Project One

Name:张楚瑜　Student ID: 201630120000x

Name:胡越洋　Student ID: 20163012000xx

Name:李晓波　Student ID: 20163012000xx

Name:王明涛　Student ID: 2016301200008

Date: Mar. 17th 2019

Part 1　The introduction

1.1The purposes

In digital voice signal processing, it's essential to extract the characteristics of a speech signal such as pitch, loudness and timber. To get these characteristics, we have some effective methods such as energy, average magnitude, zero crossings and autocorrelation and so on. Short-term spectrum is also necessary. So we design a GUI to achieve these methods to observe speech's characteristics.

1.2 Group division

In this experiment, 王明涛 design the GUI interface using matlab. 李晓波 is responsible for time-domain analysis using matlab. 胡越洋 achieve the speech's spectrogram using matlab. 张楚瑜 is responsible for the voiced/unvoiced speech and silence segmentation based on features of speech using python.

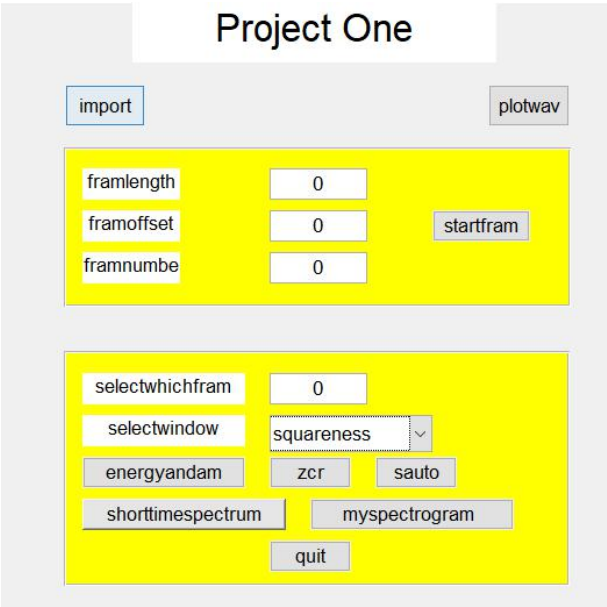
Part 2　GUI design

2.1 Platform

In our experiment, our group design page through GUI of matlab，which is used easily and has powerful data processing system.

2.2 The page

The designed interface is shown as Picture 2.2.



Picture 2.2

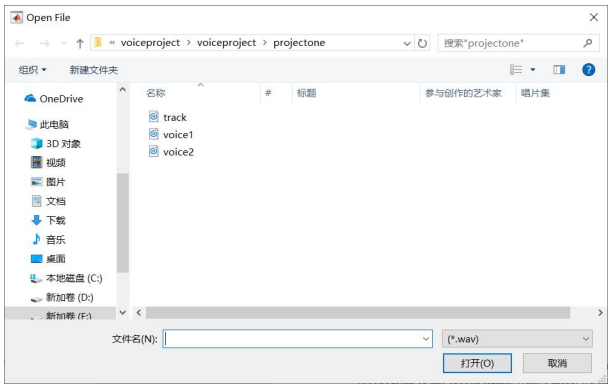
In our page, a voice signal can be framed according to frame length, frame shift which we need. We can also get the energy, average amplitude, zero-crossing rate and spectrogram of this speech signal. What's more, we can get short-term autocorrelation, short-time domain waveform and spectrum of specific frame which we can choose.

2.3 Instructions for use

2.3.1 Signal import

There is a button labeled 'import' used to import voice signal. If we click this button, there will be following interface Picture 2.3.1 pop up and we can choose the voice signal in any path.

And we can play the speech signal and get the waveform of it through the 'plotwav' button. Also, we can use it to check whether the signal is imported successfully.



Picture 2.3.1

2.3.2 Framing processing

In our page, there are two editable blank bar labeled 'framlength' 'framoffset' separately, which indicates frame length and frame shift. We can set the values we need. After you finish parameter setting, click the 'startfram' button and the voice

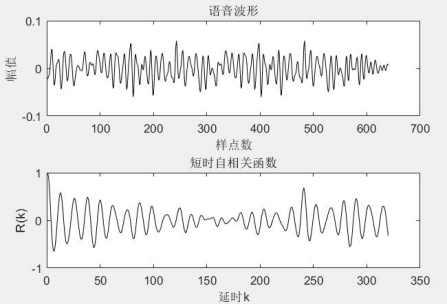
signal can be framed and the number of the frame can be shown in ‘framnumber’ bar.

There are an example as following Picture 2.3.2.

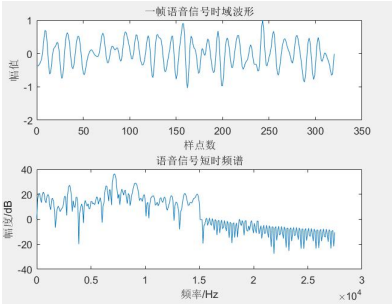
Picture 2.3.2

2.3.3 The characteristics of voice signal

After the signal framed, we can select a specific frame and window to observe its time domain waveform, spectrum and autocorrelation. For example, if the frame length is 320, frame shift is 80, the window is square and we select No.2000 to observe, we can get following result.

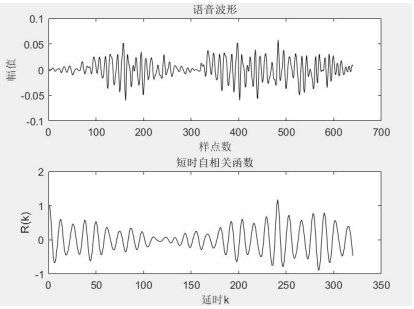


Picture 2.3.3.1 rectangular autocorrelation

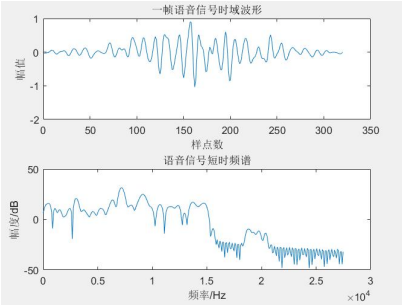


Picture 2.3.3.2 rectangular spectrum

If we just select different window such as hamming, we can get following result.



Picture 2.3.3.3 hamming autocorrelation



Picture 2.3.3.4 rectangular spectrum

Besides, we can use ‘myspectrogram’ get spectrogram of the voice signal. If you choose the different frame length and window, you will get different result. You can get zero-crossing rate through ‘zcr’ button and the result won’t be effected by the kind of window.

2.3.4 Quit

You can click ‘quit’ button to close the process.

Part 3 Time-domain analysis

According to the type of parameters analyzed, speech signal analysis can be divided into time-domain analysis and transform-domain analysis, of which time

domain analysis is the simplest and most intuitive method. It directly analyses the time-domain waveform of speech signal. The main feature parameters extracted are short-time energy and average amplitude, short-time average zero-crossing rate and short-time autocorrelation function of speech.

3.1 Short-time average energy and short-time average amplitude

Because the energy of speech signal changes with time, the energy difference between silence, unvoiced and voiced sound is obvious. Therefore, the analysis of short-time energy and short-time average amplitude can describe the change of speech characteristics.

When a rectangular window with length N is used, the short-time average energy and average amplitude of speech signal can be expressed as:

$$E_n = \sum_{m=n-(N-1)}^{n} x^2(m)$$

$$E_n = \sum_{m=n-(N-1)}^{n} \left| x\ (m) \right|$$

Figure 3.1.1 shows the short-time average energy and the short-time average amplitude of a speech signal when N = 320.

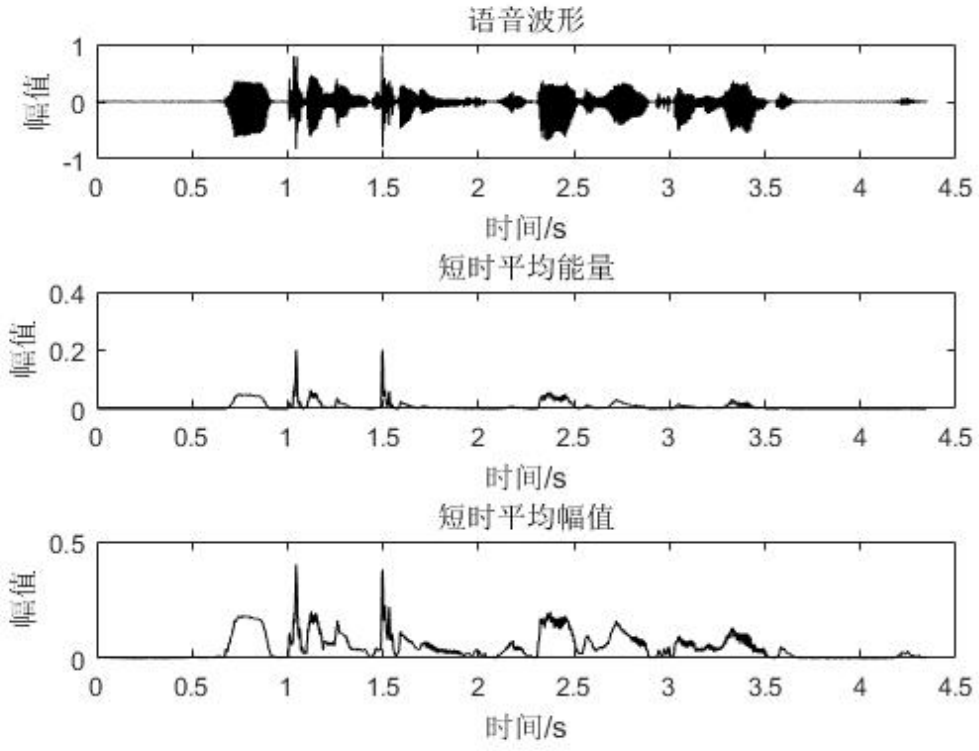


Figure 3.1.1

3.2 Short-time average zero-crossing rate

Short-time average zero-crossing rate refers to the number of times a signal

passes through a zero value per frame. Zero-crossing rate can reflect the frequency information of the signal to a certain extent. Since speech is a short-time stationary signal, the short-time average zero-crossing rate can reflect its spectral properties to a certain extent, thus a rough estimation of spectral characteristics can be obtained.
When a rectangular window with length N is used, the short-time average zero-crossing rate of speech signal can be expressed as:

$$Z_n = \frac{1}{2N} \sum_{m=n-(N-1)}^{n} \left| sgn[x(m)] - sgn[x(m)] \right|$$

Figure 3.2 shows the change curve of short-time average zero-crossing rate of a speech when N = 320.It can be seen from the figure that the difference between the short-time zero-crossing rate of voiced and unvoiced sounds is quite obvious.

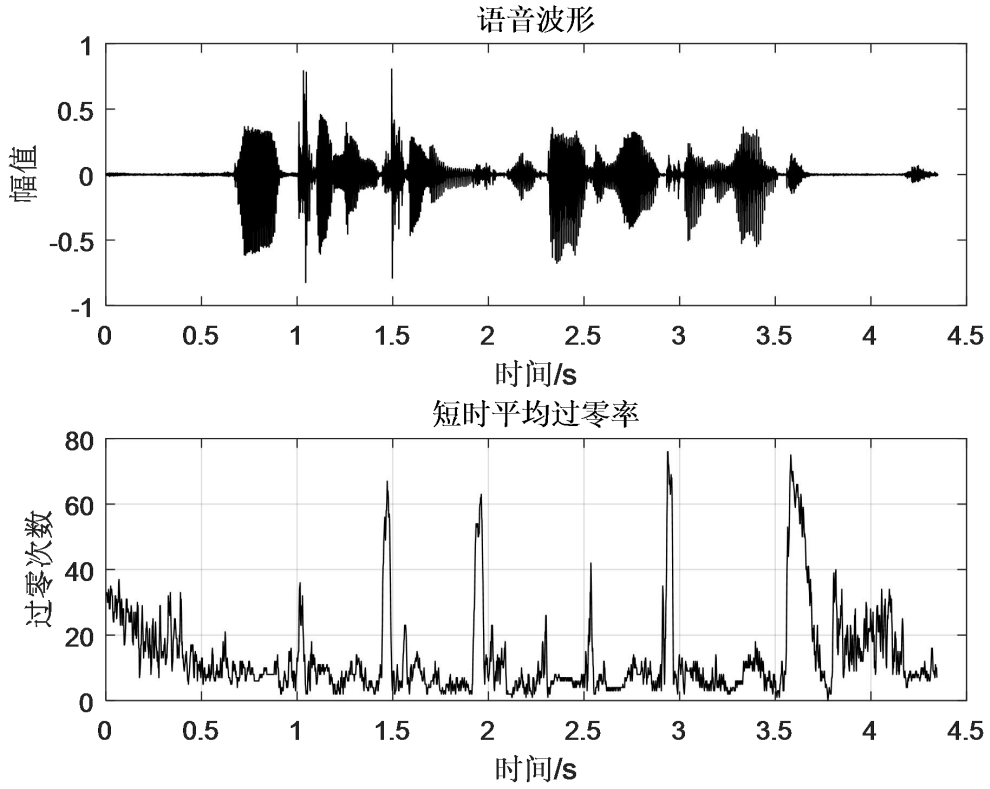


Figure 3.2.1

3.3 Short-term autocorrelation

The autocorrelation function is used to measure the similarity of the time waveform of the signal itself. The time waveform of voiced sound shows a certain periodicity, and the similarity between the waveforms is good. The time waveform of unvoiced sound shows the characteristics of random noise, and the similarity between sample points is poor. Therefore, we can use short-time autocorrelation function to measure the similarity of sounds and distinguish voiced and unvoiced sounds.

The autocorrelation function of speech signals with rectangular windows is defined as:

$$R_n(k) = R_n(-k) = \sum_{m=-\infty}^{+\infty} x(m)x(m-k)$$

Figure 3.3.1 shows a frame of voiced wave and its short-time autocorrelation when N = 320. Figure 3.3.2 shows a frame of unvoiced wave and its short-time autocorrelation when N = 320.

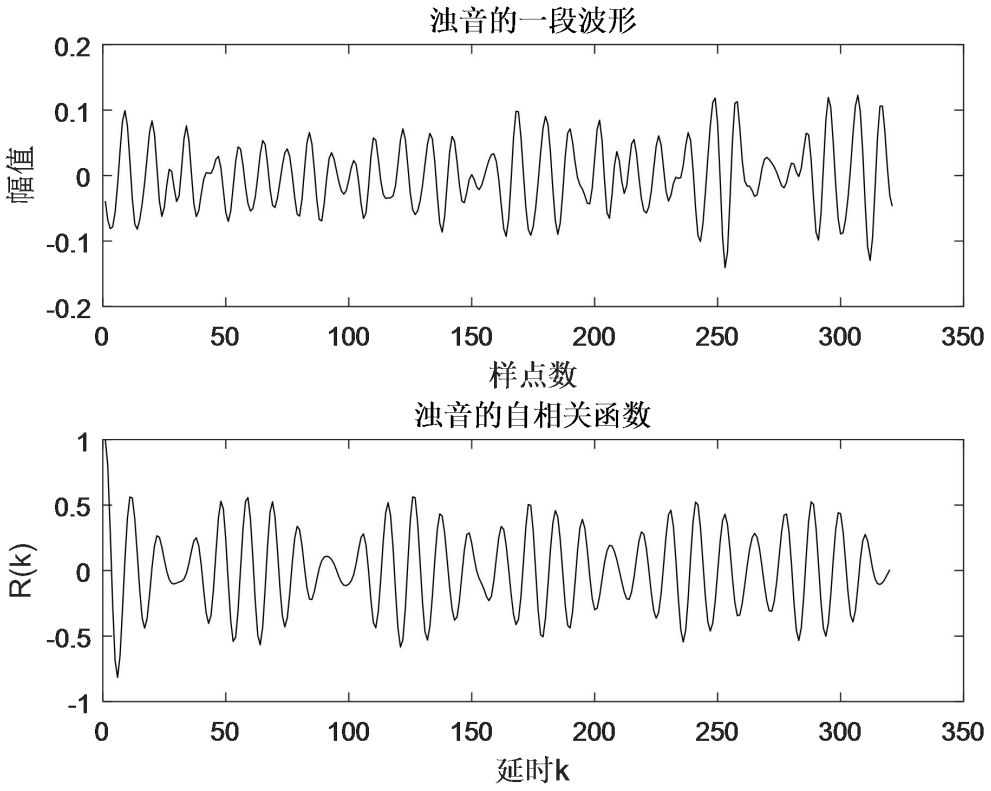


Figure 3.3.1

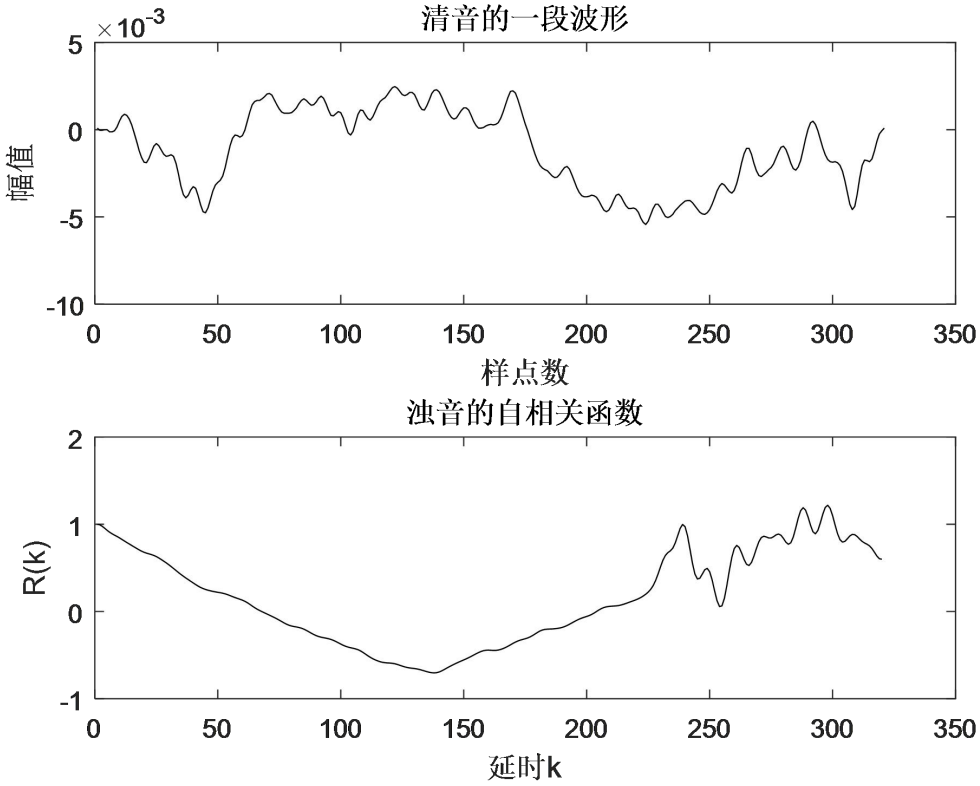


Figure 3.3.2

As seen from figs. 3.3.1 and 3.3.2, the short-time autocorrelation functions of voiced and unvoiced sound have the following characteristics:

(1) Short-time autocorrelation function can clearly reflect the periodicity of voiced signal.

(2) The short-time autocorrelation function of the voiceless sound has no periodicity, and has no obvious peak value. Its property is similar to that of noise.

At the same time, according to the repeated adjustments of parameters, it is found that window length has a significant effect on short-time autocorrelation. When setting window length N = 320, the effect is better.

Part 4 My spectrogram

A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time.

A common format is a graph with two geometric dimensions: one axis represents time or RPM, the other axis is frequency; a third dimension indicating the amplitude of a particular frequency at a particular time is represented by the intensity or color of each point in the image.

The first step of short-time Fourier transform is to add windows to the speech signal. What we need to consider is the choice of window shape and window length. Like the functions in MATLAB, spectrogram uses Hamming window by default.

$$w(n)=\begin{cases}0.54-0.46\cos\left(\dfrac{2\pi n}{M-1}\right) & 0\le n\le M-1\\ 0 & otherwise\end{cases}$$

The width of the main lobe of Hamming window is $8\pi / M$ and the attenuation of the peak value of the side lobe is $41dB$. By using the following function:
h=hamming(nsc)；

Hamming windows with NSC window length can be obtained.

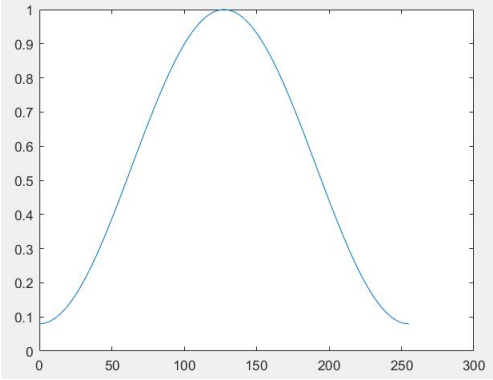


Figure 4.1 nsc=256 hamming window

Since speech signal is real, the frequency spectrum after FFT satisfies the characteristic of conjugate symmetry, so only half of it is needed to draw the spectrum map. When the number of points for fast Fourier transform is even, $rown = nfft / 2 + 1$,

When the number of points for fast Fourier transform is odd, $rown = (nfft + 1) / 2$ 。

After setting sampling frequency, window length, overlap length and FFT points, FFT is performed on each segment of data.

```
temp_S=S(index:index+nsc-1).*h';
temp_X=fft(temp_S,nff);
```

Put the results of each FFT segment into the $rown * co\ln$ matrix，After setting the time axis and frequency axis, the spectrum can be drawn.。The following are drawn and compared with spectrogram and spectrogram functions.

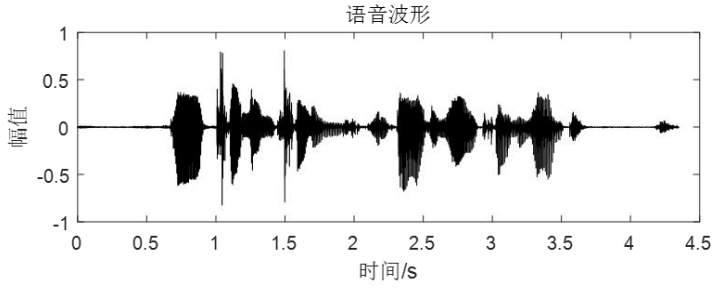


Figure 4.2 voice wavform

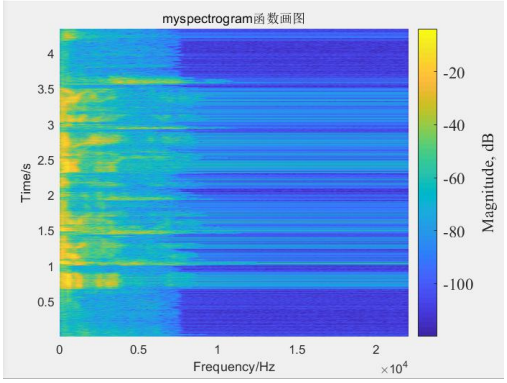


Figure 4.3 myspectrogram

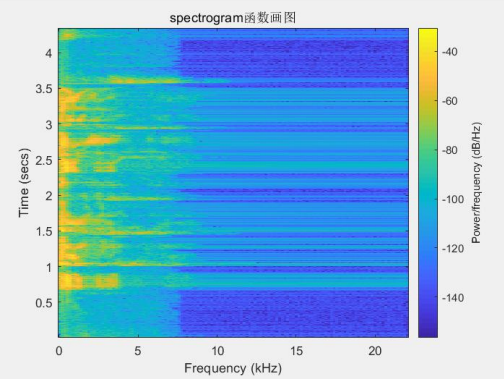


Figure 4.4 spectrogram

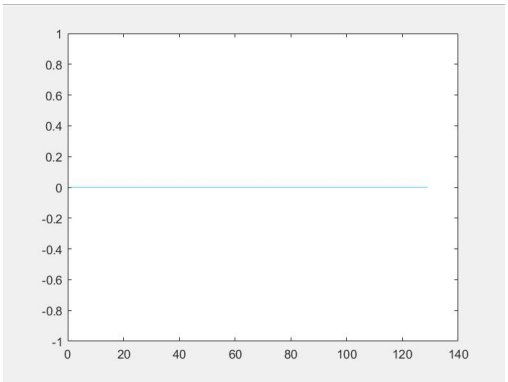


Figure 4.5 difference of myspectrogram and spectrogram

From the graph, we can see that the effect of myspectrogram function is the same as that of the spectrogram function in MATLAB.

Part 5 Voiced/Unvoiced/Silence Speech Recognition

In this part, we will try to discriminate the voiced/unvoiced speech and silence segmentation based on features of speech. Implement two methods of endpoint detection and give a brief comparison analysis of results.

5.1 Description of dataset

We give a description of dataset. The dataset is obtained from github. It includes four audio files, which are recorded at different environment, like park, room, etc. And their SNR is different. so we can compare our algorithm at different SNR. The dataset's detail is below:

	Bus stop	Cons. site	Park	Room	Overall
Dur. (min)	30.02	30.03	30.07	30.05	120.17
Avg. SNR (dB)	5.61	2.05	5.71	18.26	7.91
% of speech	40.12	26.71	26.85	30.44	31.03

sheet 1

5.2 Principle of our algorithm

We introduce some principle of our algorithm. As we know, there are too many feature in speech signal, like energy, Zero Crossing Rate, Entropy of Energy, Spectral Centroid, MFCCs, Spectral Entropy, etc. They can be classified to frequency-domain and time-domain. We adopt two time-domain feature, energy and zero crossing rate(ZCR), and one frequency-domain feature Spectral Entropy. In time- domain, the energy of voiced is high than unvoiced and silence, and unvoiced ZCR is higher than voiced and silence. In frequency-domain, Spectral Entropy of silence is higher than voiced and unvoiced.

5.3 Description of Spectral Entropy from math perspective

We give a description of Spectral Entropy from math perspective.

Spectral Entropy describes the complexity of a system. It is defined as follows:

1.Calculate the spectrum $\mathrm{X}(\omega_i)$ of signal.

2.Calculate the Power Spectral Density of signal via squaring its amplitude and normalizing by the number of bins.

$$P(\omega_i) = \frac{1}{N}\left|X(\omega_i)\right|^2$$

3.Normalize the calculated PSD so that it can be viewed as a Probability Density Function (integral is equal to 1).

$$p_i = \frac{P(\omega_i)}{\sum_i P(\omega_i)}$$

4.The Power Spectral entropy can be now calculated using a standard formula for an entropy calculation.

$$\mathrm{PSE} = -\sum_{\mathrm{i}=1}^{n} p_{\mathrm{i}} \ln p_i$$

5.4 Result

We show our results in sheet 2. Because it is a binary classification tasks, we use F1-score as the scoring standard.

	Bus stop	Cons. site	Park	Room	Mean
ZCR and energy	65.12%	41.43%	41.95%	63.25%	52.94%
Time(s)	39.20	51.15	35.84	40.71	41.73
Spectrum Entropy	47.80%	37.68%	36.77%	36.05%	39.58%
Time(s)	53.33	54.50	54.65	54.33	54.20

sheet 2 the result of our experiments. Time is feature extract time plus train time

According to results, ZCR and energy(ZE) worked better than Spectrum Entropy(SE) both in F1-score and time. However, ZE model has more hyper parameters, include high Zero crossing rate threshold, high energy threshold and low

energy threshold, which are hard to fine tune. (In our experiment, the parameter is 35,1,6).SE model only need to set one threshold(we set as 0.3 in our experiment).

We focus on ZE model. It is strange that Bus stop's results is better than Room, which SNR is three times than Bus stop.

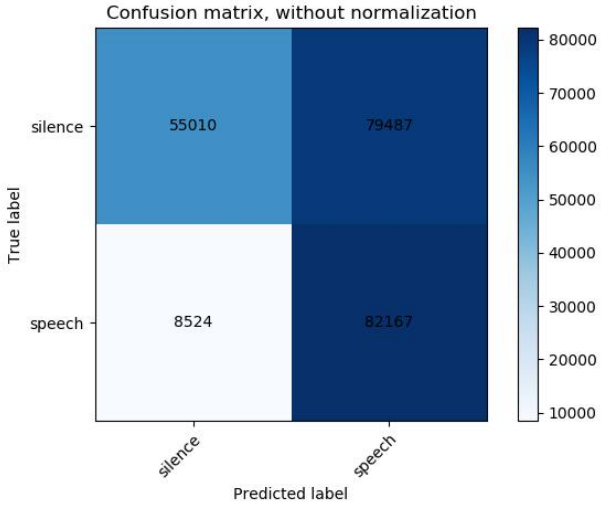


Figure 1 the confusion matrix of Bus stop, speech include voice and unvoiced.

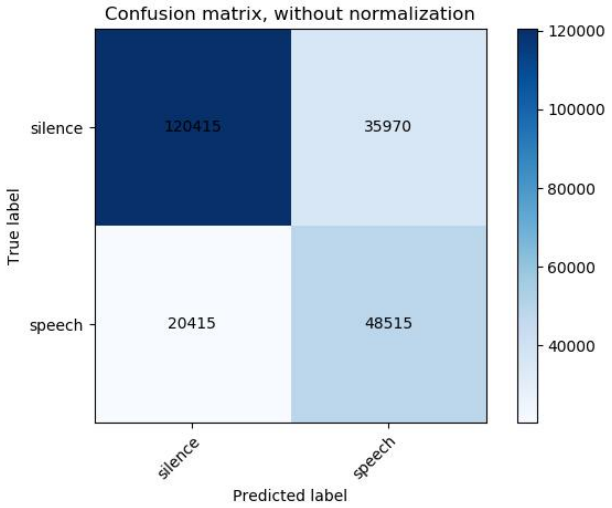


Figure 2 the confusion matrix of Room

The confusion matrix of Bus Stop and Room are different. In figure 1, 79487 silence sample are judged as speech. But in figure 2, 35970 silence sample are judged as speech, while 20415 speech sample are judged as silence. In the Room, people speak lower than Bus stop, that lead to the misjudge of speech. What's more, when I lower low energy threshold to 0.3, the result changes a lot. F1-score of Room is 67.81%, while Bus stop is 59.04%, which proves my assumption.

5.5 Conclusion

In our experiments, frequency-domain method doesn't work well when compared to time-domain, maybe I have not make good use of Spectrum Entropy. In time-domain, the result is also not good enough. With the increase of SNR, the

F1-score will increase. Code can be found in this repo.

NOTE: I have not understood Teager Energy, and have no idea of how to calculate it. I have found some paper about it, and read it later, so in our experiment, I do not consider Teager Energy.

Part 5 Discussion

In this experiment, we use the matlab and phthon to observe the speech's characteristics and voiced/unvoiced/silence speech recognition. Our group learn how to use computer to achieve these tasks. In the experimental process, we use smart phone to record a voice speech and use it to do the experiment.

However, there is a shortage in our project, which is the voiced and unvoiced recognition can't be matched to GUI because of the code is different.

I have some suggestions for our experiments. When we'll do an experiment, we must make sure which machine language to be sued. Only do like this, we can understand the other's work better.