Worcester Polytechnic Institute

ECE Program

Discrete-Time Signal And System Analysis

Project 1: Collect, Store & Analyze Human Speech Signal

Group 01

Mingxiao Zhao

Dexuan Tang

Lin Guan

Date Submitted: 2/3/2023
Date Completed: 2/3/2023
Course Instructor: Prof. Bashima Islam

Table of Contents:

1. Introduction

This project aims to help us learn about the processing of human speech audio signals including collecting, storing, and analyzing. It is the first project in the course ECE2312, focusing on mastering sophisticated software tools such as MATLAB or Python. In this lab, we used MATLAB to create an audio recorder object which records speech, produces plots of the speech's amplitude vs seconds in time domain, generate spectrogram that including energy as an axis, saves the recorded speech as WAV files, and/or loads WAV files for analysis. Also, a stereo speech file is generated by adding a column of zeros for further exploration.

2. Methodology

All code, plots, and WAV files can be found via the following Git link:
https://github.com/dexuantang/ECE2312_project_1_Group_1_C23

2.A MATLAB code explained:

At the beginning of the code, global variables are set. These variables are the sample speed of 44100 samples/sec which covers the human hearing range of 20 Hz to 20000 Hz because it is higher than the Nyquist frequency of 20000 Hz. 8 bits per channel is used and it divides the analog input signal into 512 steps. The number of channels is set to 1. The record length is set to 7 seconds since this is long enough for recording all three of the clips.

```
clear;
%% Global variable declaration
FS = 44100;
nBits = 8;
nChannels = 1;
record_length = 7;
```

Figure 1. Global variable declaration.

Then, a main function is used that calls each helper function in sequence. The operation sequence of the code is listed below:

- Let the user select audio devices
- Record the first clip
- Plot in time domain and spectrogram
- Save audio as WAV
- Load WAV and plot the spectrogram of the WAV
- Record the second clip
- Plot in time domain and spectrogram
- Save audio as WAV
- Load WAV and plot the spectrogram of the WAV
- Record the third clip
- Plot in time domain and spectrogram
- Save audio as WAV
- Load WAV and plot the spectrogram of the WAV
- Convert the third clip into stereo
- Save the stereo audio as a WAV

Inorder to let the user select the correct audio devices, the ”audiodevinfo” function is used to print out all available devices. The user is then prompted to type the correct audio device ID in the command window.

Then an “audiorecorder” object is created using the user selected input device. It then starts recording. During the recording, the program execution is blocked for the recording length using the “pause” function. This ensures that the recording process finishes correctly.

Next, the audio data is plotted in the time domain. Samples are converted to seconds using the “linspace” function to create an evenly spaced time matrix of the recording duration that has the same matrix length as the audio samples. Audio data is then plotted against the time matrix. Spectrogram is also plotted using the “spectrogram” function and the y-axis is limited to 8KHz.

Stereo audio is generated by appending zeros as the second column to the audio data.

The audio data is saved and loaded using the “audiwrite” and “audioread” function. File paths are obtained using the “uiputfile” and the “uigetfile” functions.

3. Results / Discussion

3. A Time domain plots are generated for each of the three phrases:

“The quick brown fox jumps over the lazy dog”

“We promptly judged antique ivory buckles for the next prize”

“Crazy Fredrick bought many very exquisite opal jewels”

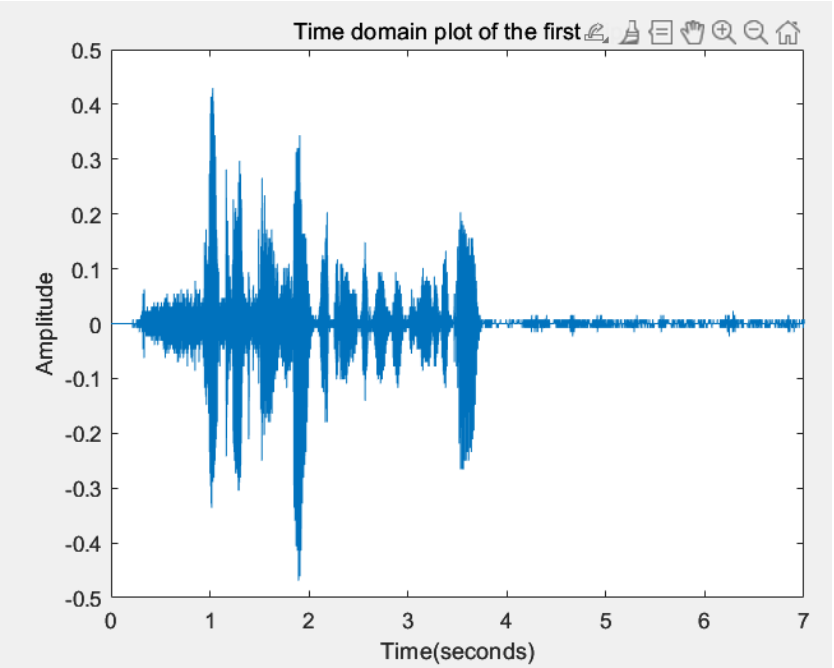

Figure 2: Plot of Time Domain for “The quick brown fox jumps over the lazy dog”

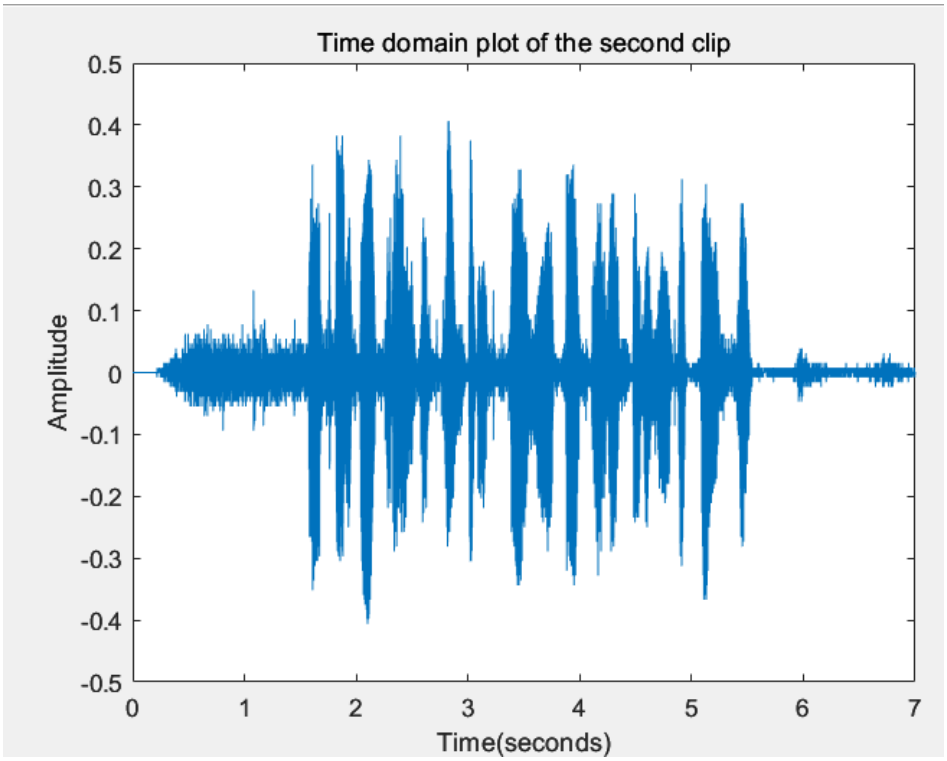

Figure 3: Plot of Time Domain for "We promptly judged antique ivory buckles for the next prize"

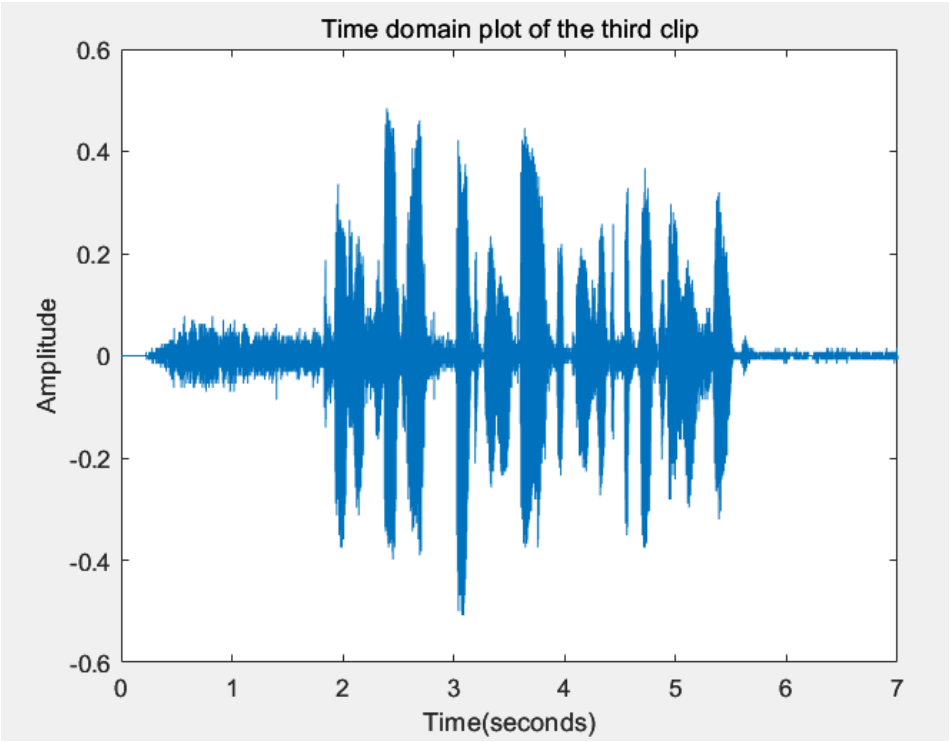

Figure 4: Plot of Time Domain for "Crazy Fredrick bought many very exquisite opal jewels"

3.B Visualization Via Spectrogram

3.B.1 Analysis of "do", "re", "mi", "fa", "so", "la", "ti", and "do".

By looking at the portion of the signal above 4000 Hz and comparing the spectrogram to one that we generated for the sounds, we were able to determine the following time segments:

- 0.75 to 1.5 second: do
- 1.5 to 2 second: re
- 2 to 2.25 second: mi
- 2.25 to 2.5 second: fa
- 2.5 to 2.75 second: so
- 2.75 to 3.25 second: la
- 3.25 to 3.75 second: ti
- 3.75 to 4.75 second: do

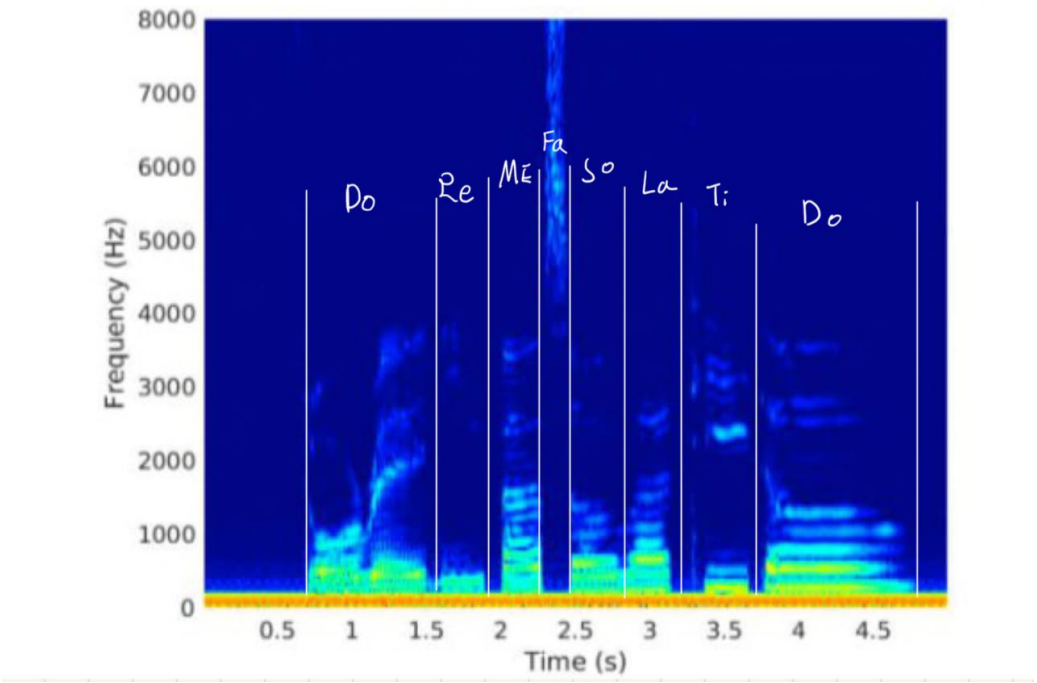

Figure 5. Spectrogram of "do", "re", "mi", "fa", "so", "la", "ti", and "do" annotated with each sound.

3.B.2 Analysis of the three clips recorded in section 3.

Sounds and words that are easily identifiable are annotated for the following spectrograms:

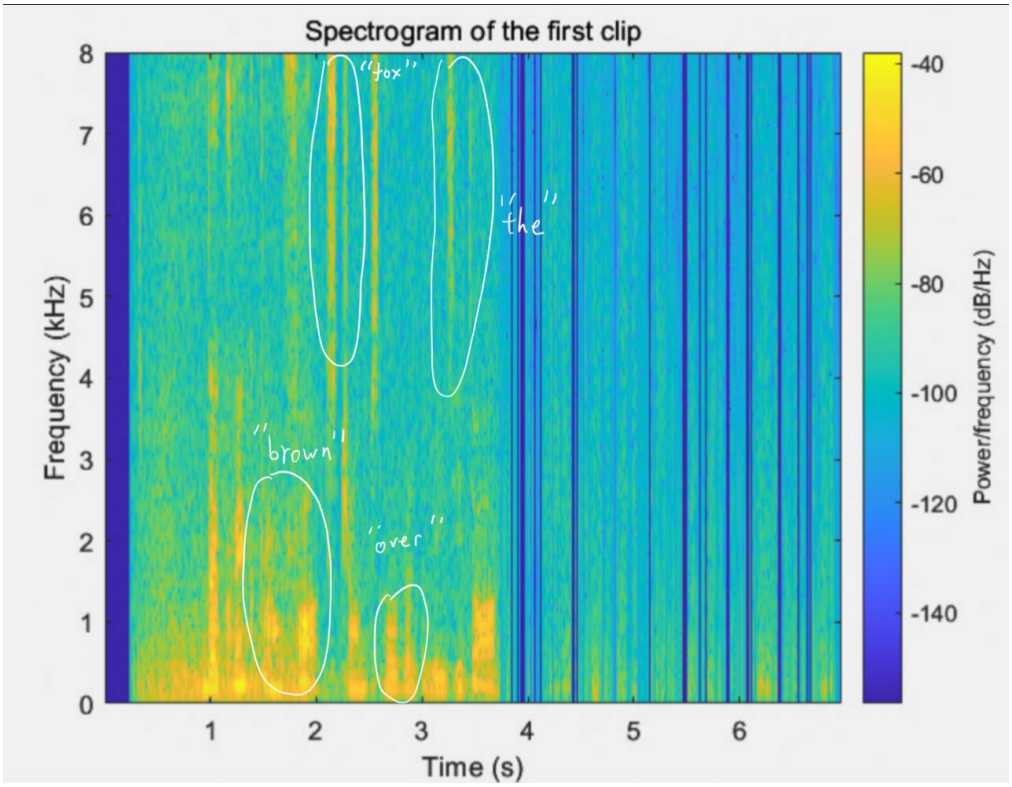

Figure 6: Spectrogram for "The quick brown fox jumps over the lazy dog"

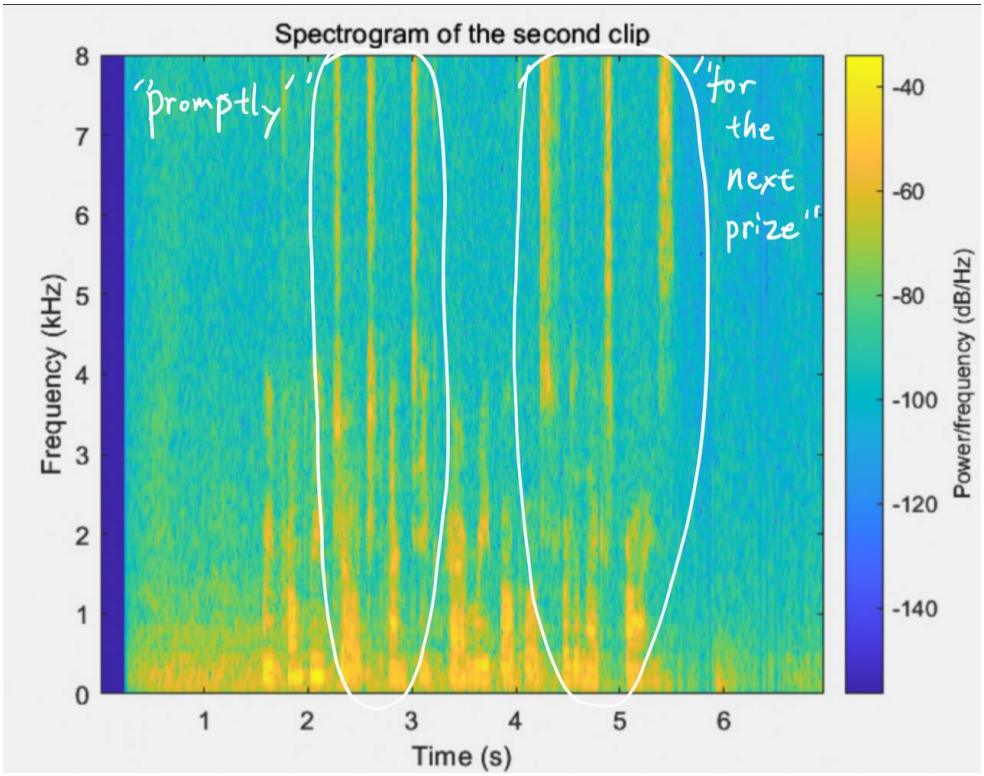

Figure 7: Spectrogram for "We promptly judged antique ivory buckles for the next prize"

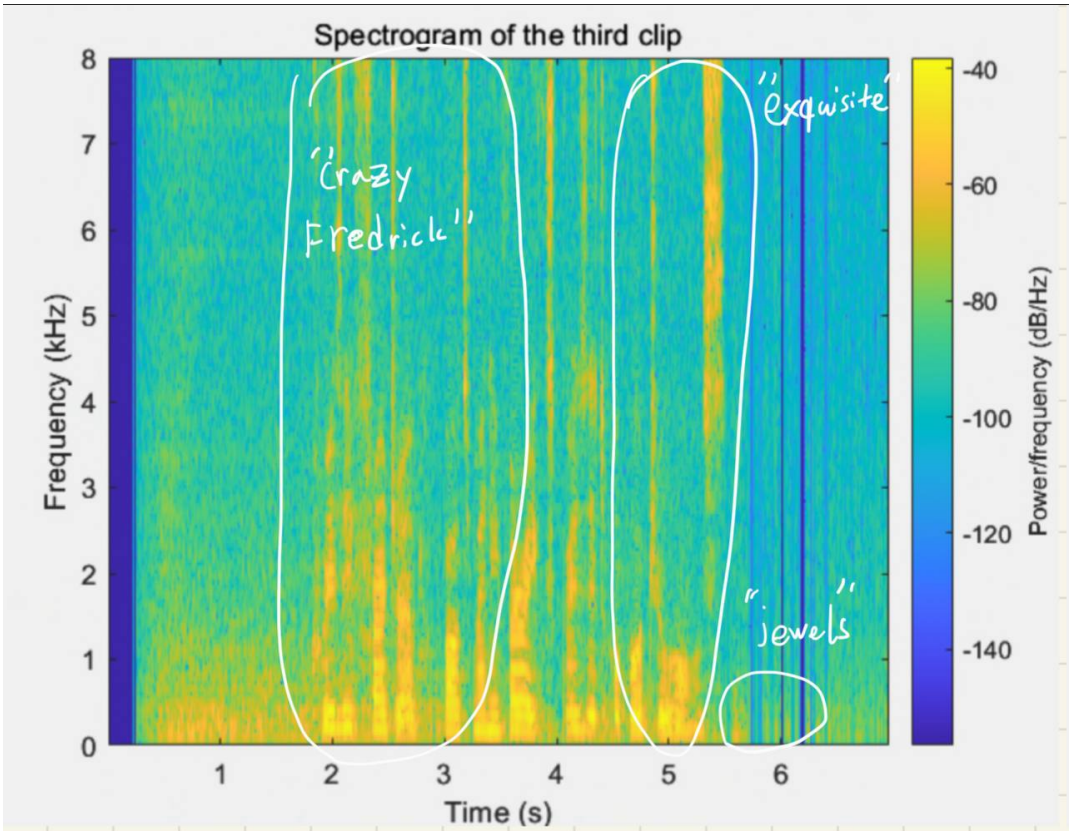

Figure 8: Spectrogram for "Crazy Fredrick bought many very exquisite opal jewels"

We noticed that words with a lot of sharp sounding consonants or round sounding vowels are easily identifiable.

3.C Saving and Loading WAV files

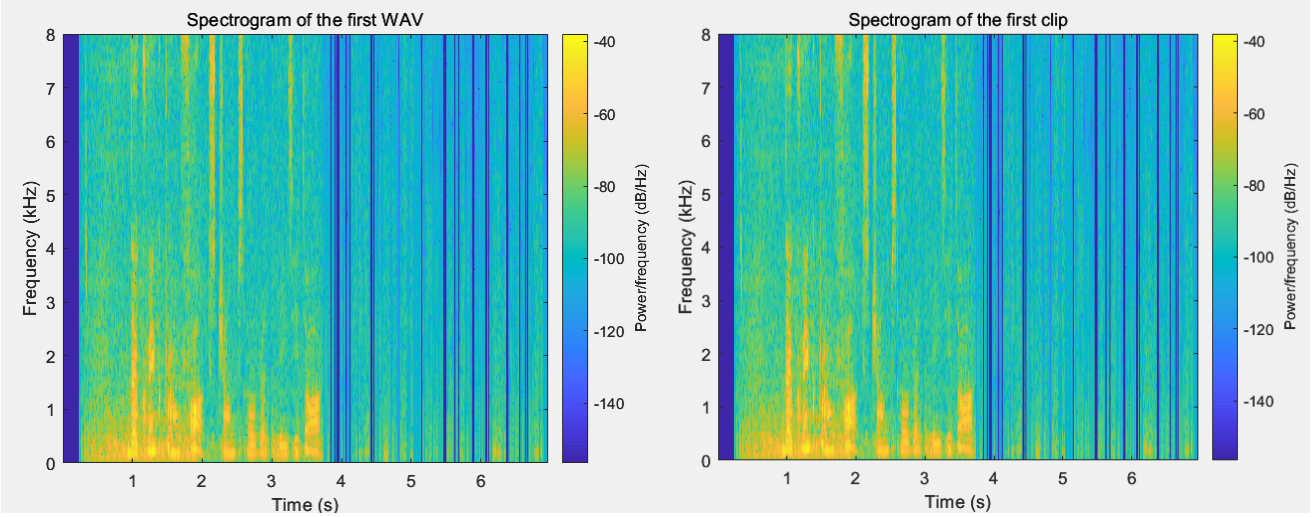

Figure 8: 1st Comparison between Spectrograms of WAV and Recorded Input

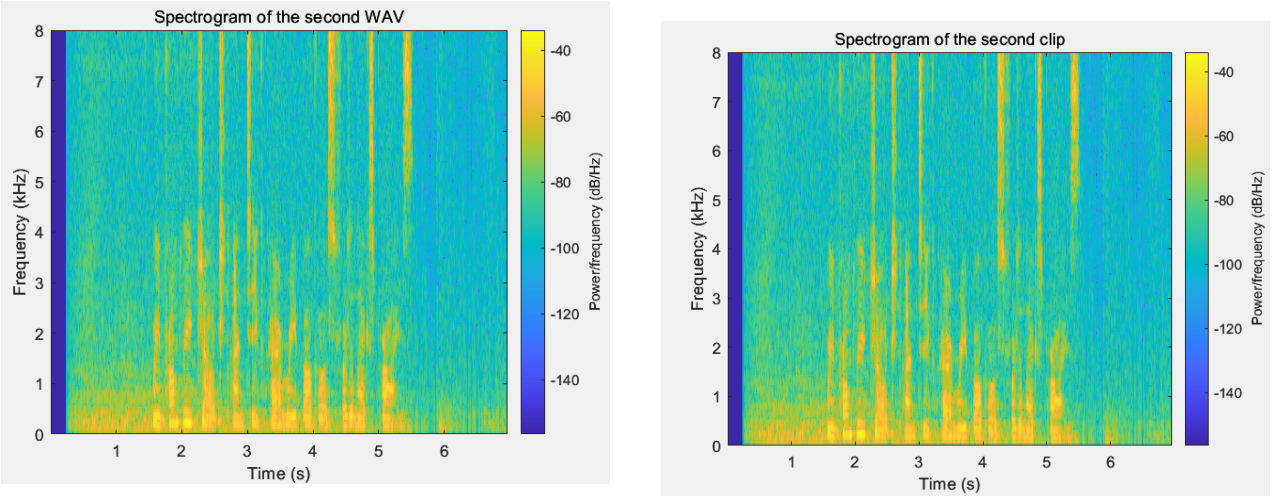

Figure 9: 2nd Comparison between Spectrograms of WAV and Recorded Input

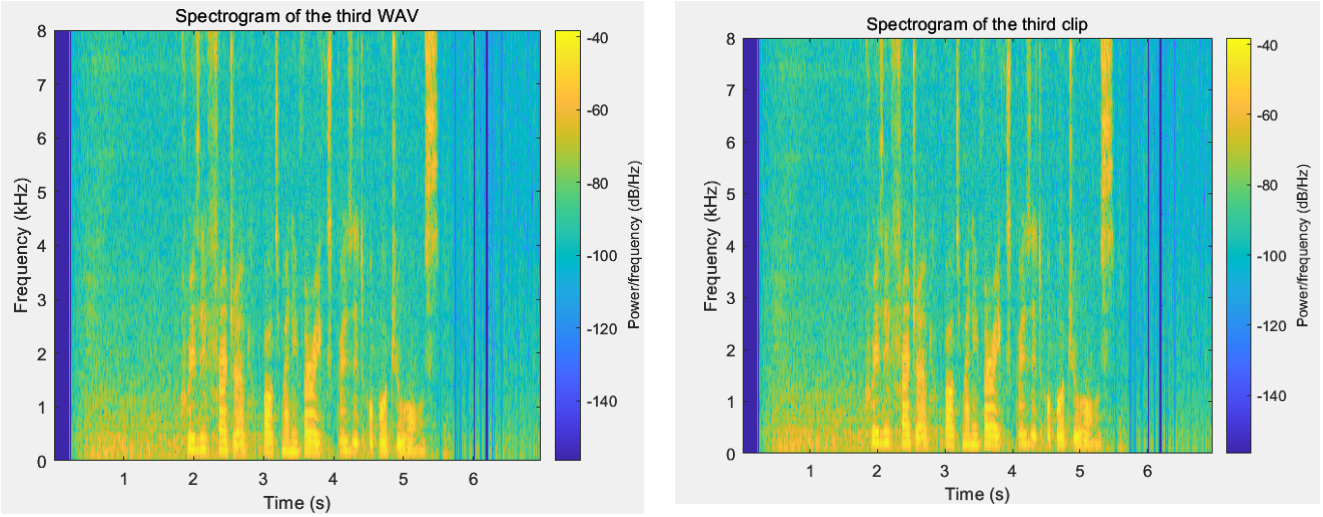

Figure 10: 3rd Comparison between Spectrograms of WAV and Recorded Input

Through 3 sets of comparisons, we can see the spectrograms of WAV files and record inputs are similar enough to be claimed as identical. This is because the WAV format is lossless and does not compress the audio data for storage.

3.D Fun with Stereo Speech File

The Stereo WAV file can be found via the following link:
https://github.com/dexuantang/ECE2312_project_1_Group_1_C23/tree/main/wavs

The stereo file only has sound through the left speaker. This makes sense because one channel is just a column of zeros. However, there are occasional clicking sounds through the right channel that could be an artifact of the output device of the computer.