

1. Wczytanie i odtworzenie sygnału audio

```
class AudioFile:
    """ https://stackoverflow.com/a/6951154/7101876 """
    chunk = 1024

    def __init__(self, file):
        """ Init audio stream """
        self.file = file
        self.waveform = wave.open(self.file, 'rb')
        self.audio = pyaudio.PyAudio()
        self.stream = self.audio.open(
            format=self.get_format(),
            channels=self.waveform.getnchannels(),
            rate=self.waveform.getframerate(),
            output=True
        )

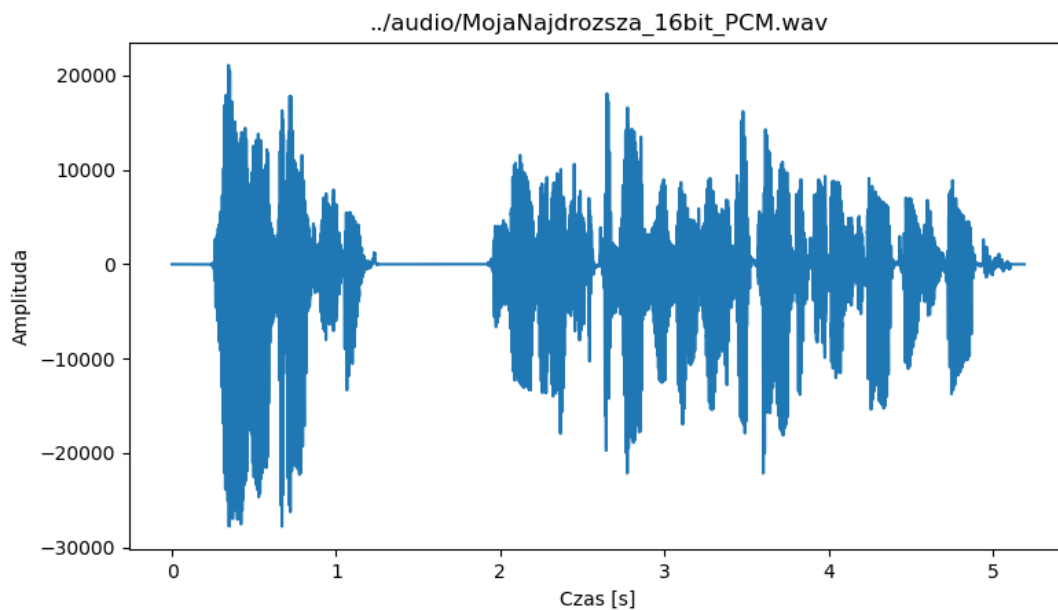
    def get_format(self):
        """ Get format """
        return self.waveform.getsampwidth() * 8

    # region Playing file

    def play(self):
        """ Play entire file in chunks """
        data = self.waveform.readframes(self.chunk)
        while len(data) > 0:
            self.stream.write(data)
            data = self.waveform.readframes(self.chunk)
```

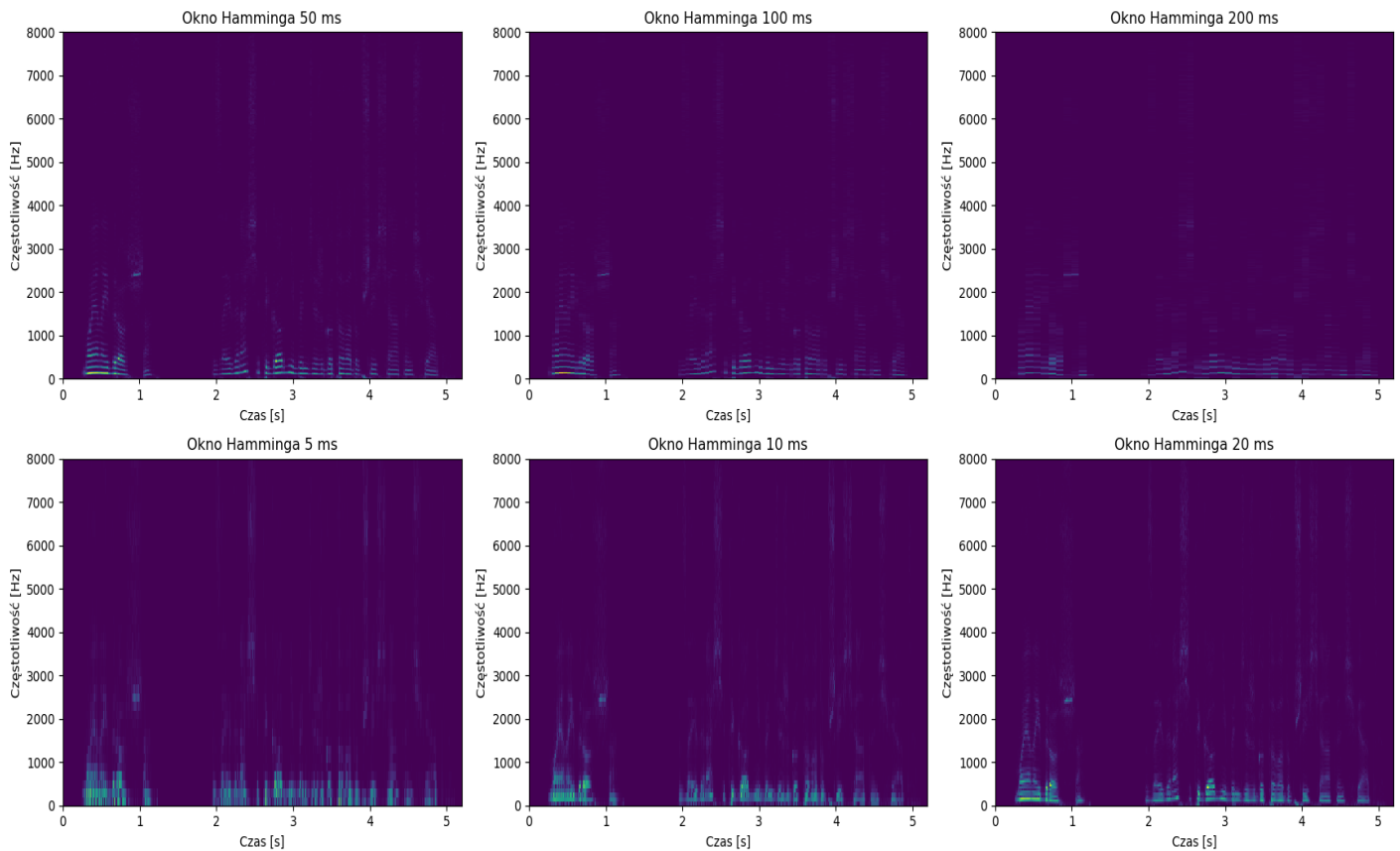
Rysunek 1. Kod ładujący i odtwarzający plik

2. Oscylogram



Rysunek 2. Oscylogram analizowanego pliku

3. Spektrogramy



Rysunek 3. Spektrogramy dla różnych długości okna

4. Uzasadnienie najlepszej długości okna

Odpowiednią długością okna dla analizy sygnału mowy z opcji podanych powyżej będzie okno o długości 20ms. Spektrogram analizy wykonanej z wykorzystaniem takiego okna ma najlepszy stosunek rozdzielczości w dziedzinie częstotliwości. Jednocześnie można dostrzec już dość wyraźnie poszczególne składowe analizowanego sygnału, co jest niemożliwe w przypadku okien o krótszej długości. Natomiast w przypadku dłuższych okien analiza wyższych składowych staje się prawie niemożliwa.