Audio Spectrogram Transformer（AST）使用及训练

```
├─checkpoint-95 # 模型文件
├─recognition # 识别音频代码
|  └─__pycache__
├─train # 训练代码
|  ├─audio
|  |  ├─cat
|  |  ├─dog
|  |  └─pig
|  └─dataset
└─__pycache__
```

1. 直接使用训练好的模型推理

首先新建一个环境

```
conda create -n your_env_name python=3.8
```

然后安装`requirements.txt`的所有python包

```
pip install -r requirements.txt
```

进入`recognition/recognition_demo.py`，展示如何使用已经训练好的模型进行推理

```
from recognition import recognize
import os

if __name__ == "__main__":
    # 如果不加这几行，会报错：OSError: [Errno 2] No such file or directory:
'test.wav'

    # 获取当前脚本所在的绝对路径
    current_script_path = os.path.abspath(__file__)
    # 获取当前脚本所在的目录
    current_script_directory = os.path.dirname(current_script_path)
    # 设置当前工作目录为脚本所在的目录
    os.chdir(current_script_directory)

    # 音频最好使用5s，因为训练集全是5s的音频
    audio_path = "test.wav"
    model_path = r"../checkpoint-95"
    predicted_labels, top_probs = recognize(audio_path, model_path)

    # 打印前五个类别及其概率
    print(predicted_labels)
    print(top_probs)
```

其中需要修改的地方：

- `audio_path`：换成需要读取的音频地址，音频长度最好为5s，如果大于5s会被截断为5s
- `model_path`：换成其他的微调后的model_path，如果没有新的模型，则默认使用`checkpoint-95`(这是目前微调过后效果最好的模型)

返回的类容是：

- `predicted_labels`：评分前五的音频标签，但是是英文（如果是中文，训练时可能报错），中英文对照关系如下

```
airplane-飞机声
birds-鸟鸣声
chicken-鸡鸣声
construction-施工声（包括工地，装修等）
dog-狗吠声
engine-引擎声（比如摩托车，拖拉机，大型机器等）
frog-蛙声
human-live-生活声（包括说话交谈，嬉笑等）
insect-虫鸣（包括蝉鸣，蛐蛐等）
rain-下雨声
sea-waves-海浪声
thunderstorm-暴风雨声（主要是包括打雷的声音）
traffic-交通声（比如车辆鸣笛等）
train-火车声
wind-风声
```

- `top_probs`：评分前五的音频标签的概率，**注意，该概率并没有实际意义，建议>0.8的视为正确，<0.8的类别舍弃**

2. 训练微调

如果有某类别声音一直识别不出来，则可以进行微调

1. 首先录取音频，最好是wav格式的音频，如果不是也可通过`train/preprocess.py`进行处理，主要包括以下几个函数，结果一定要处理成wav格式的5s为单位的音频文件

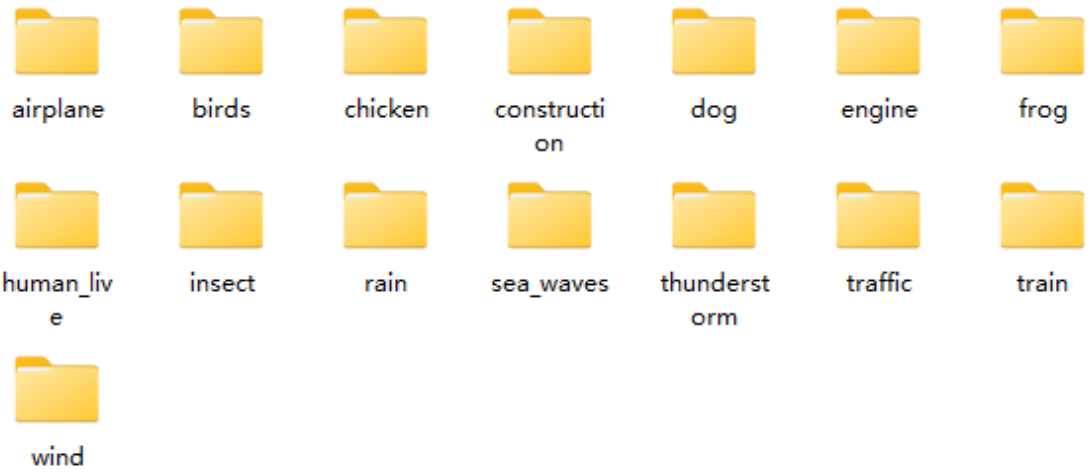
```
def convert_to_wav(input_folder, output_folder):
    """
    将输入文件夹中的音频文件转换为WAV格式，并保存到输出文件夹中
    :param input_folder:
    :param output_folder:
    :return:
    """

def split_audio(input_path, output_folder, segment_duration=5):
    """
    将音频文件切割为指定时长的片段，并保存到输出文件夹中
    :param input_path:
    :param output_folder:
    :param segment_duration:
    :return:
    """

def extract_audio_from_video(video_path, output_folder,
output_sample_rate=16000):
    """
```

```
    从视频中提取音频
    :param video_path:
    :param output_folder:
    :param output_sample_rate:
    :return:
    """
```

2. 切割好的音频进行人辨别后，进行分类，分为几种情况



   1. 如果是以上音频中某一种识别不准（比如有一种新的鸟鸣声，但是模型没有见过，识别不出来），**则直接将新的音频复制进`birds`即可**
   2. 如果是混合类别的音频识别不准（比如某地鸟鸣声经常会和蛙鸣声同时出现），**则将新的包含两种声音的音频同时复制进`birds`和`frog`**
   3. 如果是一种新的类别声音不包括在上述所有（比如想识别牛叫），**则新建一个文件夹`cow`，并把音频复制进其中即可。**

   切记，每次增加声音片段时，不宜过少也不宜过多，同一地点不同时段，5个该类别的5s音频片段为最佳

3. 训练集组织好了之后，直接进入`train/train.ipynb`，直接将整个文件运行完就可以了，然后就会出现新的训练的模型文件，选择训练完后的尾数最大的一个模型的整个文件夹，替换掉`checkpoint-95`即可（记得修改`recognition`中的模型路径）