浙江大学

边缘计算开发实践

题目 Cough Detection

姓名 _____

学号 _____

专业：电子科学与技术
姓名：
学号：
日期：2023/10/23

浙江大学实验报告

课程名称：边缘计算开发实践 指导老师：皇甫江涛 成绩：
实验名称：Cough Detection

一、 实验目标

利用 Arduino 区分噪声和咳嗽声。

二、 实验任务

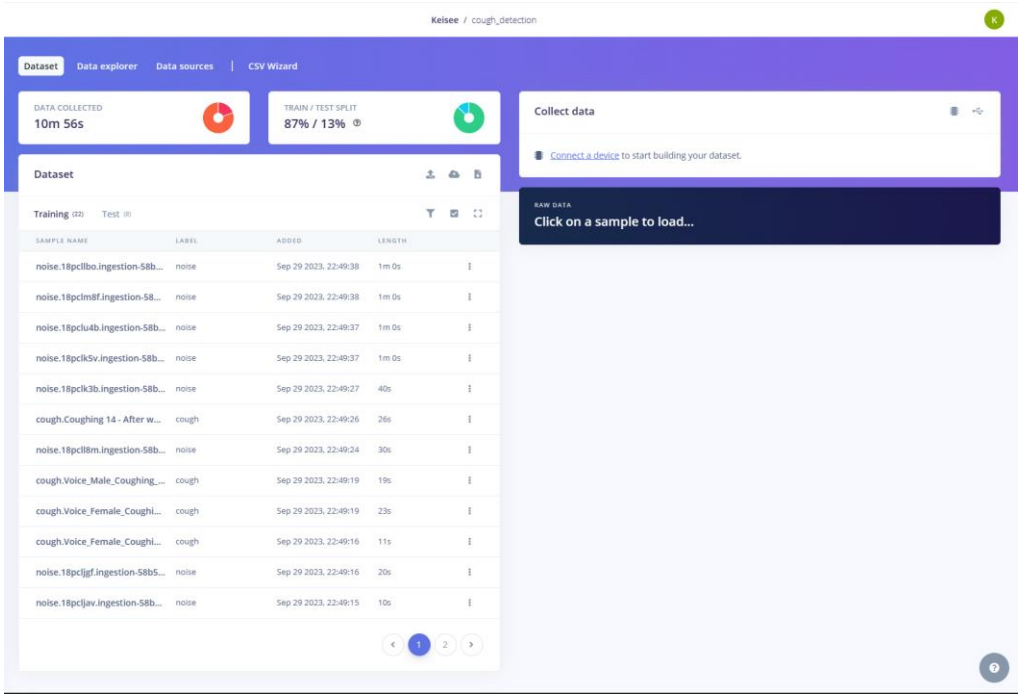
1. 获得数据集，并导入到 Edge Impulse 的工程中。
2. 在 Edge Impulse 中对数据集进行机器学习训练。
3. 将训练好的模型上传到 Arduino 上，进行模型测试。

三、 实验内容

1. 数据集的获取与导入

a) 从 https://cdn.edgeimpulse.com/datasets/cough.zip 下载数据集，解压。

b) 在 Edge Impulse 的 Dashboard 界面进行 upload existing data，导入结果在 Data acquisition 界面查看：



2. 在 Edge Impulse 网站进行模型训练

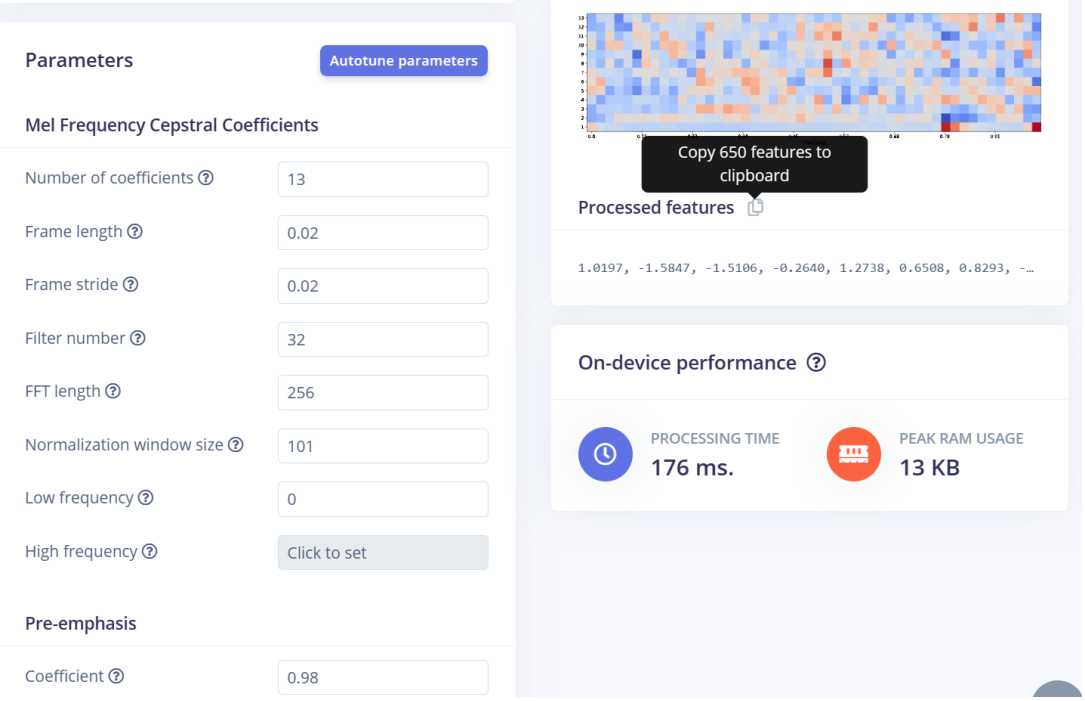
a) Create impulse

选择窗大小为 1000ms，每次增加的大小为 500ms；声音特征提取算法为 MFCC；learning block 为分类（classification）。

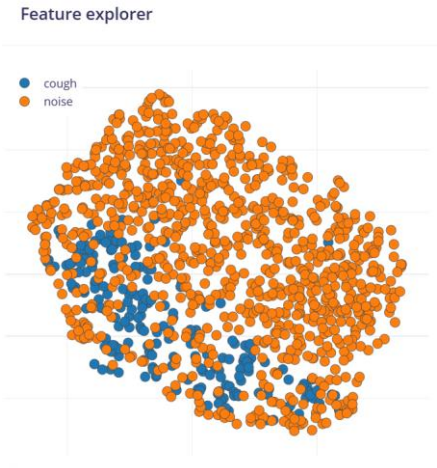
b) MFCC 特征提取

实验仅需对环境和人声作区分，因此 MFCC 的系数 13 个足矣。

具体参数如下：



由上图可知一条音频提取了 650 个特征。如果特征不够，可以增大 FFT length；本实验较为简单，650 个特征已经足够。特征提取结果：



可见经过 MFCC 处理后两种音频的特征有了明显的分簇，有利于后续的训练。

c) Classifier

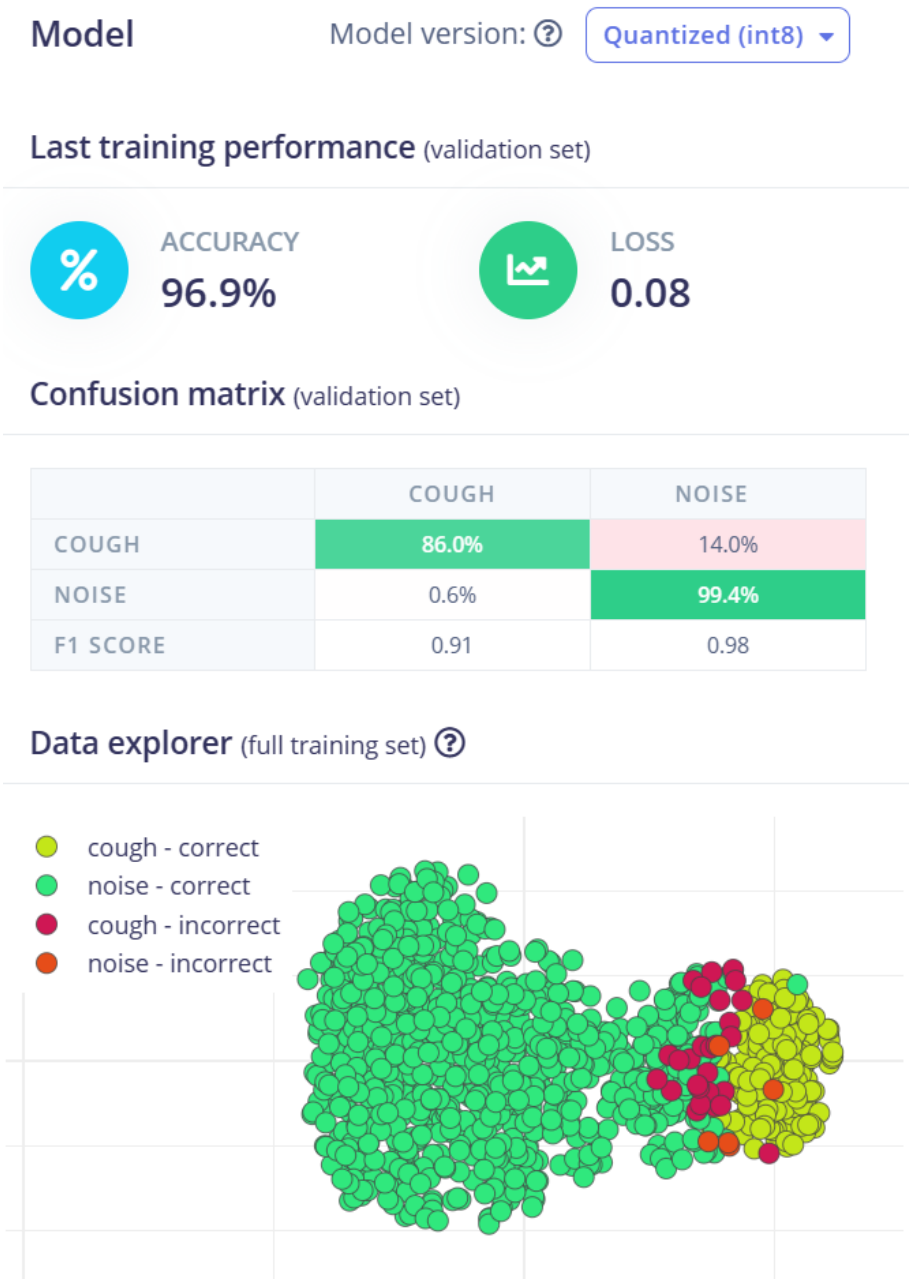
神经网络的构建代码如下：

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, InputLayer, Dropout, Flatten,
Reshape, BatchNormalization, Conv2D, MaxPooling2D, AveragePooling2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.constraints import MaxNorm
# model architecture
model = Sequential()
model.add(InputLayer(input_shape=(X_train.shape[1], ), name='x_input'))
model.add(Reshape((int(X_train.shape[1] / 13), 13, 1),
input_shape=(X_train.shape[1], )))
model.add(Conv2D(10, kernel_size=5, activation='relu', padding='same',
kernel_constraint=MaxNorm(3)))
model.add(AveragePooling2D(pool_size=2, padding='same'))
model.add(Conv2D(5, kernel_size=5, activation='relu', padding='same',
kernel_constraint=MaxNorm(3)))
model.add(AveragePooling2D(pool_size=2, padding='same'))
model.add(Flatten())
model.add(Dense(classes, activation='softmax', name='y_pred',
kernel_constraint=MaxNorm(3)))
# this controls the learning rate
opt = Adam(lr=0.005, beta_1=0.9, beta_2=0.999)
# train the neural network
model.compile(loss='categorical_crossentropy', optimizer=opt,
metrics=['accuracy'])
model.fit(X_train, Y_train, batch_size=32, epochs=9,
validation_data=(X_test, Y_test), verbose=2)
```

模型结构设计：

Layer (type)
Reshape
Conv2D
Pooling (Average)
Conv2D
Pooling (Average)
Flatten
Dense

训练结果：

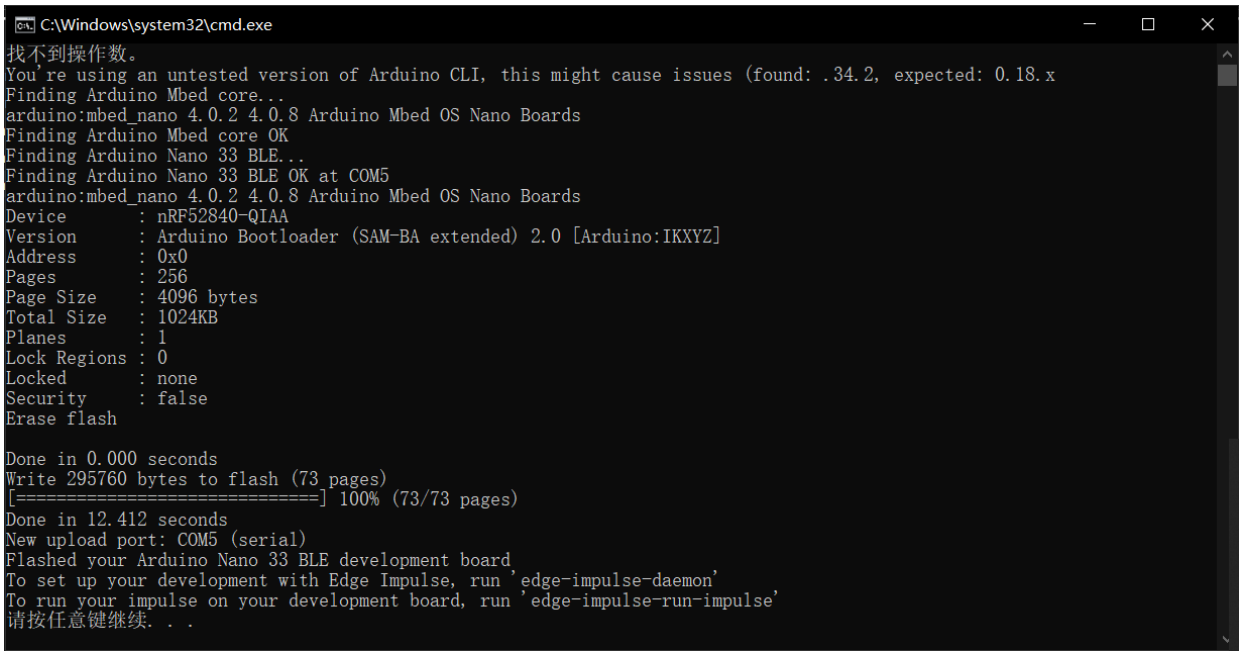


可见训练的准确度比较高，对于 Cough 信号，有少量概率被检测为 Noise 信号；但 Noise 信号不易被判断为 Cough 信号。

3. 在 Arduino Nano 33 BLE sense 上测试

a) 在 Edge Impulse 上部署模型，烧录到 Arduino Nano 33 BLE sense。

、



b) 打开命令行，输入 `edge-impulse-run-impulse` 命令，Arduino Nano 33 BLE sense 开始工作。测试结果如下，测试者发出声音时 cough 的概率提高至 0.87 以上；而不发声时，则是 noise 的概率接近于 1。综上，模型训练和测试的效果均好。