

2024仰望盃全國科學 HomeRun 實作大賽

決賽作品說明書

隊伍名稱： 生活駭客

作品名稱： 陀螺儀也可以監聽出個資？

科學概念1： 心率變異度 (HRV)

$$\text{RMSSD} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} (\text{RR}_{i+1} - \text{RR}_i)^2}$$

心率變異性是量測連續心跳速率變化程度的方法，反映自主神經系統對心臟功能的調控能力。RMSSD 是評估 HRV 常用的指標，其對副交感神經活動高度敏感，廣泛應用於監測短期數據變化。RMSSD 較高時，表示自主神經系統中的副交感神經活動相對活躍，身體處於放鬆和平衡的狀態。

本實驗利用測量出的心率運算出 RMSSD 數值作為壓力程度的指標。

科學概念2： 深度學習與點積注意力機制

$$\text{點積注意力機制公式： } a_i = \frac{q_i \cdot k_i}{d_k}$$

點積注意力機制會計算查詢(q)和鍵(k)的點積，並將其除以鍵(k)的維度，得到注意力權重(a)。在生成輸出時，模型會根據注意力權重對輸入序列的每個元素進行加權，得到最終的輸出。

本實驗使用注意力機制讓模型能夠專注於數據重要的部分，使模型學習輸入序列的長期依賴關係，如忽視在靜止狀態下陀螺儀受到環境噪聲的影響。

複賽作品說明書內文

一、發想動機：

近年來，大數據的應用已經廣泛影響了我們的日常生活，從路徑規劃、交通運輸優化，到自動駕駛、醫療診斷和金融風險管理等領域，無所不在。作為學校資訊志工隊的成員，我們對深度學習的實際應用深感興趣。

近期，諸如「iPhone 14 車禍偵測拯救性命」的報導引發了我們的好奇心。因此，我們決定將大數據深度學習和陀螺儀作為研究主題，探索其中的技術和創新，並開發出不同的系統。陀螺儀作為一種常見的感測器，廣泛應用於各個領域，包括設備平衡、運動員動作分析以及行動裝置上的使用者互動。

透過 200 份網路問卷調查發現，81% 的人會同意網頁讀取陀螺儀資訊，而儘管有部分人不同意，許多網站及應用程式對陀螺儀的存取無需經過使用者同意。如同我們與他人比個手勢就能判斷出其意義或模仿行為，陀螺儀數據是一個連續多維度的數據，我們推測能夠透過其變化推測出很多使用者的行為。這引發了我們對陀螺儀數據應用和安全性的關注。因此，我們希望透過科學方法，探索如何從陀螺儀數據中推斷出有用的信息，並洞察相關的安全疑慮。

現代的智慧手環可以計算許多健康數據，如 Garmin 手環可以計算出心律、脈搏血氧、壓力甚至是剩餘的體力。在高中生繁忙的日常生活中常常會感到時間和精力好像怎麼樣都不夠用，並從家人或自我期許中獲得莫名的壓力。我們希望能透過手環主動追蹤並分析出高中生一天都在做什麼，來解釋心率變異性（HRV）變化的原因(如壓力的來源)。

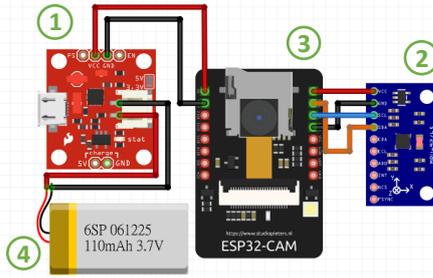
二、作品創意性：

- I. 跨領域的科學探索：整合陀螺儀、血氧心律感測器、人工智慧(深度學習)、3D 列印、硬體電路、網際網路等不同科學與科技領域。
- II. 手環耗費不到530元其動作判斷準確率即可達到96.57%，且使用者可根據自己的生活習慣，使用手邊的任意裝置進行客製化模型訓練。手環可以獨立運作，連上網路後再進行同步。
- III. 使用樹莓派運行模型：本實驗從數據採集、模型訓練到實際應用，不仰賴昂貴的 CPU 及 GPU 資源，只需使用樹莓派或手機等低階設備即可完成。
- IV. 創造性的數據分析：我們不僅監測出使用者整天的壓力變化，我們更能追蹤出使用者壓力的來源，提供使用者個人化的健康建議。
- V. 以網站視覺化呈現：從數據的蒐集到實際使用者畫面都使用網頁 GUI 呈現，讓使用者更容易操作。

三、 硬體及電路架構圖：

模型訓練手環

1. MT3608 升壓模組
2. MPU9250 陀螺儀
3. ESP32-CAM
4. 鋰電池



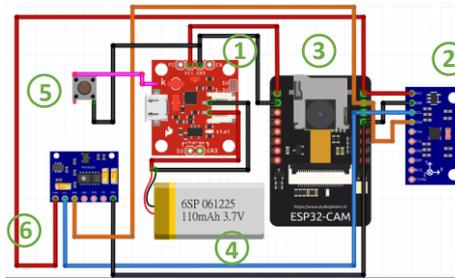
圖一：手環電路圖



圖二：手環內部實拍

日常追蹤手環

1. CKCS 升壓模組
2. MPU9250 陀螺儀
3. ESP32-CAM
4. 鋰電池
5. 按鈕
6. MAX30102 血氧心律脈搏傳感器



圖三：手環電路圖

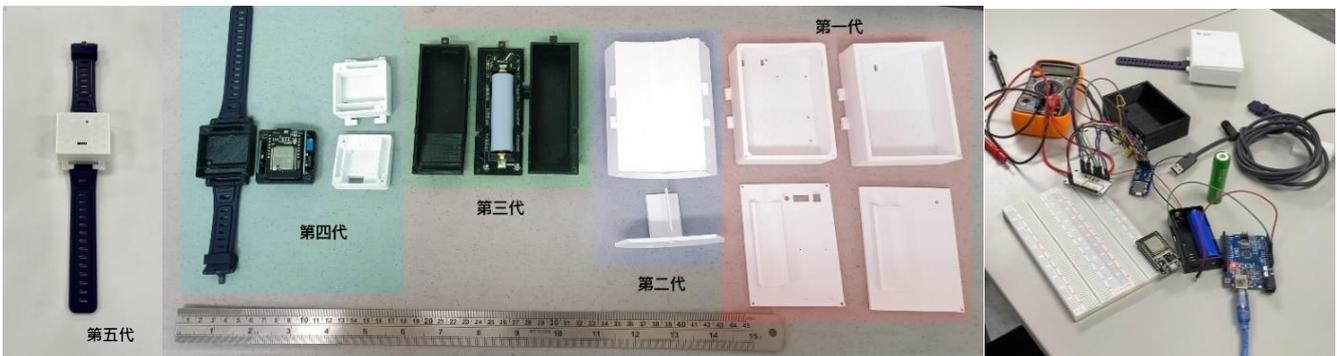


圖四：手環模型圖

我們使用 MPU9250 陀螺儀採集手部運動的陀螺儀數據，並傳送至 ESP-32 cam 連接的 SD 卡模組進行數據保存。

在元件及程式完成測試後我們使用洞洞板焊接來取代杜邦線減少體積與增加穩定性。我們使用 Shapr3D 進行 3D 模型繪製，並使用 ABS 材質進行 3D 列印，保持手環的耐用性。過程歷經了 4 代的疊代。每一代透過移除了不必要的模塊，以及重新安排布局大幅減少體積。

這個版本的手環加入了 MAX30102 血氧心律脈搏傳感器以及更換電池升壓組。除了可以使用按鈕開關以及使用 Type-C 直接充電，更可以透過 led 及時瀏覽目前手環剩餘電量。



圖五：右到左分別為第一代至第五代手環

圖六：硬體測試

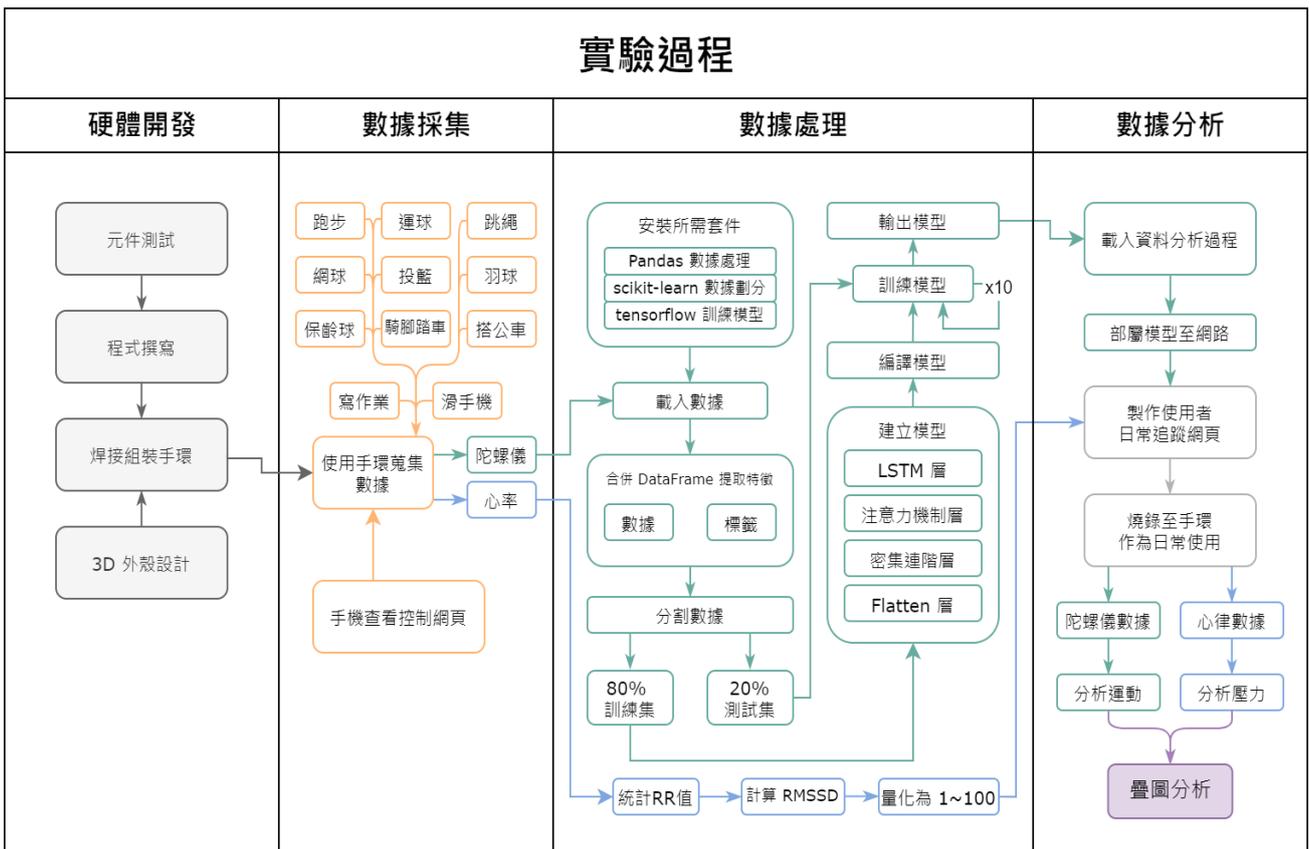
四、 作品成果說明：

在本次研究中，我們成功開發了一個運動識別模型，此模型能根據使用者手部蒐集到的陀螺儀數據識別當前的運動。此外，我們透過網頁界面視覺化模型的預測過程，展示了模型在整個運動期間不同時間點的預測機率。

成功開發運動識別模型後，我們進一步製作一款日常配戴型智慧手環。該手環能夠定時且自動採集使用者的陀螺儀數據和心律數據。連接到手機或電腦等裝置時，可以看到使用者全天的運動模式與壓力變化。透過進一步的疊圖分析，我們能夠識別出壓力的來源，從而協助使用者更有效地了解個人日常行動與壓力之間的關係。

這些成果為未來的運動監測和健康管理提供了一些新的思路和方法。我們期待這些成果能夠為人們的生活帶來一些便利和幫助。

實驗過程可以簡單分為**活動追蹤**及**壓力分析**，裡面包含**數據採集**，**數據處理**，與**數據分析**。



(一) 活動追蹤

1. 數據採集

手環電源開啟後會自動開啟 Wi-Fi 無線基地台，使用手機或樹莓派等裝置連上後即可透過 localhost 輸入本地 IP 瀏覽 ESP-32 網頁。進入網頁後可以看到每個元件的啟用狀況，下方輸入框可以控制數據的蒐集頻率以及添加數據的標籤。

我們分別配戴手環進行運球、網球揮拍、跑步、跳繩、投籃、羽球揮拍、保齡球、騎腳踏車、搭公車、寫作業、滑手機等各項活動，並從手機控制資料蒐集的開始與結束。蒐集到的數據會自動根據標籤(上述活動)分類存放在手錶的 SD 卡中。每項動作蒐集各 30 筆數據，共 330 筆數據。



圖七：陀螺儀數據蒐集

2. 數據處理

完成數據採集後我們將手環數據放入樹莓派並進行訓練。

- I. **安裝所需套件**：首先，安裝必要的套件，包括 pandas 用於數據處理，tensorflow 用於建立和訓練深度學習模型，scikit-learn 用於數據劃分。
- II. **載入資料**：定義運動資料夾的名稱，使用 os 套件來遍歷資料夾中的 CSV 檔案，並將它們讀取為 DataFrame。這些 CSV 檔案包含時間、陀螺儀和加速度計的數據。
- III. **資料前處理**：將所有的運動數據合併成一個 DataFrame，並將運動類別的名稱轉換為數值。
- IV. **特徵提取和標籤準備**：從合併後的 DataFrame 中提取特徵（時間、陀螺儀和加速度計數據）和標籤（運動類別）。
- V. **劃分訓練集和測試集**：使用 train_test_split 函數將數據劃分為訓練集和測試集。將測試集的比例設置為 20%，保留足夠的數據來評估模型的性能。
- VI. **建立模型**：使用 LSTM、注意力機制和密集連接層來建立運動分類模型。LSTM 層用於處理時間序列數據，注意力機制可以幫助模型更好地理解數據的關鍵部分，而密集連接層則用於分類。
- VII. **編譯模型**：指定優化器、損失函數和評估指標。在這裡，使用了 Adam 優化器和稀疏分類交叉熵作為損失函數。
- VIII. **訓練模型**：將訓練集的特徵和標籤傳遞給模型，並指定訓練輪數、批次大小和測試集。設置批次大小為 32 個，以提高訓練效率和充分利用 GPU 資源。
- IX. **最後**，將訓練好的模型儲存到樹莓派中，以便後續使用。我們使用 .h5 格式來儲存模型。雖然這不是最新的格式但其對於不同的裝置以及後續的格式轉換的兼容性較好。

圖八：數據蒐集控制

在模型訓練過程中曾遇到了輸入形狀不匹配的問題，因此我們將 Flatten 層添加到模型的最後一層之前，這樣可以確保輸出形狀與標籤形狀相匹配。

模型訓練程式碼：

```
import os
import pandas as pd
import tensorflow as tf
from tensorflow.keras.layers import Input, LSTM, Dense, Attention, Flatten
from tensorflow.keras.models import Model
from sklearn.model_selection import train_test_split
# 載入和處理資料
data_folders = ['badminton', 'biking', 'bowling', 'dribbling', 'jumpRope', 'run', 'shoot', 'tennis', 'phub', 'reading']
all_data = []

for folder in data_folders:
    folder_path = os.path.join('gyro/box/', folder)
    for filename in os.listdir(folder_path):
        if filename.endswith('.csv'):
            file_path = os.path.join(folder_path, filename)
            print(file_path)
            data = pd.read_csv(file_path, header=None) # 不使用第一行作為標籤
            # 提取標籤，假設檔案名稱就是標籤
            label = folder
            data['label'] = label
            all_data.append(data)

# 將所有資料合併並準備訓練資料
all_data = pd.concat(all_data, ignore_index=True)

# 將標籤轉換為數值
label_map = {label: i for i, label in enumerate(data_folders)}
all_data['label'] = all_data['label'].map(label_map)

# 提取特徵和標籤
features = all_data[['0', '1', '2', '3', '4', '5', '6']]
labels = all_data['label']

# 將資料劃分為訓練集和測試集
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2, random_state=42)

# 建立模型
input_shape = (X_train.shape[1],)
inputs = Input(shape=input_shape)
lstm_output = LSTM(64, return_sequences=True)(tf.expand_dims(inputs, axis=-1))
attention_output = Attention()(lstm_output, lstm_output)
flatten_output = Flatten()(attention_output) # 添加 Flatten 層
outputs = Dense(10, activation='softmax')(flatten_output)
model = Model(inputs, outputs)

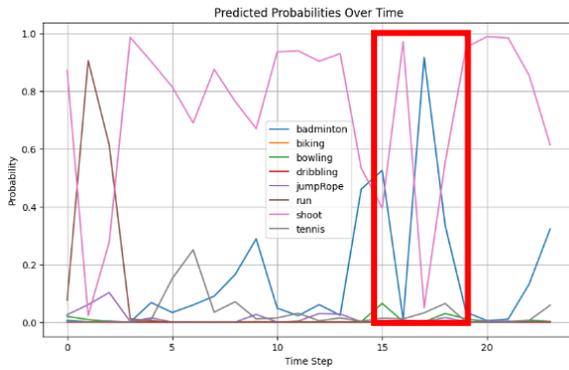
# 編譯模型
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# 訓練模型
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))

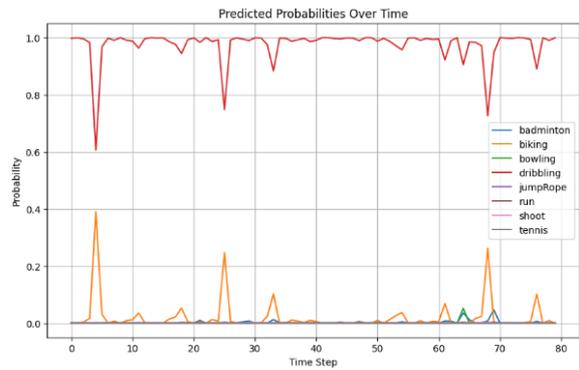
# 保存模型
model.save("model.h5")
```

3. 數據分析 - 運動模型性能評估

在日常使用時，手環可以配戴在慣用手上，根據網頁設定的採集頻率進行日常活動追蹤。這個不到 300kb 的手環模型在將數據傳入後，會預測使用者目前正在做何者運動，其運動類別預測準確率高達 92.80%。圖八至圖十一為使用 matlab 繪製折線圖顯示整個判斷過程，橫軸為數據索引，縱軸為運動類別機率。

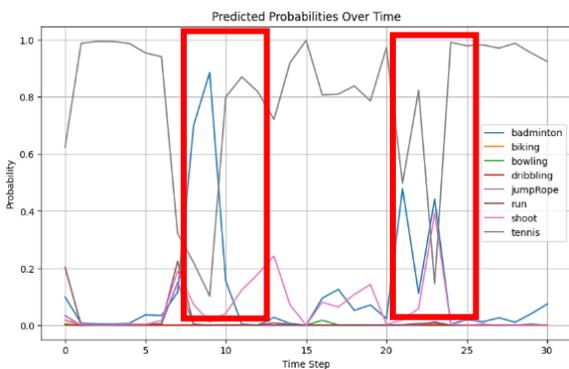


圖八：網球數據放入模型

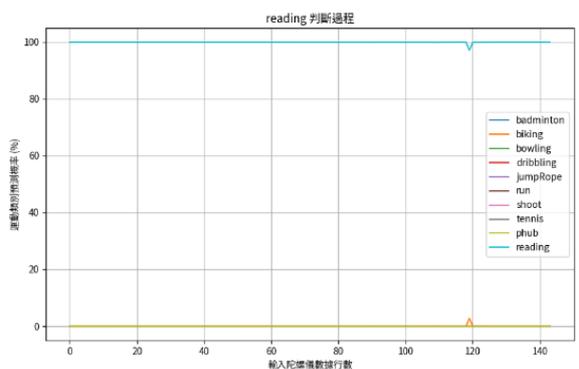


圖九：運球數據放入模型

從圖八模型的判斷過程中，我們可以看到當測試者在瞄準網球時，因與羽球的拉拍動作十分相似，因此模型一度以為是在打羽球(紅框區)，隨著之後更完整的運動過程，模型則十分確定此次運動為網球。從圖九可發現，雖然籃球運球與騎腳踏車一樣起伏不大，由於籃球運球沒有向前加速度，所以程式可以快速、輕鬆判斷。



圖十：投籃數據放入模型



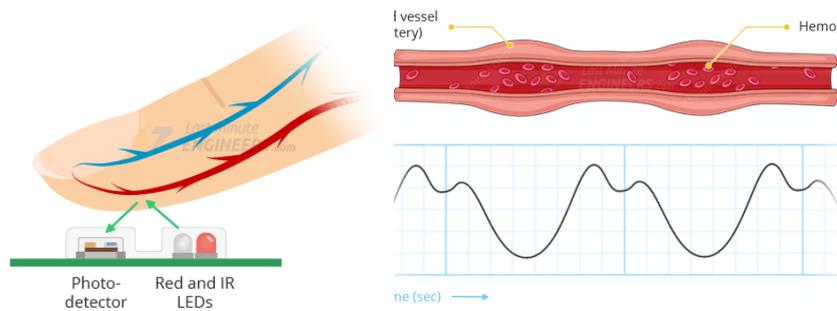
圖十一：讀書學習數據放入模型

在圖十中，我們發現投籃過程與羽球同樣有舉手與向下扣的動作，因此過程中模型有些許懷疑。而我們的模型也可以精確判斷讀書學習與與休閒滑手機，且準確率高達 96.57%，如圖十一所示。

(二) 心律分析

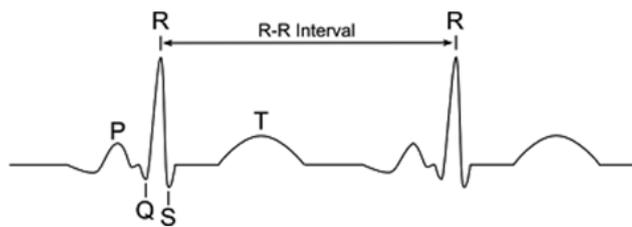
1. 數據採集

HRV 的測量通常是通過分析 ECG (心電圖) 或 PPG (光頸動脈描記術) 得到的心跳間隔的時間序列來進行的。這些測量可以提供關於個體壓力水平、健康狀態和運動恢復狀態的寶貴信息。然而我們可以透過 MAX30102 元件取得大約數值，MAX30102 對於心跳的偵測主要依賴感測器捕捉到的光訊號的變化。當心臟泵血時，血液量在血管中的變化導致經過皮膚的光吸收量改變。MAX30102 透過其內建的 LED 燈發出的光被血液吸收的量來檢測這種變化，從而推斷出心跳事件。



圖十二：MAX30102運作原理

我們調用 `getIR()` 函數持續讀取紅外（IR）通道的數據。紅外光數據反映了血液對紅外光的吸收程度，隨著心臟的跳動，這個值會周期性地變化。接著利用 `checkForBeat()` 函數來判斷是否檢測到心跳。這個函數內部通過分析紅外光數據的變化來確定是否出現了心跳。當檢測到光信號的峰值（代表心臟收縮時血液最多的時刻）後，函數會返回真（true），表示檢測到一次心跳。每當 `checkForBeat()` 返回值為 True 時，記錄當前的時間（使用 `millis()`），並與上一次心跳檢測到的時間作差，得到兩次心跳之間的時間間隔，即 RR 間期。



圖十三：心律 RR 間隔示意圖

2. 數據處理

得到 RR 值後我們通過時間序列分析計算 RMSSD，以判斷使用者目前的壓力。

由於其數學公式計算簡單，因此為了減少儲存空間的使用其計算是獨立在 ESP-32 上完成的。以下為統計與計算 RMSSD 的程式：

```
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"

MAX30105 particleSensor;

long lastBeat = 0; // Time at which the last beat occurred
bool noFinger = true; // To track if a finger is placed on the sensor
int intervals[4];
short i = 0;

void setup() {
  Serial.begin(115200);
  Serial.println("Initializing...");
}
```

```

if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) {
  Serial.println("MAX30105 was not found. Please check wiring/power.");
  while (1); // Endless Loop
}
Serial.println("Place your index finger on the sensor with steady pressure.");
particleSensor.setup(); // Configure sensor with default settings
particleSensor.setPulseAmplitudeRed(0x0A); // Turn Red LED to low to indicate sensor is running
particleSensor.setPulseAmplitudeGreen(0); // Turn off Green LED
}

void loop() {
  long irValue = particleSensor.getIR();
  if (checkForBeat(irValue) == true) {
    long delta = millis() - lastBeat;
    lastBeat = millis();
    Serial.print("delta=");
    Serial.println(delta);
    if (i < 4) {
      intervals[i++] = delta;
    }
    if (i == 4) {
      float rmssd = calcRMSSD(intervals, i);
      Serial.print("RMSSD: ");
      Serial.println(rmssd);
      i = 0; // Reset the index for new measurements
    }
  }

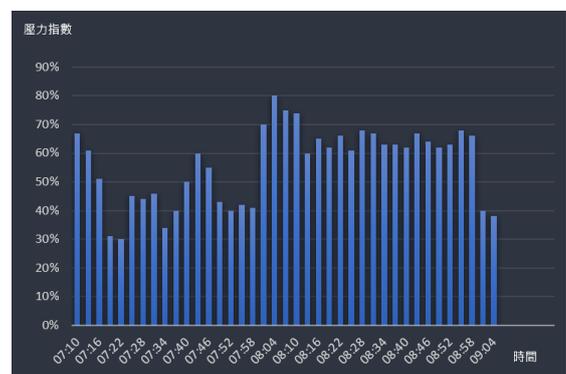
  if (irValue < 50000) {
    if (noFinger) {
      Serial.println("No finger detected.");
      noFinger = false;
    }
  } else {
    if (!noFinger) {
      Serial.println("Finger detected.");
      noFinger = true;
    }
  }
}

float calcRMSSD(int intervals[], int length) {
  float sum = 0;
  for (int j = 1; j < length; j++) {
    float diff = intervals[j] - intervals[j - 1];
    sum += diff * diff;
  }
  return sqrt(sum / (length - 1));
}

```

3. 數據分析

本實驗將蒐集到最高的 RMSSD 值定義為壓力值 100，最低的 RMSSD 值定義為壓力值 0，並將之後得到的 RMSSD 值反映在 RMSSD 與壓力值的迴歸直線上，從而得到其對應的壓力值，最後可以得出時間與壓力指數對應圖（圖十四）。

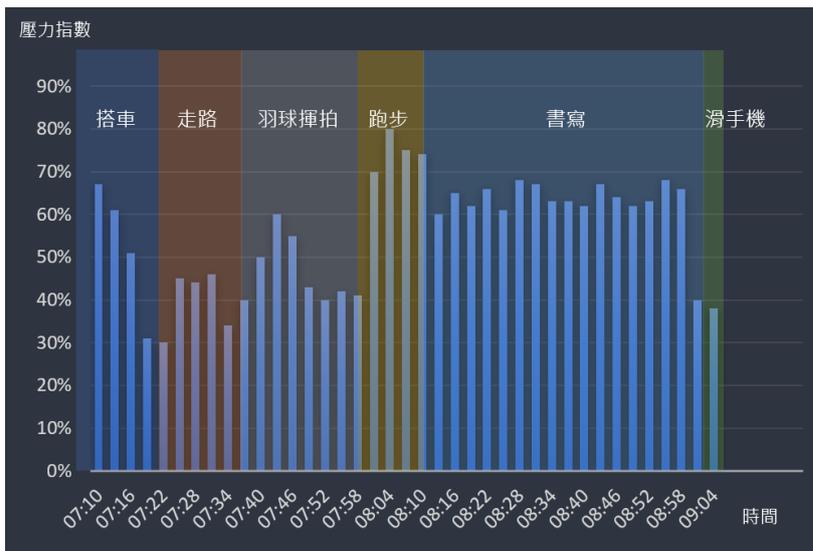


圖十四：時間與壓力指數對應圖

(三) 綜合比較

手環在開機後就會每分鐘進行一次數據採集，包括連續 30 秒的心律數據以及 5 秒的陀螺儀數據採集。HRV 值會在採集後被立即計算出來，與陀螺儀數據一起儲存至 Micro-SD 裡面。當手機或樹莓派等裝置連上 ESP-32 網頁後會同步時間並為數據加上時間標記。並使用手機的運算資源運行行為分析。

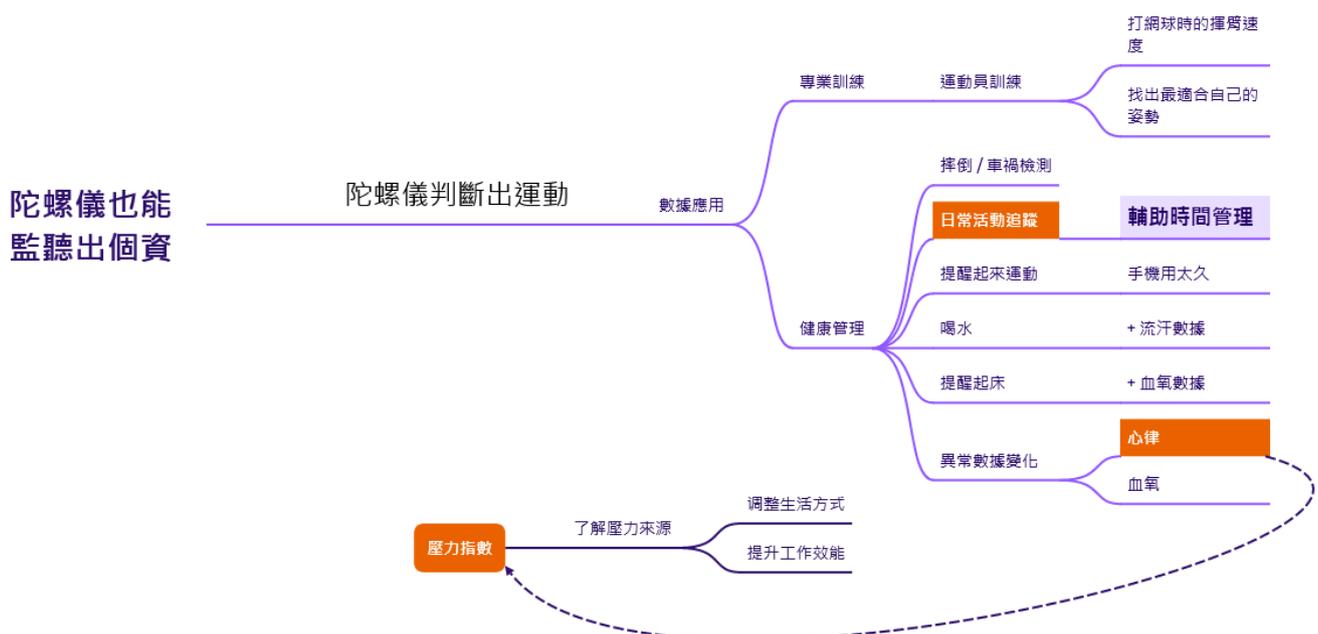
下圖為高二學生早上 7:10~9:10 活動與其對應的壓力數據，分別為出門搭車、走路、運動、上課學習及下課滑手機，從圖中可以分析出壓力最大時間為 8:05~8:10 上課時間前趕回教室，而上課時間壓力普遍較高，下課滑手機壓力指數雖然下降但也高達 40%。



圖十五：活動類別與其對應的壓力指數



圖十六：以網頁 GUI 呈現日常活動分析



圖十七：目前成果與未來展望 (橘色部分為已完成)

五、 結論

在這次的實驗中我們不斷的疊代數據蒐集方式及模型的訓練方式來提升便利性及準確性。如將陀螺儀數據增加了加速度，從三維提升到六維，使模型準確率提升了 10%。而手錶從原本比 Arduino UNO 還大 (尺寸：100x75x35，單位：mm；重量：193gw) 變成比 ESP-32 還小 (尺寸：41x35x19，單位：mm；重量：69gw)。

我們訓練的機器學習模型可用不到 530 元就能夠了解使用者一整天的生活習慣及健康數據，從而取得使用者的個人資訊。這項技術具有應用於運動和健康管理方面的潛力，在運動方面，我們可以透過手環監測運動過程並提供即時反饋和建議，並可運用其資訊輔助自己時間與健康管理。在健康方面，我們可以利用此裝置追蹤使用者的日常生活中的壓力來源並提供個人化的健康建議。也可以得知生活上時間的分配，例如:運動時間、睡眠時間、休閒時間、學習時間等，當然，這項技術若被不法使用，可能會導致嚴重的「資安問題」。

六、 參考文獻：

1. Mordechai, I. (2022, August 21). *GAIROSCOPE: Injecting Data from Air-Gapped Computers to Nearby Gyroscopes*. Arxiv. <https://arxiv.org/abs/2208.09764>
2. Olivia, C. (2015, July 22). *Motivic Toposes*. Arxiv. <https://arxiv.org/abs/1507.06271>
3. kannan, B., & Jestin, J. (2015, September 26). *A Prototype Malayalam to Sign Language Automatic Translator*. Arxiv. <https://arxiv.org/abs/1412.7415>
4. Lisa, vaas. (2015, September 15). *Smartwatch Sensors Can Be Used to Eavesdrop on the Keys You're Typing*. Sophos Nes. <https://pse.is/5q5bcu>
5. Maryam, M., ehsan toreini, T., Siamak f, shahandashti, & Feng, H. (2017, April 7). *Stealing PINs via Mobile Sensors: Actual Risk versus User Perception*. International Journal of Information Security. <https://link.springer.com/article/10.1007/s10207-017-0369-x>
6. 心率變異性 (HRV) 和心血管疾病危險因素的關係. JooiUp. (n.d.). <https://www.jooiup.com/knowledge/content/1262>
7. 瞭解 GARMIN 裝置上的 HRV 狀態. (n.d.). Garmin 台灣. <https://www.garmin.com.tw/minisite/garmin-technology/health-science/hrv-status/>