

IOT期末專題

題目:森林非法活動偵測系統

組員: 電機所碩二 楊璿臻、地理五 黃彥嘉

指導教授: 淡江大學電機系 廖書漢 教授



Introduction

- 目標:
 - 透過聲音辨識的方法，檢測森林中盜伐事件的發生
- 需要解決的難處:
 - 聲音辨識模型: 如何得到良好的辨識模型
 - 傳輸方法: 需要及時性而森林中訊號傳輸不良

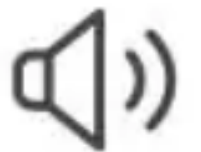


音訊辨識

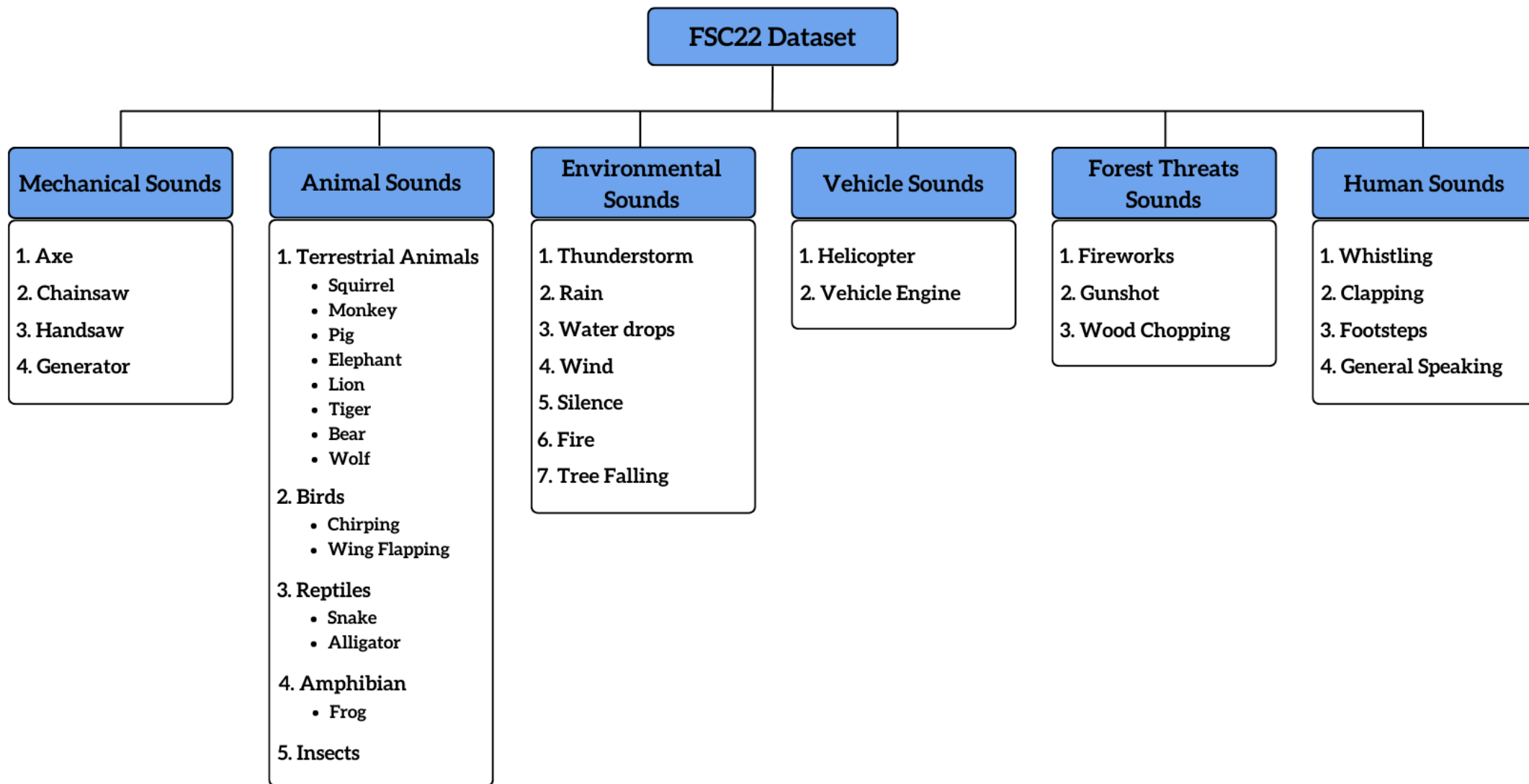


Dataset

- **Dataset : FSC22 - A dataset for forest sound classification**
- **Source: Bandara, M., Jayasundara, R., Ariyaratne, I., Meedeniya, D., & Perera, C. (n.d.). FSC22.**
- **Contents:**
 - 包含2025個label過的5秒音訊片段檔案(.wav)
 - 聲音類型分為六大類：
 - 機械聲、動物聲、環境聲、車輛聲、森林威脅聲和人類聲音
 - 目前這些大類劃分為27個子類別，每個類別75個音訊樣本



Dataset



Model

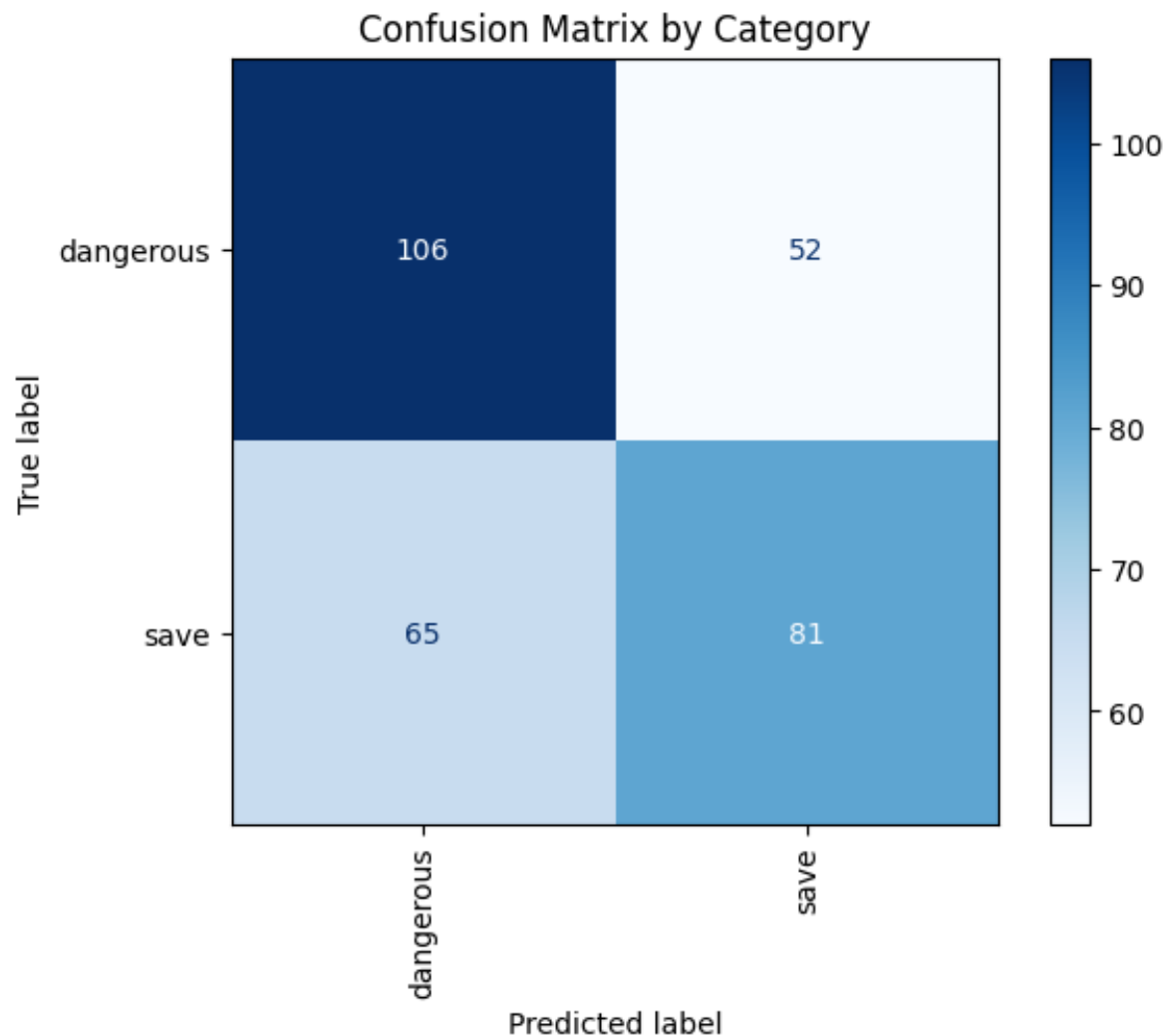
- **Model: YAMNet**
- **Source:** Plakal, M., & Ellis, D. (n.d.). *YAMNet: Pretrained deep net for audio event classification*.
(<https://github.com/tensorflow/models/blob/master/research/audioset/yamnet/>)
- **About Model:**
 - YAMNet是一個預訓練的深度神經網路，基於AudioSet-YouTube語料庫，採用Mobilenet_v1深度可分離卷積架構。

Model Fine-Tuning

- Data processing:
 - 將5秒音檔資料隨機擷取一秒
 - 轉成Mel頻譜並轉對數
 - 最後透過計算每個頻帶的平均值降維
- Split the dataset into: 80% training, 10% validation, 10% testing
- Hyperparameter:
 - Loss function: Cross Entropy Loss
 - Optimizer: Adam
 - Learning rate = 0.001
 - Epochs = 15
 - Batch size = 64

Performance

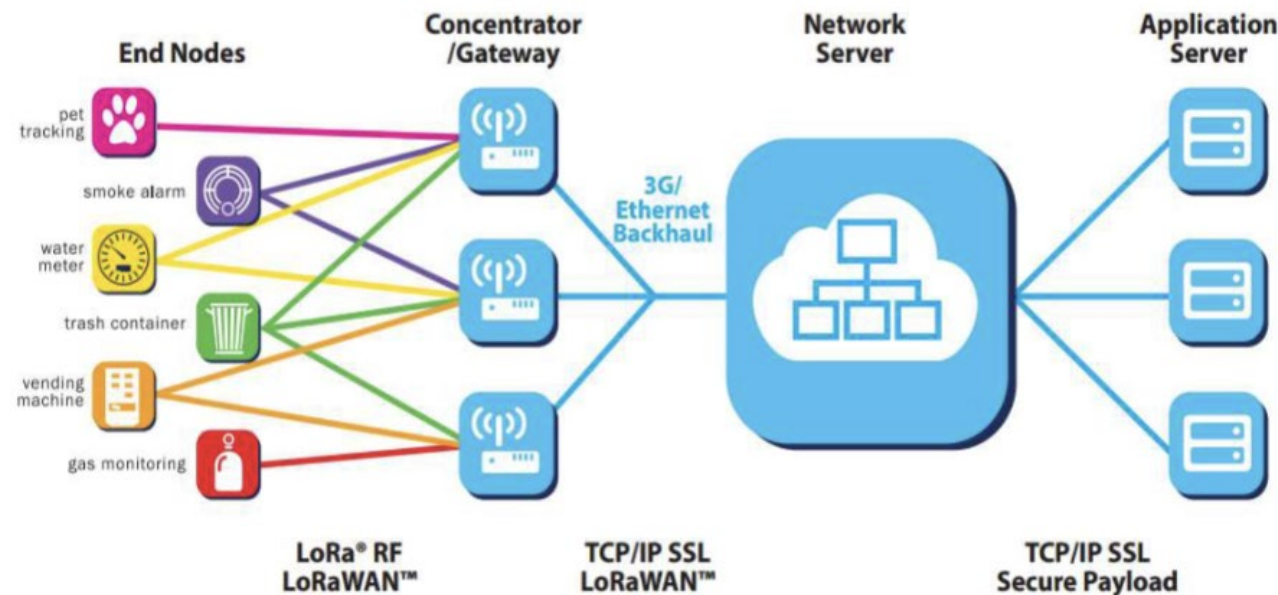
- 最後我們將27個類別分成2類:
 - 危險: 電鋸、斧頭等
 - 安全: 平靜無聲、動物聲、環境聲
- 模型表現如右圖:



訊號傳輸方法

Core Technology-LoRa

- What is LoRa(Long Range):
 - 是由 Semtech 開發的一種無線調變技術，專為長距離、低功耗、低數據速率應用設計。
 - 採用 Chirp Spread Spectrum (CSS) 調變技術，實現長距離通訊。
 - 適合物聯網 (IoT) 應用，支援低速率、高延遲容忍度的數據傳輸。



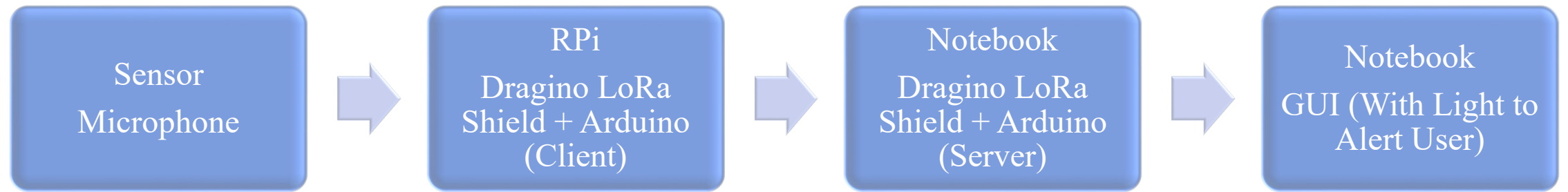
Core Technology-LoRa

- 主要特點
 - 低功耗：適用於電池驅動的設備，延長設備壽命。
 - 長距離：典型通訊範圍達數公里，郊區最遠可達 15 公里。
 - 抗干擾能力：利用 CSS 調變提高抗干擾性能，適合密集設備的環境。
 - 低成本：採用非授權頻段（如 433 MHz、868 MHz、915 MHz）。
- 應用範圍：
 - 智慧城市：路燈、垃圾管理、交通監控。
 - 農業與環境：土壤監測、水資源管理、天氣站。
 - 工業物聯網：資產追蹤、機器監控、能源管理。

Core Technology-LoRa

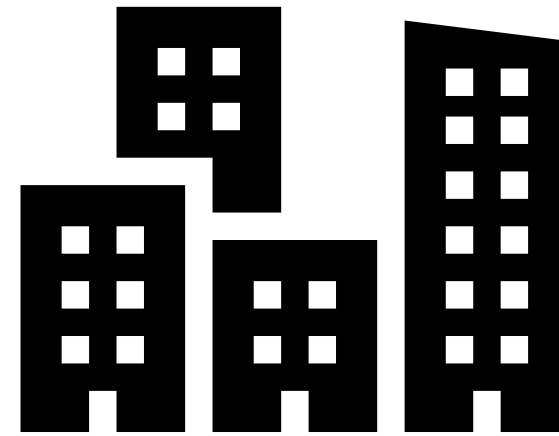
- 運作原理
 - 基礎結構：由 LoRa 節點、LoRa Gateway 和網路伺服器組成。
 - 上傳流程：資料從感測器經 LoRa 節點傳輸至 LoRa Gateway，再經由網路伺服器處理。
 - 雙向通訊：支援上行與下行通訊，滿足遠端設備控制需求。
- LoRa 的優勢
 - 長距離與穿透力強，適合在障礙物密集區域使用。
 - 可擴展性強，支援數千台設備連接同一網關。
- 專題中的應用：
 - 本專題研究使用 LoRa 作為核心通訊技術實現資料傳輸。實現低功耗長距離傳輸，有效解決森林中無線通訊的問題。

Overview



End Nodes (In Forest)

Office



Hardware

- Arduino*2
- Dragino LoRa Shield (含天線) *2
- USB傳輸線
- RaspberryPi 4B *1
- Microphone(Yeti)



Dragino LoRa Shield (含天線) *2



Microphone(Yeti)

Recording

- RPi透過外接麥克風，每隔30秒錄製接近6秒的音訊內容

```
def record_audio(duration, sample_rate=16000, channels=1, chunk_size=1024):  
    """  
    使用 pyaudio 錄製音訊並返回音訊數據。  
    :param duration: 錄音時間 (秒)  
    :param sample_rate: 音訊採樣率 (預設 16000 Hz)  
    :param channels: 聲道數量 (預設單聲道)  
    :param chunk_size: 單次讀取的音訊塊大小  
    :return: 錄製的音訊數據 (numpy 陣列)  
    """  
  
    p = pyaudio.PyAudio()  
  
    # 開啟音訊流  
    stream = p.open(format=pyaudio.paInt16,  
                    channels=channels,  
                    rate=sample_rate,  
                    input=True,  
                    frames_per_buffer=chunk_size)  
  
    print(f"開始錄製 {duration} 秒音訊...")  
  
    frames = []  
    # 錄製  
    for i in range(0, int(sample_rate / chunk_size * duration)):  
        data = stream.read(chunk_size)  
        frames.append(np.frombuffer(data, dtype=np.int16))  
  
    # 停止  
    stream.stop_stream()  
    stream.close()  
    p.terminate()  
    print("錄製完成。")  
    return np.hstack(frames)
```

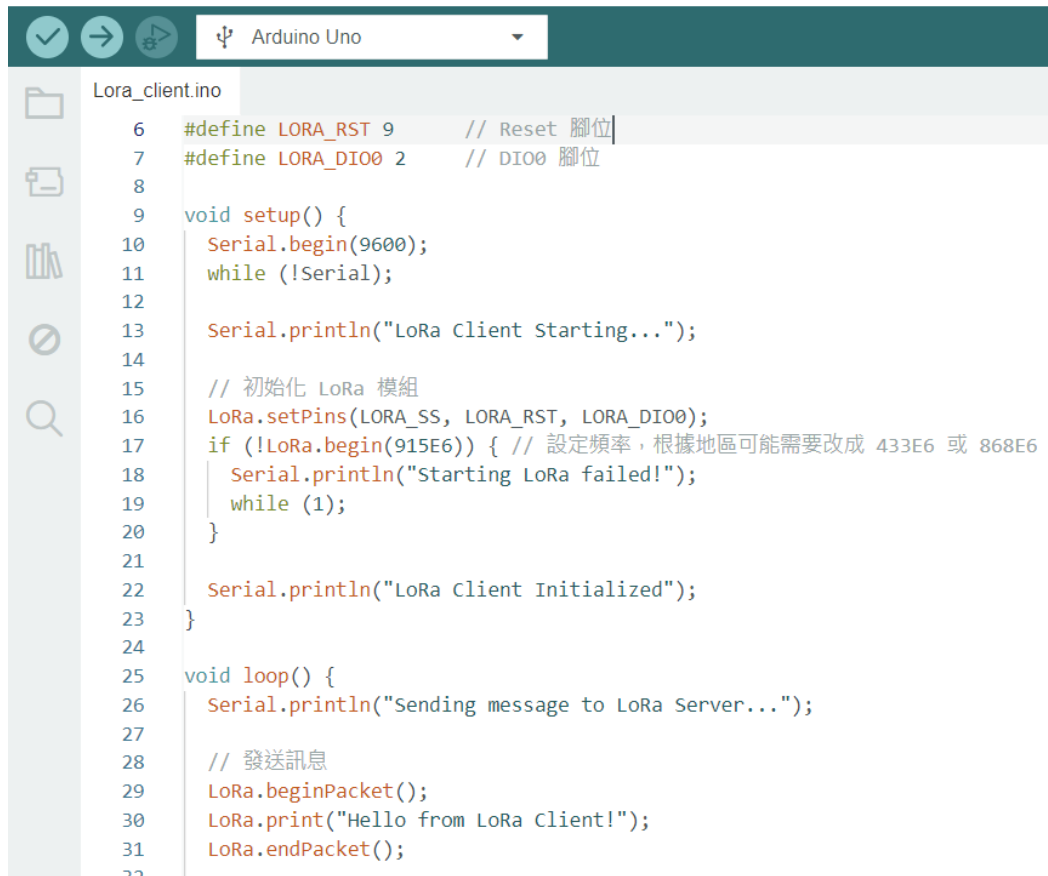
Data Processing

- 將5秒多的音訊，去掉後面多的部分截成5秒後，隨機從中間擷取1秒
- 將音訊在RPi上轉成Mel頻譜並轉對數，計算每個頻帶的平均值降維
- 結果通常為一串負的2位整數部分小數：ex: -xx.xx...

```
[ -14.25169  -12.988901 -14.952024 -17.644434 -19.986778 -21.40833  
  -20.098137 -20.222565 -20.131214 -18.62408  -19.34272  -21.171993  
  -23.923738 -25.862526 -27.685764 -28.728048 -29.442167 -28.234364  
  -28.166548 -28.669239 -29.239353 -28.974365 -28.22696  -28.718573  
  -30.134346 -29.71614  -32.67496  -33.057663 -34.280758 -35.10585  
  -35.898636 -35.583923 -36.77786  -37.810955 -37.357704 -38.157524  
  -39.943954 -41.921474 -42.90444  -42.112686 -42.64201  -44.12635  
  -45.007866 -44.50277  -44.634186 -45.570477 -45.651684 -45.736946  
  -45.51947  -45.600586 -46.571472 -47.74471  -48.67644  -49.40889  
  -49.520893 -49.650055 -49.166306 -50.04866  -49.98298  -50.106842  
  -50.601593 -51.359184 -55.089798 -66.98772 ]
```


LoRa Transmission

- Client (In Forest)-Send Data



```
Lora_client.ino
6  #define LORA_RST 9      // Reset 腳位
7  #define LORA_DIO0 2    // DIO0 腳位
8
9  void setup() {
10     Serial.begin(9600);
11     while (!Serial);
12
13     Serial.println("LoRa Client Starting...");
14
15     // 初始化 LoRa 模組
16     LoRa.setPins(LORA_SS, LORA_RST, LORA_DIO0);
17     if (!LoRa.begin(915E6)) { // 設定頻率，根據地區可能需要改成 433E6 或 868E6
18         Serial.println("Starting LoRa failed!");
19         while (1);
20     }
21
22     Serial.println("LoRa Client Initialized");
23 }
24
25 void loop() {
26     Serial.println("Sending message to LoRa Server...");
27
28     // 發送訊息
29     LoRa.beginPacket();
30     LoRa.print("Hello from LoRa Client!");
31     LoRa.endPacket();
32 }
```

- Server (In Office)-Receive Data



```
Lora_server.ino
1  #include <SPI.h>
2  #include <LoRa.h>
3
4  void setup() {
5      // 啟動串列通訊
6      Serial.begin(9600);
7      while (!Serial);
8
9      // 初始化 LoRa 模組
10     if (!LoRa.begin(915E6)) { // 使用對應的頻率 (例如 868 MHz 或 915 MHz，視地區而定)
11         Serial.println("LoRa初始化失敗");
12         while (1);
13     }
14     // Serial.println("LoRa已初始化，開始接收訊息...");
15 }
16
17 void loop() {
18     // 檢查是否有來自 LoRa 發送端的訊息
19     int packetSize = LoRa.parsePacket();
20     if (packetSize) {
21         String received = "";
22         while (LoRa.available()) {
23             received += (char)LoRa.read();
24         }
25         // 將接收到的訊息發送到串列監視器
26         Serial.print("接收到的資料：");
27         Serial.println(received);
28     }
29 }
30 }
```

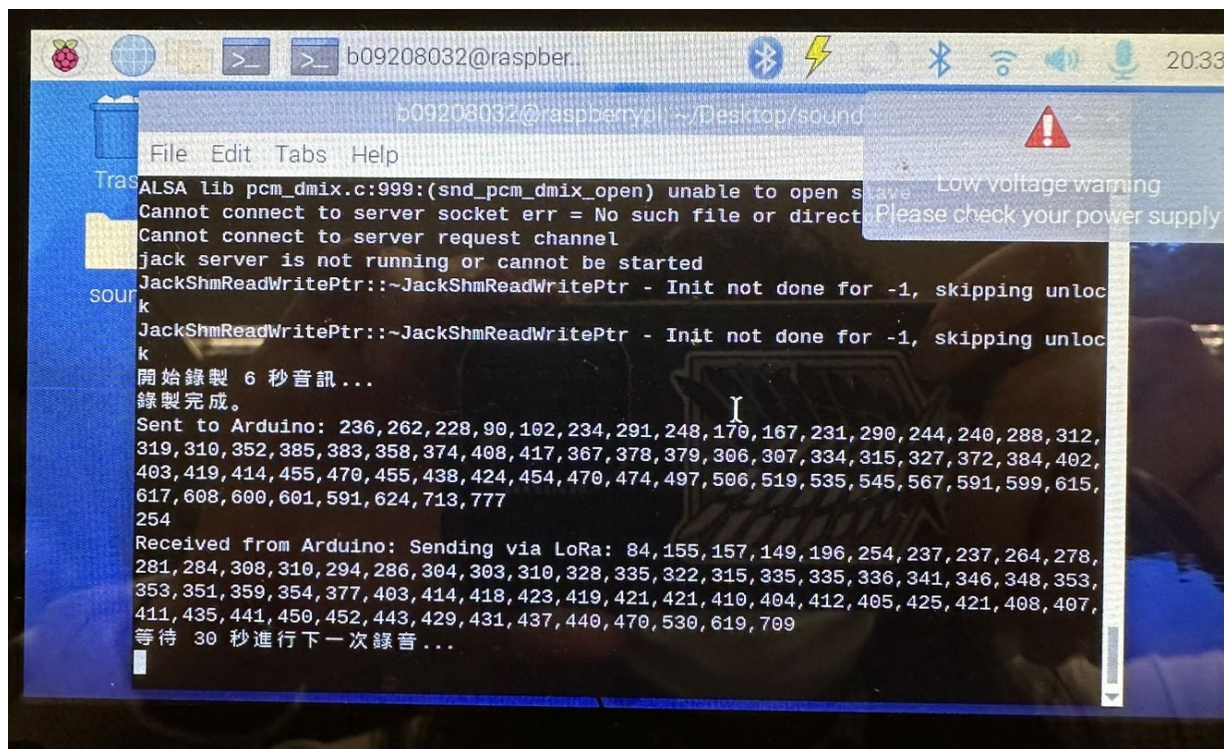
LoRa Transmission

- 由於LoRa每次只能傳255個字元: 把所有數 $\times (-10)$ 並取整數部分
- 通常會變成一個三位數的正整數

```
Sent to Arduino: 236,262,228,90,102,234,291,248,170,167,231,290,244,240,288,312,  
319,310,352,385,383,358,374,408,417,367,378,379,306,307,334,315,327,372,384,402,  
403,419,414,455,470,455,438,424,454,470,474,497,506,519,535,545,567,591,599,615,  
617,608,600,601,591,624,713,777
```

LoRa Transmission

- 將數列傳給連結在RPi上接有LoRa的Ardiuno
- Ardiuno在接收到資料後回傳收到的資料，以便確認



The screenshot shows a terminal window on a Raspberry Pi. The window title is "b09208032@raspberrypi". The terminal output includes the following text:

```
ALSA lib pcm_dmix.c:999:(snd_pcm_dmix_open) unable to open slave
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
開始錄製 6 秒音訊...
錄製完成。
Sent to Arduino: 236,262,228,90,102,234,291,248,170,167,231,290,244,240,288,312,
319,310,352,385,383,358,374,408,417,367,378,379,306,307,334,315,327,372,384,402,
403,419,414,455,470,455,438,424,454,470,474,497,506,519,535,545,567,591,599,615,
617,608,600,601,591,624,713,777
254
Received from Arduino: Sending via LoRa: 84,155,157,149,196,254,237,237,264,278,
281,284,308,310,294,286,304,303,310,328,335,322,315,335,335,336,341,346,348,353,
353,351,359,354,377,403,414,418,423,419,421,421,410,404,412,405,425,421,408,407,
411,435,441,450,452,443,429,431,437,440,470,530,619,709
等待 30 秒進行下一次錄音...
```

A "Low voltage warning" dialog box is overlaid on the terminal, stating: "Low voltage warning. Please check your power supply."

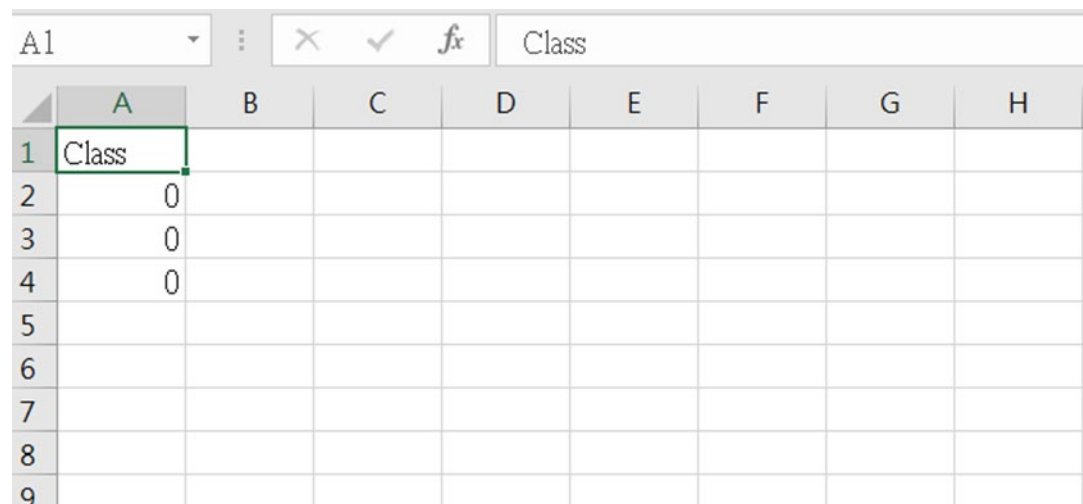
Sound Detection

- 透過LoRa傳給中央電腦後，
再把所有數 $\times (-1/10)$ 做數據還原
- 得到精確度到小數點後1位的訊號值
- 將數列轉成符合模型的數據格式進行辨識

```
C:\> Users > user > Desktop > 物聯網期末專題 > model.py > ...
1  from torch_vggish_yamnet import yamnet
2  import multiprocessing
3  import torch.nn as nn
4  import numpy as np
5  import torch
6  import pandas as pd
7  #from openpyxl import Workbook, load_workbook
8  import csv
9  # from Lora_server import data_array
10 import time
11 import importlib
12 import warnings
13 warnings.filterwarnings('ignore')
14
15 while True:
16     import Lora_server
17     importlib.reload(Lora_server)
18     # from Lora_server import data_array
19
20     data_list = Lora_server.data_array
21
22     while len(data_list) < 64:
23         data_list.append(0)
24     new_list = []
25     for i in data_list:
26         n = (i * -1) / 10
27         new_list.append(n)
28     # print(new_list)
29     # 轉換為 NumPy 陣列
30     mel_mean = np.array(new_list)
31     mel_tensor = torch.tensor(mel_mean, dtype=torch.float32)
32     # print(f"mel_tensor shape before unsqueeze: {mel_tensor.shape}")
33     # print(mel_tensor)
34
35     class YAMNetClassifier(nn.Module):
36         def __init__(self, num_classes=27):
37             super(YAMNetClassifier, self).__init__()
38             self.yamnet = yamnet.YAMNet(yamnet.Constants.FEATURE_DIM, yamnet.Constants.NUM_CLASSES)
```

Detection Result

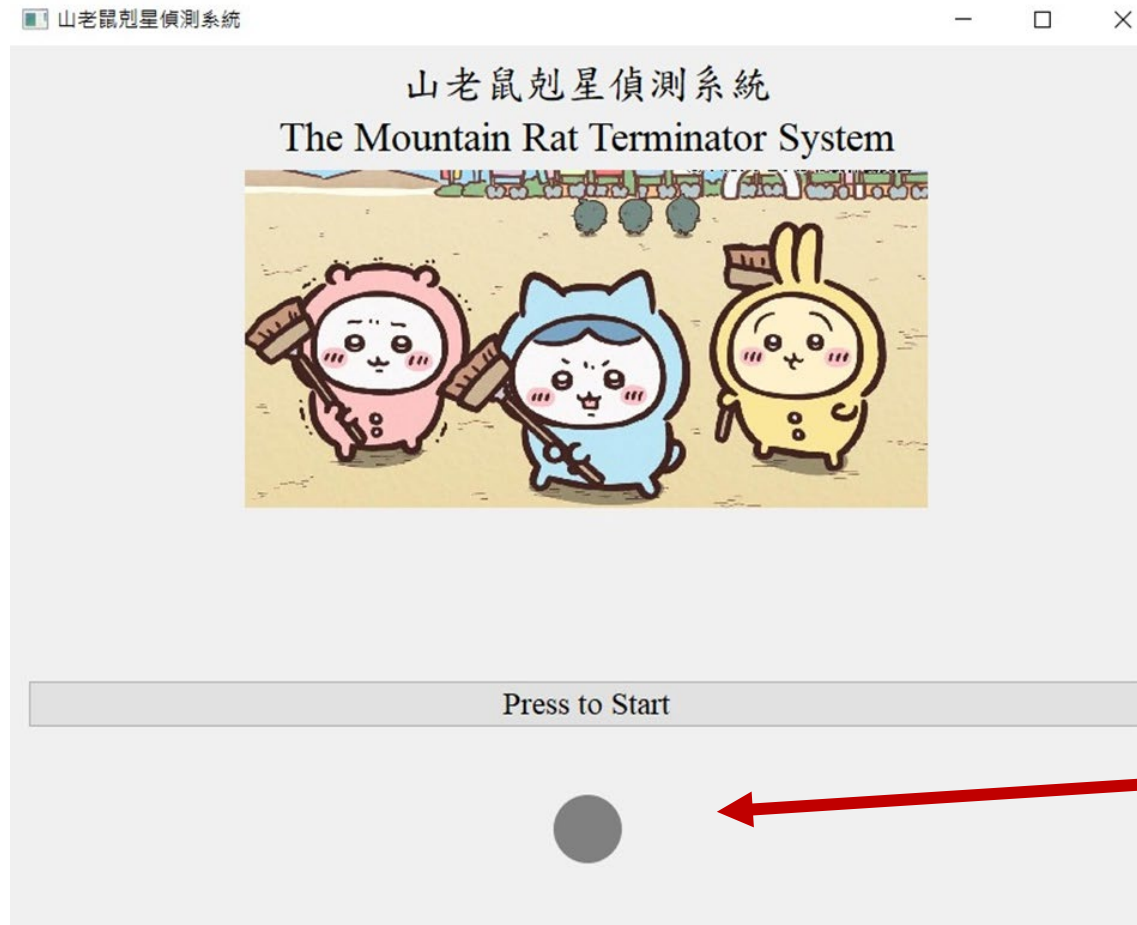
- 將辨識到的值存到一個csv檔中
- 然後再同步讀取csv資料顯示到UI介面顯示
- 一直持續執行以上內容直到中斷程式



	A	B	C	D	E	F	G	H
1	Class							
2	0							
3	0							
4	0							
5								
6								
7								
8								
9								

User Interface (UI)

- Graphical User Interface on Notebook

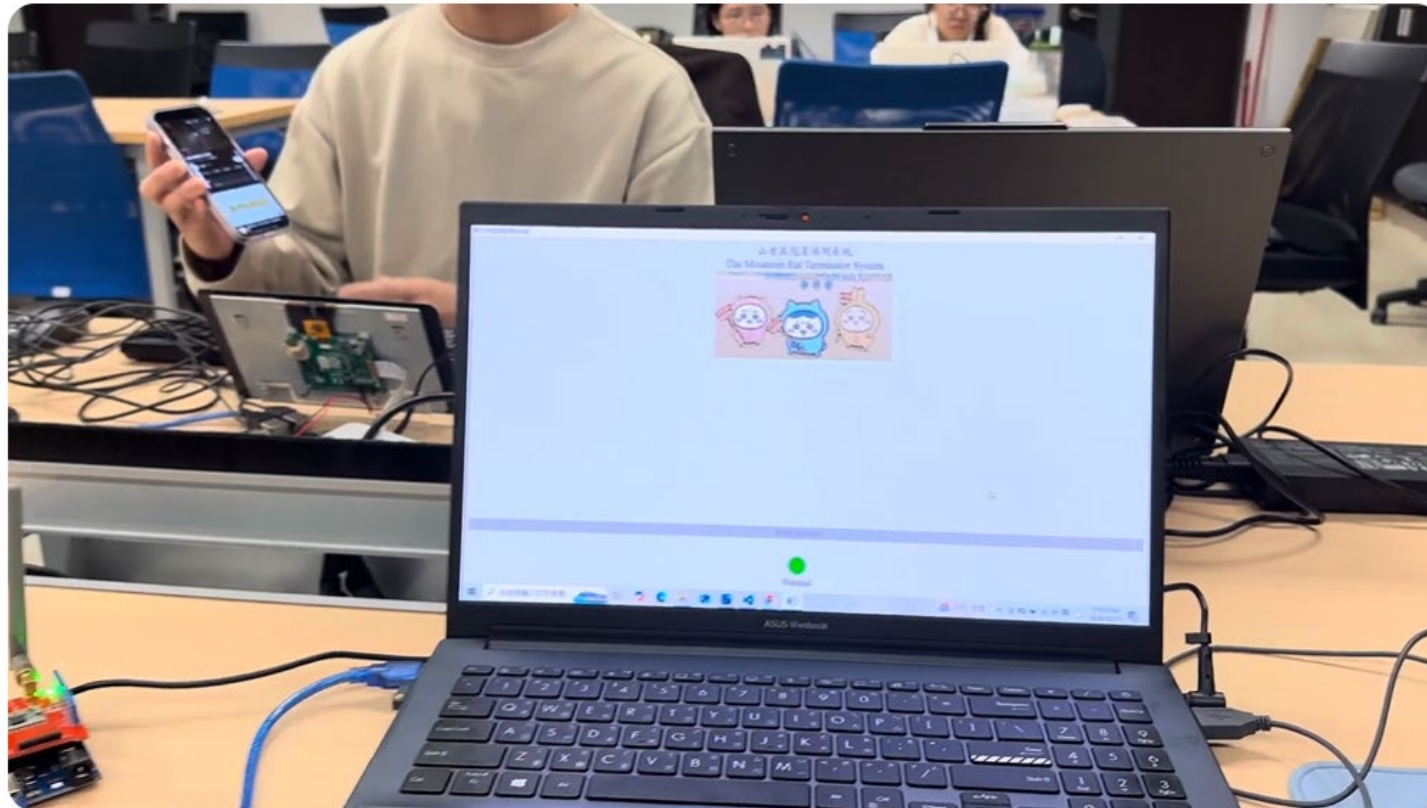


Traffic Light

- Gray-Waiting...
- Green-Normal
- Red-Warning

Demo

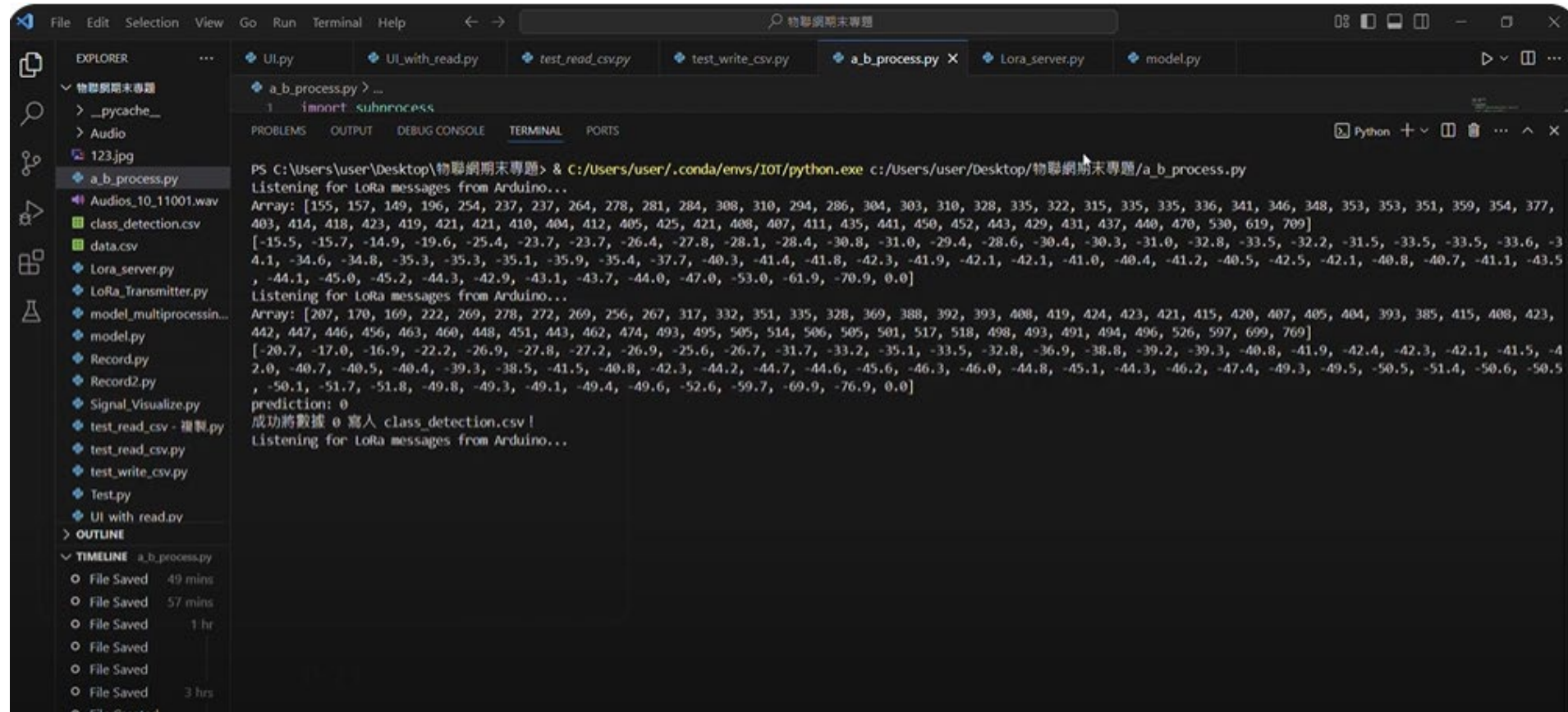
- Video: <https://youtu.be/3iSb10vEZIE?si=1nAOwTmZadfgtm7j>



2024 Fall IOT期末專題-山老鼠剋星系統 (作者:楊璿臻、黃彥嘉)

Demo

- Video: https://youtu.be/D_B_GAvpTaQ?si=3gozo5IuGtg2iCVC



```
PS C:\Users\user\Desktop\物聯網期末專題> & C:/Users/user/.conda/envs/IOT/python.exe c:/Users/user/Desktop/物聯網期末專題/a_b_process.py
Listening for LoRa messages from Arduino...
Array: [155, 157, 149, 196, 254, 237, 237, 264, 278, 281, 284, 308, 310, 294, 286, 304, 303, 310, 328, 335, 322, 315, 335, 335, 336, 341, 346, 348, 353, 353, 351, 359, 354, 377,
403, 414, 418, 423, 419, 421, 421, 410, 404, 412, 405, 425, 421, 408, 407, 411, 435, 441, 450, 452, 443, 429, 431, 437, 440, 470, 530, 619, 709]
[-15.5, -15.7, -14.9, -19.6, -25.4, -23.7, -23.7, -26.4, -27.8, -28.1, -28.4, -30.8, -31.0, -29.4, -28.6, -30.4, -30.3, -31.0, -32.8, -33.5, -32.2, -31.5, -33.5, -33.5, -33.6, -3
4.1, -34.6, -34.8, -35.3, -35.3, -35.1, -35.9, -35.4, -37.7, -40.3, -41.4, -41.8, -42.3, -41.9, -42.1, -42.1, -41.0, -40.4, -41.2, -40.5, -42.5, -42.1, -40.8, -40.7, -41.1, -43.5
, -44.1, -45.0, -45.2, -44.3, -42.9, -43.1, -43.7, -44.0, -47.0, -53.0, -61.9, -70.9, 0.0]
Listening for LoRa messages from Arduino...
Array: [207, 170, 169, 222, 269, 278, 272, 269, 256, 267, 317, 332, 351, 335, 328, 369, 388, 392, 393, 408, 419, 424, 423, 421, 415, 420, 407, 405, 404, 393, 385, 415, 408, 423,
442, 447, 446, 456, 463, 460, 448, 451, 443, 462, 474, 493, 495, 505, 514, 506, 505, 501, 517, 518, 498, 493, 491, 494, 496, 526, 597, 699, 769]
[-20.7, -17.0, -16.9, -22.2, -26.9, -27.8, -27.2, -26.9, -25.6, -26.7, -31.7, -33.2, -35.1, -33.5, -32.8, -36.9, -38.8, -39.2, -39.3, -40.8, -41.9, -42.4, -42.3, -42.1, -41.5, -4
2.0, -40.7, -40.5, -40.4, -39.3, -38.5, -41.5, -40.8, -42.3, -44.2, -44.7, -44.6, -45.6, -46.3, -46.0, -44.8, -45.1, -44.3, -46.2, -47.4, -49.3, -49.5, -50.5, -51.4, -50.6, -50.5
, -50.1, -51.7, -51.8, -49.8, -49.3, -49.1, -49.4, -49.6, -52.6, -59.7, -69.9, -76.9, 0.0]
prediction: 0
成功將數據 0 寫入 class_detection.csv!
Listening for LoRa messages from Arduino...
```


Conclusion

- Future Work
 - 模型優化
 - 結合其他方法 (如遙測技術等...)
 - 設定停止鍵
 - 實際實驗

Reference

- Plakal, M., & Ellis, D. (n.d.). YAMNet: Pretrained deep net for audio event classification.
Retrieved from:
<https://github.com/tensorflow/models/blob/master/research/audioset/yamnet/>
- Bandara, M., Jayasundara, R., Ariyaratne, I., Meedeniya, D., & Perera, C. (n.d.). **FSC22**.
- V. Singh, K. C. Ray and S. Tripathy, "An Efficient Method and Hardware System for Monitoring of Illegal Logging Events in Forest," in IEEE Systems Journal, vol. 18, no. 1, pp. 40-50, March 2024, doi: 10.1109/JSYST.2023.3333677.
- LoRa上課講義
- 廖教授提供的資料
- ChatGPT

Thanks for your attention