计算机科学与技术学院神经网络与深度学习课程实验报告

实验题目：trigger word detection		学号：201900130015
日期：2021	班级：　智能班	姓名：李德锋
Email：ldf2878945468@163.com		

实验目的：

构建语音识别项目

合成和处理录音以创建训练/开发数据集

训练触发词检测模型并进行预测

实验软件和硬件环境：

Intel(R) Core(TM) i7-8550U CPU
华为云

实验原理和方法：

根据提示和公式原理补全代码
根据给出的答案检测是否正确

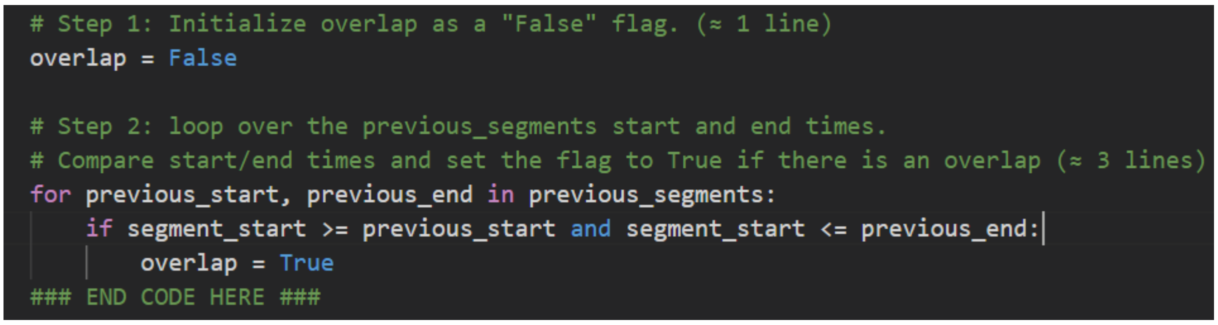
实验步骤：（不要求罗列完整源代码）

1. 导入所需要的包
2. 创建语言数据集

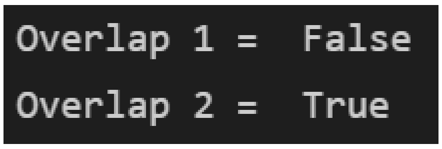
测试数据，设置频谱图时间步长为 5511

3. 生成单个训练

1). 使用辅助函数，将 10 秒音频离散化为 10000 步
2). 检测新的时间段是否与先前的段重叠

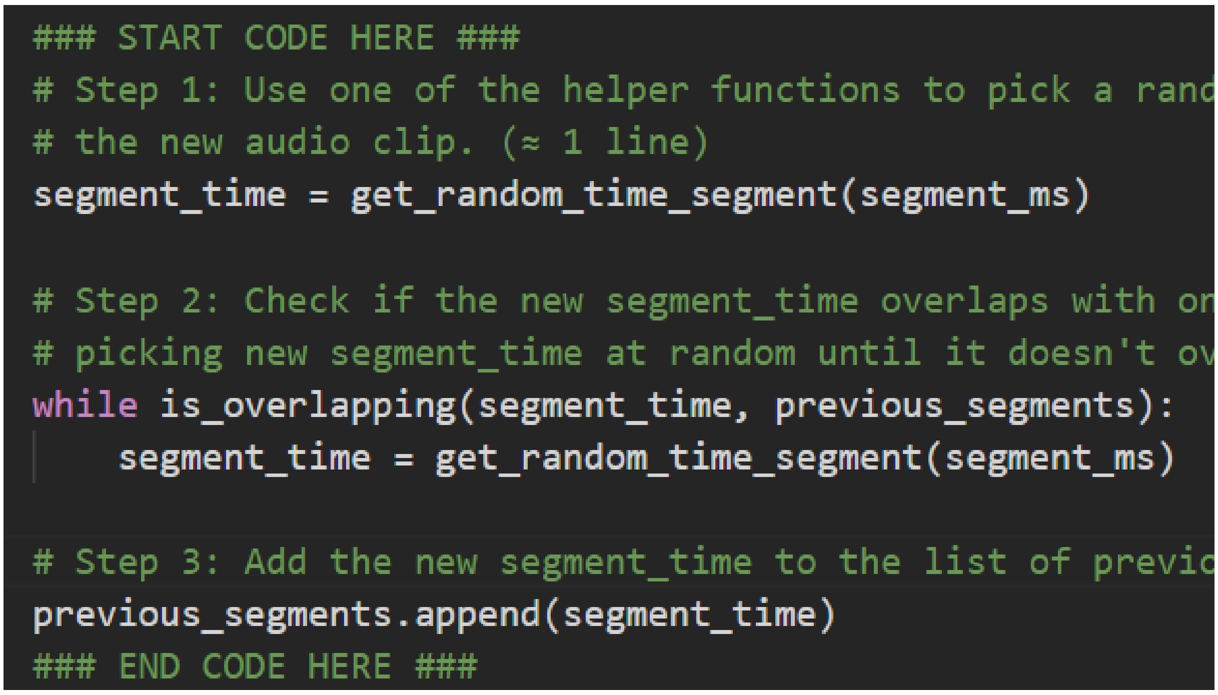


```
# Step 1: Initialize overlap as a "False" flag. (≈ 1 line)
overlap = False

# Step 2: loop over the previous_segments start and end times.
# Compare start/end times and set the flag to True if there is an overlap (≈ 3 lines)
for previous_start, previous_end in previous_segments:
    if segment_start >= previous_start and segment_start <= previous_end:
        overlap = True
### END CODE HERE ###
```

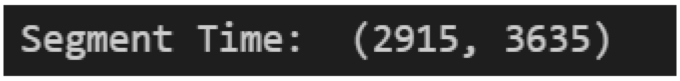


```
Overlap 1 =  False
Overlap 2 =  True
```

3). 在随机时间插入 10s 背景中，保证不会有重叠。首先确定插入的随机时间段，检查新的段时间是否与先前的段时间重叠，将新的分段时间添加到先前的分段列表中

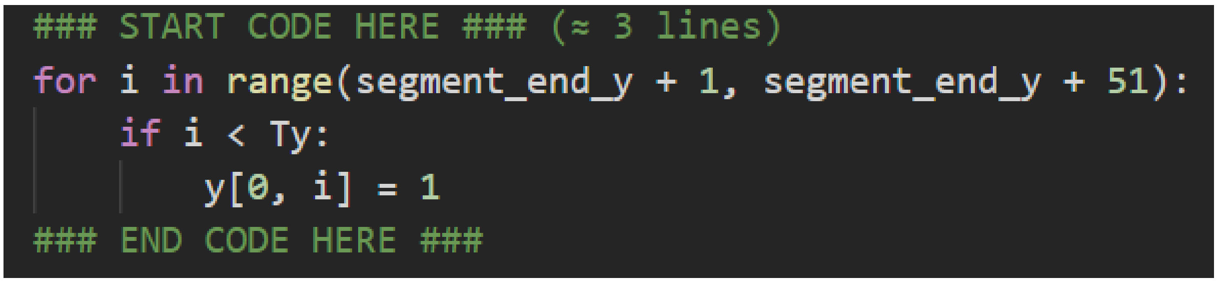


```
### START CODE HERE ###
# Step 1: Use one of the helper functions to pick a rand
# the new audio clip. (≈ 1 line)
segment_time = get_random_time_segment(segment_ms)

# Step 2: Check if the new segment_time overlaps with on
# picking new segment_time at random until it doesn't ov
while is_overlapping(segment_time, previous_segments):
    segment_time = get_random_time_segment(segment_ms)

# Step 3: Add the new segment_time to the list of previo
previous_segments.append(segment_time)
### END CODE HERE ###
```



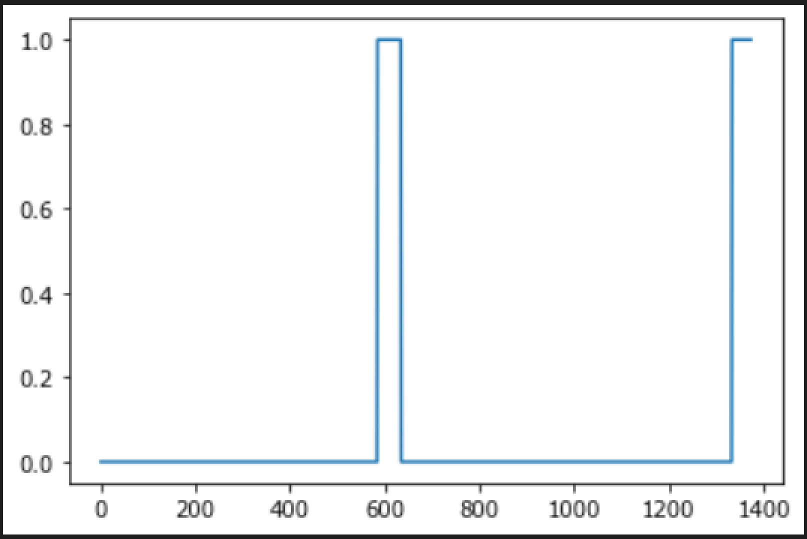
```
Segment Time:  (2915, 3635)
```

4). 更新标签向量 y, 50 个输出步骤的标签设置为 1

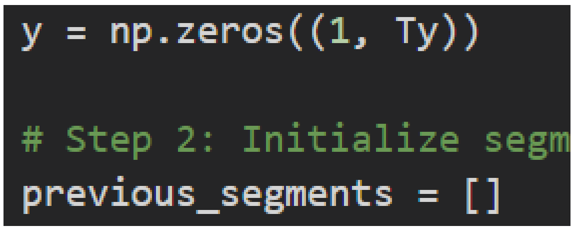


```
### START CODE HERE ### (≈ 3 lines)
for i in range(segment_end_y + 1, segment_end_y + 51):
    if i < Ty:
        y[0, i] = 1
### END CODE HERE ###
```

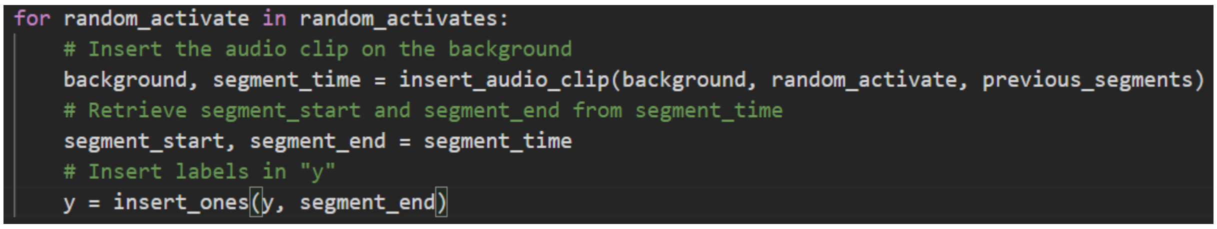
```
sanity checks: 0.0 1.0 0.0
```



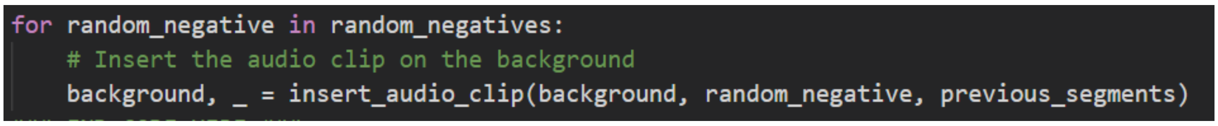
5).创建一个新的训练示例：将 y 标签初始化，将 existing segments 的集合初始化为空列表，随机插入音频



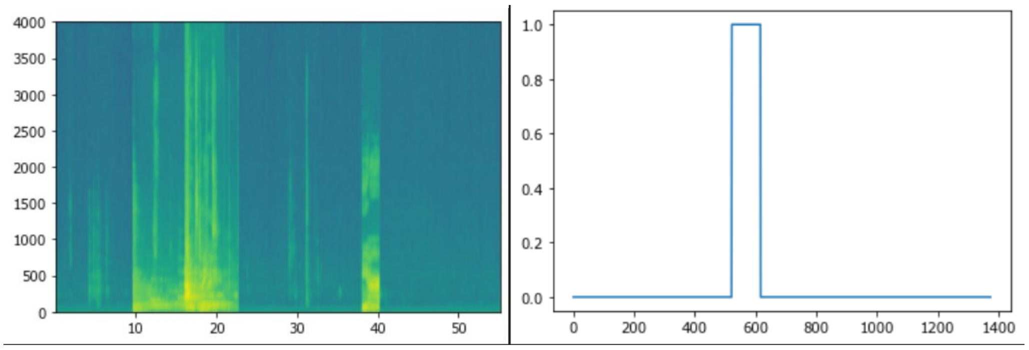
```
y = np.zeros((1, Ty))

# Step 2: Initialize segm
previous_segments = []
```



```
for random_activate in random_activates:
    # Insert the audio clip on the background
    background, segment_time = insert_audio_clip(background, random_activate, previous_segments)
    # Retrieve segment_start and segment_end from segment_time
    segment_start, segment_end = segment_time
    # Insert labels in "y"
    y = insert_ones(y, segment_end)
```



```
for random_negative in random_negatives:
    # Insert the audio clip on the background
    background, _ = insert_audio_clip(background, random_negative, previous_segments)
```

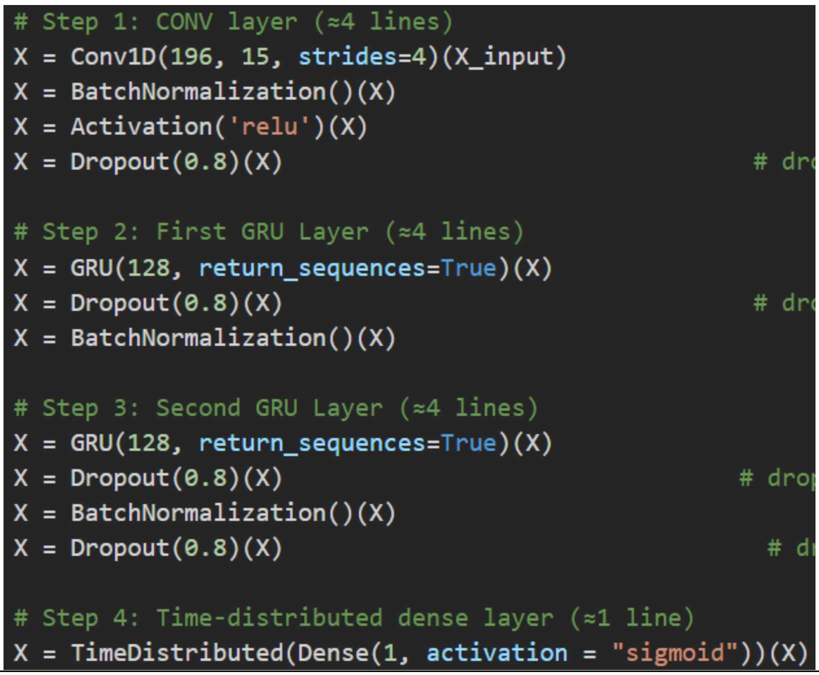


4. 完整训练集

1). 导入数据和包

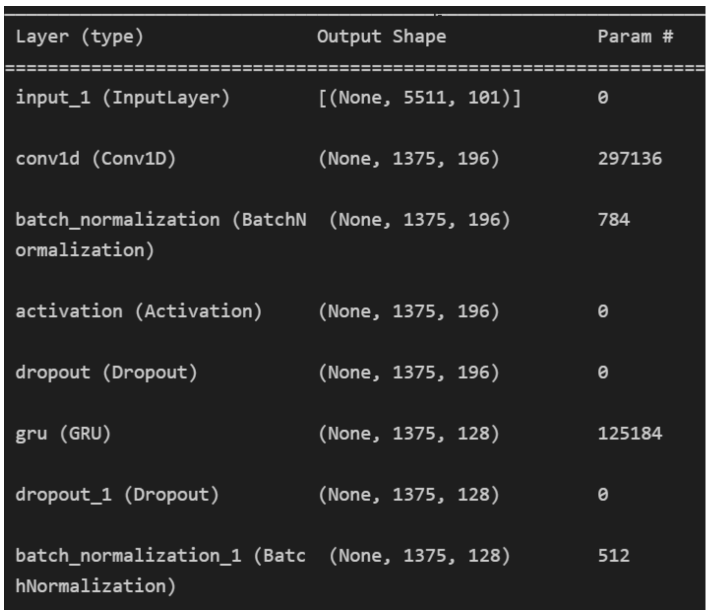
2). 建立模型

使用单向 RNN，实现 CONV 层，第一个 GRU 层，第二个 GRU 层，创建时间分布

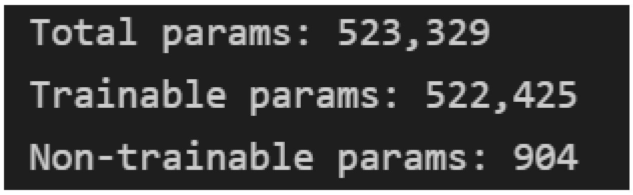


```
# Step 1: CONV layer (≈4 lines)
X = Conv1D(196, 15, strides=4)(X_input)
X = BatchNormalization()(X)
X = Activation('relu')(X)
X = Dropout(0.8)(X)                                    # dr

# Step 2: First GRU Layer (≈4 lines)
X = GRU(128, return_sequences=True)(X)
X = Dropout(0.8)(X)                                    # dr
X = BatchNormalization()(X)

# Step 3: Second GRU Layer (≈4 lines)
X = GRU(128, return_sequences=True)(X)
X = Dropout(0.8)(X)                                   # drop
X = BatchNormalization()(X)
X = Dropout(0.8)(X)                                     # d

# Step 4: Time-distributed dense layer (≈1 line)
X = TimeDistributed(Dense(1, activation = "sigmoid"))(X)
```

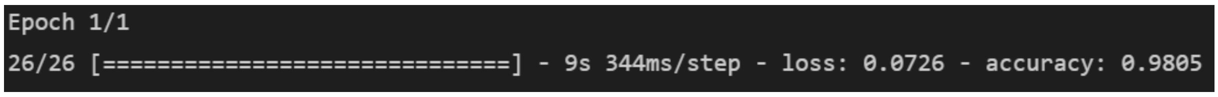
打印模型摘要以跟踪形状



```
Layer (type)                Output Shape              Param #
=================================================================
input_1 (InputLayer)        [(None, 5511, 101)]       0

conv1d (Conv1D)             (None, 1375, 196)         297136

batch_normalization (BatchN  (None, 1375, 196)        784
ormalization)

activation (Activation)     (None, 1375, 196)         0

dropout (Dropout)           (None, 1375, 196)         0

gru (GRU)                   (None, 1375, 128)         125184

dropout_1 (Dropout)         (None, 1375, 128)         0

batch_normalization_1 (Batc  (None, 1375, 128)        512
hNormalization)
```

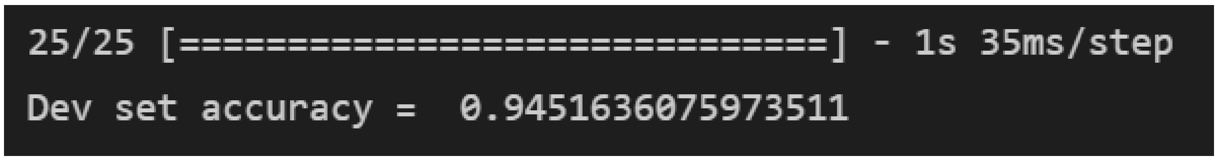


```
Total params: 523,329
Trainable params: 522,425
Non-trainable params: 904
```

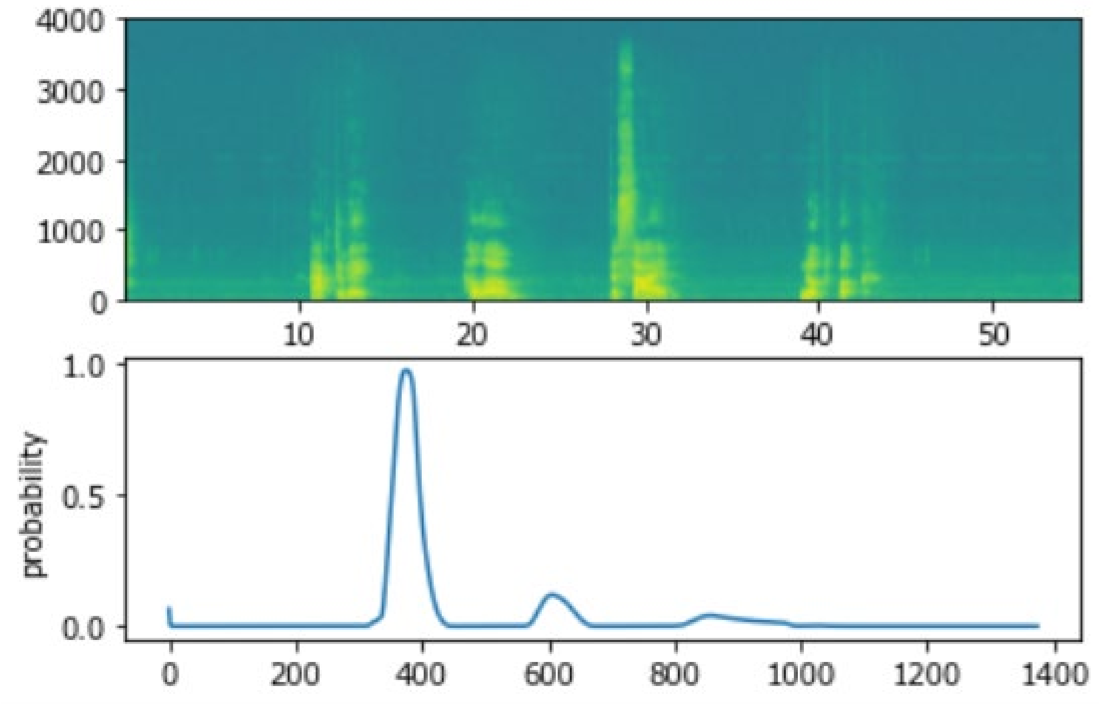


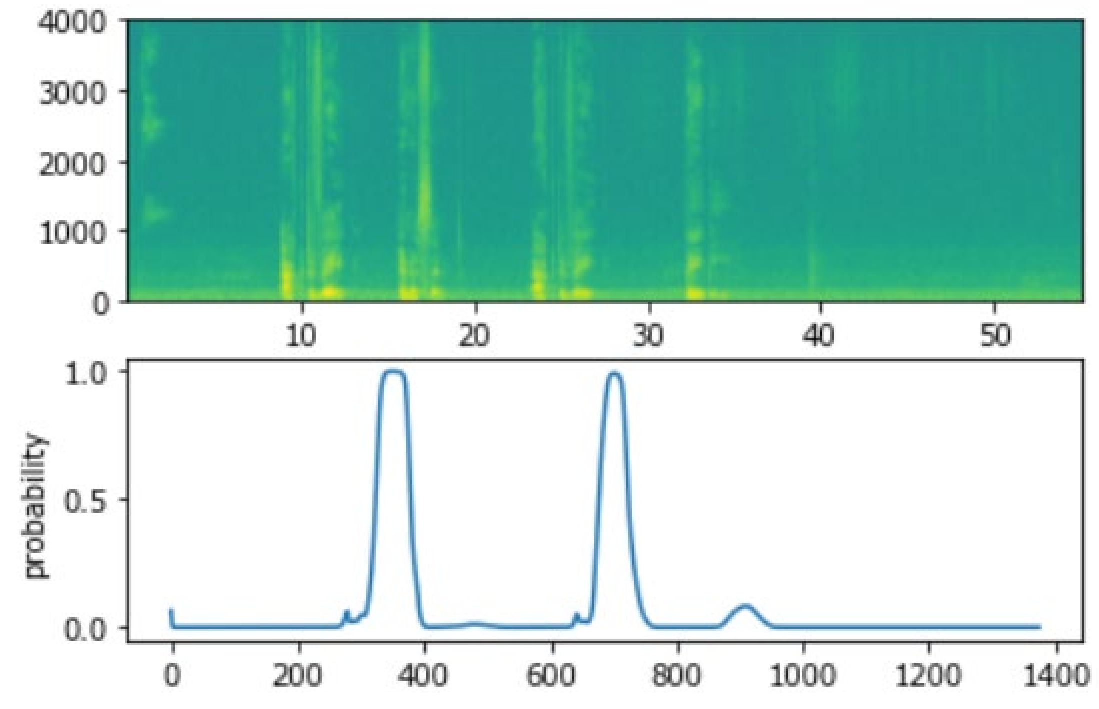
```
Epoch 1/1
26/26 [==============================] - 9s 344ms/step - loss: 0.0726 - accuracy: 0.9805
```

测试



```
25/25 [==============================] - 1s 35ms/step
Dev set accuracy =  0.9451636075973511
```





测试自己样例

结论分析与体会：

学会了如何构建语音识别项目

学会了合成和处理录音以创建训练

训练触发词检测模型并进行预测

就实验过程中遇到和出现的问题，你是如何解决和处理的，自拟 1－3 道问答题：

结果与答案不符合

发现数据包的版本不一样，和同学讨论后发现是正常现象