Analiza i przetwarzanie dźwięku

Sprawozdanie z projektu 2 - analiza częstotliwościowa

Aleksander Malinowski Damian Skowroński

31 maja 2023

1 Wprowadzenie

W tej części rozwijamy dalej projekt o funkcjonalności związane z dźwiękiem w dziedzinie częstotliwości. W tym sprawozdaniu skupimy się początkowo na dokumentacji kodu, a następnie wykorzystamy nowe funkcjonalności w celu opisu i wyciągnięcia wniosków z kilku nagrań.

2 Dokumentacja funkcji umożliwiających analizę częstotliwościową

Funkcje z pierwszego projektu dotyczyły dziedziny czasu i przenieśliśmy je do pliku *time_domain.py*. Nadal implementujemy rozwiązanie w języku Python, a wykorzystane pakiety to głównie: *numpy*, *scipy*, *matplotlib*, *plotly* (pełen wypis pakietów w pliku *environment.yml*. Funkcje z tej części znajdują się w pliku *frequency_domain.py*. Funkcje z obu plików można łatwo zaimportować poprzez wykonanie kodu:

```
import sys
sys.path.append('..')
from functions.time_domain import *
from functions.frequency_domain import *
```

Funkcje ogólnie stosują się do nazewnictwa `compute_<cecha>()`, kiedy mają coś policzyć i `visualise_<cecha>()`, kiedy zwracają wykres (w przeciwieństwie do funkcji z dziedziny czasu, które stosowały się do nazewnictwa `get_<cecha>()` i `plot_<cecha>`). Do funkcji tworzących wykresy można podać argumenty: `fig`, `subplot_row` i `subplot_col`, aby zintegrować je z pozostałymi w dużej figurze.

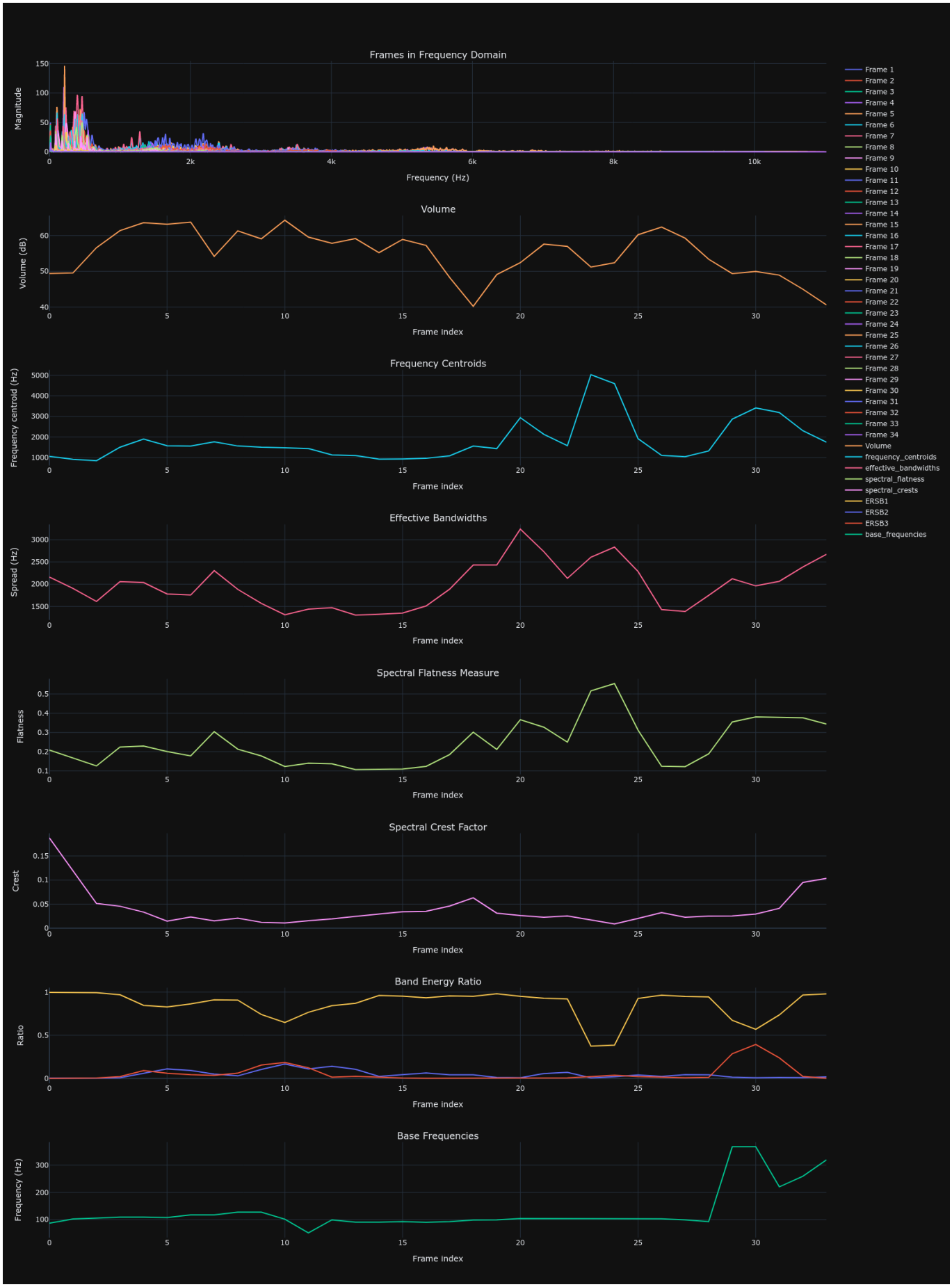
Generalnie najważniejszą funkcją, której output łączy inne funkcje, jest `visualise_all()`. Wystarczy do niej podać w argumentach podzielone ramki, częstotliwość próbkowania, liczbę próbek i liczbę klatek, aby otrzymać interaktywne wykresy związane z analizą częstotliwościową. Output przedstawiony jest na rysunku 1. W następnych sekcjach zawarte są informacje dotyczące poszczególnych funkcjonalności.

2.1 Transformacja ramek do dziedziny częstotliwości

Funkcja `transform_frames_to_frequency_domain()` służy do przekształcania ramki sygnału dźwiękowego z dziedziny czasu do dziedziny częstotliwości za pomocą szybkiej transformacji Fouriera (wykorzystane scipy.fft.fft). Argumenty:

- frames - tablica zawierająca ramki dźwięku, może być jednowymiarowa lub dwuwymiarowa
- frame_rate - częstotliwość próbkowania, wyrażona w hercach (Hz)
- N_ - długość pojedynczej ramki
- window_type - typ funkcji okienkowej, która ma zostać zastosowana. Opcjonalny argument, wartości domyślne to None, co oznacza brak okna. Do wyboru wiele funkcji, m.in.: 'boxcar', 'triang', 'blackman', 'hamming', 'hann'. Pełny opis możliwych okien pod tym linkiem.

Funkcja zwraca tablicę, która zawiera przekształcone ramki dźwięku z dziedziny czasu na dziedzinę częstotliwości. Jeśli argument frames jest jednowymiarowy, zwracany jest wynik przekształcenia Fourier'a dla pojedynczej ramki. W przeciwnym razie, funkcja iteruje po wszystkich ramkach i zwraca przekształcone ramki jako tablicę dwuwymiarową.



Rysunek 1: Output z funkcji visualise_all() dla dwusekundowego nagrania *zdanie_3*

2.2 Rysowanie wykresu sygnału w dziedzinie czasu

Za generowanie wykresu ramki w dziedzine czasu odpowiada funkcja `visualise_frames()`. Funkcja tworzy wykresy dla każdej ramki dźwięku w dziedzinie częstotliwości. Używa funkcji `np.abs` do wyliczenia amplitudy dla każdej częstotliwości. Następnie tworzy oś częstotliwości używając funkcji `np.fft.fftfreq`. Wykres widać jako pierwszy na rysunku 1.Argumenty:

- fft_frames - tablica dwuwymiarowa, zawierająca przekształcone ramki dźwięku z dziedziny czasu na dziedzinę częstotliwości
- frame_rate: częstotliwość próbkowania, wyrażona w hercach (Hz)
- n_ - liczba ramek dźwięku do wyświetlenia
- N_ - długość pojedynczej ramki
- fig - obiekt rysunku wykresu. Opcjonalny argument, wartości domyślne to None, co oznacza, że zostanie utworzony nowy obiekt rysunku.
- subplot_row, subplot_column - jeśli fig jest podany to odpowiadają za pozycję wykresu.

2.3 Volume

Funkcja `compute_volume()` przyjmuje jedną ramkę dźwiękową w dziedzinie częstotliwości i oblicza jej głośność. Domyślnie głośność jest obliczana w skali liniowej, ale jeśli in_db jest ustawione na True, funkcja zwraca wartość głośności w decybelach. Jeśli spl jest ustawione na True, funkcja zwraca wartość głośności w poziomie ciśnienia akustycznego (SPL) w decybelach.

Funkcja `visualise_volume()` oblicza głośność dla każdej ramki wykorzystując funkcję `compute_volume()` i rysuje wykres głośności. Domyślnie wykresy są rysowane w skali SPL w decybelach, ale jeśli in_db jest ustawione na True, wykresy są rysowane w skali liniowej w decybelach. Wykres można zobaczyć na rysunku 1 jako drugi.

2.4 Frequency centroid

Funkcja `compute_frequency_centroid()` oblicza centroid częstotliwościowy dla danej ramki dźwiękowej. Centroid częstotliwościowy, zwany także centroidem spektralnym, jest miarą środka ciężkości spektrum częstotliwościowego ramki. Funkcja przyjmuje dwie tablice: `magnitude` - wartości amplitudy dla każdej częstotliwości i `freq_axis` - oś częstotliwości. Argument `N_` oznacza długość ramki. Funkcja oblicza centroid częstotliwościowy jako iloczyn ważony amplitud i odpowiadających częstotliwości, a następnie dzieli przez sumę amplitud. Zwraca wartość centroidu częstotliwościowego.

Funkcja `visualise_frequency_centroids` oblicza centroidy częstotliwościowe dla każdej ramki i rysuje wykres centroidów. Wykres można zobaczyć na rysunku 1 jako trzeci.

2.5 Effective bandwidth

Funkcja `compute_effective_bandwidth` oblicza szerokość pasma efektywnego dla danej ramki. Szerokość pasma efektywnego, zwana także rozproszeniem spektralnym, jest miarą rozproszenia częstotliwości w spektrum częstotliwościowym ramki. Funkcja przyjmuje dwie tablice: `magnitude` - wartości amplitudy dla każdej częstotliwości i `freq_axis` - oś częstotliwości. Argument `N_` oznacza długość ramki. Funkcja oblicza centroid częstotliwościowy przy użyciu funkcji `compute_frequency_centroid` opisanej w poprzedniej sekcji, a następnie oblicza szerokość pasma efektywnego jako pierwiastek kwadratowy z sumy ważonych kwadratów różnicy między częstotliwościami a centroidem, podzielonej przez sumę amplitud. Zwraca wartość szerokości pasma efektywnego.

Funkcja `visualise_effective_bandwidths` oblicza szerokości pasma efektywnego dla każdej ramki używając funkcji `compute_effective_bandwidth` i rysuje wykres szerokości pasma efektywnego dla nagrania. Wykres można zobaczyć na rysunku 1 jako czwarty.

2.6 Spectral flatness

Funkcja `compute_spectral_flatness` oblicza płaskość spektralną dla danej ramki. Płaskość spektralna jest miarą równomierności rozkładu energii w spektrum częstotliwościowym ramki. Funkcja przyjmuje tablicę `magnitude`, która zawiera wartości amplitudy dla każdej częstotliwości, oraz argument `N_`, który oznacza długość ramki. Funkcja oblicza płaskość spektralną jako geometryczną średnią wartości amplitudy podzieloną przez średnią wartość amplitudy. Zwraca wartość płaskości spektralnej.

Funkcja `visualise_spectral_flatness` oblicza płaskości spektralne dla każdej ramki używając funkcji `compute_spectral_flatness` i rysuje wykres płaskości spektralnej. Wykres można zobaczyć na rysunku 1 jako piąty.

2.7 Spectral crest

Funkcja `compute_spectral_crest` oblicza wierzchołek spektralny dla danej ramki. Funkcja przyjmuje tablicę `magnitude`, która zawiera wartości amplitudy dla każdej częstotliwości, oraz argument `N_`, który oznacza długość ramki. Funkcja oblicza wierzchołek spektralny jako stosunek największej wartości amplitudy do sumy amplitud. Zwraca wartość wierzchołka spektralnego.

Funkcja `visualise_spectral_crest` oblicza wierzchołki spektralne dla każdej ramki używając funkcji `compute_spectral_crest` i rysuje wykres wierzchołków spektralnych. Wykres można zobaczyć na rysunku 1 jako szósty.

2.8 Band energy ratio

Funkcja `compute_band_energy_ratio` oblicza stosunek energii pasmowej dla trzech przedziałów częstotliwości w stosunku do całkowitej energii dla danej ramki. Funkcja przyjmuje `frame_magnitude`, które zawiera wartości amplitudy dla każdej częstotliwości, `freq_axis`, które zawiera osie częstotliwości, oraz `N_`, oznaczające długość ramki. Następnie funkcja oblicza energię dla każdego z trzech pasm (`band_1_energy` (0-630 Hz), `band_2_energy` (630-1720 Hz), `band_3_energy` (1720-4400 Hz)) poprzez sumowanie kwadratów amplitud w odpowiednich przedziałach częstotliwości. Pełna energia jest obliczana jako suma kwadratów wszystkich amplitud. Na koniec funkcja zwraca trzy wartości: stosunek energii pasma 1 do pełnej energii (ERSB1), stosunek energii pasma 2 do pełnej energii (ERSB2) i stosunek energii pasma 3 do pełnej energii (ERSB3).

Funkcja `visualise_band_energy_ratio` oblicza stosunki energii pasmowej dla każdej ramki i rysuje wykresy tych stosunków dla każdego pasma na współdzielonym wykresie. Wykres można zobaczyć na rysunku 1 jako siódmy.

2.9 Base frequencies

Funkcja `compute_base_frequency` oblicza podstawową częstotliwość dla danej ramki. Funkcja przyjmuje `frame`, który zawiera próbki dźwiękowe ramki, `sampling_frequency`, oznaczającą częstotliwość próbkowania, oraz opcjonalne parametry `min_freq` i `max_freq`, które określają minimalną i maksymalną oczekiwaną podstawową częstotliwość. Wewnątrz funkcji obliczane jest realne cepstrum przez przeprowadzenie odwrotnej transformaty Fouriera (IFFT) na logarytmie z amplitudy FFT ramki. Następnie wybierane są granice quefrency (odwrotność częstotliwości) na podstawie podanych `min_freq` i `max_freq`. Lokalne maksimum cepstrum jest wyznaczane w tym zakresie i zapisywane jako `local_max_quefrency`. Ostatecznie podstawowa częstotliwość jest obliczana jako odwrotność quefrency lokalnego maksimum. Funkcja zwraca również realne cepstrum i podstawową częstotliwość.

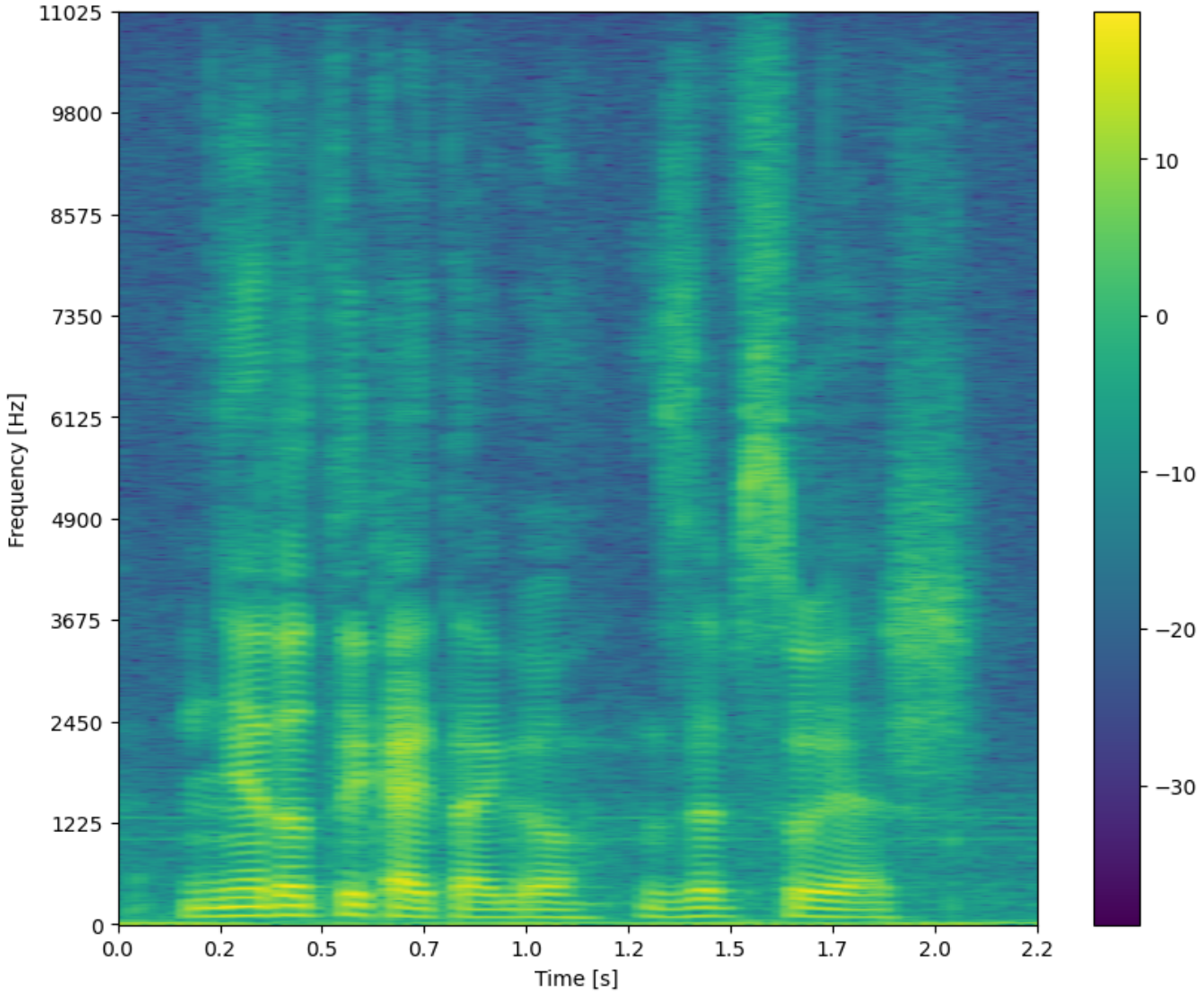
Funkcja `visualise_base_frequency` oblicza i rysuje podstawowe częstotliwości dla każdej ramki. Dla każdej ramki wywoływana jest funkcja `compute_base_frequency` w celu obliczenia podstawowej częstotliwości. Obliczone podstawowe częstotliwości są przedstawiane na wykresie. Wykres można zobaczyć na rysunku 1 jako wykres ósmy.

2.10 Spectrogram

Funkcja `visualise_spectrogram` służy do wizualizacji spektrogramu dźwiękowego dla danego pliku. Funkcji przyjmuje argumenty:

- `path` - ścieżka do pliku
- `percent_frame_size` - procentowe rozmiary ramki
- `percent_hop_length` - procentowe przesunięcie ramki
- `window_type` - typ okna wykorzystanego przy transformacji Fouriera
- `figsize` - rozmiar spektrogramu, krotka
- `time_in_frames` - czy oś X jako indeks ramek (True), czy jako czas w sekundach (False), boolean
- `log_amplitude` - czy wartości widma amplitudowego przeliczane na logarytm dziesiętny, boolean

Funkcja korzysta z funkcji `read_wave` (*time_domain.py*), `split_to_frames` (*time_domain.py*) i `transform_frames_to_frequency_domain` w celu kolejno: wczytania pliku o podanej ścieżce `path`, podziale na ramki w zależności od podanych `percent_frame_size` i `percent_hop_length` i ostatecznie wykonania transformacji Fouriera przy wykorzystaniu podanego `window_type`. Z tak przygotowanych ramek otrzymuje ich amplitudy i przedstawia spektrogram używając funkcji `imshow` z pakietu *matplotlib*. Przykładowy spektrogram został przedstawiony na rysunku 2.



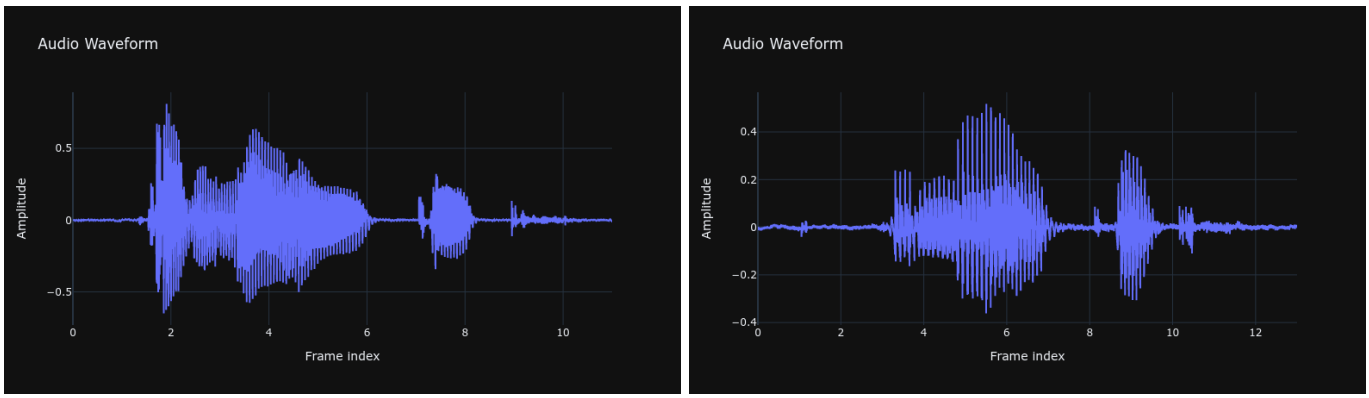
Rysunek 2: Output z funkcji visualise_spectrogram dla dwusekundowego nagrania *zdanie_3*

3 Przykłady użycia wizualizacji dla różnych nagrań

W tej sekcji pokażemy jak można wykorzystywać wizualizacje do wyciągania wniosków z klipów audio.

3.1 Porównanie męskiego i żeńskiego głosu

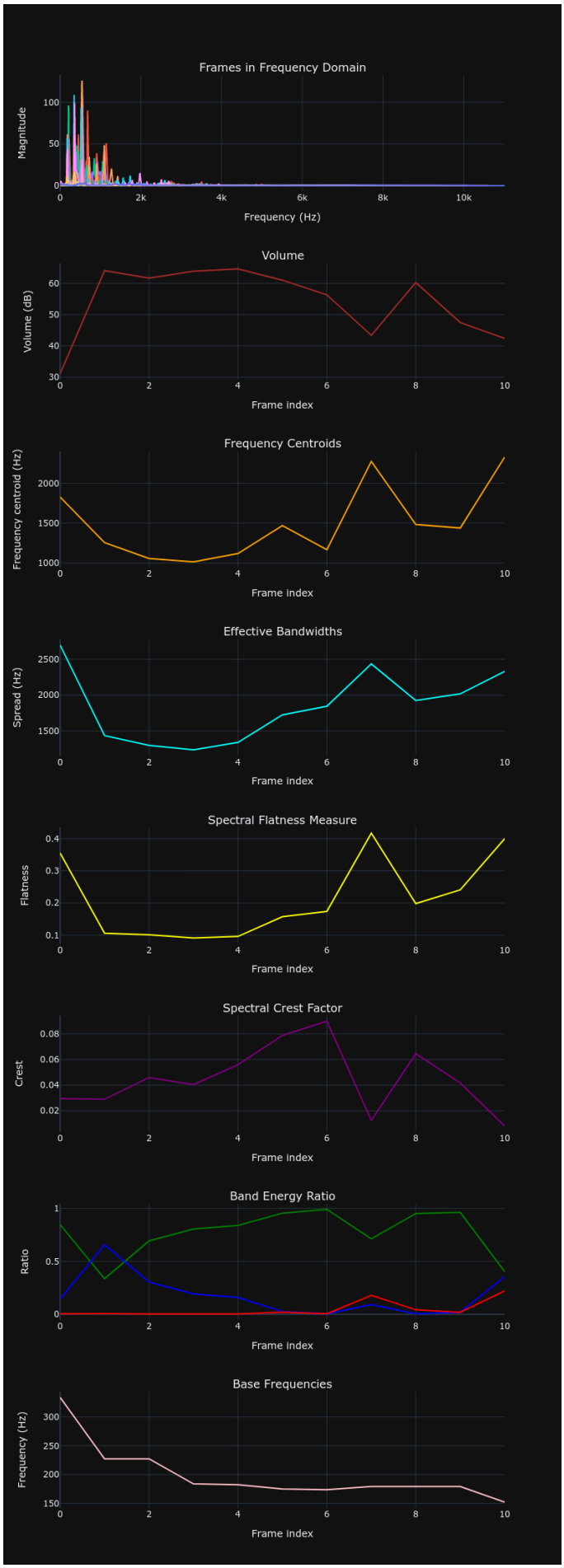
Do porównania wykorzystamy nagrania mówienia słowa "omoltyk" głosu męskiego i żeńskiego. Oba nagrania są znormalizowane (wizualizację można zobaczyć lepiej w pliku *examples/omoltyk_comparison.ipynb*).



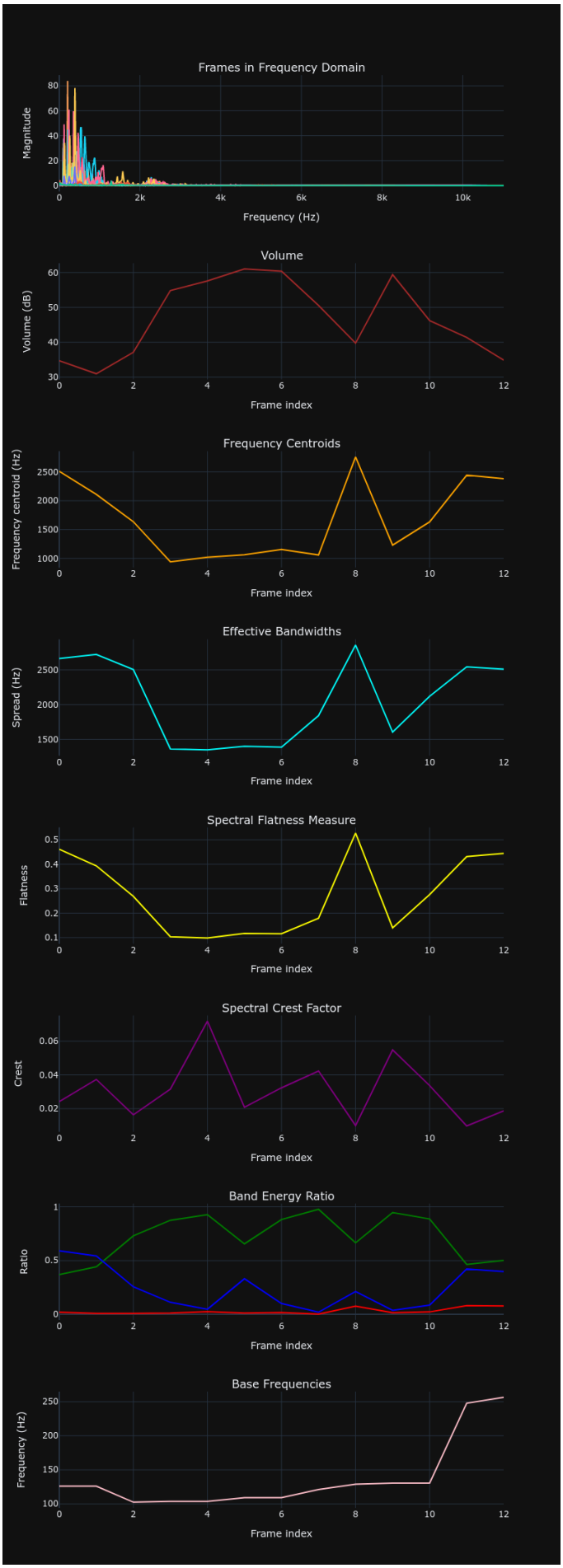
(a) głos żeński

(b) głos męski

Rysunek 3: Wizualizacja nagrania "omoltyk"

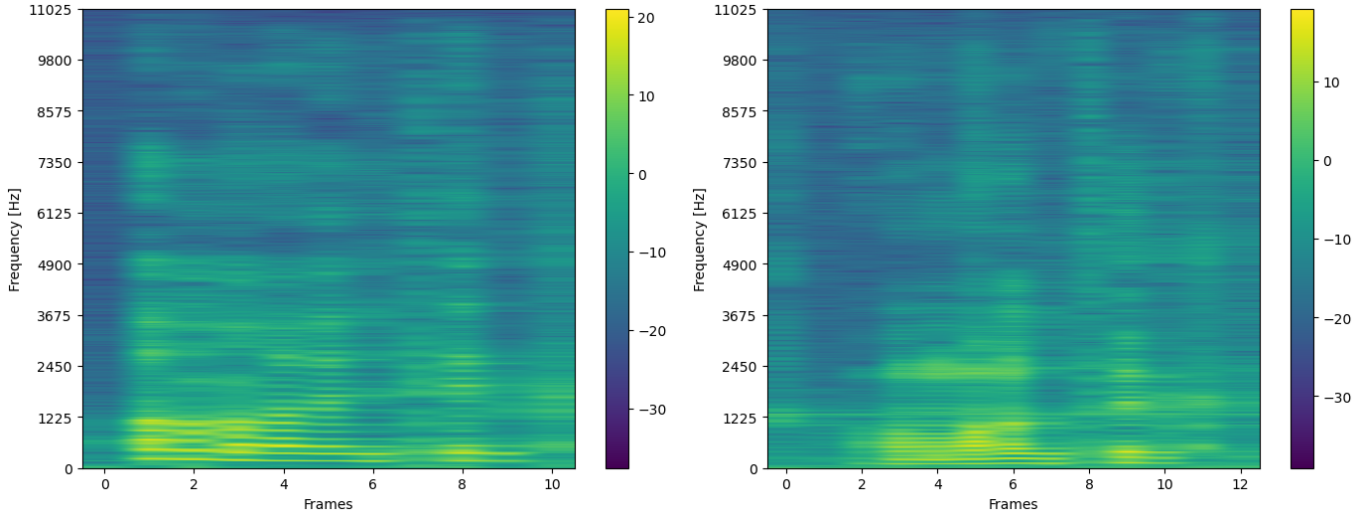


(a) głos żeński



(b) głos męski

Rysunek 4: Wizualizacja cech z dziedziny częstliwości dla nagrania ”omoltyk”



(a) głos żeński

(b) głos męski

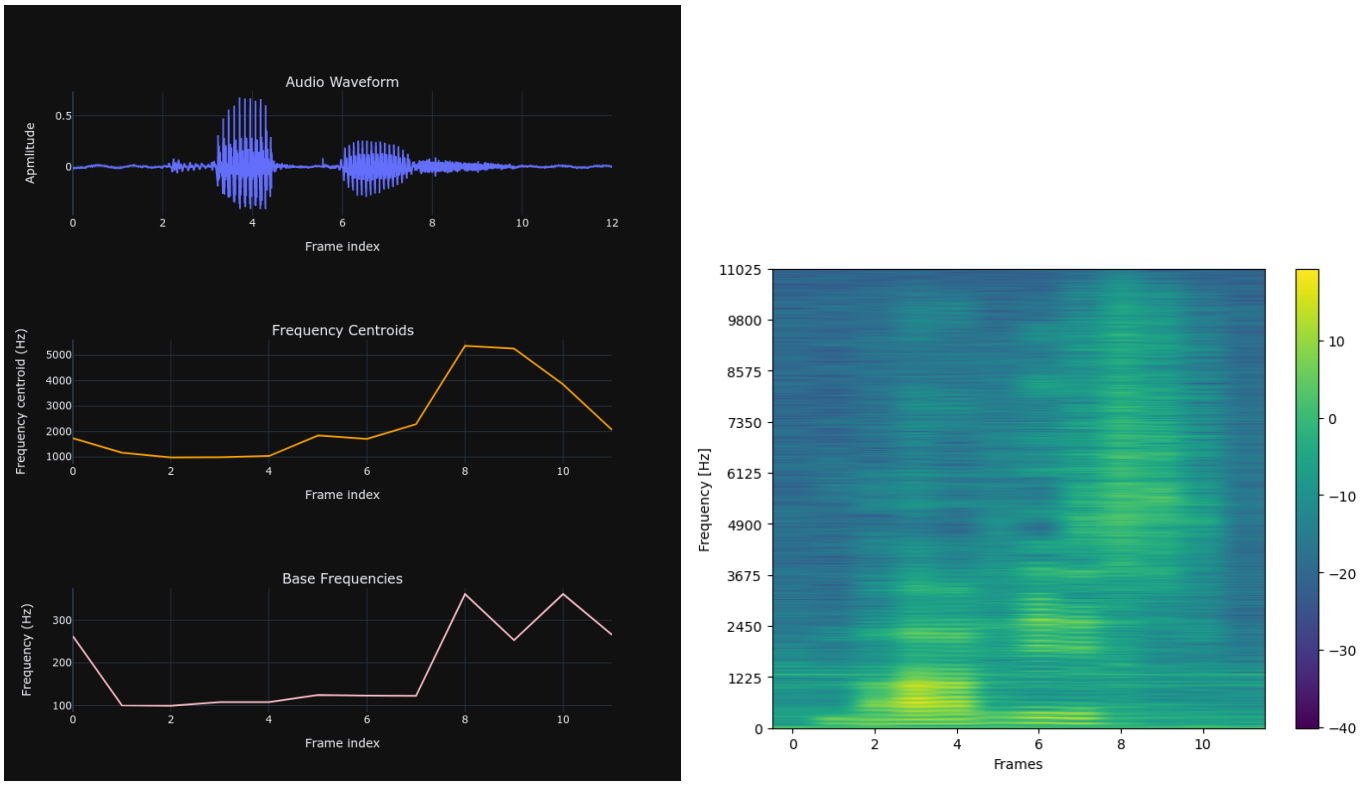
Rysunek 5: Spektrogramy dla nagrania "omoltyk"

Na rysunku 3 widać, że słowo zostało zaintonowane w inny sposób. W przypadku głosu żeńskiego (wykres a) widać, że dużo wagi jest w pierwszym "o". Reszta wykresu nie różni się znacząco.

Na rysunku 4 można zobaczyć na przykład, że na wykresie *Frames in Frequency Domain* głos męski (wykres b) jest bardziej "ściśnięty" do 0 na osi X niż głos żeński (wykres a). Dodatkowo na wykresie *Base Frequencies* dla głosu żeńskiego podstawowe częstotliwości są zwykle w około 180 Hz, natomiast dla głosu męskiego są one znacznie niższe, to znaczy w przedziale 100-130Hz. Wyniki zgadzają się z tezą, że głos męski zwykle ma niższą częstotliwość niż głos żeński.

3.2 Zmiana tonu podstawowego

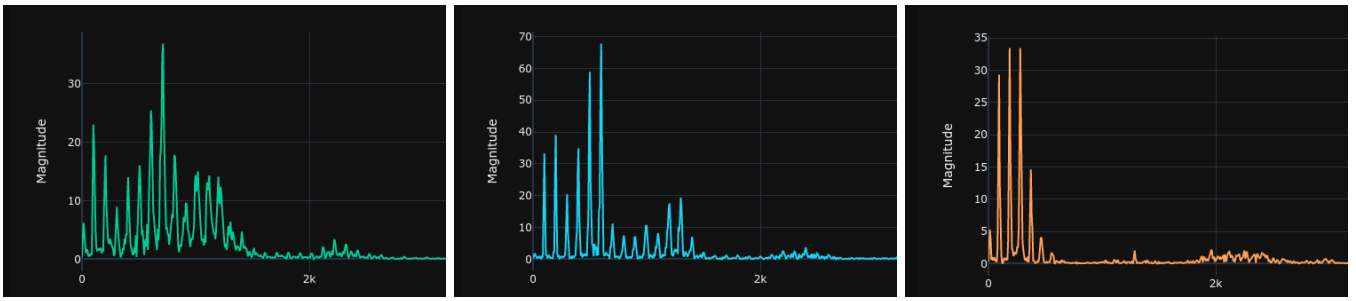
Na rysunku 6 przedstawione są wykresy dla nagrania "bapis" przez głos męski. Widać jak dla ostatniej głoski "s" (około 8 ramki) częstotliwości są bardzo wysokie. Ton podstawowy idzie w górę z 100Hz na 300Hz. Ogólnie dla pozostałych nagrań ton podstawowy dla tego samego lektora pozostaje w tym samym przedziale.



Rysunek 6: Wykresy dla nagrania "bapis"

3.3 Samogłoski i formanty

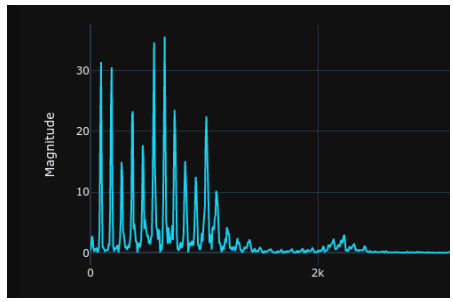
Na rysunku 7 pokazane są wykresy częstotliwości ramek, w których występowały dane samogłoski w różnych słowach. Wykresy są zrobione w ten sposób, że na przykład dla głoski ”a” mamy zrzuty ekranu z wykresów z dwóch słów jeden pod drugim (czyli jakby w 1 i 2 rzędzie obrazków). Widać, że ramki odpowiadające danym samogłoskom są do siebie bardzo podobne i można poszczególną samogłoskę odróżnić od pozostałych. Wykresy wraz ze spektrogramami można zobaczyć w pliku *examples/vowels_male.ipynb*.



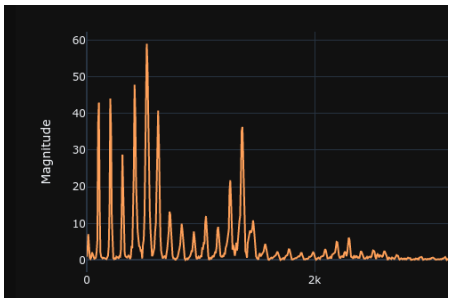
(a) ”a”

(b) ”e”

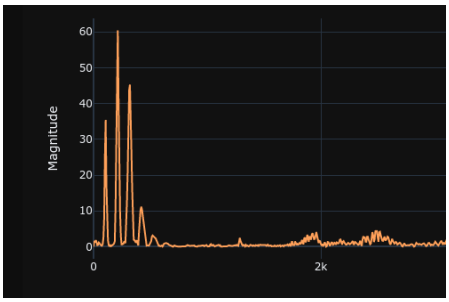
(c) ”i”



(d) ”o”



(e) ”u”

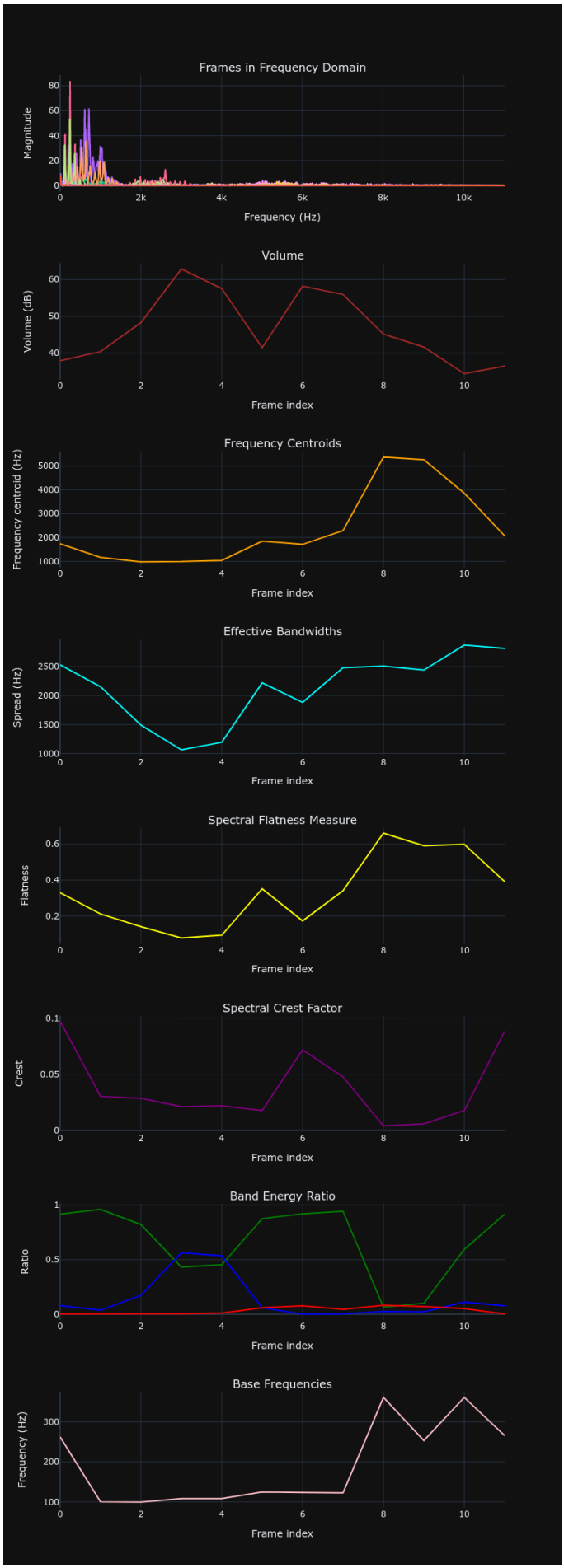


(f) ”y”

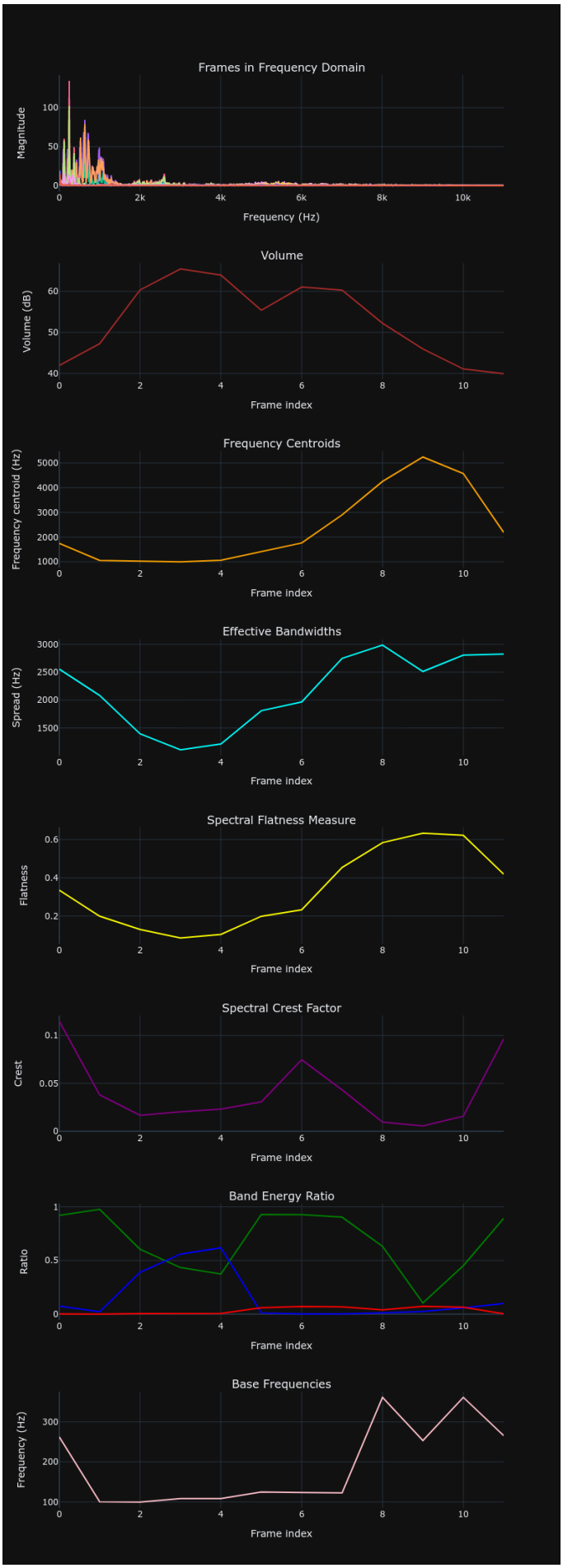
Rysunek 7: Wykresy częstotliwości dla poszczególnych samogłosek.

3.4 Różnice w użyciu funkcji okienkowych

Na rysunku 8 można zauważyć, że wybór funkcji okienkowej ma niemałe znaczenie dla wyników. W tym przypadku wybór funkcji ”hann” wydaje się lepszy ponieważ na wykresach otrzymujemy większą szczegółowość. Wykresy dla prostokątnej funkcji są bardziej gładkie.



(a) window function = "hann"

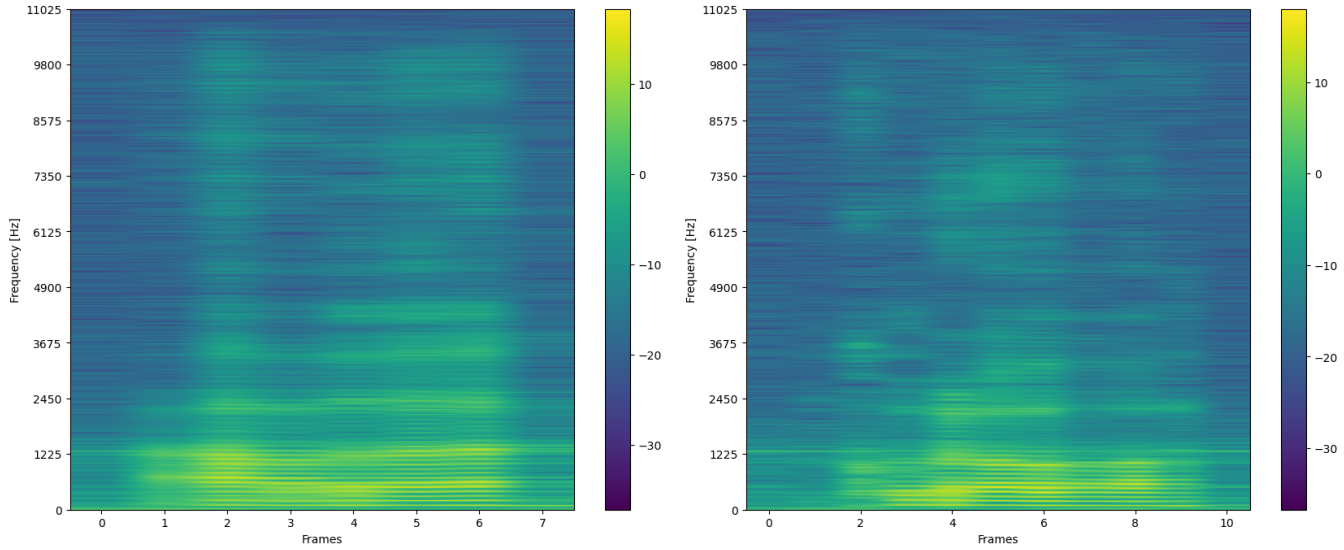


(b) window function = "rectangular"

Rysunek 8: Wykresy cech z dziedziny częstotliwości dla nagrania "bapis" w zależności of funkcji okienkowej.

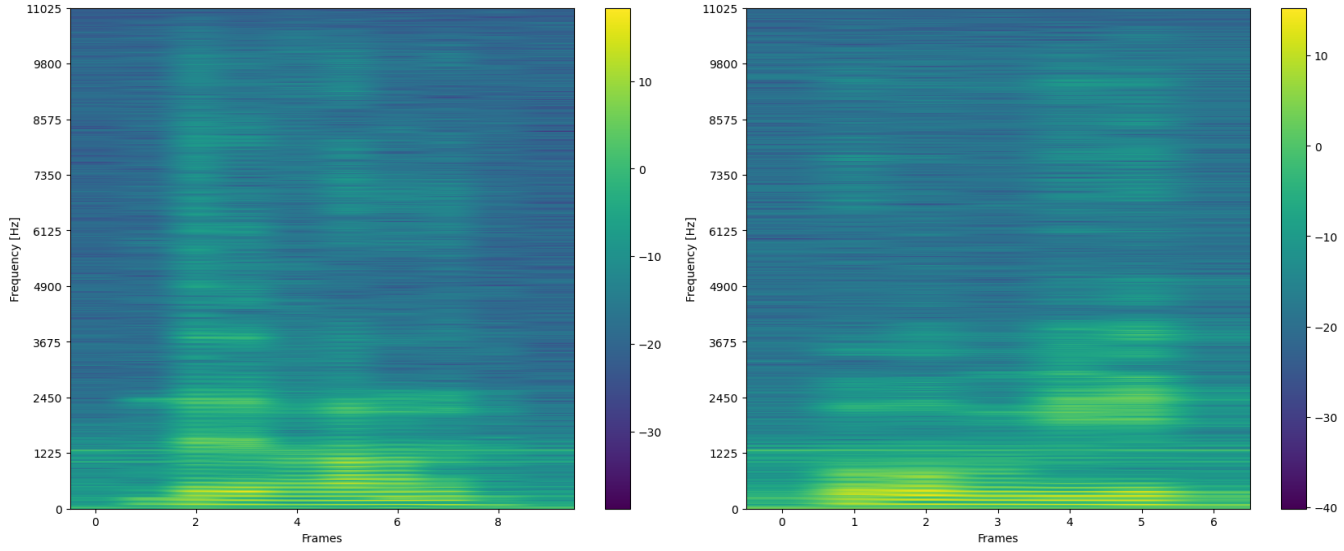
3.5 Różne głoski na spektrogramach

Na rysunku 9 przedstawione są spektrogramy dla różnych słów.



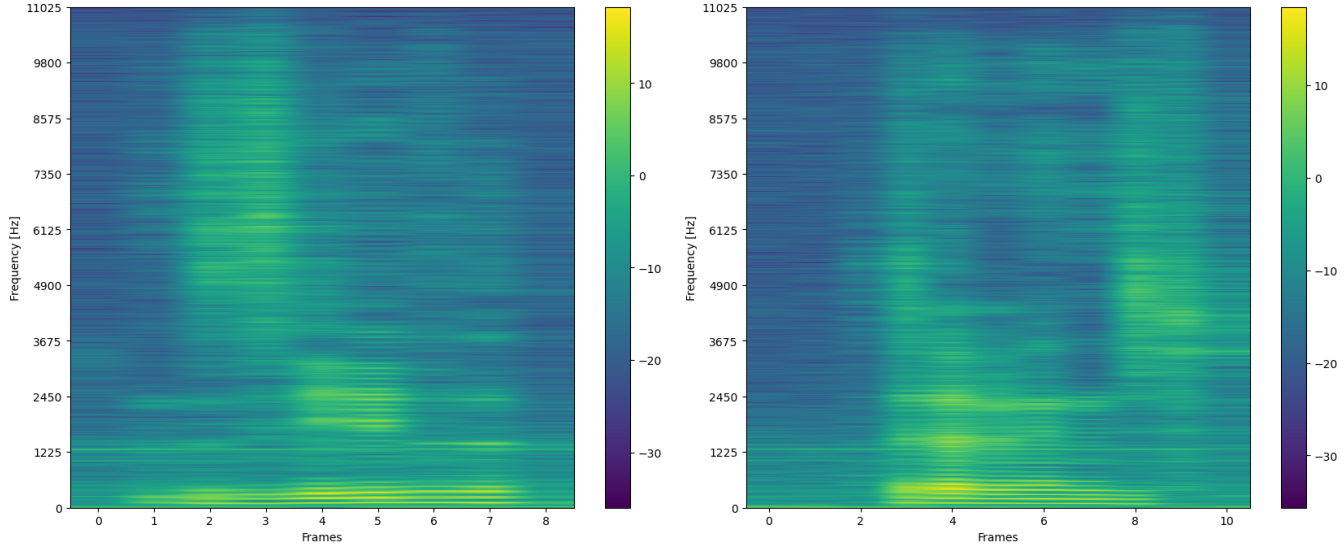
(a) ”abe”

(b) ”kurowa”



(c) ”dywan”

(d) ”ugi”



(e) ”zid”

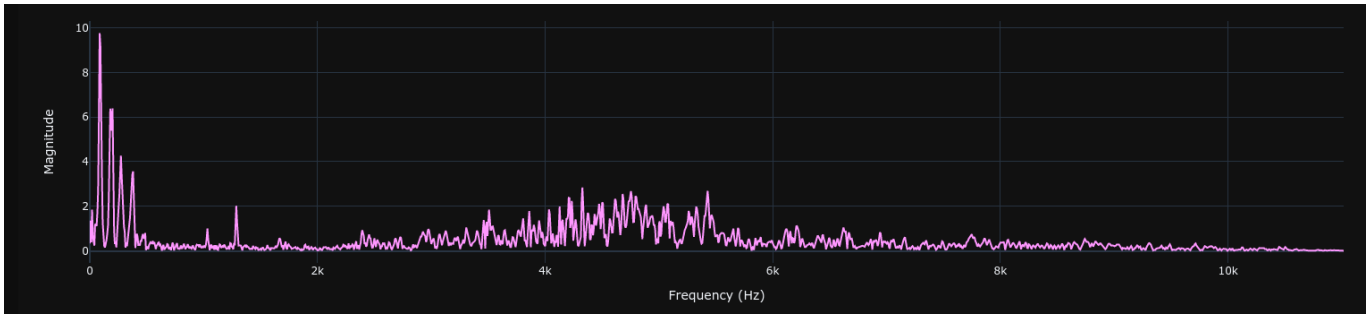
(f) ”dziewięć”

Rysunek 9: Spektrogramy dla różnych wyrazów

Na spektrogramach łatwo jest znaleźć momenty, w których wymawiane są samogłoski, albo spółgłoski szczelinowe. Samogłoski zwykle zajmują całe spektrum, co widać na rysunku 9 jako pionowe rozjaśnienia. Widać też, że każda samogłoska ma trochę inny "obraz". Spółgłoski szczelinowe można rozpoznać jako rozjaśnienia na wyższych częstotliwościach. Na przykład dla "z" w słowie "zid", albo "ć" w słowie "dziewięć".

3.6 Rozpoznanie spółgłoski szczelinowej

Na rysunku 10 można zobaczyć częstotliwości odpowiadające głosce "ć" na końcu słowa "dziewięć". Widać, całkiem dużą aktywność w przedziale 2kHz-6kHz, która nie dzieje się w pozostałych ramkach.



Rysunek 10: Caption