# Sprawozdanie z projektu I

Analiza i przetwarzanie dźwięku

Wojciech Klusek, Aleksander Kuś

31.03.2023

## 1 Opis aplikacji

W tej sekcji przedstawiony zostanie opis interfejsu aplikacji i użytych rozwiązań.

### 1.1 Wybór technologii

Aplikacja została napisana w języku Python z wykorzystaniem Jupyter Notebook. Wybór języka został podyktowany łatwością pisania rozwiązań w tym języku i dużą dostępnością bibliotek do rysowania wykresów oraz wczytywania plików audio.

Do wczytania plików `.wav` użyliśmy biblioteki `wave`, która pozwala w łatwy sposób wgrać plik `.wav` i przetworzyć go do formy umożliwiającej dalszą analizę. Do rysowania wykresów użyliśmy biblioteki `matplotlib` ze względu na bardzo dobrą dokumentację i łatwość w tworzeniu wykresów. Do działania na tablicach wykorzystaliśmy bibliotekę `numpy` ze względu na łatwość użycia i nasze doświadczenie z tym rozwiązaniem.

Wykorzystanie technologii Jupyter Notebook pozwala nam w łatwy sposób generować raporty z wymaganymi informacjami i wykresami dla konkretnych plików audio bez konieczności tworzenia graficznego interfejsu użytkownika. Wszystkie rysunki przedstawione w raporcie zostały wygenerowane z notatnika naszego programu z jądrem Python w wersji 3.9.13 z zainstalowanymi wyżej wymienionymi bibliotekami.

### 1.2 Opis parametrów

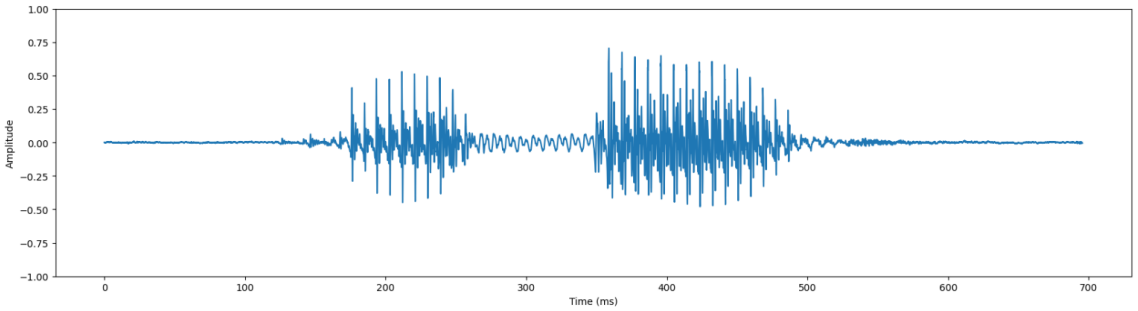
W drugiej komórce notatnika znajdują się parametry obliczeń, takie jak:

- FRAME_DURATION - długość pojedynczej ramki w milisekundach
- INPUT_FILE - nazwa pliku wejściowego z rozszerzeniem `.wav`
- ZCR_SILENCE_THRESHOLD i VOLUME_SILENCE_THRESHOLD - wartości brane pod uwagę przy detekcji ciszy
- LAG - parametr brany pod uwagę w obliczaniu częstotliwości tonu podstawowego

### 1.3 Opis interfejsu

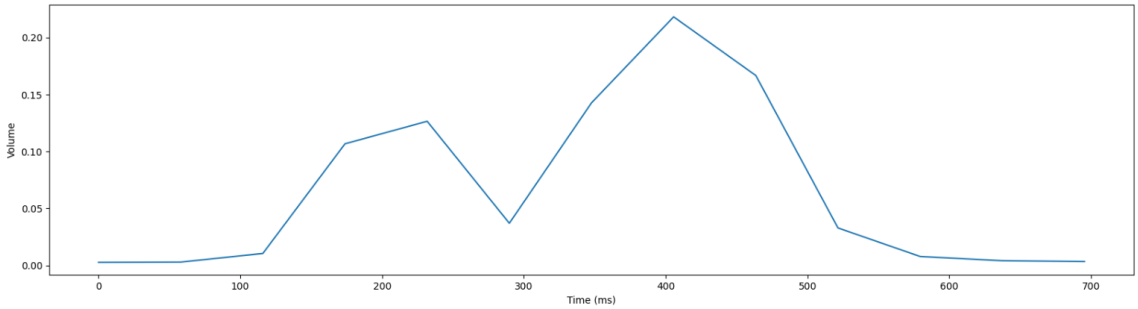
W dalszych częściach aplikacji kolejne komórki odpowiadają za obliczanie i demonstrację poszczególnych cech sygnału dźwiękowego. Po uruchomieniu notatnika dla przykładowego nagrania wykonanego w sali laboratoryjnej `aba_1.wav` program pokazuje następujące rezultaty:

1. Przebieg czasowy pliku:



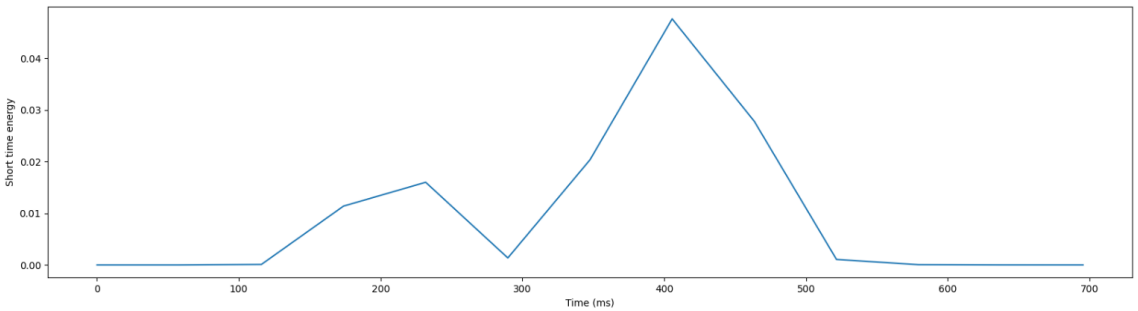

Rysunek 1: Wykres przebiegu czasowego dla przykładowego pliku.

2. Głośność:



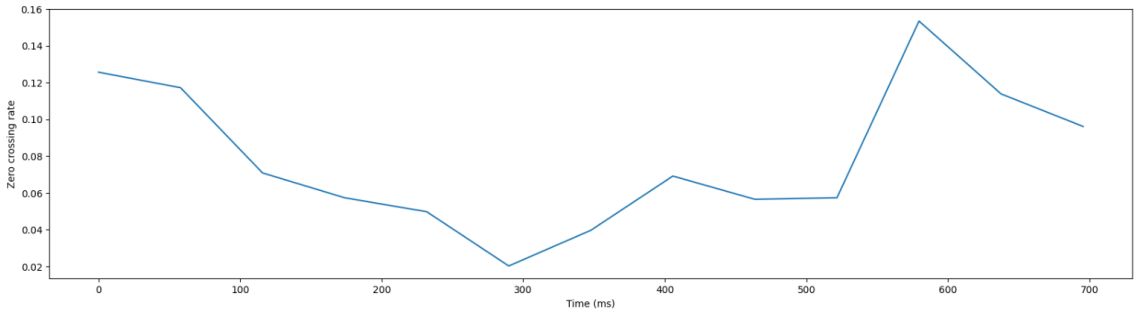

Rysunek 2: Wykres głośności dla przykładowego pliku.

3. Short time energy (STE):



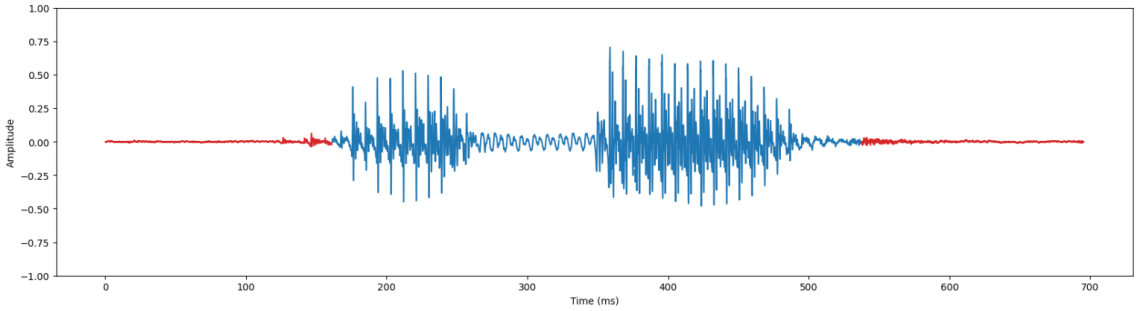

Rysunek 3: Wykres STE dla przykładowego pliku.

4. Zero crossing rate (ZCR):



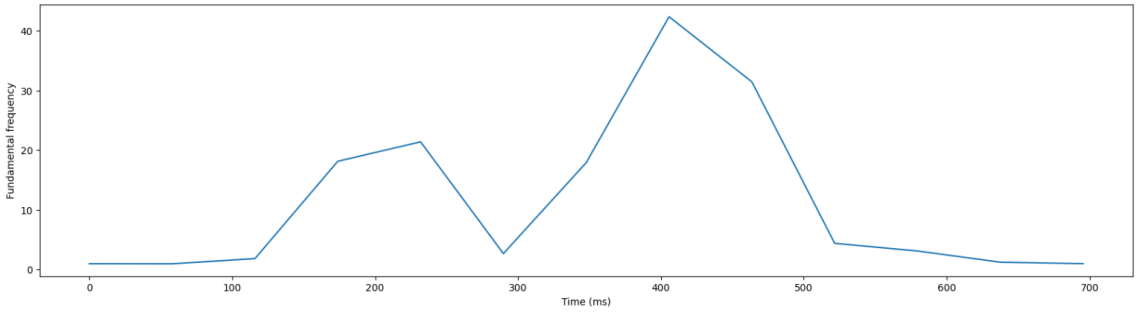

Rysunek 4: Wykres ZCR dla przykładowego pliku.

5. Detekcja ciszy:




Rysunek 5: Detekcja ciszy dla przykładowego pliku.

6. Częstotliwość tonu podstawowego:




Rysunek 6: Wykres częstotliwości tonu podstawowego dla przykładowego pliku.

Na powyższych rysunkach przedstawiono wyniki działania naszego programu dla przykładowego pliku wejściowego. Na rysunku 1 przedstawiono przebieg czasowy pliku audio z dokładnością dla każdej próbki. Na rysunkach 2, 3 i 4 przedstawiono kolejno głośność, short time energy oraz zero crossing rate w dziedzinie czasu dla każdej ramki. Rysunek 5 zawiera przebieg czasowy pliku z zaznaczonymi na czerwono wykrytymi obszarami ciszy. Rysunek 6 zawiera wykres częstotliwości tonu podstawowego. W notatniku pod wykresami 2, 3, 4 i 6 znajduje się informacja o średniej wartości danej cechy dla wszystkich ramek.

# 2 Opis metod

## 2.1 Głośność

Głośność dźwięku jest określana jako pierwiastek średniej energii sygnału audio w danym fragmencie:

$$p(n) = \sqrt{\frac{1}{N} * \sum_{i=0}^{N-1} s_n^2(i)}$$

Można zauważyć, że głośność jest bezpośrednio powiązana z amplitudą fali dźwiękowej. Bez względu na to, czy używa się formy znormalizowanej czy nie, różnice między wykresami głośności dotyczą jedynie zakresu wartości, a nie samej zależności między amplitudą a głośnością.

## 2.2 Short time energy

Podniesienie głośności do kwadratu daje miarę określaną jako Short Time Energy (STE):

$$STE(n) = \frac{1}{N} * \sum_{i=0}^{N-1} s_n^2(i) = p(n)^2$$

Short time energy umożliwia odróżnienie fragmentów mowy, które są dźwięczne (np. samogłoski) od tych, które są bezdźwięczne (np. spółgłoski). Wartości short time energy dla bezdźwięcznych fragmentów mowy są znacznie niższe niż dla dźwięcznych fragmentów, co pozwala na ich wykrycie i odróżnienie.

## 2.3 Zero crossing rate

Liczba przejść amplitudy przez zero w jednej ramce sygnału audio:

$$Z(n) = \frac{2}{N} * (\sum_{i=1}^{N-1} |sign(S_n(i)) - sign(S_n(i-1))|)$$

Ta miara jest bardzo skuteczna w rozpoznawaniu fragmentów mowy, które są dźwięczne lub bezdźwięczne, ponieważ typowy bezdźwięczny dźwięk ma niską energię, ale wysoką wartość współczynnika ZCR (ang. Zero Crossing Rate), który mierzy, jak często sygnał audio zmienia znak w danym fragmencie. Kombinacja współczynnika ZCR i głośności pozwala na sklasyfikowanie bezdźwięcznych fragmentów mowy o niskiej energii jako ciszy.

## 2.4 Częstotliwość tonu podstawowego

- Funkcja autokorelacji:

$$R_n(l) = \sum_{i=0}^{N-l-1} s_n(i) * s_n(i+l)$$

- Funkcja AMDF (Average Magnitude Difference Function)

$$R_n(l) = \sum_{i=0}^{N-l-1} s_n(i) * s_n(i+l)$$

Częstotliwość tonu podstawowego F0 (Fundamental Frequency) to podstawowa częstotliwość drgań struny głosowej lub innych źródeł dźwięku, która determinuje wysokość dźwięku. F0 jest mierzona w hercach (Hz) i określa, ile razy na sekundę drgają struny głosowe lub inny źródło dźwięku, co skutkuje powstawaniem odpowiedniej wysokości dźwięku.

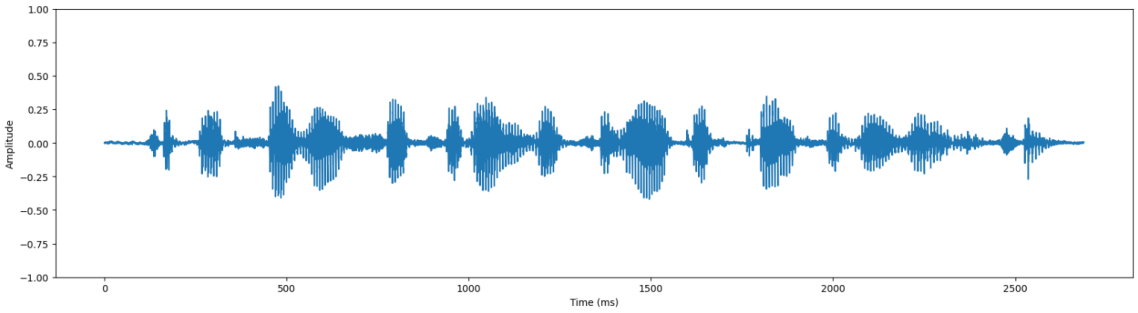
## 2.5 Detekcja ciszy

Rozpoznawanie ciszy odbywa się poprzez wykorzystanie dwóch parametrów: głośności (Volume) i ZCR (Zero Crossing Rate). Aby zaklasyfikować ramkę jako ciszę, należy sprawdzić, czy wartości głośności i ZCR dla tej ramki są poniżej pewnego poziomu. My wybraliśmy parametry: 0.5 dla ZCR oraz 0.02 dla głośności. Oznacza to, że jeśli dla danej ramki ZCR i Volume będą mniejsze od tych wartości, to taka ramka zostanie zaklasyfikowana jako cisza.

# 3 Prezentacja wyników działania

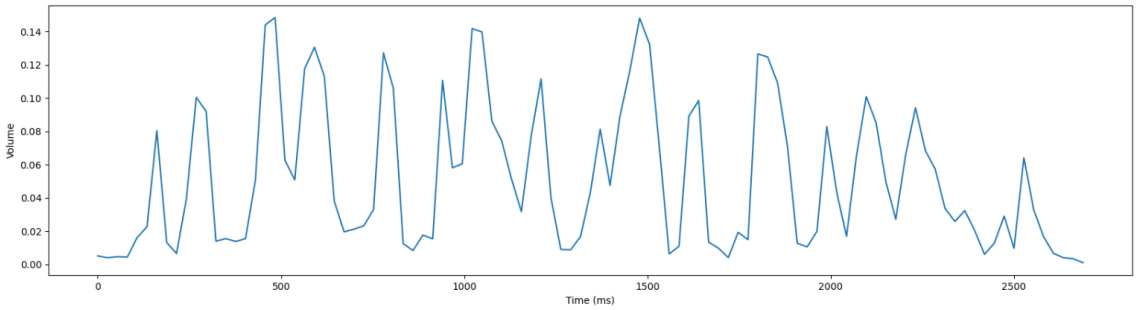
W tej sekcji zaprezentujemy wyniki działania naszego programu przy zmianie długości ramki dla jednego pliku.

Poniżej przedstawiono, jak zmieniają się generowane wykresy dla długości ramek 10ms i 40ms dla tego samego pliku audio. Na rysunku 7 przedstawiono przebieg czasowy pliku audio, dla którego wykonywany jest ten test. Przebieg czasowy jest niezależny od długości ramki.

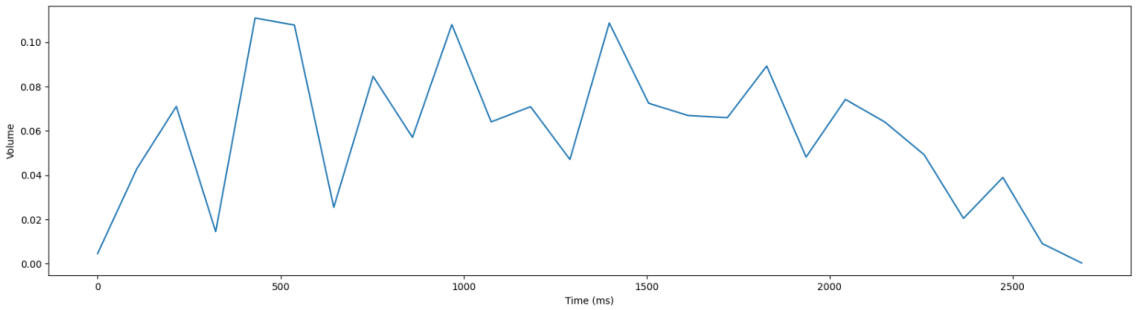



Rysunek 7: Wykres przebiegu czasowego dla pliku przy testowaniu długości ramki.

## 3.1 Głośność

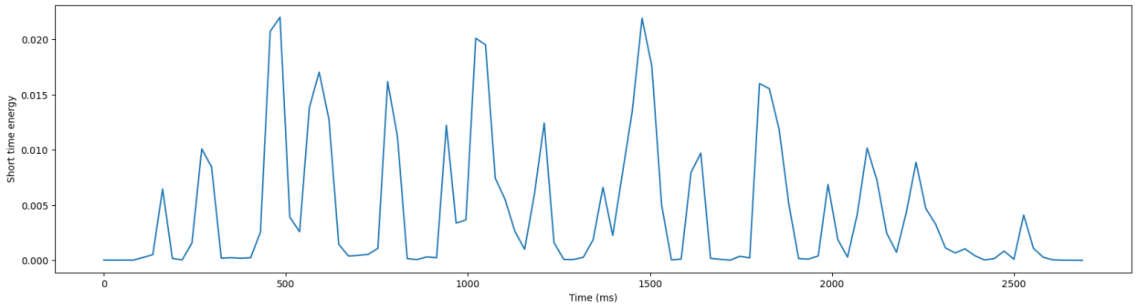



Rysunek 8: Wykres głośności dla długości ramki 10ms.

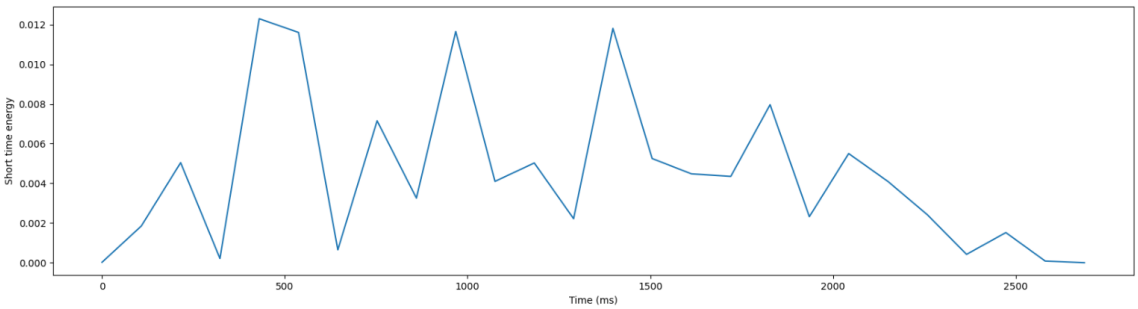



Rysunek 9: Wykres głośności dla długości ramki 40ms.

## 3.2 Short time energy

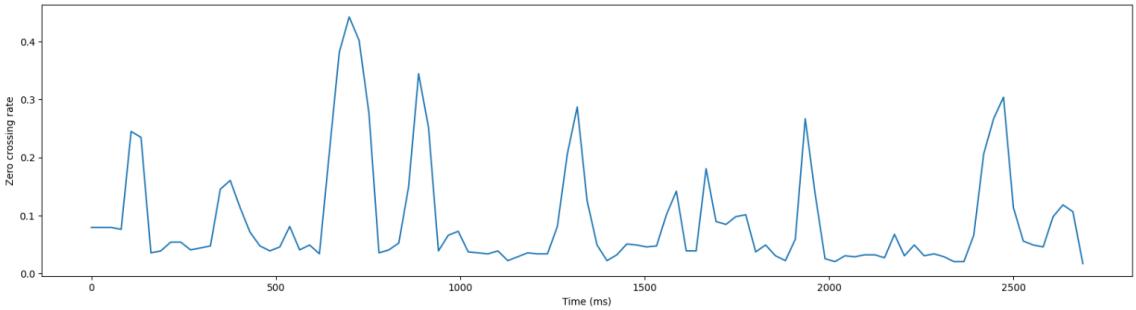



Rysunek 10: Wykres short time energy dla długości ramki 10ms.

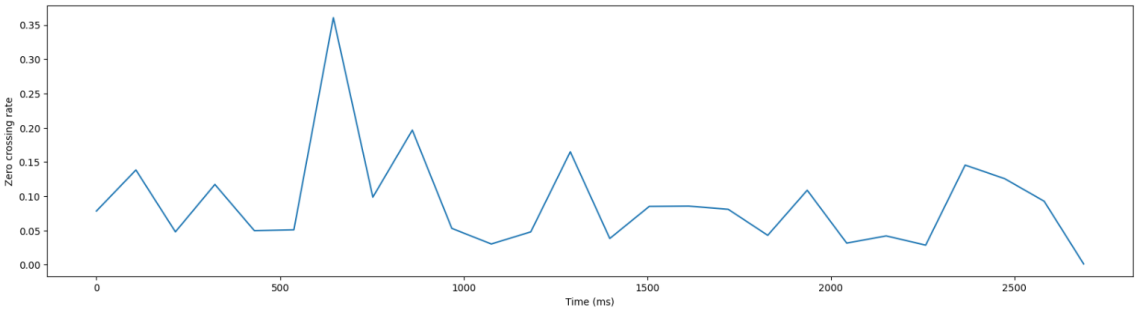



Rysunek 11: Wykres short time energy dla długości ramki 40ms.

## 3.3 Zero crossing rate

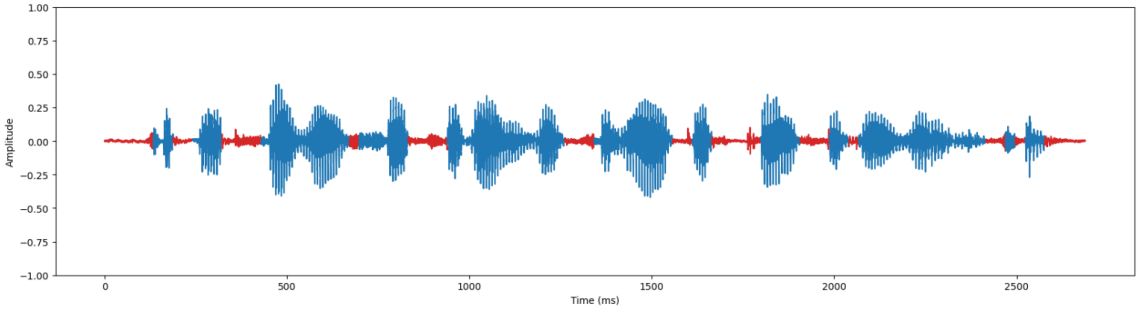



Rysunek 12: Wykres zero crossing rate dla długości ramki 10ms.

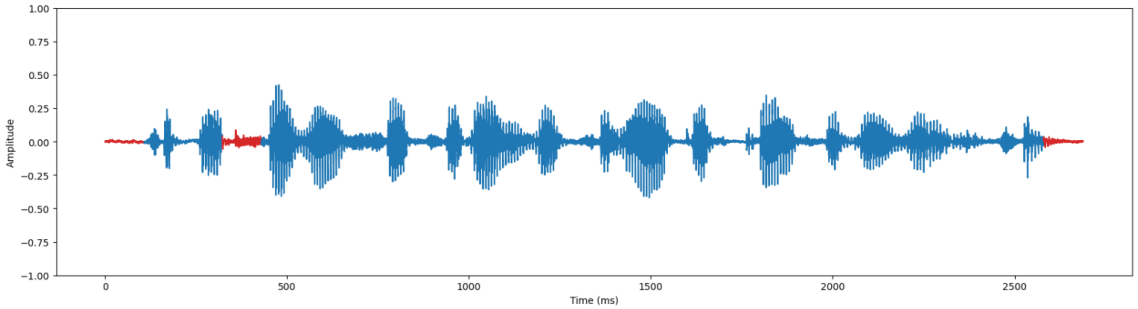



Rysunek 13: Wykres zero crossing rate dla długości ramki 40ms.

## 3.4 Detekcja ciszy

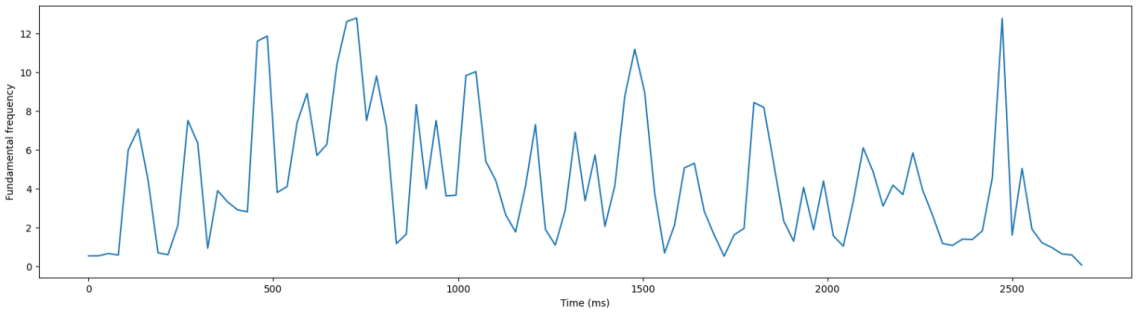



Rysunek 14: Detekcja ciszy dla długości ramki 10ms.

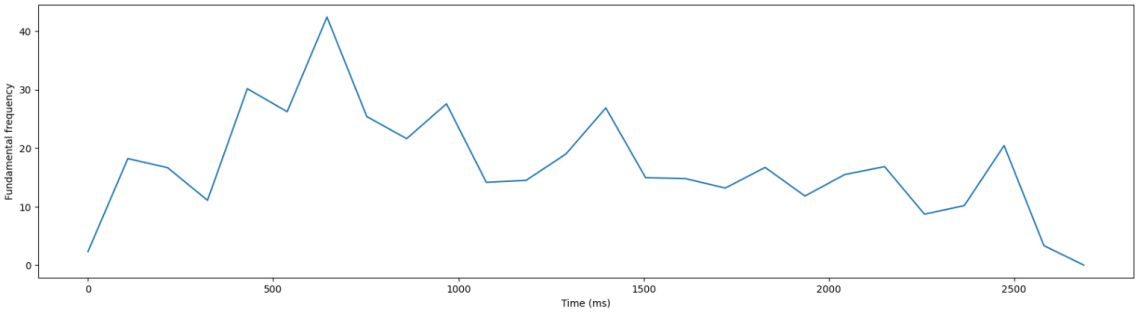



Rysunek 15: Detekcja ciszy dla długości ramki 40ms.

## 3.5 Częstotliwość tonu podstawowego




Rysunek 16: Wykres częstotliwości tonu podstawowego dla długości ramki 10ms.




Rysunek 17: Wykres częstotliwości tonu podstawowego dla długości ramki 40ms.

### 3.6 Porównanie

Porównując ze sobą wykresy przedstawione na rysunkach 8 i 9, 10 i 11, 12 i 13, 14 i 15 oraz 16 i 17 możemy powiedzieć, że zwiększenie długości ramki powoduje na spadek dokładności obliczanych cech. Dla przykładu w detekcji ciszy dla ramek 10-milisekundowych na rysunku 14 wykryto więcej momentów, które mogłyby zostać uznane jako cisza, niż na rysunku 15 dla ramek 40-milisekundowych.

## 4 Wnioski

- Aplikacja została napisana w języku Python z wykorzystaniem bibliotek takich jak wave, matplotlib i numpy, co umożliwiło łatwe wczytywanie plików audio, rysowanie wykresów i efektywne operowanie na tablicach.
- W przypadku notatnika Pythona, aplikacja może być łatwo rozwijana i dostosowywana do potrzeb użytkownika, umożliwiając bardziej zaawansowane analizy i manipulacje na danych audio.
- Wyświetlanie parametrów na poziomie ramki i w dziedzinie czasu na wykresie umożliwia użytkownikowi dokładną analizę i interpretację wyników, a także wykrycie nieprawidłowości w pliku audio.