**语音识别技术实验报告**

**姓名 陆昊宇 ­­­­ 学号 19220432 ； 22 级 04 班**

**成 绩\_\_\_\_\_\_\_\_\_\_\_\_**

**课程名称：语音识别技术 教师签字\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
| **实验内容** | **语音降噪算法** | **实验时间** |  |
| **实验目的** | 1. 根据双门限法的原理，编写Python函数，并基于测试语音C4\_1\_y.wav 实现端点检测。 2. 编写基本谱减法函数：根据基本谱减法的原理，编写函数，并基于测试语音C6\_1\_y.wav 叠加不同信噪比的噪声，进行降噪测试。 3. 编写基本维纳滤波函数：根据基本维纳滤波法的原理，编写函数，并基于测试语音C6\_2\_y.wav 叠加不同信噪比的噪声，进行降噪测试。 | | |
| **实验工具**  （说明使用的库及其版本） | # 实验环境配置 !pip install soundfile librosa matplotlib pystoi pesq import numpy as np import soundfile as sf import librosa import matplotlib.pyplot as plt from IPython.display import Audio | | |
|  | 1. **相关算法原理描述**  端点检测（双门限法） **双门限检测流程**：   谱减法降噪 其中为过减因子，为噪声下限 维纳滤波降噪 其中为先验信噪比估计   1. **代码**  端点检测（双门限法） def endpoint\_detection(  file\_path, frame\_len=400, step=160,   ITL=0.3, ITU=0.2, ZCR\_th=50, min\_silence=10 ):  # 读取音频  y, sr = sf.read(file\_path)  if y.ndim > 1: y = y[:,0] # 转为单声道    # 预加重  y = np.append(y[0], y[1:] - 0.97 \* y[:-1])    # 分帧  frames = []  n\_frames = (len(y) - frame\_len) // step + 1  for i in range(n\_frames):  frames.append(y[i\*step : i\*step+frame\_len])    # 计算短时能量和过零率  energy = np.array([np.sum(frame\*\*2) for frame in frames])  zcr = np.array([0.5 \* np.sum(np.abs(np.diff(np.sign(frame)))) for frame in frames])    # 归一化  energy = (energy - np.min(energy)) / (np.max(energy) - np.min(energy))    # 端点检测  in\_speech = False  speech\_segments = []  silence\_count = 0  start = 0    for i in range(len(energy)):  if not in\_speech:  if energy[i] > ITL or zcr[i] > ZCR\_th:  start = max(0, i - 2)  in\_speech = True  else:  if energy[i] < ITU and zcr[i] < ZCR\_th:  silence\_count += 1  if silence\_count >= min\_silence:  end = i - silence\_count  speech\_segments.append((start\*step, end\*step))  in\_speech = False  silence\_count = 0  else:  silence\_count = 0    return speech\_segments, y, energy, zcr 谱减法降噪 def spectral\_subtraction(noisy, sr, n\_fft=512, hop\_length=160, win\_length=400, noise\_frames=5):  # 计算STFT  D = librosa.stft(noisy, n\_fft=n\_fft, hop\_length=hop\_length,   win\_length=win\_length, window='hann')  mag = np.abs(D)    # 估计噪声谱  noise\_profile = np.mean(mag[:, :noise\_frames], axis=1, keepdims=True)    # 谱减法  mag\_enhanced = np.maximum(mag - noise\_profile, 0)    # 重建信号  D\_enhanced = mag\_enhanced \* np.exp(1j \* np.angle(D))  enhanced = librosa.istft(D\_enhanced, hop\_length=hop\_length,   win\_length=win\_length, window='hann')    return enhanced 维纳滤波降噪 def wiener\_filter(noisy, sr, n\_fft=512, hop\_length=160, win\_length=400, noise\_frames=5):  # 计算STFT  D = librosa.stft(noisy, n\_fft=n\_fft, hop\_length=hop\_length,  win\_length=win\_length, window='hann')  power = np.abs(D)\*\*2    # 估计噪声功率  noise\_power = np.mean(power[:, :noise\_frames], axis=1, keepdims=True)    # 维纳滤波  snr\_prior = np.maximum((power - noise\_power) / noise\_power, 1e-6)  H = snr\_prior / (1 + snr\_prior)    # 应用滤波  D\_enhanced = D \* H  enhanced = librosa.istft(D\_enhanced, hop\_length=hop\_length,  win\_length=win\_length, window='hann')    return enhanced   1. **运行结果**  端点检测（双门限法）  谱减法降噪 原始音频    加噪音频    降噪音频   维纳滤波降噪 原始音频    加噪音频    降噪音频     1. **实验结论及收获建议**  * 掌握了语音端点检测的基本原理 * 理解了传统降噪算法的实现差异 | | |