulacja progu ciszy** – dostępny jest suwak pozwalający na ustawienie progu głośności uznawanego za ciszę.
- **Wybór długości ramki analizy** – użytkownik może wybrać długość ramki analizy spośród wartości: 10ms, 20ms, 30ms i 40ms.
- **Analiza sygnału audio** – aplikacja pozwala na analizowanie sygnału w podziale na ramki lub klipy (jednosekundowe fragmenty).
- **Wizualizacja przebiegu sygnału i cech** – aplikacja generuje wykresy prezentujące kluczowe parametry sygnału audio.
- **Zapisywanie wyników analizy** – użytkownik może zapisać obliczone cechy sygnału do pliku `.csv` lub `.txt`.

## 1.2 Interfejs użytkownika

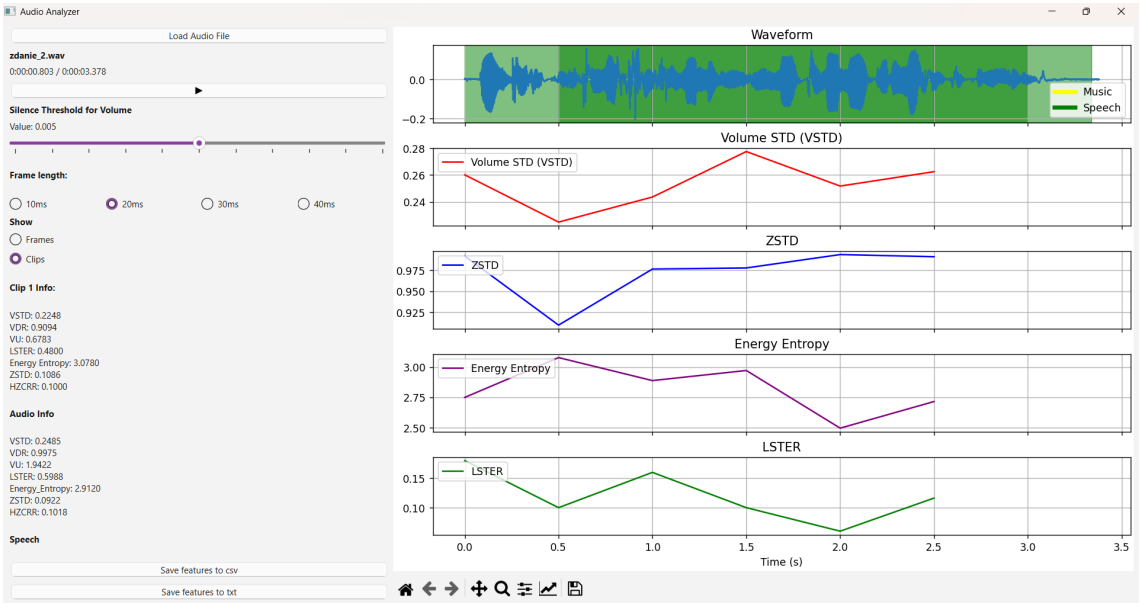
Interfejs aplikacji składa się z dwóch głównych sekcji:

1. **Lewy panel** – zawiera sekcję ładowania pliku, odtwarzacz audio, ustawienia progu ciszy, wybór długości ramki oraz opcję analizy sygnału w podziale na ramki lub klipy. Dodatkowo wyświetlane są podstawowe informacje o aktualnie analizowanym fragmencie dźwięku.
2. **Prawy panel** – zawiera wykresy przedstawiające analizowane cechy sygnału na poziomie ramki lub klipu. Znajduje się tu również pasek narzędzi do nawigacji po wykresach.

Na rysunkach poniżej przedstawione zostały widoki dla analizy dźwięku na poziomie ramki (Rysunek 1) oraz na poziomie klipu (Rysunek 2). Warto dodać, że wszystkie cechy przedstawione w dokumentacji na wykresach zostały obliczone na podstawie ramek o długości 20 ms (step również długości ramki) oraz klipów trwających 1 s (step o długości 0.5s).




Rysunek 1: Interfejs użytkownika przy analizie na poziomie ramek, zdanie_2.wav: *Niebieski słoń fruwał nad jeziorem pełnym gorącej czekolady (głos żeński).* (F0 liczone na podstawie autokorelacji).




Rysunek 2: Interfejs użytkownika przy analizie na poziomie klipów, zdanie_2.wav: *Niebieski słoń fruwał nad jeziorem pełnym gorącej czekolady (głos żeński).*

Sekcja *Audio Info* znajduje się w lewej dolnej części aplikacji i wyświetla wartości cech sygnału audio, które zostały obliczone na podstawie zaznaczonego fragmentu sygnału. Zmiana zakresu zaznaczenia na wykresie (wartości na osi x) powoduje automatyczną aktualizację wyników analizy w sekcji *Audio Info*. Znajdują się tam te same informacje co w sekcji *Clip Info* oraz informacja czy cały zaznaczony obszar reprezentuje mowę czy muzykę.

# 2 Funkcje techniczne

## 2.1 Funkcja `load_audio` (`app.py AudioAnalyzer`)

Funkcja `load_audio` umożliwia użytkownikowi wczytanie pliku audio w formacie `.wav`. Po wybraniu pliku, audio jest ładowane przy użyciu biblioteki `librosa`, a także ustawiany jest próg ciszy i długość ramki analizy. Dodatkowo, plik jest przygotowywany do odtwarzania w aplikacji.

### 2.2 Funkcja `process_audio (app.py AudioAnalyzer)`

Funkcja `process_audio` przetwarza załadowany plik audio, wyodrębniając z niego cechy (przy użyciu funkcji `extract_features` oraz `extract_mini_clip_features`). Na podstawie tych cech generowane są wykresy ilustrujące analizowane parametry audio. Funkcja resetuje również informacje o ramkach i oblicza czas trwania audio.

### 2.3 Funkcja `plot_audio (app.py AudioAnalyzer)`

Funkcja `plot_audio` rysuje wykresy przedstawiające różne cechy akustyczne klipu audio. Na pierwszym wykresie wyświetlana jest fala dźwiękowa, a na kolejnych wykresach rysowane są wartości cech na poziomie ramki (Rysunek 1) lub klipu (Rysunek 2). W zależności od ustawienia, wykresy mogą wskazywać obszary ciche i dźwięczne (w funkcji `plot_wavelength_frames`) lub obszary muzyki i mowy (w funkcji `plot_wavelength_clips`). Wartości te są wyświetlane w formie wykresów na osi czasu, umożliwiając wizualną analizę zmian cech w czasie.

## 3 Funkcje do ekstrakcji cech dźwiękowych

### 3.1 Funkcja: `extract_features (feature_extractor.py)`

Funkcja `extract_features` służy do ekstrakcji cech na poziomie ramek z sygnału audio. Procesuje dane wejściowe w postaci sygnału audio, dzieląc je na ramki i obliczając różne cechy, takie jak energia krótkoterminowa (STE), objętość (volume), wskaźnik przejść przez zero (ZCR), oraz różne wskaźniki dla cichych i głosowych fragmentów.
**Obliczane cechy:**

- **STE (Short-Time Energy):** Energia krótkoterminowa ramki audio. Obliczana jako średnia kwadratów próbek w ramce.

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

  Gdzie $x_i$ to próbka w ramce, a $N$ to długość ramki.

- **Volume (Głośność):** Pierwiastek z energii krótkoterminowej.

$$\text{Volume} = \sqrt{\text{STE}}$$

  Reprezentuje ogólną głośność sygnału w danej ramce.

- **ZCR (Zero-Crossing Rate):** Wskaźnik przejść przez zero, obliczany jako średnia liczba przejść przez zero w ramce podzielona przez jej długość.

$$\text{ZCR} = \frac{1}{2}\frac{1}{N}\sum_{i=1}^{N-1} |\text{sgn}(x_i) - \text{sgn}(x_{i+1})|$$

  Gdzie $\text{sgn}(x)$ to funkcja znakowa.

- **Silent Ratio (Wskaźnik Ciszy):** Przed klasyfikacją na obszary ciche, wpierw obliczane są najmniejsze i największe wartości głośności (min_vol i max_vol). Zmienna odpowiadająca za próg ciszy (vol_threshold) jest wybierane suwakiem, z domyślną wartością 0.09. Ramka jest uznawana za cichą (*silent_ratio = 1*), gdy spełnia warunki:

$$\text{vol} < \text{min\_vol} + (\text{max\_vol} - \text{min\_vol}) \times \text{vol\_threshold}$$

  oraz

$$\text{ZCR} > 0.01.$$

- **Voiced Ratio (Wskaźnik Głosek Dźwięcznych)**: Ramka jest uznawana za zawierającą głoski dźwięczne (*voiced*), gdy spełnia warunki:

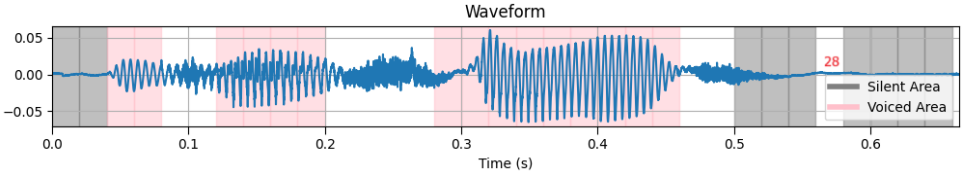
$$\text{ZCR} < 0.15$$

$$\text{vol} > \text{min\_vol} + (\text{max\_vol} - \text{min\_vol}) \times \text{vol\_threshold}$$

$$F_0 < 1000 \text{ Hz}.$$

Jeśli wszystkie warunki są spełnione, *voiced_ratio* wynosi 1, w przeciwnym razie 0.

Przykład z Rysunku 3 przedstawia czytane przez żeński głos słowo "dziesięć"zawierające zarówno głoski dźwięczne (’dzi’,’e’,’ę’) jak i bezdźwięczne (’si’,’ć’).




Rysunek 3: Przebieg czasowy dla pliku dziesiec_2.wav z oznaczonymi obszarami dźwięcznymi

Jak widać, dobrane parametry skutecznie identyfikują zarówno ciszę, jak i dźwięczne fragmenty wypowiedzi. Pewne niedokładności można zauważyć w ramach 5 i 6, gdzie występuje dźwięk „zi” z „dzi” – w naturalnej mowie mógł on brzmieć podobnie do „si”, co mogło wpłynąć na jego błędną klasyfikację jako głoski bezdźwięcznej. Algorytm miał również trudność z wykryciem ciszy w ramce nr 28, co wynikało z tego, że obecny tam szum, mimo bardzo niskiej głośności, miał wartości powyżej zera, powodując zerową wartość ZCR i tym samym błędne oznaczenie ramki jako zawierającej dźwięk.

- **F0 (Fundamental Frequency):** Funkcja `estimate_f0` szacuje częstotliwość podstawową ($F_0$) ramki audio za pomocą dwóch metod: AMDF (Average Magnitude Difference Function) i autokorelacji.

  - **AMDF (Average Magnitude Difference Function)**
    AMDF mierzy różnicę amplitudową sygnału dla przesuniętych wersji i znajduje lag, który minimalizuje tę różnicę:

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

    gdzie $x_i$ to próbka sygnału, $l$ to lag, a $N$ to długość ramki.
    Najlepszy lag wybierany jest jako:

    $$\text{best\_lag} = \arg \min_{l \in [\text{min\_lag}, \text{max\_lag}]} \text{AMDF}(l)$$

    gdzie min_lag ma domyślną wartość 40, a max_lag 320. Częstotliwość podstawowa to:

    $$F_0^{\text{AMDF}} = \frac{\text{sr}}{\text{best\_lag}}$$

  - **Autokorelacja**
    Autokorelacja mierzy podobieństwo sygnału do jego przesuniętej wersji:

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

    Następnie normalizujemy wyniki i wybieramy pierwszy znaczący szczyt:

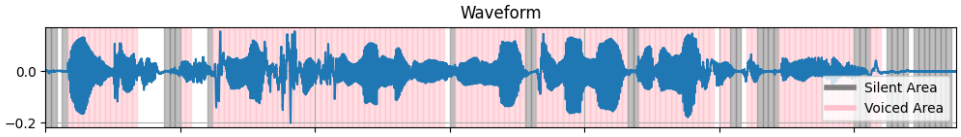
    $$\text{peak\_lag} = \arg\max_l \text{Autocorr}_{\text{norm}}(l), \quad l > 0$$

    Częstotliwość podstawowa wynosi:

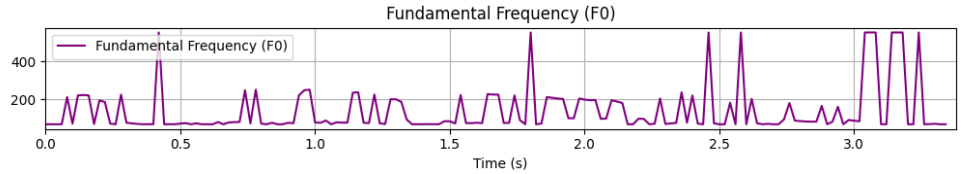
    $$F_0^{\text{Autocorr}} = \frac{\text{sr}}{\text{peak\_lag}}$$

Funkcja `estimate_f0(frame, sr, type)` domyślnie zwraca $F_0$ z AMDF, ale można wymusić użycie autokorelacji, ustawiając `type='autocorr'`.

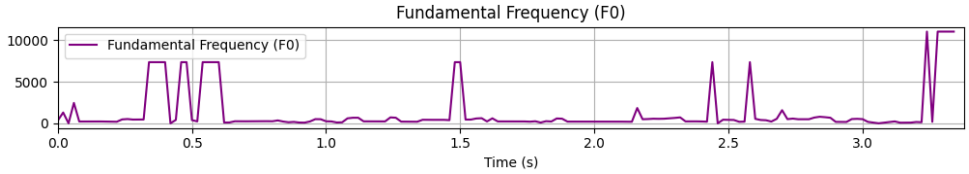
Na rysunkach poniżej zostały przedstawione porównania wyników estymacji częstotliwości podstawowej z użyciem autokorelacji i AMDF, zarówno w przypadku mowy (Rysunek 3) i muzyki (Rysunek 4).




(a) Waveform

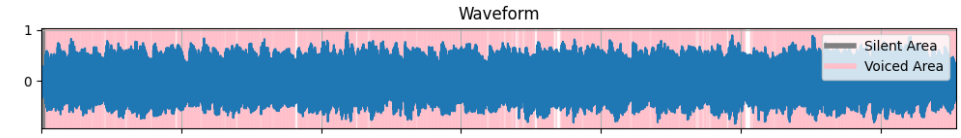



(b) F0 obliczone na podstawie AMDF

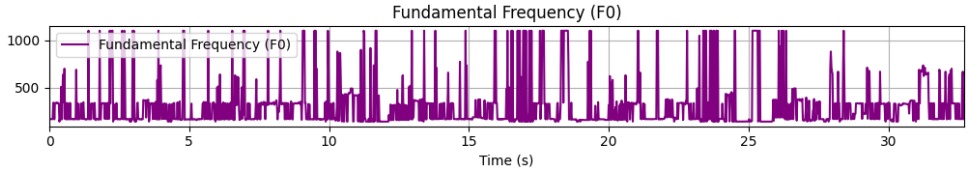



(c) F0 obliczone na podstawie autokorelacji

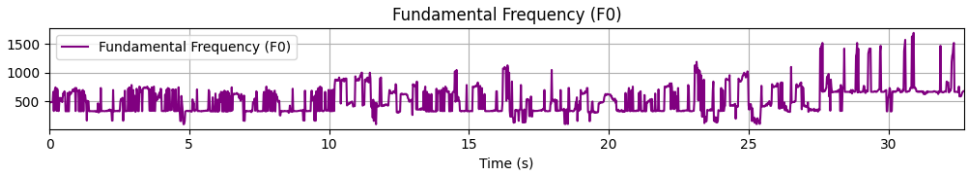
Rysunek 4: Porównanie metod obliczania częstotliwości podstawowej (F0) dla zdanie_2.wav: *Niebieski słoń fruwał nad jeziorem pełnym gorącej czekolady (głos żeński).*




(a) Waveform




(b) F0 obliczone na podstawie AMDF




(c) F0 obliczone na podstawie autokorelacji

Rysunek 5: Porównanie metod obliczania częstotliwości podstawowej (F0) dla piano_music.wav

Jak widać, zarówno w przypadku mowy jak i muzyki obie metody działają poprawnie, chociaż można dostrzec pewne różnice. Część szczytów pokrywa się, podczas gdy inne się nie zgadzają. Dodatkowo, warto zauważyć, że w przypadku autokorelacji na wykresie muzycznym, nie występują aż tak duże skoki (wartości z reguły nie przekraczają 1500 Hz), podczas gdy w przypadku mowy czasami skaczą do wartości rzędu 10000 Hz. Dzieje się to prawdopodobnie z uwagi na mniejszą ilość szumów i cichych obszarów w analizowanym sygnale muzycznym.

## 3.2 Funkcja: `extract_clip_features (feature_extractor.py)`

Funkcja `extract_clip_features` przetwarza cechy na poziomie ramek na cechy na poziomie klipu, uwzględniając różne statystyki oparte na głośności, energii i ZCR.
**Obliczane cechy:**

- **VSTD (Volume Standard Deviation):** Odchylenie standardowe głośność znormalizowane przez maksymalną głośność.

$$\text{VSTD} = \frac{\text{std}(V)}{\max(V)}$$

  Gdzie $V$ to wektor głośności w klipie.

- **VDR (Volume Dynamic Range):** Zakres dynamiczny głośności, znormalizowany przez maksymalną głośność.

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

- **VU (Volume Undulation):** Suma różnic między szczytami a dolinami głośność, odzwierciedlająca zmienność głośności.

$$\text{VU} = \sum |\Delta V|$$

- **LSTER (Low Short-Time Energy Ratio):** Proporcja ramek o energii krótkoterminowej mniejszej niż połowa średniej energii.

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

- **Energy Entropy:** Entropia energii w segmentach STE. W aktualnej implementacji każdy jednosekundowy klip jest dzielony na 10 segmentów. Entropia mierzy niejednorodność rozkładu energii w segmencie.

$$\text{Energy Entropy} = -\sum p_i \log_2(p_i)$$

  Gdzie $p_i$ to znormalizowana energia w danym segmencie.

- **ZSTD (ZCR Standard Deviation):** Odchylenie standardowe ZCR w klipie.

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

- **HZCRR (High ZCR Rate):** Proporcja ramek o ZCR wyższej niż 1.5 razy średnia wartość ZCR.

$$\text{HZCRR} = \frac{\sum_{i=1}^{N} \mathbf{1}(ZCR_i > 1.5 \times \text{avg}(ZCR))}{N}$$

## 3.3 Funkcja: `extract_mini_clip_features (feature_extractor.py)`

Funkcja `extract_mini_clip_features` rozbija cechy ramek na mini-klipy (1-sekundowe klipy z 50% nakładania się) i oblicza cechy na poziomie tych mini-klipów.

## 3.4 Funkcja: `is_music (app.py AudioAnalyzer)`

Funkcja `is_music` klasyfikuje klip audio jako muzykę lub mowę na podstawie wartości trzech cech akustycznych:

- **LSTER** (Low Short-Time Energy Ratio) Wskazuje, jak często energia sygnału spada poniżej pewnego progu. Niższe wartości są charakterystyczne dla muzyki.

- **HZCRR** (High Zero-Crossing Rate Ratio) Ocenia, jak często sygnał przekracza oś poziomą, co jest związane z obecnością szybkich zmian w sygnale. Niższe wartości sugerują muzykę.

- **ZSTD** (Zero-Crossing Rate Standard Deviation) Mierzy zmienność częstości przejść przez zero. Stabilniejsze wartości (niższy ZSTD) są typowe dla muzyki.

Algorytm klasyfikuje klip jako muzykę, jeśli spełniony jest jeden z poniższych warunków:

$$\text{LSTER} \leq 0.39 \wedge \text{HZCRR} < 0.15$$
$$\text{lub} \quad \text{LSTER} \leq 0.39 \wedge \text{ZSTD} < 0.04$$
$$\text{lub} \quad \text{HZCRR} < 0.09 \wedge \text{ZSTD} < 0.037$$

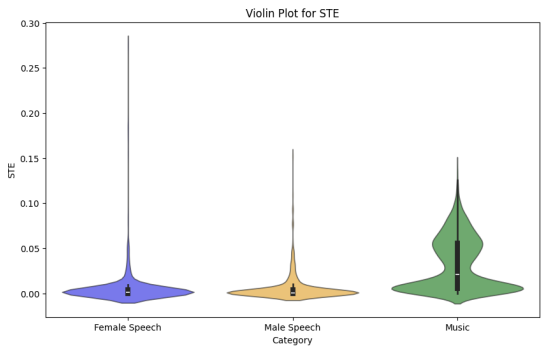
Progi te zostały dobrane na podstawie próbek z danych (dostępnych w repozytorium w odpowiednich folderach). Jak widać w sekcji 4.2 na Rysunku 9, gdzie są zaznaczone wartości progowe dla cech akustycznych. Te progi umożliwiają skuteczną klasyfikację klipów audio jako muzykę lub mowę. Jeśli którykolwiek z tych warunków zachodzi, funkcja zwraca `True` (muzyka), w przeciwnym razie `False` (mowa).

# 4 Porównanie wyników

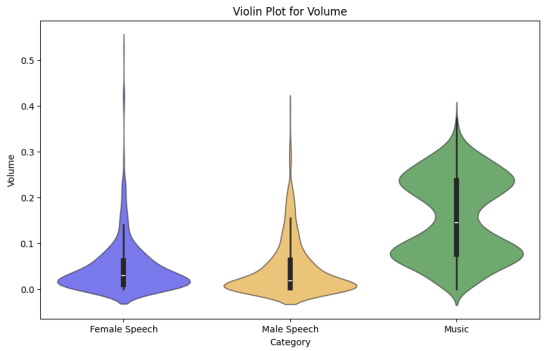
Wszystkie pliki użyte w analizie oraz wartości cech wygenerowane na ich podstawie zostały umieszczone w odpowiednich folderach na GitHubie. Dla kategorii mowy żeńskiej i męskiej użyto po 6-8 krótkich nagrań, tworzonych podczas zajęć, natomiast w przypadku muzyki, do analizy wybrano około 30-sekundowy klip z muzyką fortepianową z akompaniamentem oraz fragmenty utworu „Kung Fu Fighting". Nagrania zostały podzielone na trzy kategorie: mowa żeńska, mowa męska i muzyka, a dla każdej z nich narysowano rozkłady cech. W poniższej sekcji się na omówieniu głównych różnic, które mogą występować pomiędzy kategoriami, takich jak mowa i muzyka, mowa męska i żeńska, a także zidentyfikujemy cechy, które najlepiej charakteryzują każdą z nich.

## 4.1 Cechy na poziomie ramki

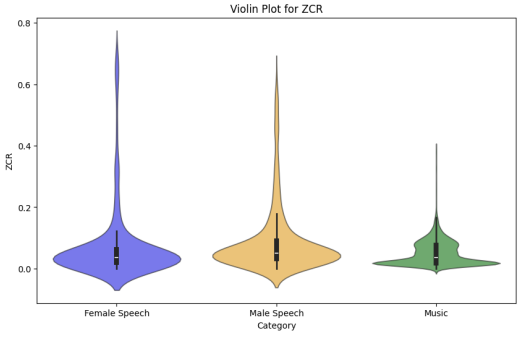
W analizie cech na poziomie ramek skupiono się na tym jak rozkładają się STE, Volume, ZCR i F0, co zostało przedstawione na wykresach poniżej.



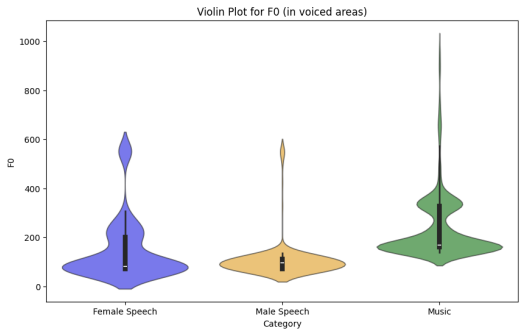

(a) Short Time Energy




(b) Volume




(c) Zero Crossing Rate

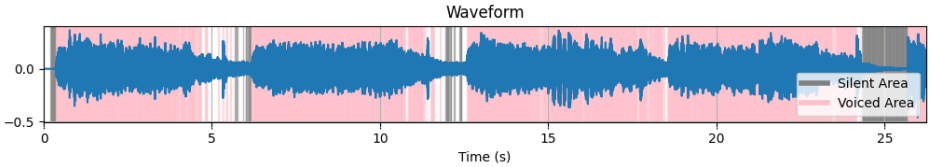



(d) Fundamental Frequency (in voiced areas)

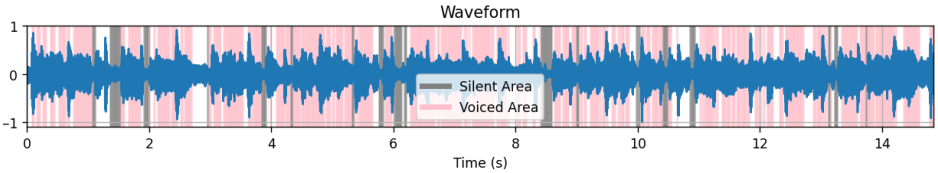
Rysunek 6: Porównanie wartości na pozimie ramek dla mowy żeńskiej, męskiej i muzyki

Jak widać, rozkłady tych cech dla mowy żeńskiej i męskiej są zasadniczo bardzo podobne (Rysunki 5a, 5b, 5c). Jedyną wyraźniejszą różnicą występującą między typami mowy jest rozkład częstotliwości podstawowej (Rysunek 5d), gdzie w przypadku mowy żeńskiej zauważalnie więcej jest wartości w górnej części wykresu. Jest to zgodne z intuicją, gdyż częstotliwości podstawowe w przypadku mowy męskiej są z reguły niższe niż w przypadku mowy żeńskiej.

W przypadku muzyki można zaobserwować szerszy zakres wartości F0 i głośności w porównaniu do mowy. F0 w muzyce zaczyna się nieco wyżej, co jest efektem szerszego zakresu częstotliwości w muzyce instrumentalnej i wyższych tonów generowanych przez instrumenty. Ponadto, w muzyce występuje więcej wartości o wyższym STE i głośności niż w przypadku mowy, co może wynikać z większej dynamiki i intensywności dźwięków w muzyce, gdzie nagłe zmiany energii są częstsze, zwłaszcza w przypadku instrumentów muzycznych. Duży wpływ na te cechy mógł mieć w tym przypadku utwór "Kung Fu Fighting", którego wykresy falowe zostały przedstawione poniżej.




(a) intro_kung_fu.wav

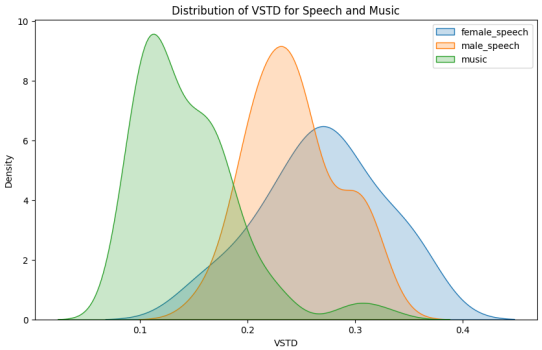



(b) music_sample.wav

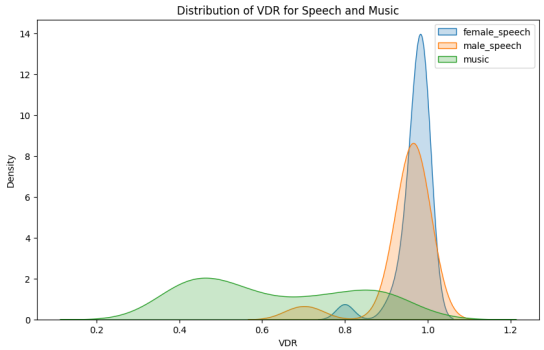
Rysunek 7: Fragmenty utworu "Kung Fu Fighting"użyte podczas analizy cech utworów muzycznych

## 4.2 Cechy na poziomie klipu

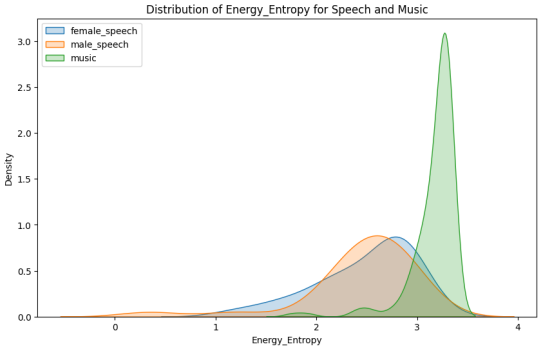
Analiza cech na poziomie klipu obejmuje dłuższe fragmenty dźwięku (1-sekundowe klipy), co pozwala lepiej uchwycić jego ogólną charakterystykę. W przeciwieństwie do analizy ramek, gdzie obserwujemy szybkie zmiany sygnału, tutaj oceniamy jego zmienność w dłuższym okresie. Dzięki temu możliwe jest porównanie rozkładów cech dla mowy żeńskiej, męskiej i muzyki, co pomaga określić, które z nich są najbardziej użyteczne do klasyfikacji tych kategorii.



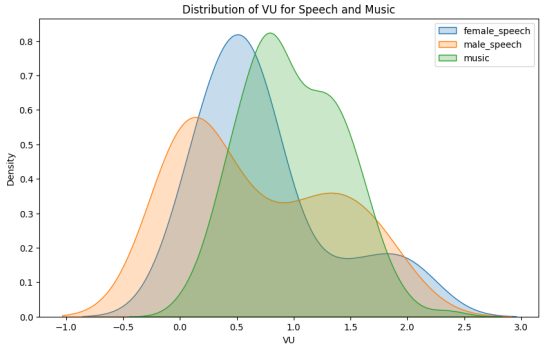

(a) Volume Standard Deviation




(b) Volume Dynamic Range



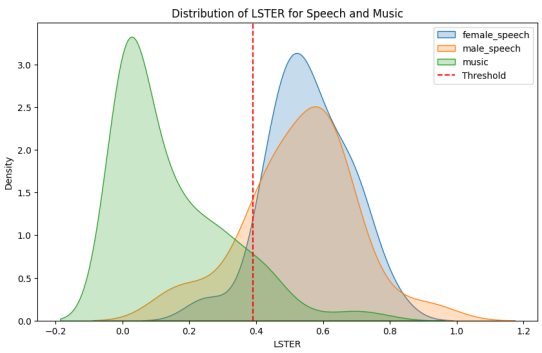

(c) Energy Entropy



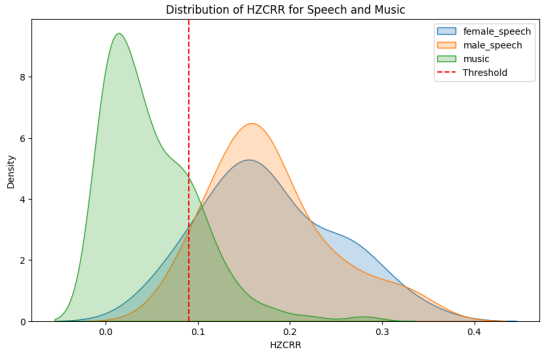

(d) Volume Undulation

Rysunek 8: Porównanie wartości na pozimie klipu bazujących na głośności oraz 'energy entropy' dla mowy żeńskiej, męskiej i muzyki

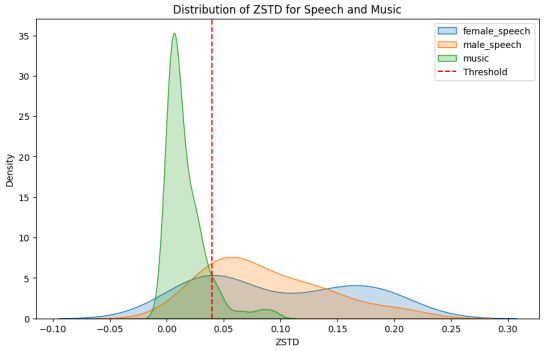
Chociaż na podstawie powyższych cech przedstawionych na wykresach można by również rozróżnić muzykę i mowę, w implementacji zdecydowano się na trzy inne, które są częściej stosowane w tradycyjnych podejściach. Progi (thresholdy) dla wartości LSTER, HZCRR oraz ZSTD zostały wybrane około miejsc styku (przecięcia) rozkładów muzyki i mowy. Są zaznaczone na czerwono i przy mojej implementacji działają na większości rozważanych przykładów.




(a) Low Short Time Energy Ratio




(b) High Zero Crossing Rate Ratio




(c) Standard Deviation of the ZCR

Rysunek 9: Porównanie wartości na pozimie klipu użytych do klasyfikacji muzyka-mowa

Jak widać, dla muzyki wartości tych cech są zazwyczaj poniżej wyznaczonych progów, co wynika z mniejszej zmienności w muzyce, szczególnie w zakresie głośności i struktury dźwięku. Z kolei dla mowy wartości cech są zazwyczaj wyższe, ponieważ mowa charakteryzuje się szerszym zakresem zmienności, szczególnie w odniesieniu do tonacji, intensywności oraz obecności pauz, które są częstym elementem w mowie, zwłaszcza w naturalnych, niezmontowanych wypowiedziach.

# 5 Podsumowanie

Projekt "Audio Analyzer"to aplikacja, która umożliwia szczegółową analizę dźwięków w formacie .wav, koncentrując się na wyciąganiu cech sygnałów audio w dziedzinie czasu. Dzięki zaimplementowanym funkcjom, udało się przeprowadzić kompleksową analizę mowy i muzyki, obejmującą różnorodne cechy dźwiękowe, takie jak energia krótkoterminowa (STE), głośność, wskaźnik przejść przez zero (ZCR) oraz inne parametry charakteryzujące sygnał.
W trakcie realizacji projektu, zdobyłam praktyczną wiedzę na temat różnych metod analizy dźwięku oraz sposobów reprezentowania sygnałów audio w postaci wykresów. Zrozumiałam, jak kluczowe jest dobranie odpowiednich parametrów do analizy, takich jak długość ramki czy próg ciszy, które wpływają na dokładność i interpretację wyników. Ponadto, za pomocą klasyfikacji sygnałów na mowę i muzykę, mogłam lepiej zrozumieć, jak różne cechy akustyczne (np. zmienność głośności, energia) mogą pomóc w identyfikacji charakterystyki dźwięku.
Dzięki przeprowadzonej analizie, udało się uzyskać interesujące wnioski, takie jak różnice w rozkładzie cech dźwiękowych między mową a muzyką, a także wpływ długości analizowanych fragmentów na jakość wyników. Projekt nie tylko umożliwił lepsze zrozumienie teoretycznych aspektów analizy dźwięku, ale także pozwolił na praktyczne zastosowanie tych zasad w pracy z rzeczywistymi plikami audio.

# 6 Źródła (audio)

- Zasoby własne (nagrane podczas zajęć)
- https://pixabay.com/sound-effects/search/piano/
- https://archive.org/details/017kungFuFightingoriginalByCarlDouglas

# Literatura


[1] J. Rafalko, "Cechy sygnału audio w dziedzinie czasu,"2024.
URL: https://pages.mini.pw.edu.pl/ rafalkoj/www/?Dydaktyka:2024

[2] Hao Jiang, "A Comprehensive Study on Audio Feature Extraction for Speech and Music Classification,"*Proceedings of the ACM Multimedia Conference*, 2001.
URL: https://www.hao-jiang.net/papers/conference/acmm01.pdf

[3] ChatGPT, OpenAI, "Audio Analyzer Application Documentation,"OpenAI, 2025.
URL: https://www.openai.com/chatgpt