哈爾濱工業大學

实验报告

实 验（一）

题　　目 语音信号的端点检测

专　　业 人工智能

学　　号 1190202107

班　　级 1903602

学　　生 姚舜宇

指 导 教 师 郑铁然

实 验 地 点 格物 213

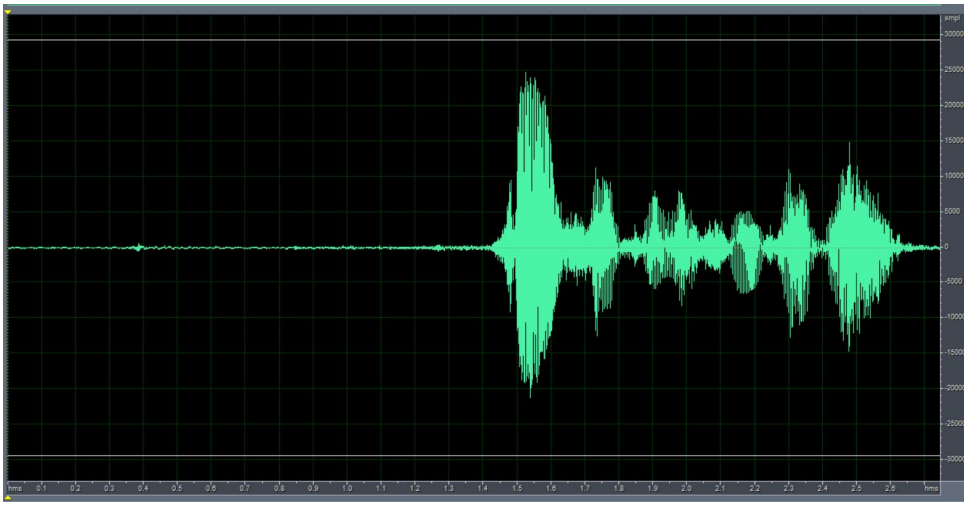
实 验 日 期 2021.12.4

计算机科学与技术学院

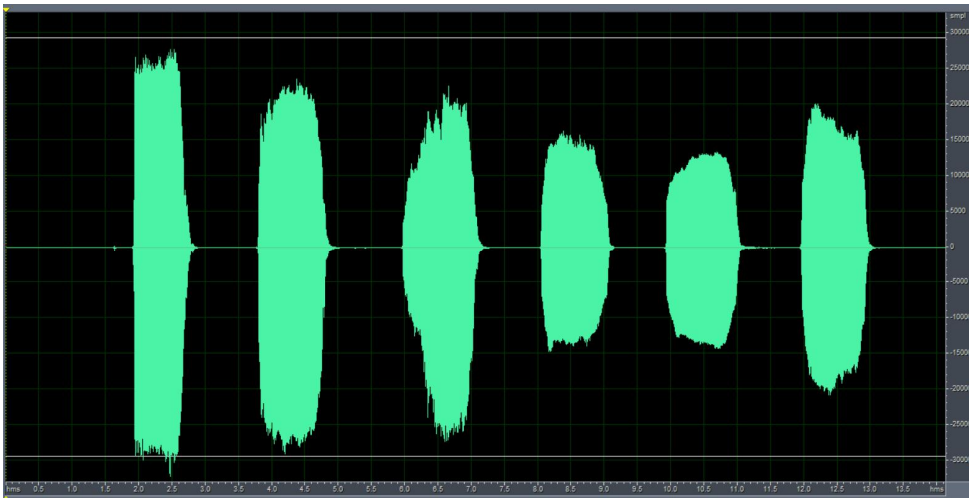
一、 语音编辑和处理工具的使用

1.1 语音文件的时域波形截图

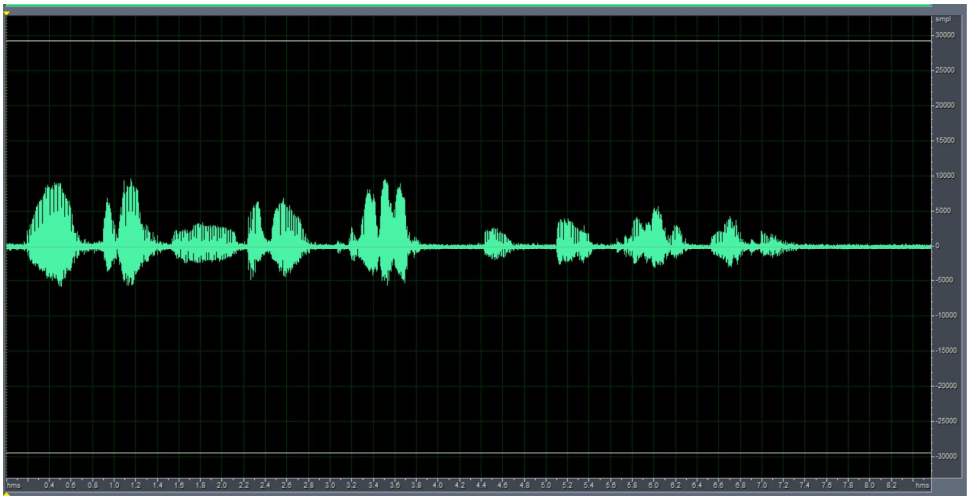
1.wav



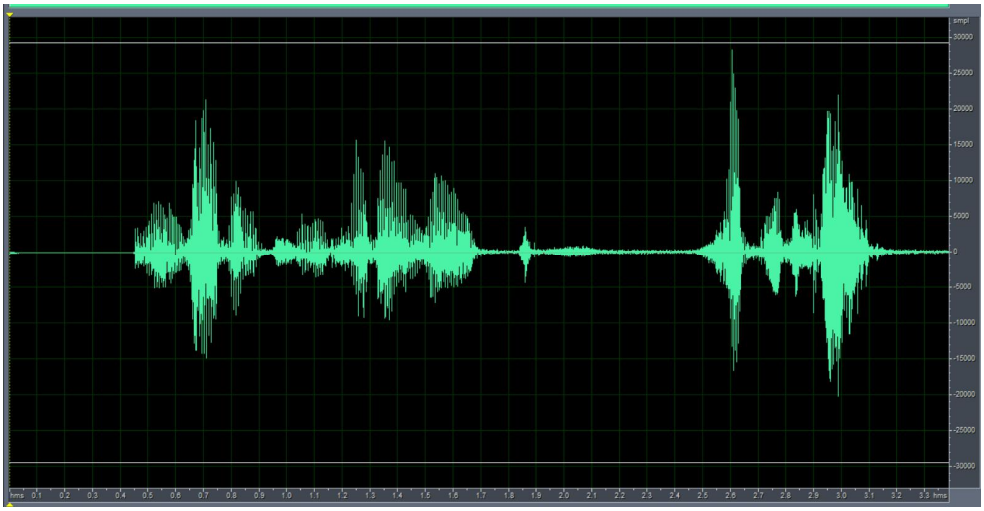
2.wav



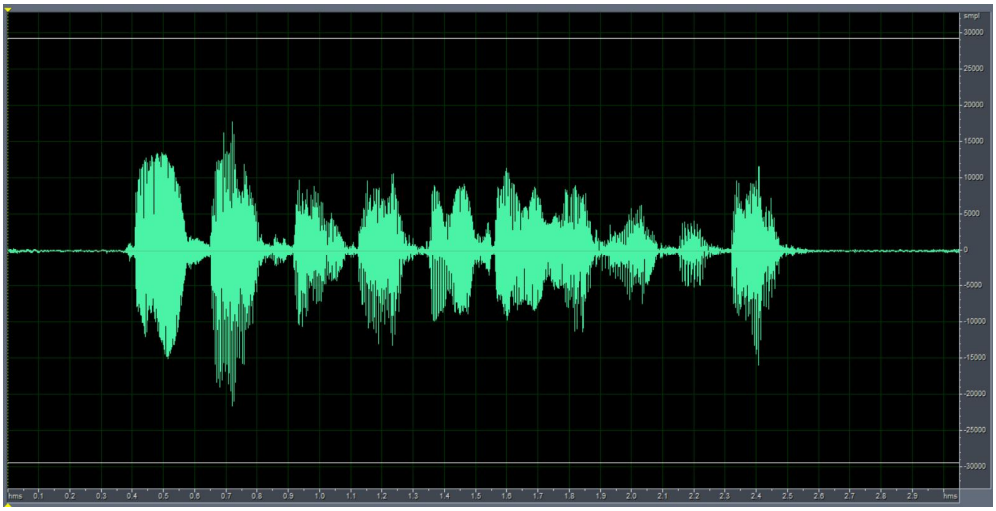
3.wav



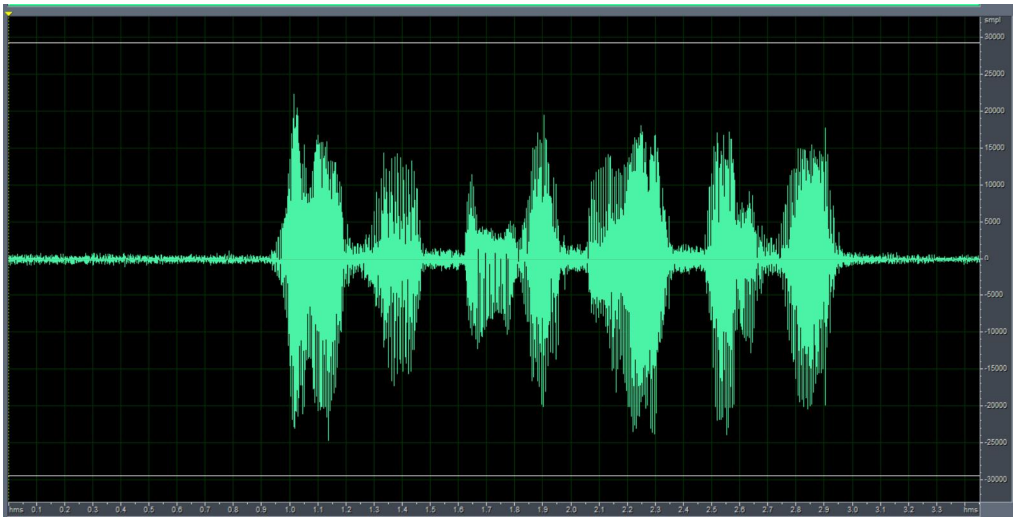
4.wav



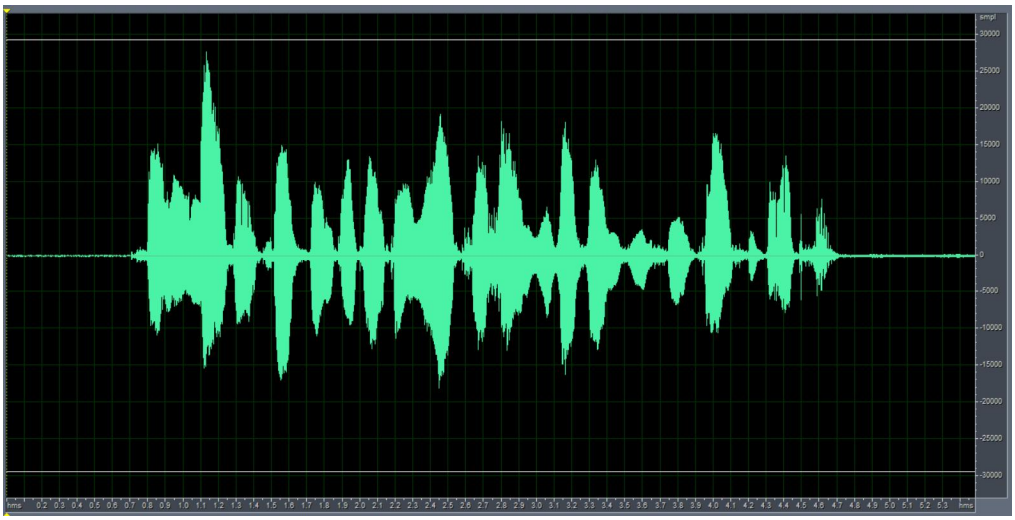
5.wav



6.wav

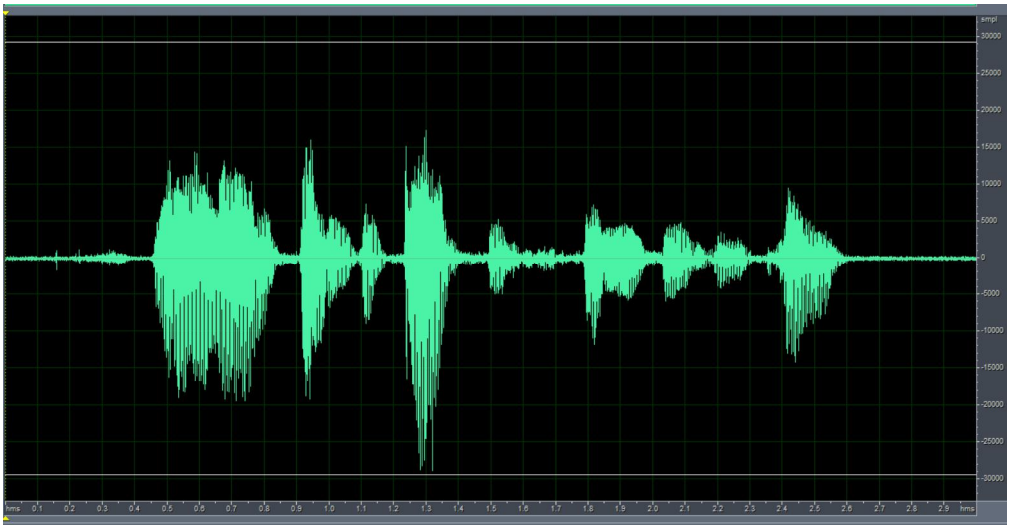


7.wav

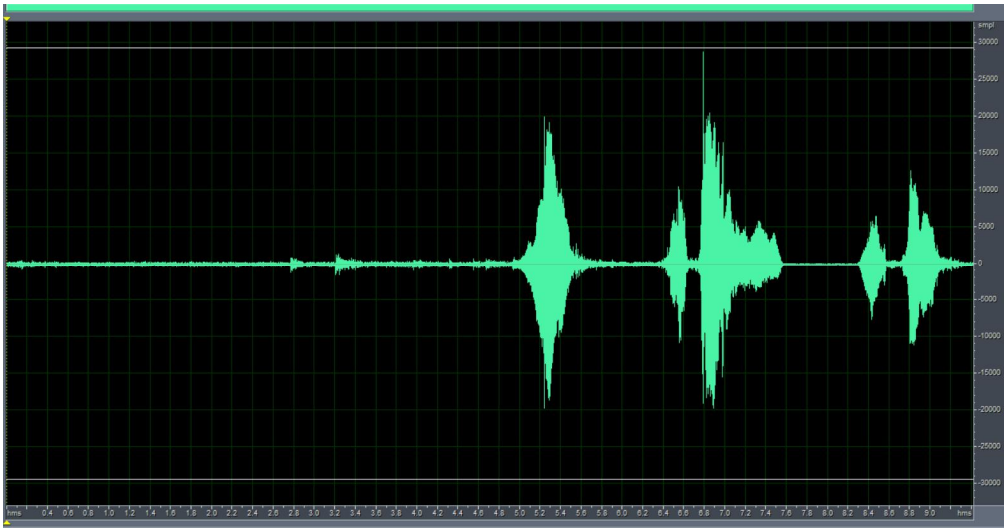


8.wav

9.wav

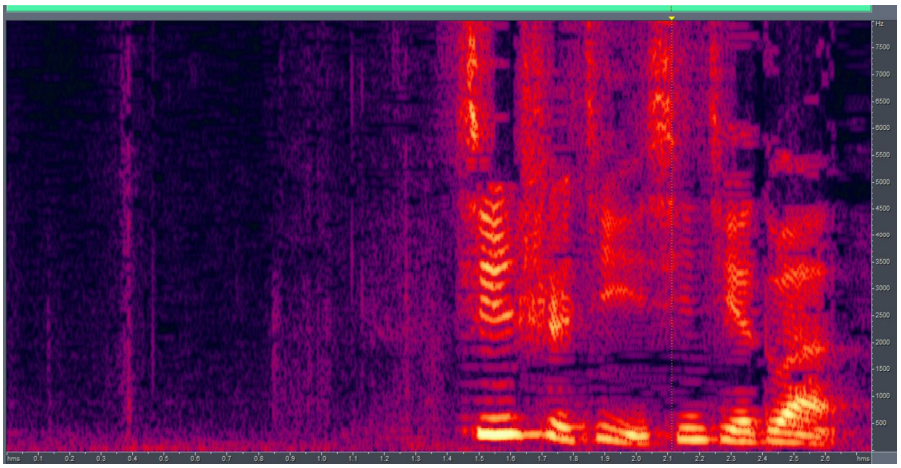


10.wav

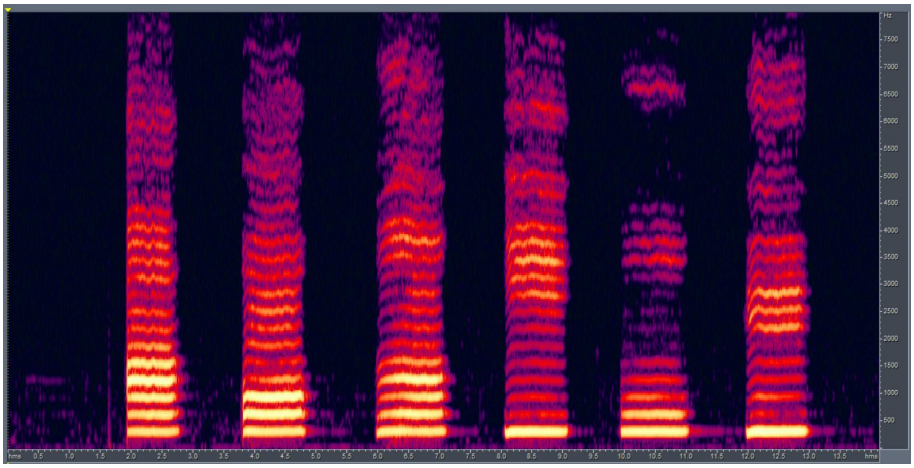


1.2 语音文件的语谱图截图

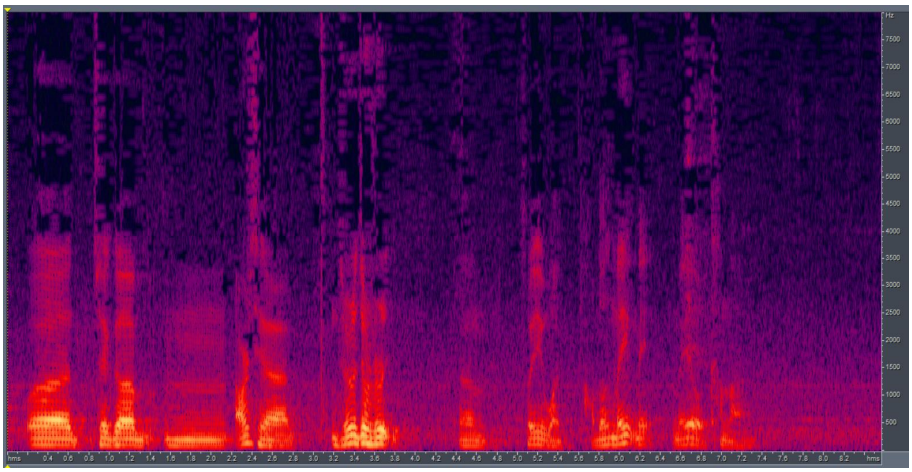
1.wav



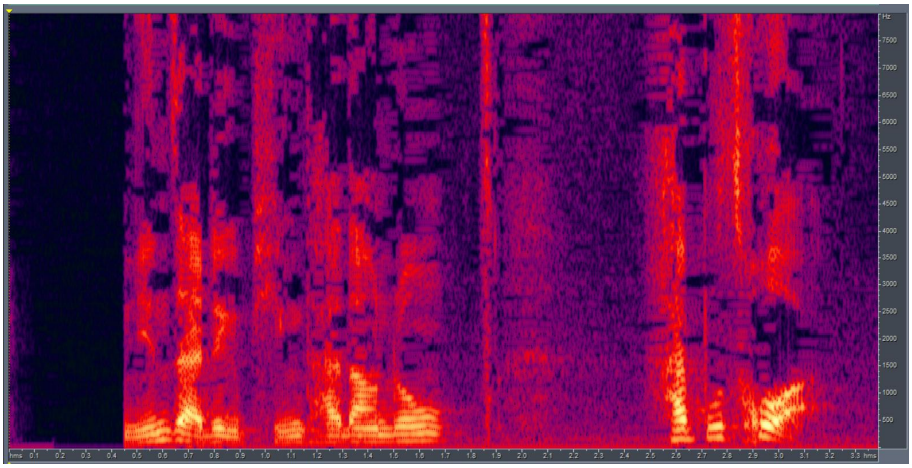
2.wav



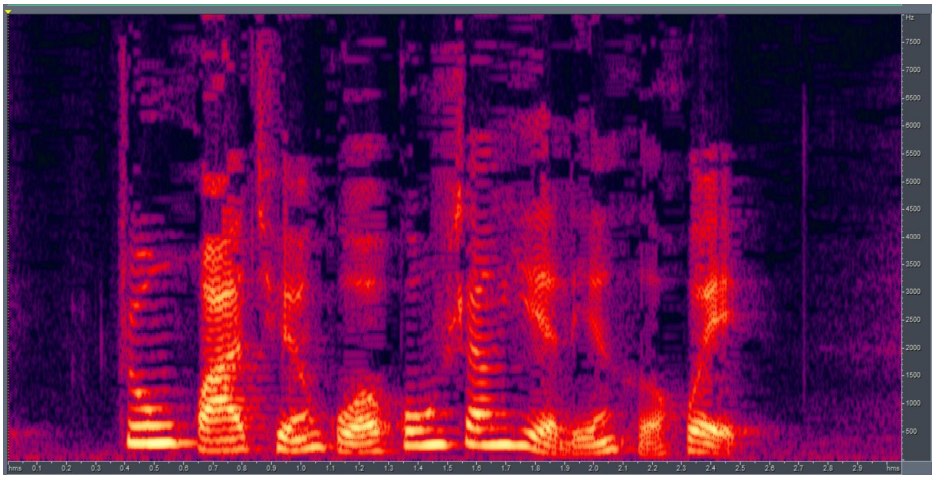
3.wav



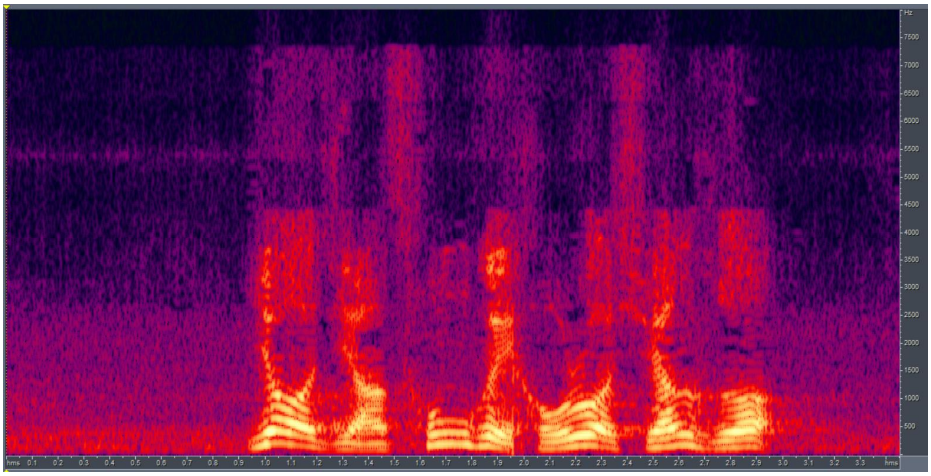
4.wav



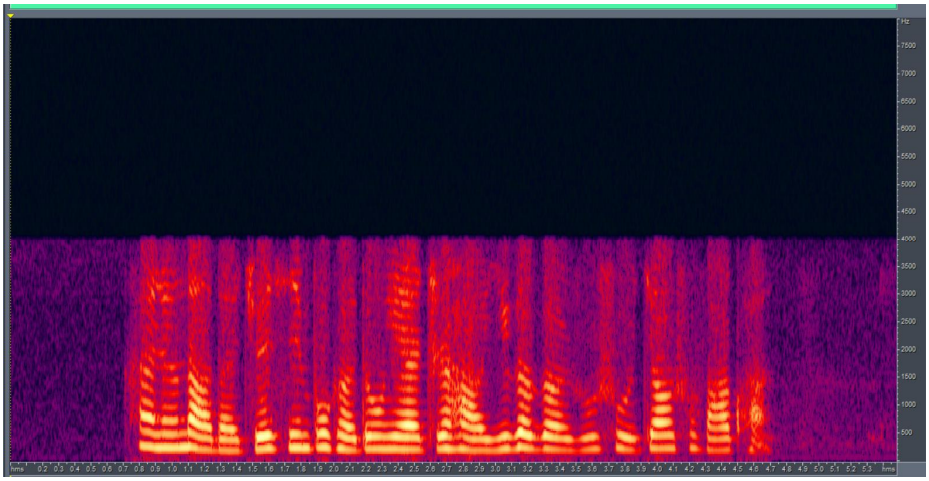
5.wav



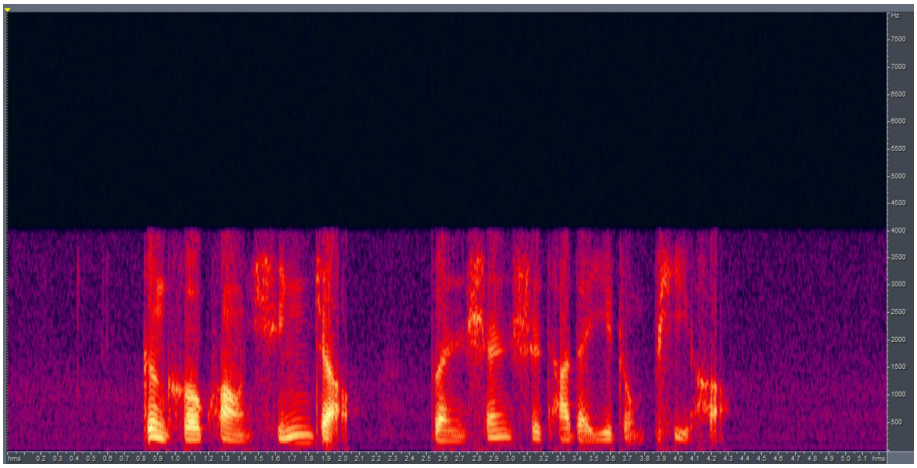
6.wav



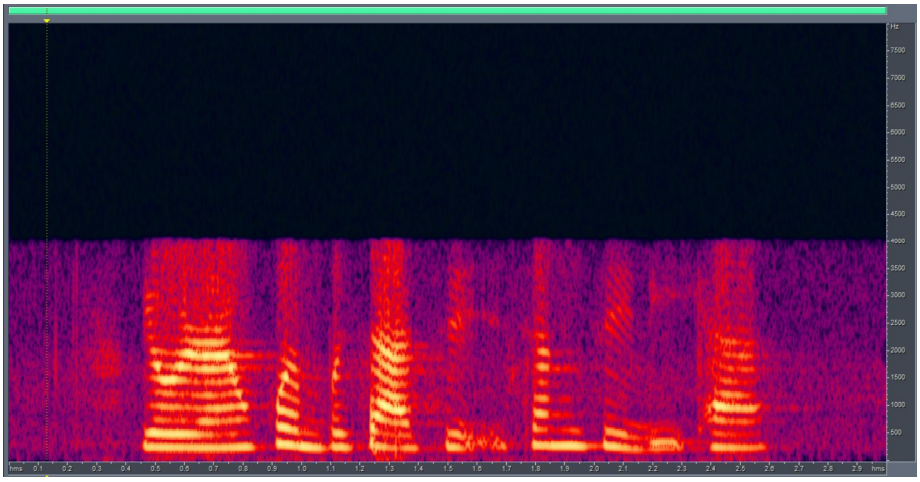
7.wav



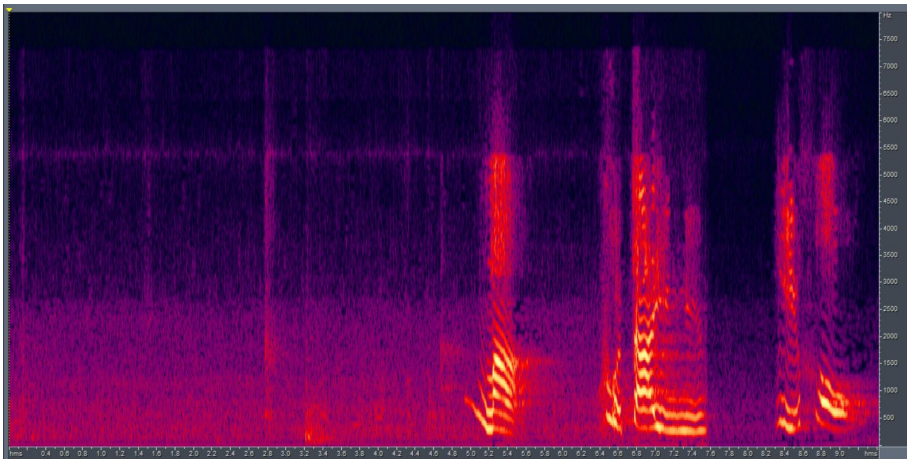
8.wav



9.wav

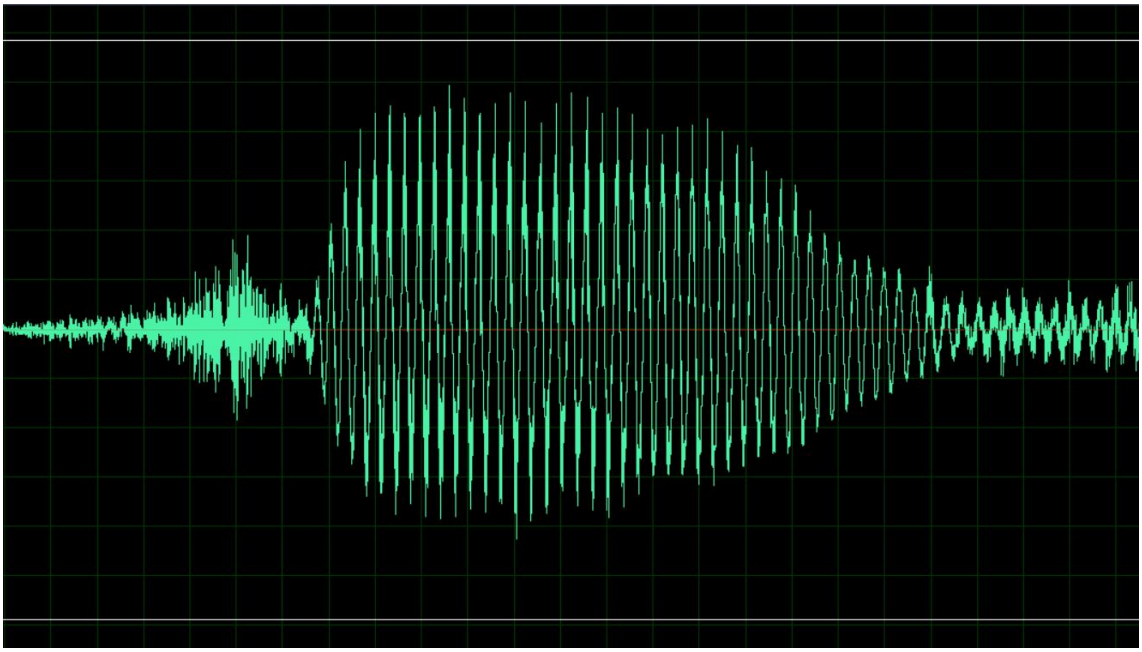


10.wav

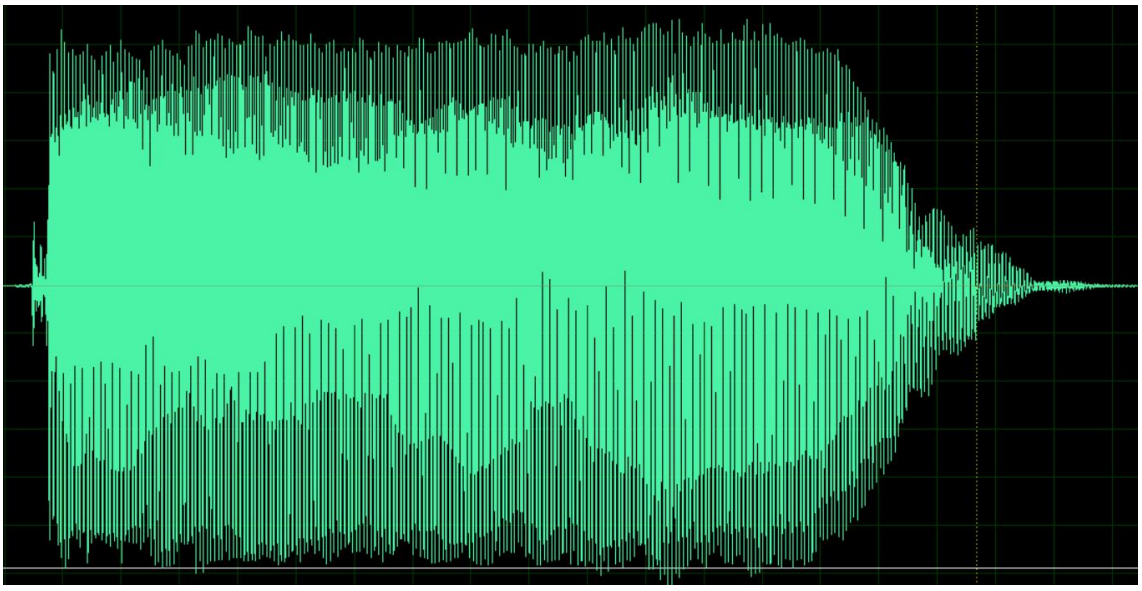


1.3 第一个音节的时域波形截图

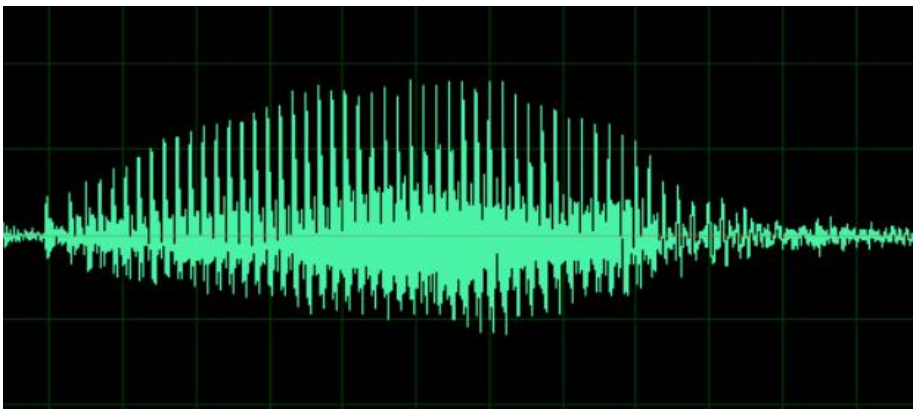
1.wav



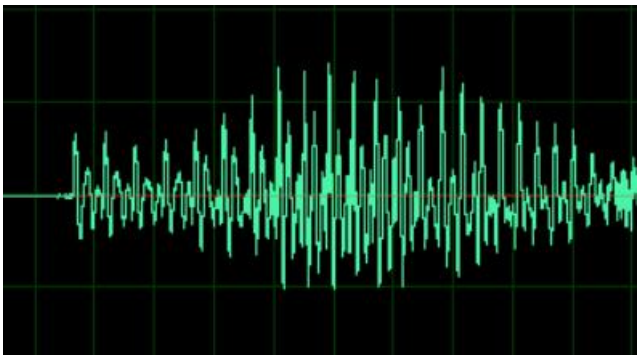
2.wav



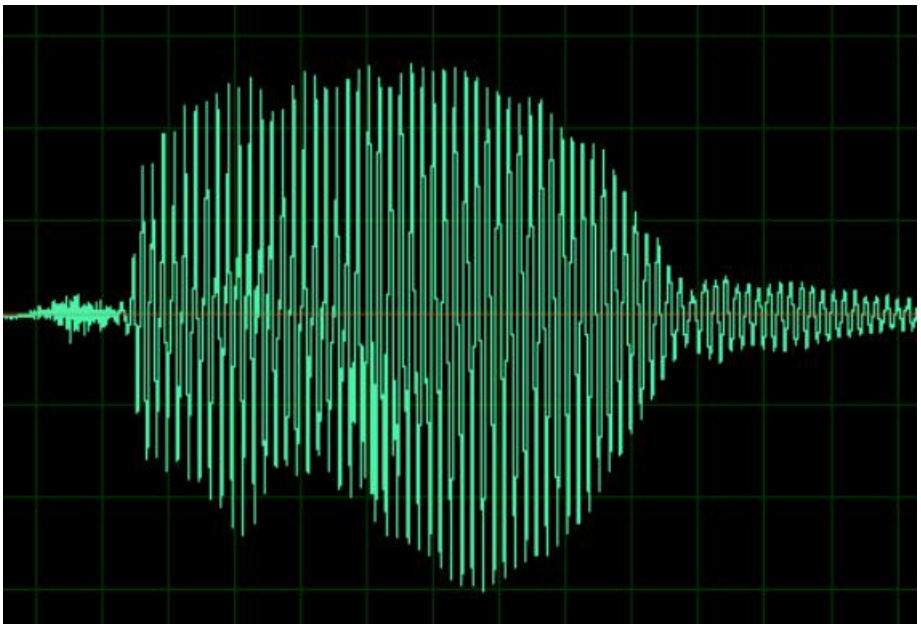
3.wav



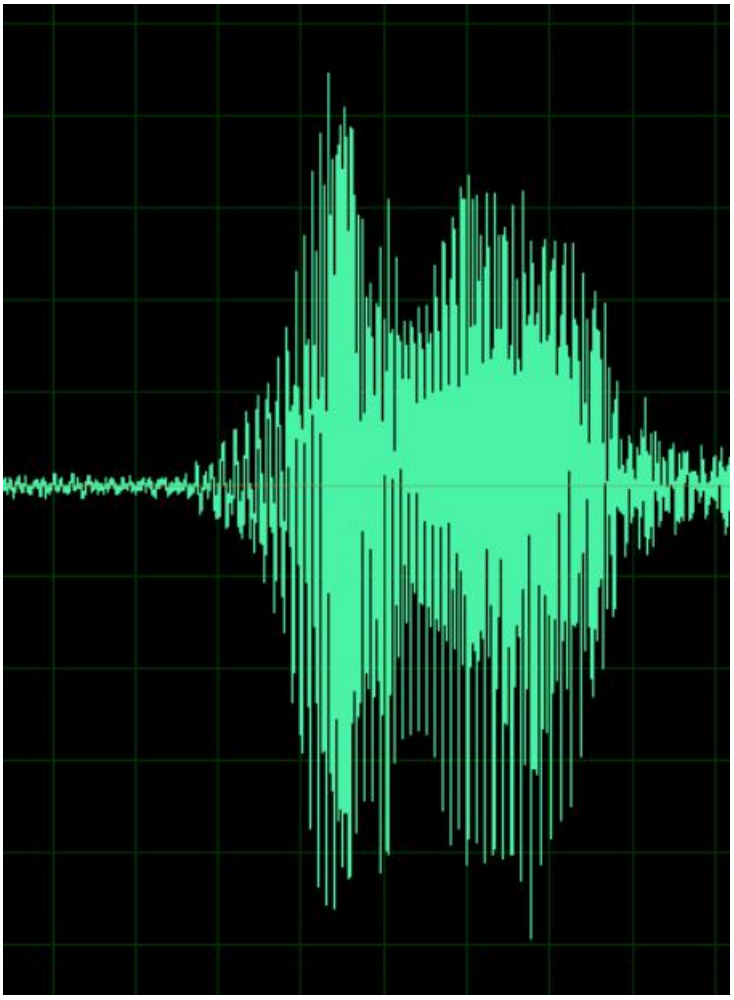
4.wav



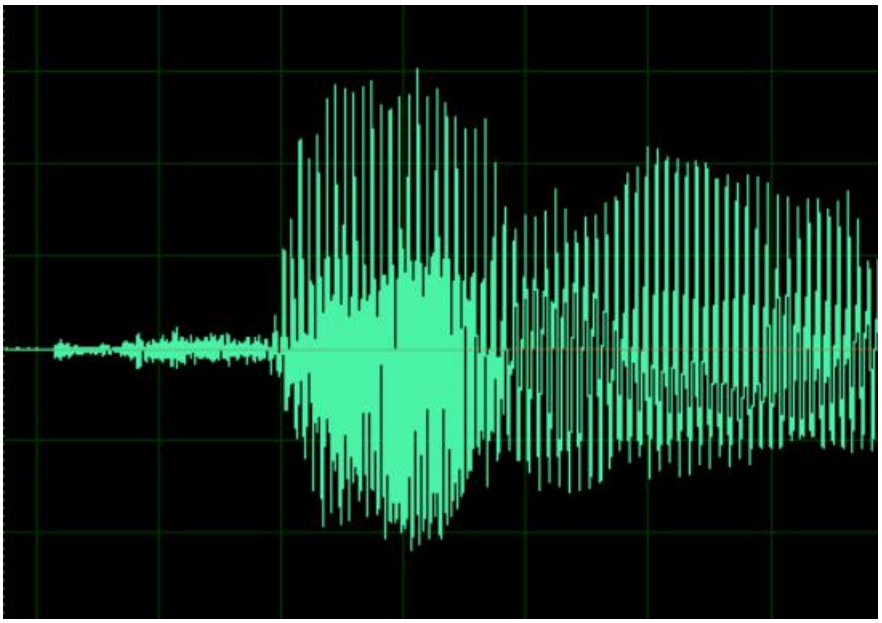
5.wav



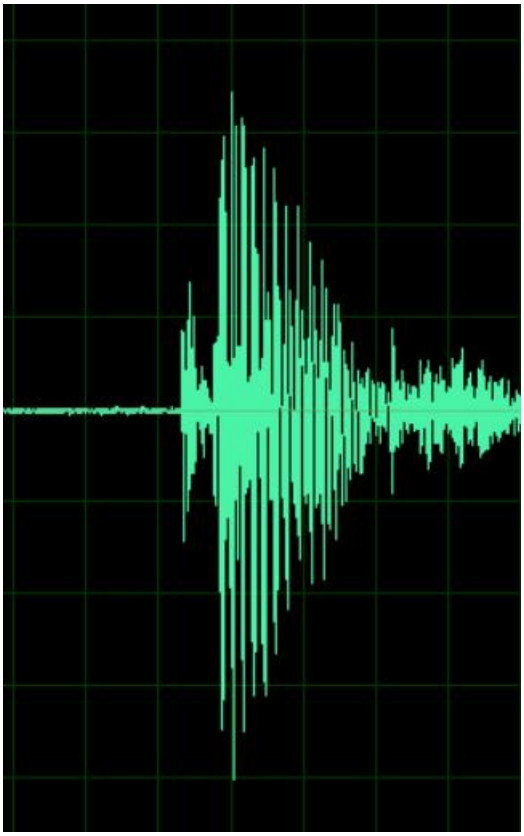
6.wav



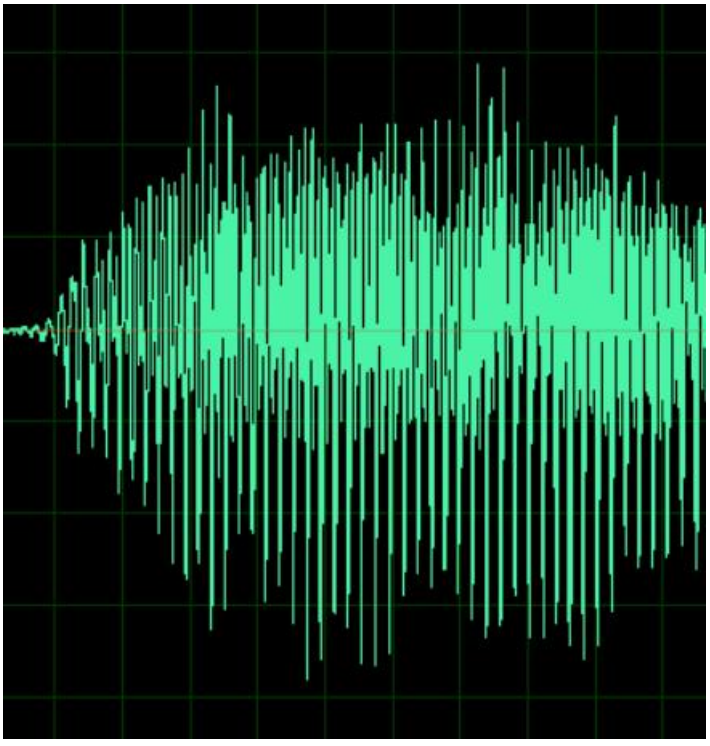
7.wav



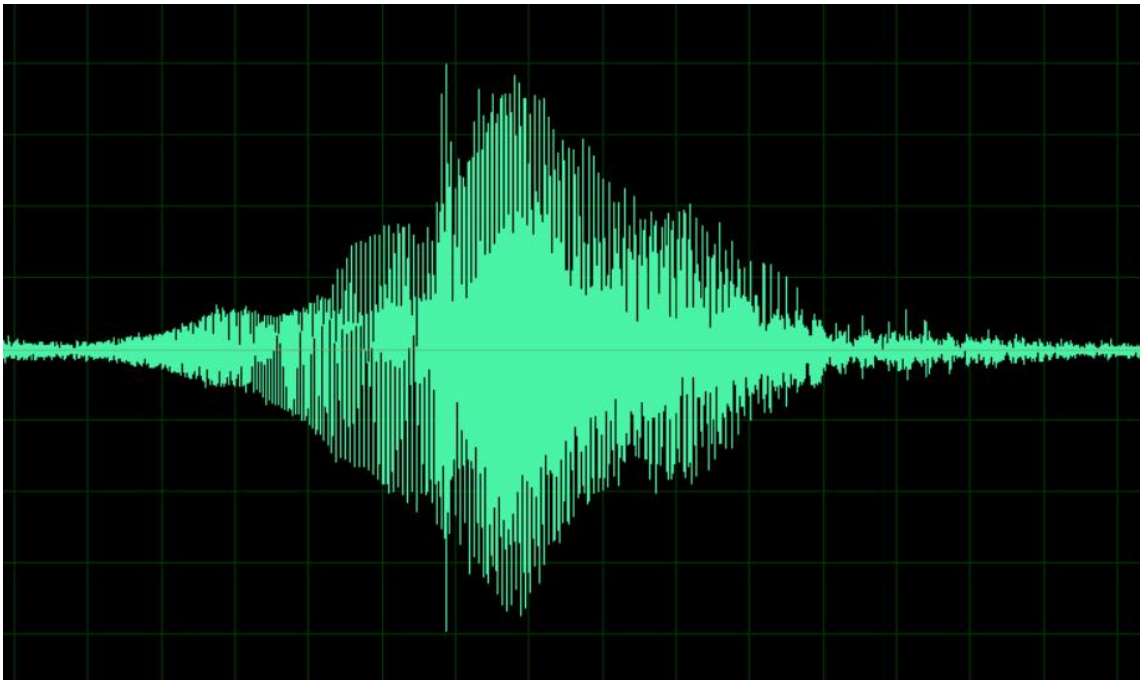
8.wav



9.wav



10.wav



1.4 语料的格式

采样频率　=16000HZ
量化比特数=16bit
声道个数　=1(mono)

二、能量和过零率特征提取

2.1 给出特征提取算法，给出概要性介绍，标明所采用的开发工具

开发工具：
Python3.9，pycharm2021.1.3，操作系统 Windows10 64 位

从文件中提取数据：

```
def readfile(filepath):
    f = wave.open(filepath, 'rb')
    global params
    params = f.getparams()
    str_data = f.readframes(params[3])
    f.close()
    wave_data = np.fromstring(str_data, dtype=np.short)
    return wave_data
```

计算短时能量：
使用下方公式进行计算：

$$E_n = \sum_{m=-\infty}^{\infty} [x(m)w(m)]^2$$

```
def cal_energy(wave_data):
    step = 256
    wave_data_len = len(wave_data)
    frame_num = math.ceil(wave_data_len / step)
    energy = []
    for i in range(frame_num):
        frame = wave_data[np.arange(i * step, min(i * step + step, wave_data_len))]
        frame_mat = np.mat(frame)
        sum = np.longlong(frame_mat) @ np.longlong(frame_mat.T)
        energy.append(sum[0, 0])
    return energy
```

能量特征属于时域特征，因此这里的窗口函数 w(m)是方窗，对于每一个要计算的帧而言为是 1，因此可以忽略。由于每一帧里有 256 个采样点，数据是离散的，因此计算时使用平方求和的形式，每一帧采样点平方和对应的就是该帧的短时能量可使用矩阵对内积进行加速计算。

计算短时过零率：

使用下方公式进行计算：

$$Z_n = \sum_{m=-\infty}^{\infty} \left|\text{sgn}[x(m)] - \text{sgn}[x(m-1)]\right| w(m)$$

其中

$$\text{sgn}[x(n)] = \begin{cases} 1 & x(n) \geq 0 \\ -1 & x(n) < 0 \end{cases}, \quad w(n) = \begin{cases} 1/2N & 0 \leq n \leq N-1 \\ 0 & \text{其它} \end{cases}$$

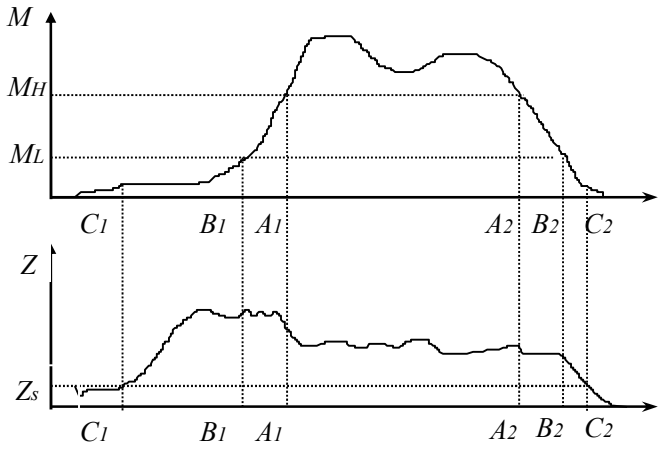
```
def cal_0cross_rate(wave_data):
    step = 256
    wave_data_len = len(wave_data)
    frame_num = math.ceil(wave_data_len / step)
    zcr = []

    def sgn(list):
        for i in range(len(list)):
            list[i] = 1 if list[i] >= 0 else -1
        return list

    for i in range(frame_num):
        frame = wave_data[np.arange(i * step, min(i * step + step, wave_data_len))]
        zcr.append(sum(abs(sgn(frame[0:-1]) - sgn(frame[1::]))) / (2 * 256))
    return zcr
```

三、 端点检测算法

3.1 给出端点检测算法，给出概要性介绍，标明所采用的开发工具

端点检测算法就是找到语音的端点，取出语音中的静音部分。

本次实验中，如果利用双门限法进行的端点检测，算法一共分为三步，三个阈值，包括两个短时能量阈值和一个过零率阈值。算法过程如下：



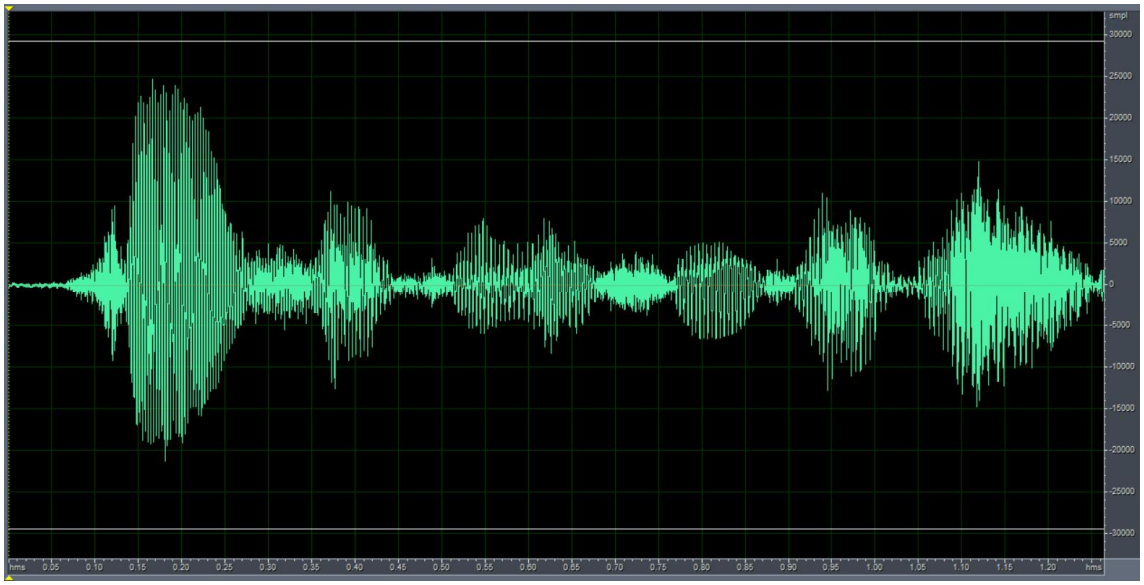
首先取一个较高的短时能量作为阈值 MH，利用这个阈值找出语音中的浊音部分，即图中的 A1 到 A2 区间。

然后取一个较低的能量阈值 ML，利用这个阈值从 A1A2 向两端进行搜索，将较低能量段的语音部分也加入到语音段，进一步扩大语音段的范围。即在图中 B1 到 B2 也是语音段。

第三步是利用短时过零率阈值 ZS，由于语音的两端部分是辅音即清音部分，也是语音中的一部分，但是辅音的能量与静音部分的能量均较低，然而过零率比静音部分高出很多。为区分开二者，将利用短时能量区分完的语音段继续向两端进行搜索，短时过零率大于 3 倍 Zs 的部分，则认为是语音的清音部分。将该部分加入语言段，就是求得的语音段，即如图 C1-C2 部分。

四、 计算检测正确率

4.1 “1.wav”语料去除静音后的时域波形截图



4.2 正确率

正确检出文件的个数：10

正确率=100%

五、 总结

10.1 请总结本次实验的收获

首先熟悉了软件 cooledit 的使用，通过查看波形图对语音的特征有了一些基本的了解。
通过编写代码计算语音的能量、过零率，对语音信号的特点更加了解了。

10.2 请给出对本次实验内容的建议

无。