

实验一 自动识别技术——语音识别系统的设计与实现（2）

1. 音频文件可视化

将音频信息数字化实际上是语音识别的第一步，数字化之后的音频信息更方便处理，同时，可以通过图像可视化地表示出来。

1.1 Numpy 模块

Numpy 模块是 Python 一个运行速度非常快的数学库，提供了线性代数、傅里叶变化、随机数生成等功能，对于大量的数据操作，用其更方便。

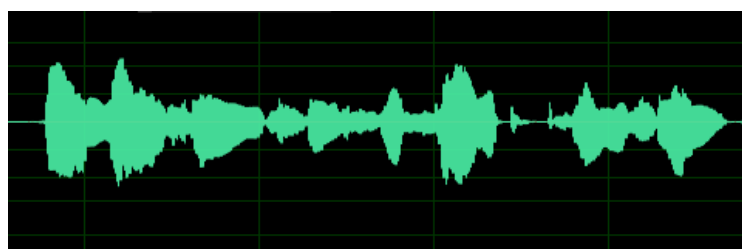
1.2 Matplotlib 模块

Matplotlib 是 Python 最著名的绘图库，适合交互式制图。

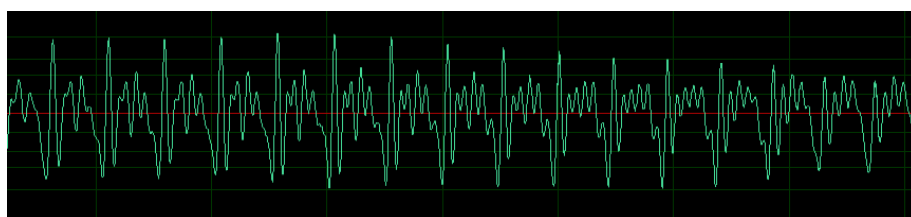
1.3 时域图

语音的频域分析和时域分析是两种常用的全面分析信号特征的方法。

时域图：语音波形。表现的是一段音频在一段时间内音量的变化，其横轴是时间方向，纵轴是振幅方向。



从图中可以得到各个音的起始位置，但很难看出更加有用的信息。但如果我们将其放大到 100ms 的场景下，可以得到下图所示。



时域分析适用于分析信号的瞬时特性，比如声音的起伏、振幅、声音强度、脉冲、波形等。

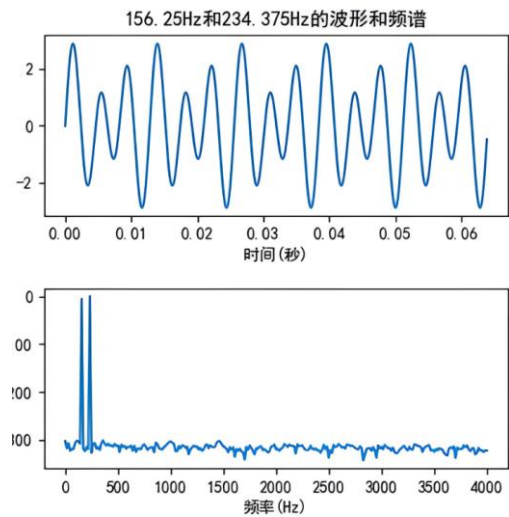
1.4 频谱图

时域分析可以直观的显示信号的瞬时变化，分析音高等时间特征，但无法揭示信号的频率特性。语音识别的基本原理是先分析声波中各种频率的组成及其频率的时变模式，再

结合语音数据库中的数据进行匹配运算，得出“语音”所包括的语言信息。

频谱图将一个声音中存在的不同频率成分以直观的图像形式展示出来，横轴为频率，纵轴为能量（强度）。常使用傅里叶变换（Fourier Transform）将语音信号从时域转换为频域。傅里叶变换可以将一个信号分解为一系列的正弦和余弦函数。对频谱数据可视化，我们可以获取语音信号在不同频率上的能量分布情况，从而了解语音信号的频率特征。

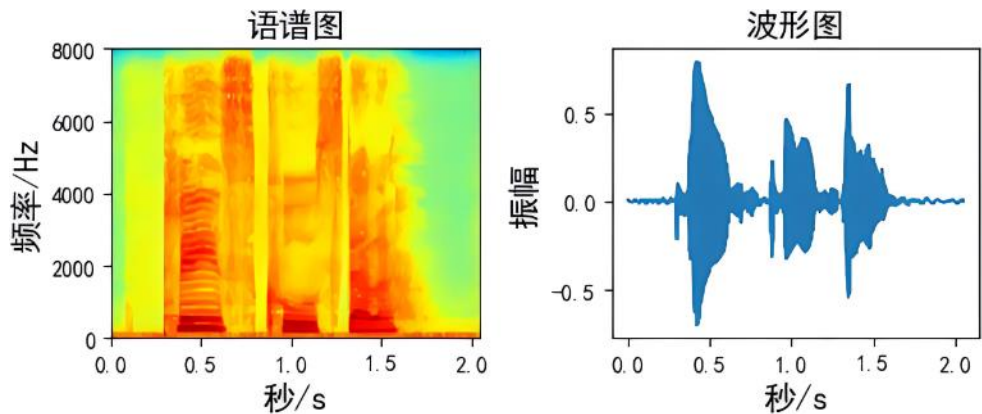
下图展示了频率为 156.25Hz 和 234.375Hz 的两个正弦波叠加后的信号时域图和频谱图



根据频谱图的形状、峰值、频率分量等特征进行分析。可以观察频谱图上的频率分布情况、能量集中区域，进一步进行音频处理、故障诊断、信号调制等方面的工作。

1.5 语谱图

声波信号由时域图谱变换到频域图谱，使得信号丢失了时间信息，而发音的先后顺序也是非常重要的，这就需要另一种表现音频特征的方法---语谱图。



语谱图的横坐标为时间，纵坐标为对应时间点的频率。坐标中的每个点用不同颜色表示，颜色越亮表示频率越大，颜色越淡表示频率越小。

2. 实验内容及要求

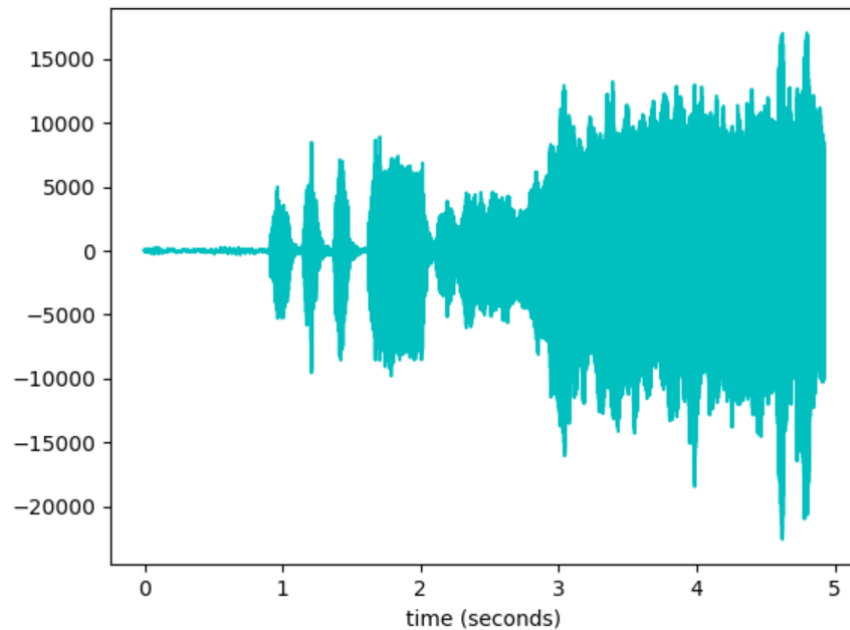
1) 实验内容

➤ 熟悉 Numpy 和 matplotlib 库的用法

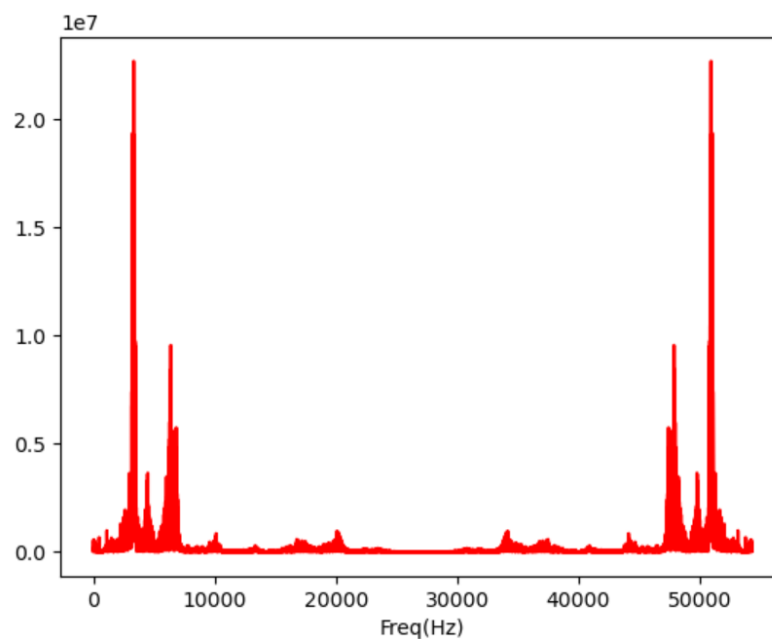
- (1) 利用 numpy 定义一个 5*5 大小的矩阵，使用 matshow 函数绘制矩阵。
- (2) 将(1)中的矩阵通过三维视图的形式显示出来。提示：可以使用 Axes3D 对象的 plot_surface()方法。

➤ 显示音频，即将采样的音频信号按照某种规则以曲线的形式展示出来。

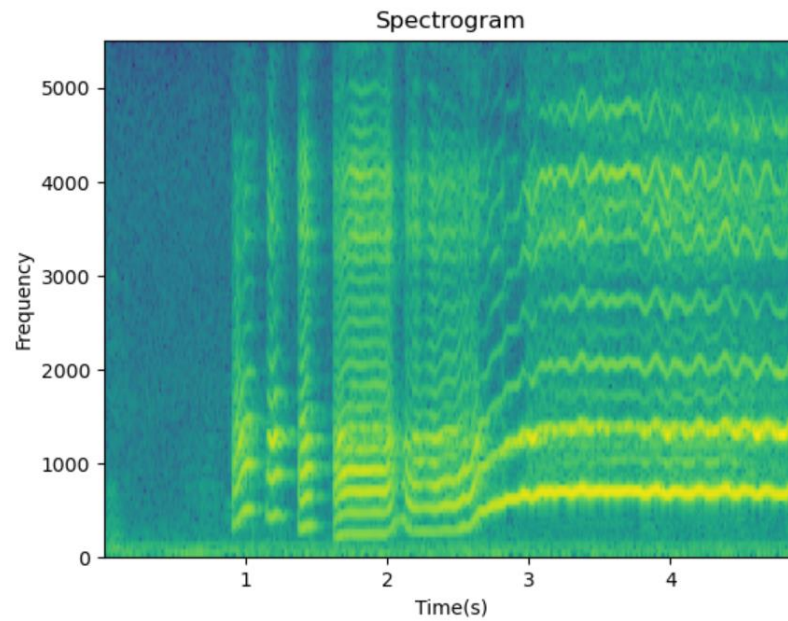
- (1) 制作时域图：利用 numpy、wave、matplotlib 模块，为上节实验课生成的 wav 音频文件创建时域图，展示随时间变化的能量关系。



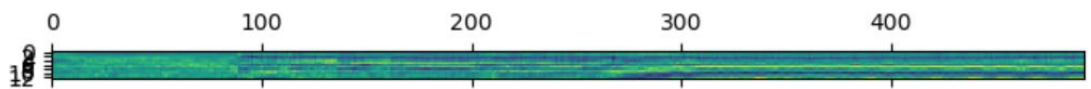
- (2) 制作频域图：将时域图转换为频域图需要用到傅里叶变换（FFT），即将一个信号分解为不同频率的组合，同时给出每一种频率的大小，通过 numpy.fft 模块中的 fft 函数实现傅里叶变换。



- (3) 制作语谱图：通过 matplotlib.pyplot 模块中的 specgram 函数，实现绘制上节实验课生成的 wav 音频文件的语谱图。



- (4) 利用 Python_speech_features 模块中的 mfcc 函数实现对音频提取 MFCC 特征，制作提取 MFCC 特征后的音频语谱图。



2) 上传要求

- (1) 运行效果图
- (2) 源程序文件