

Project 1:基于线性分类器和语音短时特征的简单语音端点检测

- 整个项目的文件结构：
 - root
 - Project1
 - Utils.py
 - Dataset_construct.py
 - Detection.py
 - Test_AUC_EER_Score.py
 - Evaluate.py（评估接口，由助教提供）
 - Vad_utils.py（读入标签接口，由助教提供，仅在Test_AUC_EER_Score.py中为模拟环境使用，其余为自己实现的读入函数）
 - Model_save
 - Model.pkl
 - Input:（由Dataset_construct.py生成）
 - Features:
 - Dev_features.npy
 - Test_features.npy
 - Train_features.npy
 - Labels:
 - Dev_labels.npy
 - Train_labels.npy
 - Output:
 - Test_prediction.txt（test数据集的预测结果）
 - Train_prediction.txt（train数据集（前500个wav）的预测结果，仅用于代替test进行验证）
 - VAD
- Utils.py：
 - 提供原始的获取输入函数get_input()
 - 语音信号的预处理和特征提取
- Dataset_construct.py：.
 - 分别构建dev、test和train数据集的特征向量和标签集，并保存到特定路径
 - 需要在main里面选择三个部分中的一个uncomment并运行，另外两个保持被注释状态
- Detaction.py：
 - 包含三个部分：

- Dev_main(): 读取dev特征向量和标签, 训练模型并保存, 指标评估
- Test_main(): 读取test特征向量, 读取模型, 进行预测并保存结果到./output/test_prediction.txt
- Train_main() (作为测试集的替代): 读取前500个train特征向量和标签, 读取模型, 进行预测并保存结果到./output/train_prediction.txt, 进行指标评估
- 需要运行哪个部分直接在main函数里面uncomment, 另外两个comment即可
- Test_AUC_EER_Score.py:
 - 目的是尽可能模拟助教的测试环境, 将train数据集 (前500个) 作为test训练集的替代进行指标评估
 - 使用助教提供的read_label_from_file()接口分别读取数据集提供的train_labels.txt和上面生成的./output/train_prediction.txt, 进行一些简单的长度补全后, 使用提供的get_metrics()接口计算平均AUC和EER得分
 - 设计该脚本的目的是, 之前的评估全部是直接基于帧进行预测的, 并没有将以帧为单位的预测结果转化为以秒为单位的txt格式, 因此该步骤对于结果的泛化性验证是必要的。
- 完整的项目演示可以参见“项目完整演示视频”。