Voice Recognition System for Speech Impaired People
Domain:- Speech Processing

PROJECT REPORT

Submitted by

BL.EN. U4AIE19050	Peddinti Srinivasa Sri Kalyan
BL.EN. U4AIE19054	Ruthvik K
BL.EN. U4AIE19069	VISHAL.C

in partial fulfilment for the award of the degree
0f

BACHELOR OF TECHNOLOGY

IN

"ARTIFICIAL INTELLIGENCE" ENGINEERING

For the subject

19MAT117- Mathematics for Intelligent Systems

AMRITA SCHOOL OF ENGINEERING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

BANGALORE 560 035

July-2020

AMRITA VISHWA VIDYAPEETHAM

ABSTRACT

This version Voice Recognition System mainly focuses on people with cerebral paralysis i.e. those people who can produce only sound but they do not pronounce an understandable structure in their phrases. When we hear someone and we can't understand what he is saying, but we can connect the few sounds that we understand with the words we know and taking into account its form, tone, accent, and pronunciation, we can assign an equivalent word. Based on certain characteristics related to the frequency, we can find out the actual words the speech impaired person wants to tell. Our aim is to build such a system based on speech processing which can be efficient in helping us to understand those people who are Impaired. MATLAB and Python has the flexibility to implement complex algorithms for digital processing of signal and hence can be used to realize Voice Recognition. We can use MATLAB and Python for digitalization of a sound file, as we working in the frequency domain, we represent our data into signals using Discrete Fourier Transformation (DFT). A way of applying DFT is by using Fast Fourier Transformation (FFT). As the voice signals has many problems such as noise and other distortions, etc. We first try to stabilize the signal by normalization of the signal and by using FFT, then we get the main signal. To implement the voice recognition, we will use the range of 500 Hz to 4000 Hz and try our best to get the best results. We will use power spectrum analysis and power spectrogram to compare the similarities between two or more voice samples and also print the word (Speech to Text) using Speech Recognition module in python.

Keywords: - Fast Fourier Transformation, Discrete Fourier Transformation, Power Spectral Analysis, Spectrogram, Speech to text.

INTRODUCTION

When we hear someone and we cannot understand what he is saying, we connect the few sounds that we understand with the ones that we know, and taking into account its form, tone, accent and pronunciation, we assign an equivalent word. This presents a problem, how can we know what is the equivalent word to the incomplete phrase? When we spend a great deal of time with the handicapped person, our ear can find some characteristics related with his frequency in the way that the person talks. By seeing the graphics of frequencies (spectrograph) we can find the actual word that impaired person has said by comparing the with the standard word.

MATLAB and Python has the flexibility to implement complex algorithms for digital processing of signal and hence can be used to realize Voice Recognition. In our project we used MATLAB to do spectral analysis of audio signal and we used Python for analysis of spectrograph.

Analysis of any signal can be done mainly in two domains namely as: Time Domain Analysis and Frequency/Spectral Doman analysis. While time domain analysis of a signal tells mainly about the amplitude of the signal, spectral or frequency domain analysis tells about the individual frequency component present in the signal. Frequency domain also tells about the phase information of the system at different frequencies. Fourier Transform also has several applications in speech processing. Fourier transform is also helpful in filter designing in order to remove any specific band or frequency component present in the speech signal. Fourier Transform can also act as a feature vector for speaker recognition application. Finally we use Spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time which an be used to identify spoken words phonetically, and also to analyse.

We also used Speech Recognition module in Python to convert the given Speech into

METHODOLOGY

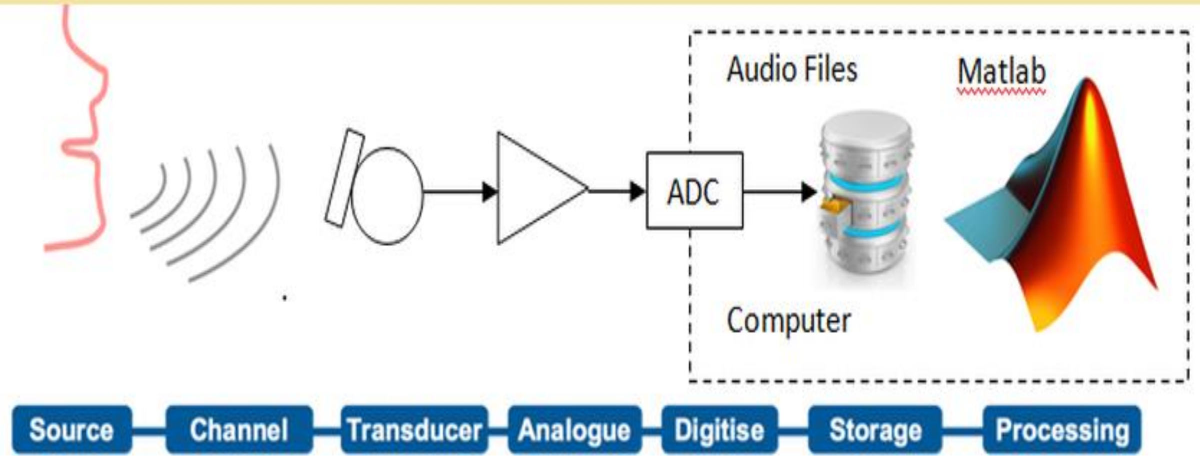
Spectrogram Analysis in Python: -

- Loading Audio into Python:
  import scipy.io.wavfile as wav package is used for audio file .wav handling.
- Visualizing Audio:
  The scipy package provides all built functions required for computing and visualizing FFT.
- Visualizing Audio:
  The scipy package provides all built functions required for computing and visualizing FFT.
  This visualization is called the time-domain representation of a given signal. This shows us the loudness (amplitude) of sound wave changing with time.
  These amplitudes are not very informative, as they only talk about the loudness of audio recording. To better understand the audio signal, it is necessary to transform it into the frequency-domain. The frequency-domain representation of a signal tells us what different frequencies are present in the signal. Fourier Transform is a mathematical concept that can convert a continuous signal from time-domain to frequency-domain.
- Fourier Transform (FT)
  An audio signal is a complex signal composed of multiple 'single-frequency sound waves' which travel together as a disturbance(pressure-change) in the medium. When sound is recorded, we only capture the resultant amplitudes of those multiple waves. Fourier Transform is a mathematical concept that can decompose a signal into its constituent frequencies. Fourier transform does not just give the frequencies present in the signal, It also gives the magnitude of each frequency present in the signal.
- Spectrogram
  Suppose you have an audio file in which someone is speaking a phrase (for example: How are you). Your recognition system should be able to predict these three words in the same order (1. 'how', 2. 'are', 3. 'you'). If you remember, in the previous exercise we broke our signal into its frequency values which will serve as features for our recognition system. But when we applied FFT to our signal, it gave us only frequency values and we lost the track of time information. Now our system won't be able to tell what was spoken first if we use these frequencies as features. We need to find a different way to calculate features for our system such that it has frequency values along with the time at which they were observed. Here Spectrograms come into the picture.

Visual representation of frequencies of a given signal with time is called Spectrogram. In a spectrogram representation plot — one axis represents the time, the second axis represents frequencies and the colors represent magnitude (amplitude) of the observed frequency at a particular time. The following screenshot represents the spectrogram of the same audio signal we discussed earlier. Bright colors represent strong frequencies. Similar to earlier FFT plot, smaller frequencies ranging from (0–1kHz) are strong(bright).

- By analyzing two voices that is the standard voice with the experimental voice, if they are similar, we will obtain a similar spectrogram and thus conclude that we have got that word. Here we use Matplotlib to plot the spectrogram and Numpy to deal with arrays, Linear Algebra, summation, appending and truncating values.

For Power Spectral Analysis in MATLAB: -



- Reading audio:
  [y, fs] = audioread(filename): This command is used to read any file format files supported by Matlab. It also returns the sound data and sampling frequency used while recording the signal.
- FFT Computation in Matlab:

Matlab provides built in command to find the Fourier transform of the signal. Several other commands are also required in order to plot the FT of the signal.

1. Y = fft(x, n) : This command is used to determine the FFT of the sequence x and n points. Generally n is the no. greater than length of x. Normally it is chosen in multiple of 2n.
2. Y = abs(x) : This command returns the absolute value of the sequence x.

- Matlab Code for FFT Plot of Speech Signal:
  Code for FFT computation and plotting the single sided spectrum of the input sequence

```
%% Read signal
[data, fs] = wavread('filename’);

%% take Fourier transform and plot single sided spectrum
l = length(data);
NFFT = 2^nextpow2(l);
f = fs/2*linspace(0,1,NFFT/2+1);
xf = abs(fft(data, NFFT));

plot(f, xf(1:NFFT/2+1));
```

Power spectral density (PSD) describes that variation of power present in the signal as a function of frequency. In other words it tells which frequency components contain higher power and which frequency contains lower power. It shows the variation in the form of a graph with frequency of x-axis and power (db/Hz or watt/Hz) on y-axis. There are several ways to compute PSD in theory as well as in Matlab.

PSD estimate using Periodogram Command: PSD calculation can be done by simply by periodogram command in Matlab. It takes data, window and sampling frequency as input and returns the PSD and frequency of the data. Matlab code for PSD calculation using periodogram is given as follows:

```
[psdestx, Fxx] = periodogram(data, rectwin(length(data)), length(data), fs);
plot(Fxx, 10*log10(psdestx)); grid on;
xlabel('Hz'); ylabel('Power/Frequency (dB/Hz)');
title('Periodogram Power Spectral Density Estimate');
```

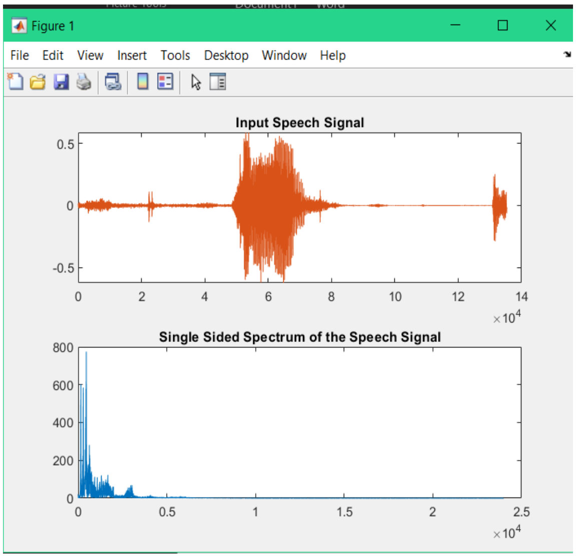
Welch’s Method for PSD Estimate: Welch’s method is an improved way of estimating the Power Spectral Density of input Data. It is better than the periodogram method as it reduces the amount of noise in estimated PSD at the cost of frequency resolution.

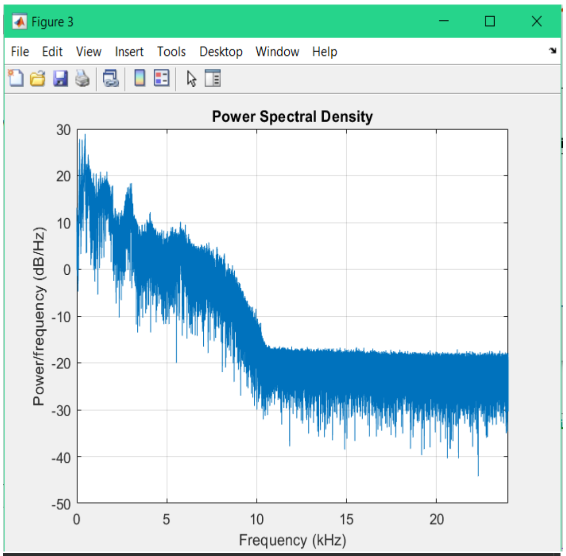
Code:

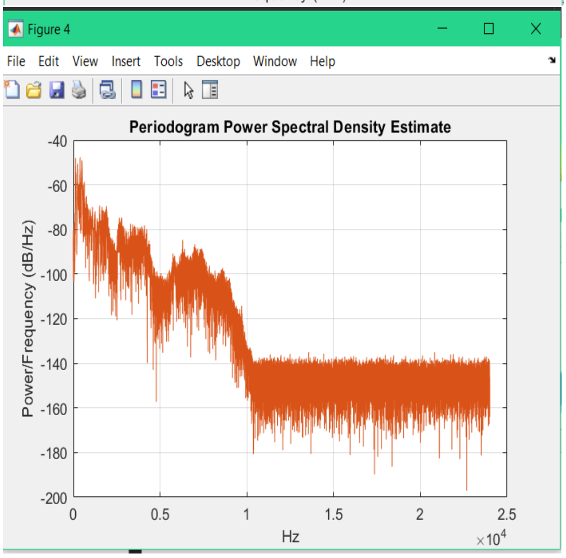
```
h = spectrum.welch; % create welch spectrum object
d = psd(h, data,'Fs', fs);
figure;
plot(d);
```

RESULTS

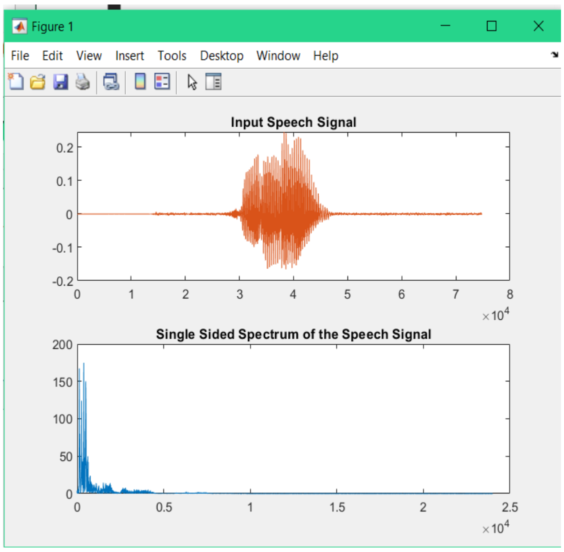
HELLO1

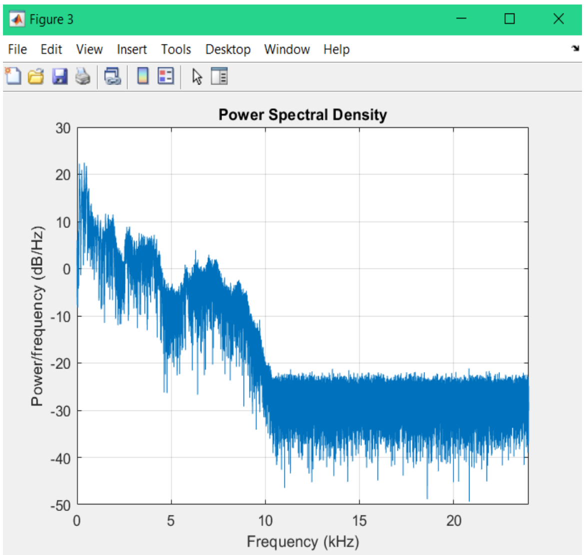


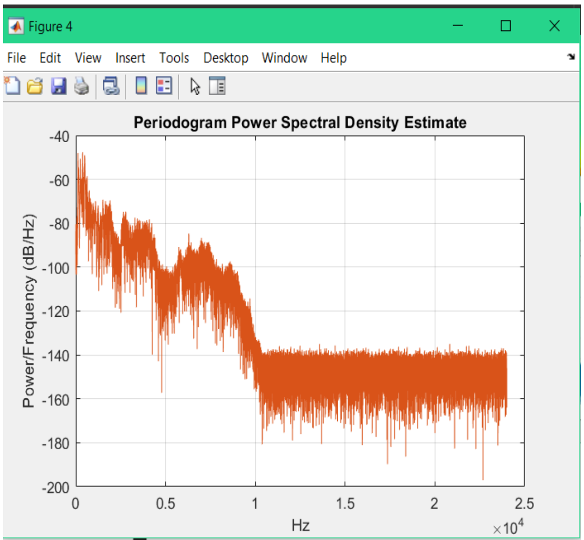


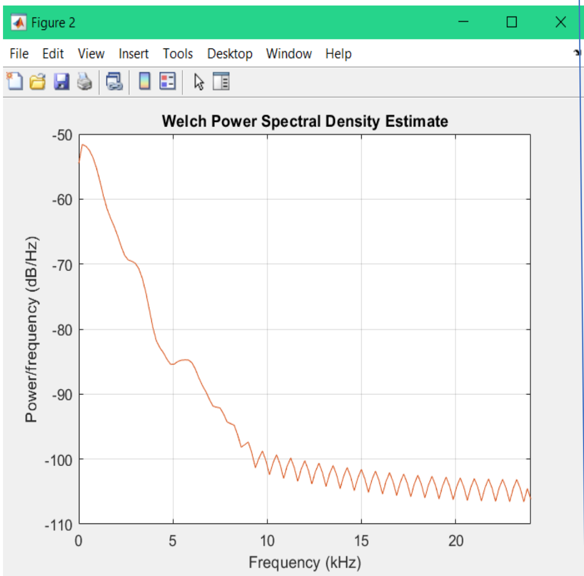


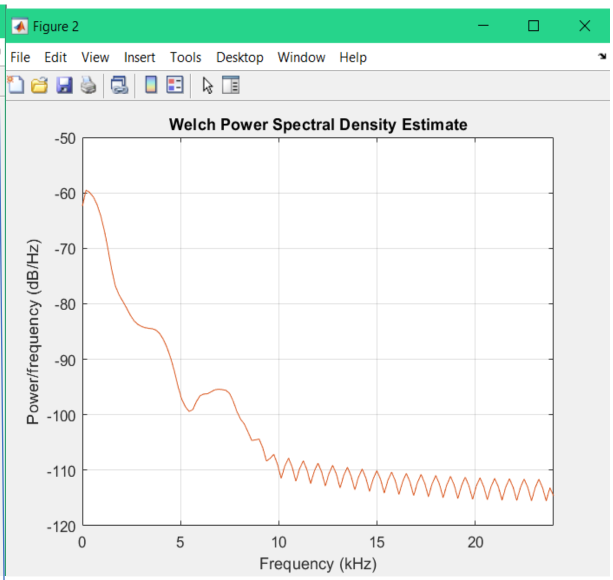
HELLO 2

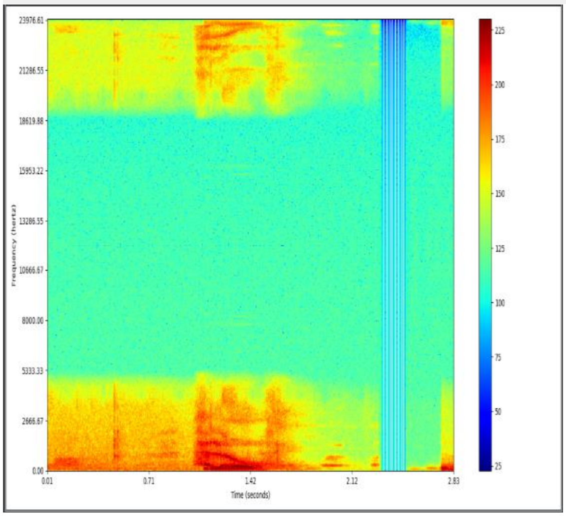


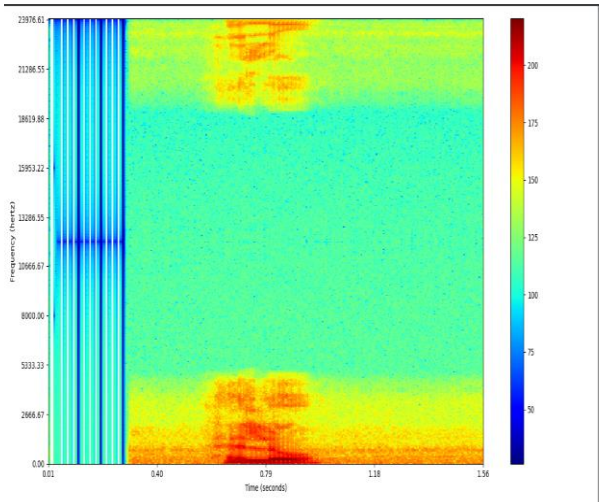




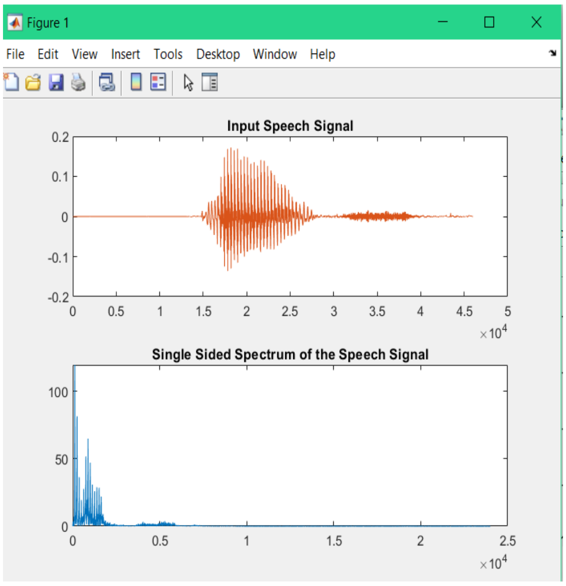




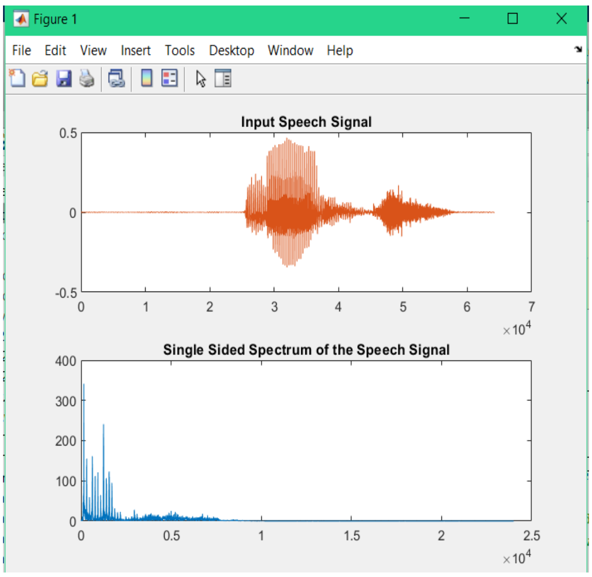




MATHS 1



MATHS 2



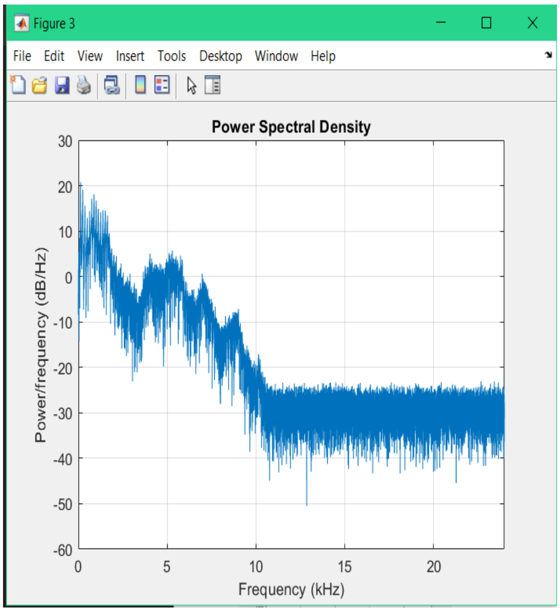


Figure 3
File Edit View Insert Tools Desktop Window Help
Power Spectral Density
Power/frequency (dB/Hz)
Frequency (kHz)

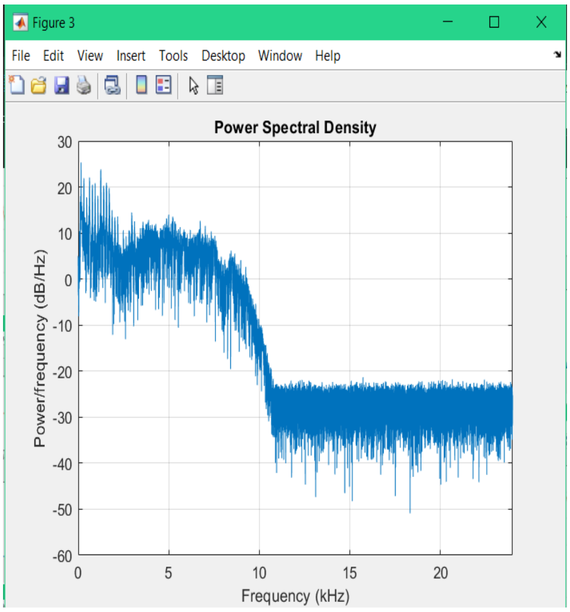


Figure 3
File Edit View Insert Tools Desktop Window Help
Power Spectral Density
Power/frequency (dB/Hz)
Frequency (kHz)

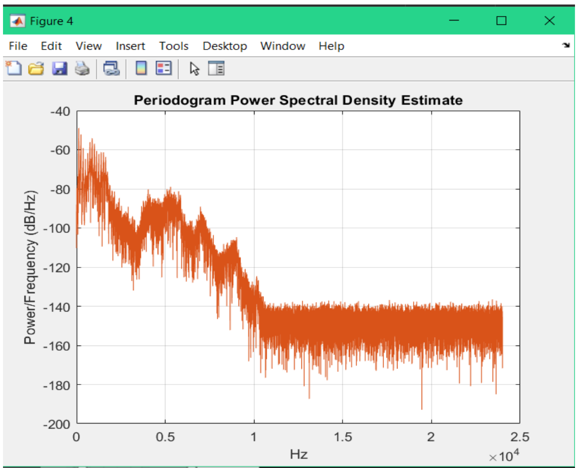


Figure 4
File Edit View Insert Tools Desktop Window Help
Periodogram Power Spectral Density Estimate
Power/Frequency (dB/Hz)
Hz

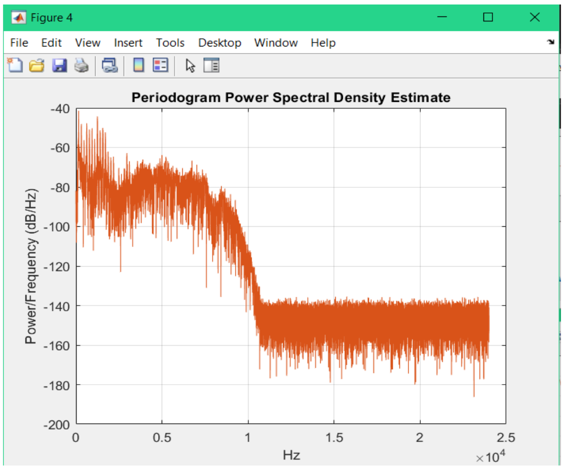


Figure 4
File Edit View Insert Tools Desktop Window Help
Periodogram Power Spectral Density Estimate
Power/Frequency (dB/Hz)
Hz

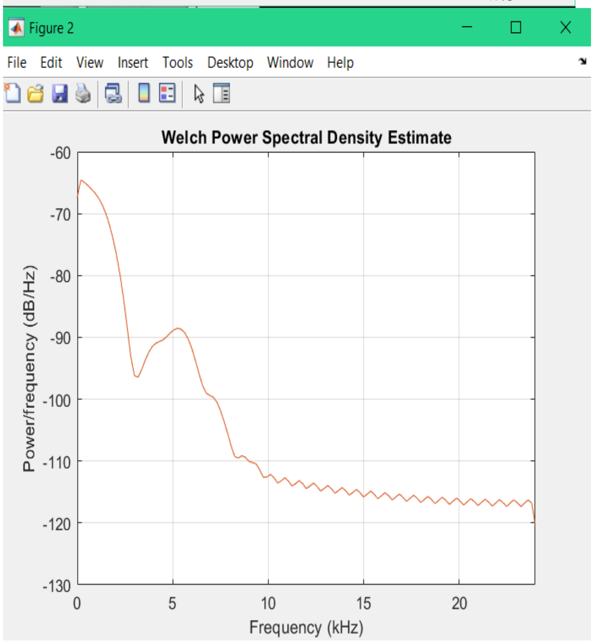


Figure 2
File Edit View Insert Tools Desktop Window Help
Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
Frequency (kHz)

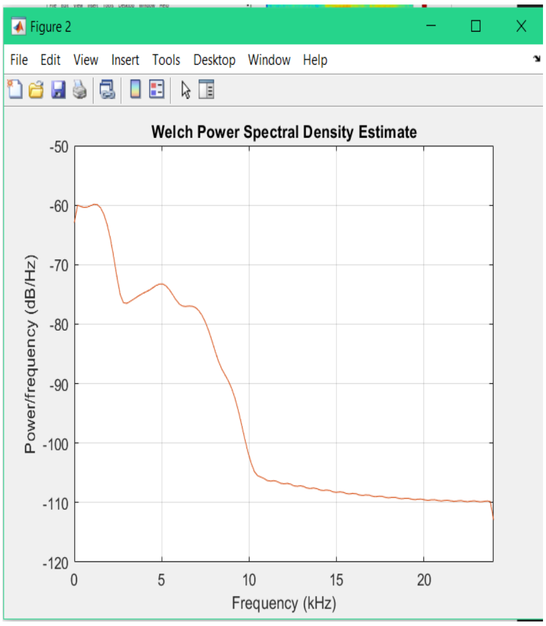
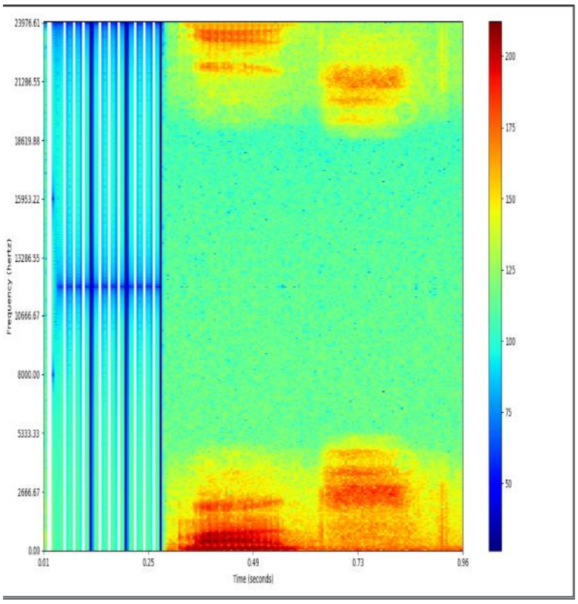
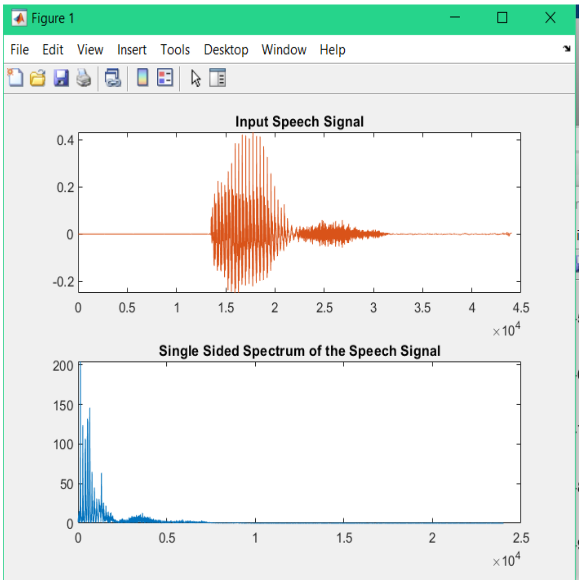
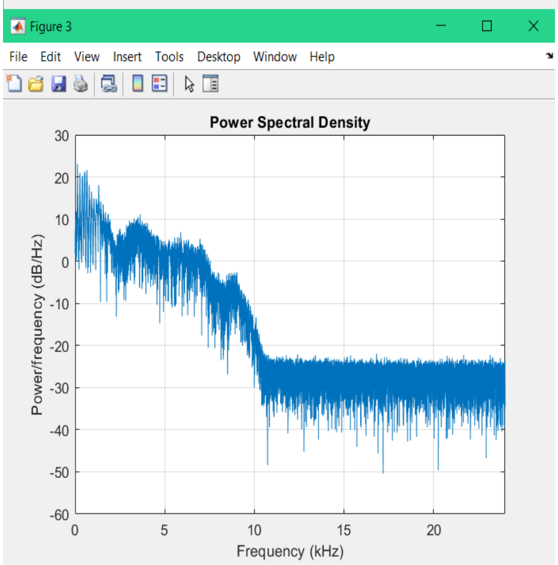


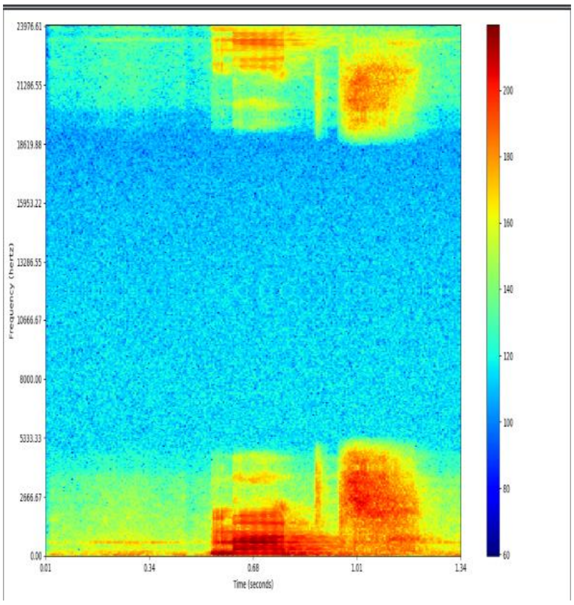
Figure 2
File Edit View Insert Tools Desktop Window Help
Welch Power Spectral Density Estimate
Power/frequency (dB/Hz)
Frequency (kHz)



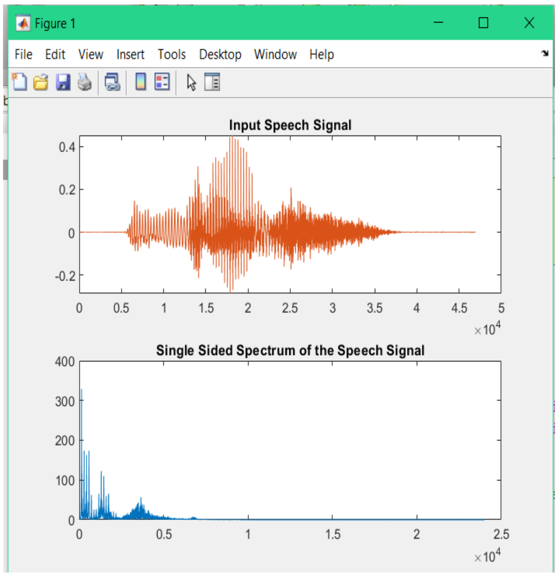
BRUSH 1

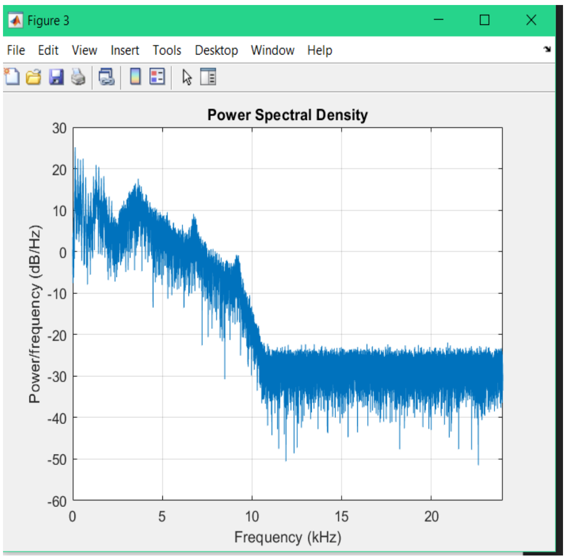


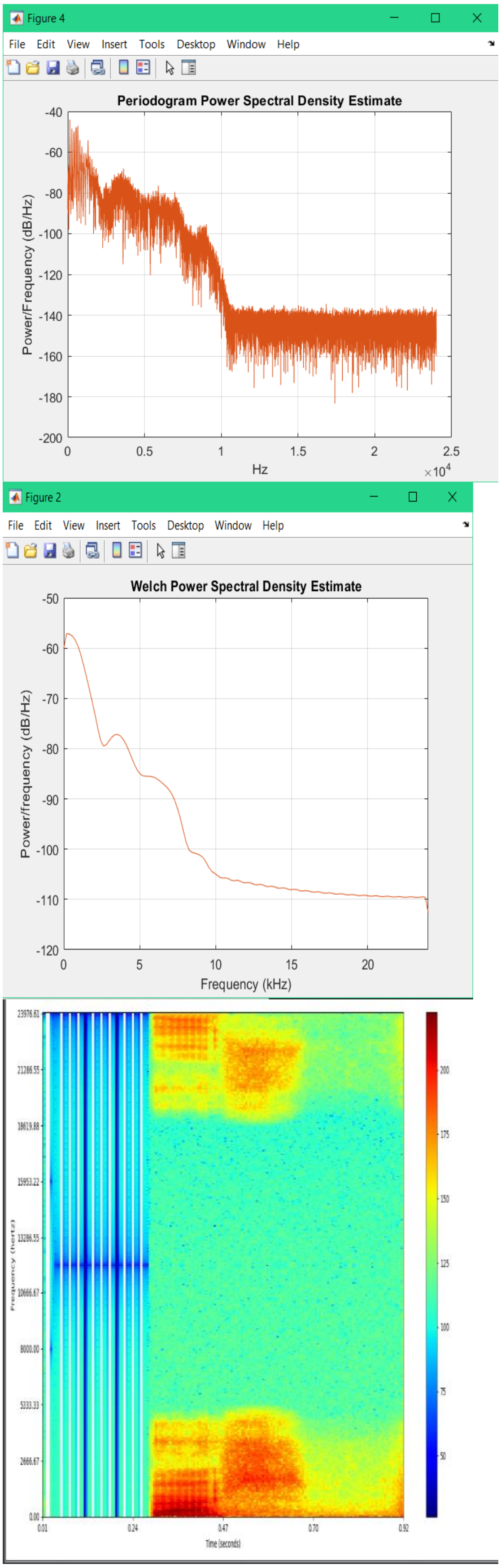




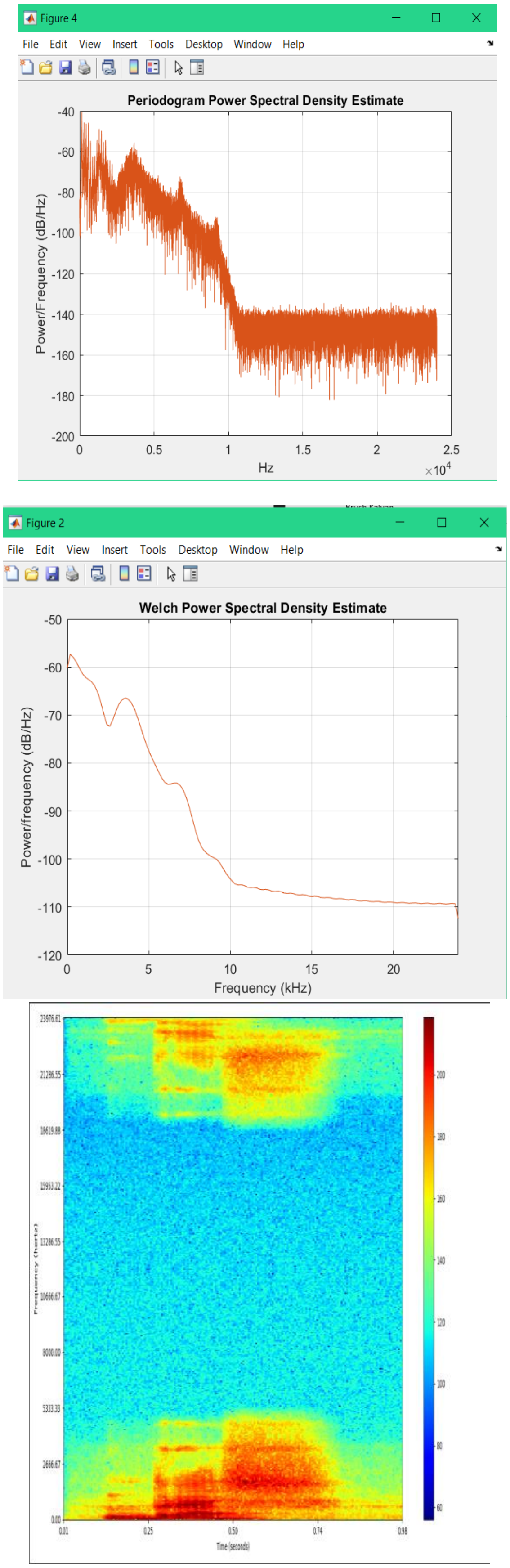
BRUSH 2







LEARN1



LEARN2

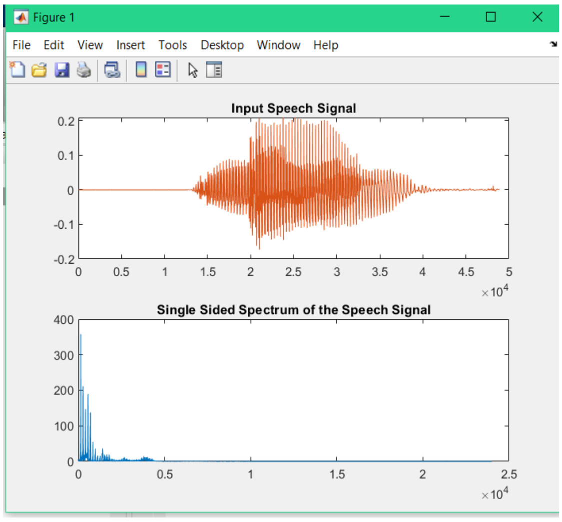


Figure 1
File Edit View Insert Tools Desktop Window Help
Input Speech Signal
Single Sided Spectrum of the Speech Signal

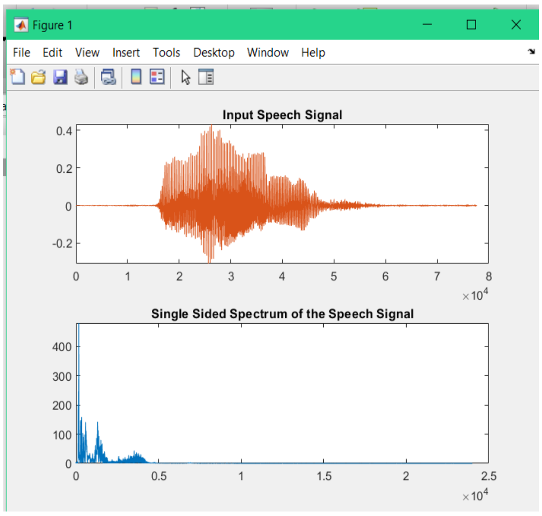


Figure 1
File Edit View Insert Tools Desktop Window Help
Input Speech Signal
Single Sided Spectrum of the Speech Signal

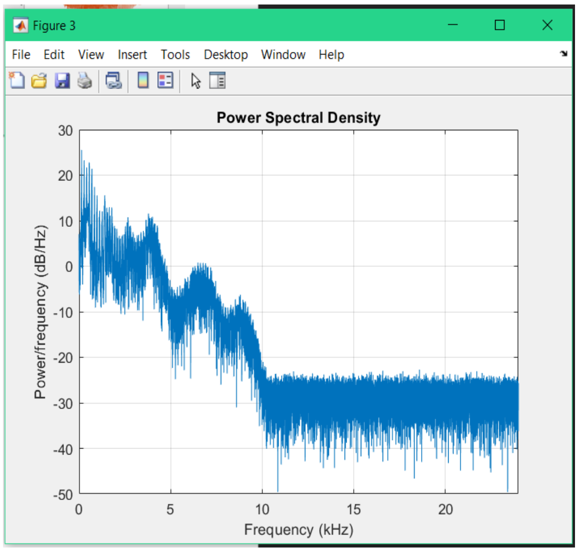


Figure 3
File Edit View Insert Tools Desktop Window Help
Power Spectral Density
Power/frequency (dB/Hz)
Frequency (kHz)

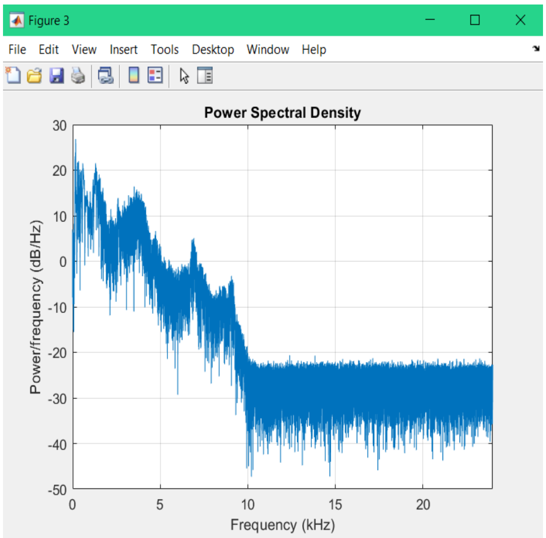


Figure 3
File Edit View Insert Tools Desktop Window Help
Power Spectral Density
Power/frequency (dB/Hz)
Frequency (kHz)

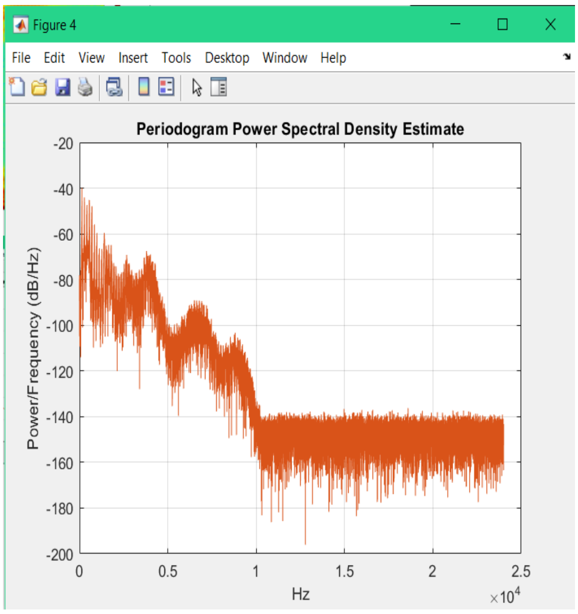


Figure 4
File Edit View Insert Tools Desktop Window Help
Periodogram Power Spectral Density Estimate
Power/Frequency (dB/Hz)
Hz

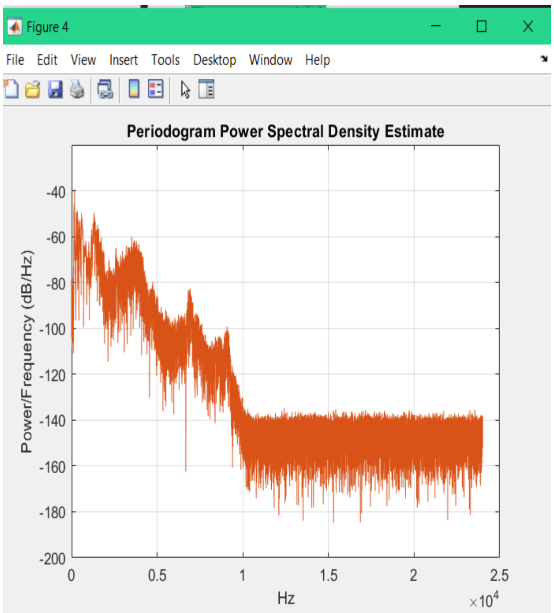
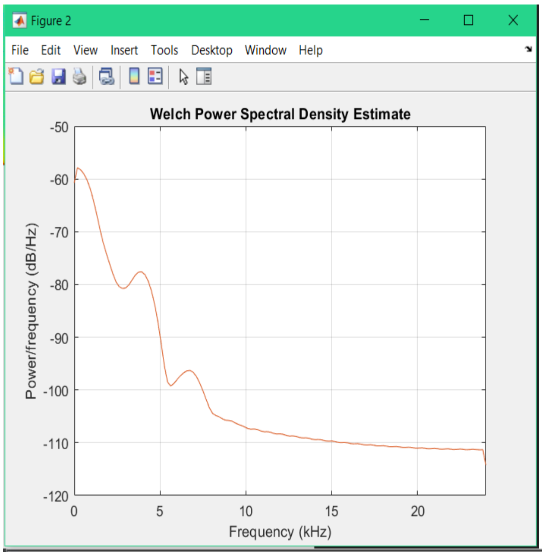
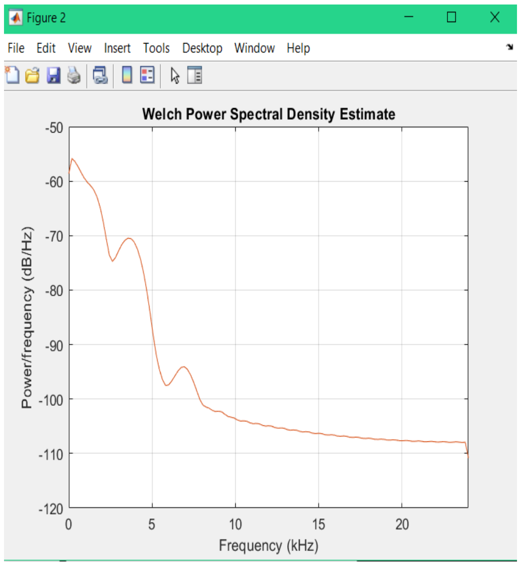
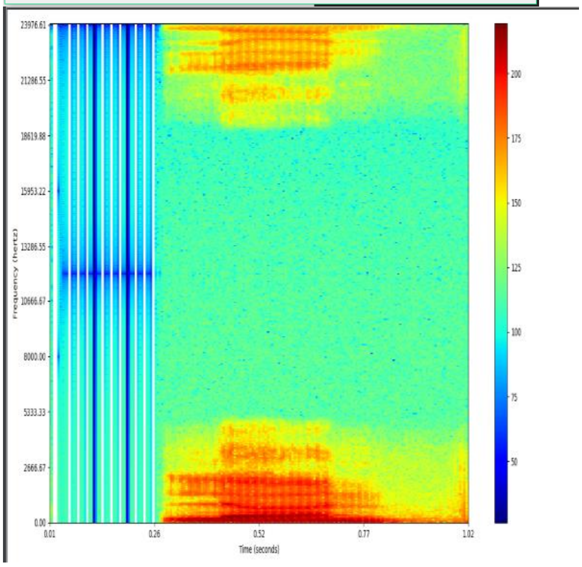
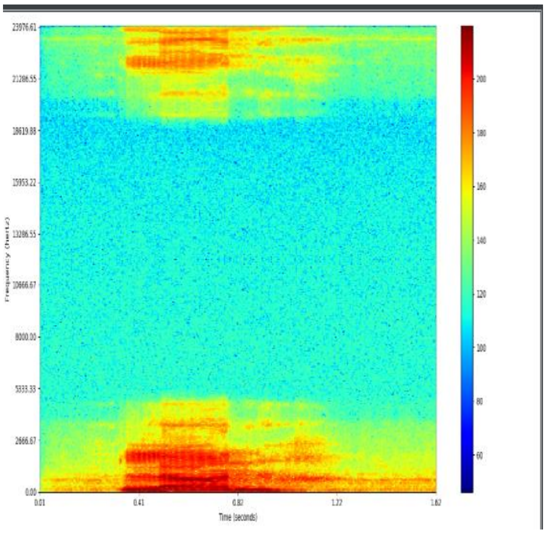


Figure 4
File Edit View Insert Tools Desktop Window Help
Periodogram Power Spectral Density Estimate
Power/Frequency (dB/Hz)
Hz









CONCLUSION AND FUTURE SCOPE

By using Power Spectral Analysis and Spectrogram it is possible to recognize and identity the words by comparing the given word with the standard word. Sometimes it happens that some words may have same frequency which results in similar spectrogram and thus it makes us difficult to identify those words. But still research is going on in the field of speech processing and one day we can find other such characteristics other than frequency and intensity which can solve the problems still faced in this domain. We can also train a deep learning model and build a speech recognition system of our own.