

实验报告

一 实验数据预处理

1. 在提取 MFCC 特征以及训练 GMM 模型时，需要使用 .wav 格式的音频文件，所以，我们首先把训练集（train 文件夹）、测试集（dev 文件夹）、预测集（eval 文件夹）下的 .flac 格式的音频文件转化为 .wav 格式的音频文件。我使用 ffmpeg 这一多媒体处理工具来完成 .flac 文件到 .wav 文件的转换。

代码如下：

```
def flac_to_wav(filepath, savedir):
    filename = filepath.replace('.flac', '.wav')
    savefilename = filename.split('\\')
    print(savefilename)
    save_dir = savedir + '\\' + savefilename[-1]
    print(save_dir)
    cmd = r"D:\ffmpeg\ffmpeg-4.4-full_build\bin\ffmpeg.exe -i " + filepath +
    ' ' + save_dir
    os.system(cmd)

if __name__ == "__main__":
    # 批量处理，将音频文件从 .flac 格式变为 .wav 格式
    savedir = r"D:\zju_deepfake\train\dev"      #用户可以改变，为输出文件的位置
    path = r"D:\zju_deepfake\res"              #用户可以改变，为输入文件的位置
    for root, dirs, files in os.walk(path):
        for name in files:
            filepath = root + "\\" + name
            print("filepath"+filepath)
            if filepath.split('.')[-1] == "flac":
                flac_to_wav(filepath, savedir)
```

说明：ffmpeg 多媒体处理工具下载网址：ffmpeg <https://ffmpeg.org/>

2. 因为要分别训练真假语音的 GMM 模型，所以需要将训练集分成 2 个文件夹，一个存放真实语音（true 文件夹），一个存放合成语音（false 文件夹）。

二 实验思路

1. 首先在真实语音训练集（true 文件夹）上提取 MFCC 语音特征，训练出一个真实语音 GMM 模型 1：T.out。然后，在合成语音训练集（false 文件夹）上提取 MFCC 语音特征，训练出一个合成语音 GMM 模型 2：F.out。



2. 在进行预测时，对于每一条语音数据，分别用 GMM 模型 1 (T.out)、GMM 模型 2 (F.out) 进行训练，取得分高的为最后结果。

3. 对该实验中所给的测试集 (dev 文件夹) 进行预测，并与实验所给的测试集的结果 (dev.txt 文件) 进行比较，计算出准确率。可以尝试改变参数，体会其对准确率的影响。

4. 根据实验要求，分别对预测集 (eval1 文件夹、eval2 文件夹) 中的语音数据进行预测，并将结果按照实验规定格式输出。

三 实验步骤

1. 首先训练真实语音的 GMM 模型，在命令行输入：

```
python speaker-recognition.py -t enroll -i "./true" -m "T.out"
```

2. 训练合成语音的 GMM 模型，在命令行输入：

```
python speaker-recognition.py -t enroll -i "./false" -m "F.out"
```

3. 分别用真实语音 GMM 模型和合成语音进行预测，得分高的为预测结果，在命令行输入：

```
python speaker-recognition.py -t predict -i "./testall"
```

4. 用给定的数据集 dev，测试训练出的模型的准确率，在命令行输入：

```
python speaker-recognition.py -t predict -i "./testall"
```

四 算法说明

1. 特征提取

本实验采用 python_speech_features 包来提取 MFCC 特征。

2. 模型训练

本实验采用 sklearn 库来搭建 GMM 模型，

GaussianMixture.fit() 方法可以从训练数据中拟合出一个高斯混合模型。

scores= GaussianMixture.predict() 方法，可以为每个样本分配最适合的高斯分布模型。在预测时，分别调用 GMM 模型 1 (T.out)、GMM 模型 2 (F.out) 得到对应的两个 scores，取较大的 scores 所用的模型标签为实验的预测结果。