

臺北市立大安高級工業職業學校  
電子科

專題報告

長照輔助系統  
人體動態姿態辨識

學生 組長：邱紹濤  
組員：何冠頤  
組員：黃奕智  
組員：楊佳翰

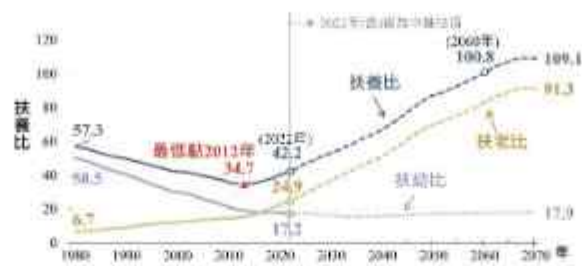
指導老師：張顯盛老師

中華民國 112 年 1 月 號

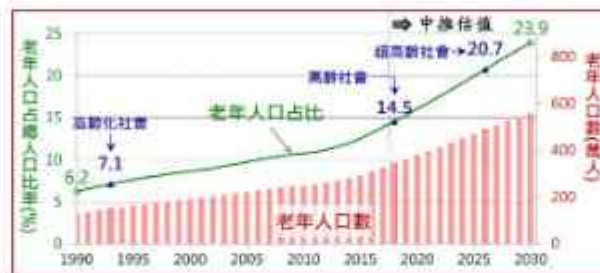
臺北市立大安高工 電子科  
專題報告：

# 摘要(Abstract)

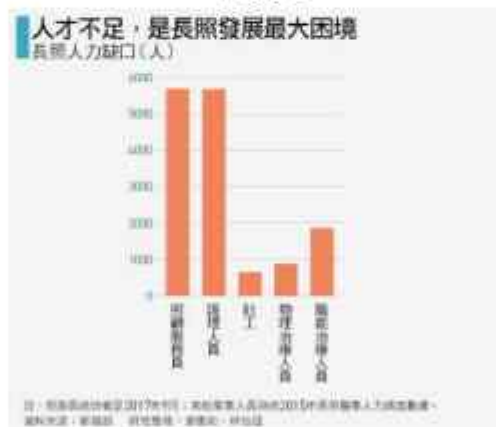
▼表 1



▼表 2



▼表 3



台灣現在已經邁入高齡社會，老年人口的比例正在快速增加，根據表(1)研究推估台灣將在 2026 年邁入超高齡社會，屆時台灣將有 20.7%的人口是老年人口，伴隨著這個現象，根據表(2)台灣的扶養比(每 100 個青壯年人口所需扶養(幼年人口+老年人口)數)也在急劇增長，為此，政府提出了許多老人及幼兒的照護政策，如我們熟知的長期照護。我們今天的主題便是長期照護輔助，因為根據表(3)，台灣的長照的工作人員是不夠的，我們想利用機器及電腦觀測人的動作，若老人有問題，長照人員就可以即時趕到，這樣就可以節省照顧同一老人所需要的人力，來減少台灣長照人員的缺口。

關鍵字:ESP32 CAM、機器學習、骨架辨識

# 目錄

|         |                         |    |
|---------|-------------------------|----|
| 第一章、    | 前言 .....                | 7  |
| 1-1、    | .....專題製作背景及目的          | 7  |
| 1-1-1、  | 背景 .....                | 7  |
| 1-1-2、  | 目的 .....                | 7  |
| 1-2、    | 專題製作方法 .....            | 7  |
| 1-2-1、  | 方法 .....                | 7  |
| 1-2-2、  | 步驟 .....                | 7  |
| 1-2-3、  | 進度 .....                | 7  |
| 1-3、    | 預計成果 .....              | 7  |
| 第二章、    | 理論探討 .....              | 7  |
| 2-1、    | 硬體 .....                | 7  |
| 2-1-1、  | ESP32 CAM .....         | 7  |
| 2-1-2、  | 攝像頭 .....               | 8  |
| 2-1-3、  | SG90 Servo 伺服馬達 .....   | 8  |
| 2-1-4、  | FT232RL .....           | 9  |
| 2-2、    | 軟體 .....                | 9  |
| 2-2-1、  | Python .....            | 9  |
| 2-2-2、  | Arduino .....           | 9  |
| 2-2-3、  | TensorFlow .....        | 9  |
| 2-2-4、  | OpenCV .....            | 10 |
| 2-2-5、  | Keras .....             | 10 |
| 2-2-6、  | Anaconda .....          | 10 |
| 2-2-7、  | Line Notify .....       | 11 |
| 2-2-8、  | Firebase .....          | 11 |
| 2-2-9、  | App Inventor .....      | 11 |
| 2-2-10、 | Tinkercad .....         | 12 |
| 2-2-11、 | Altium Designer .....   | 12 |
| 2-2-12、 | Autodesk Inventor ..... | 12 |
| 第三章、    | 專題設計 .....              | 13 |
| 3-1、    | 流程圖 .....               | 13 |

|             |                   |           |
|-------------|-------------------|-----------|
| 3-2、        | 甘特圖.....          | 13        |
| 3-3、        | 硬體部分.....         | 14        |
| 3-4、        | 軟體部分.....         | 16        |
| <b>第四章、</b> | <b>專題成果.....</b>  | <b>31</b> |
| 4-1         | 整體外觀.....         | 31        |
| 4-2         | 運作方式.....         | 31        |
| 4-3         | 問題與解決方法.....      | 31        |
| <b>第五章、</b> | <b>結論與建議.....</b> | <b>32</b> |
| 5-1         | 結論.....           | 32        |
| 5-2         | 建議.....           | 32        |
| 5-3         | 未來展望.....         | 32        |
| <b>第六章、</b> | <b>參考文獻.....</b>  | <b>33</b> |
| <b>第七章、</b> | <b>附件.....</b>    | <b>34</b> |

## 圖目錄

|       |                              |    |
|-------|------------------------------|----|
| ▲圖 1  | ESP32CAM 圖表 1.....           | 8  |
| ▲圖 2  | 攝像頭 圖表 2.....                | 8  |
| ▲圖 3  | SG90 SERVO 伺服馬達 圖表 3.....    | 8  |
| ▲圖 4  | FT232RL 圖表 4.....            | 9  |
| ▲圖 5  | PYTHON 圖表 5.....             | 9  |
| ▲圖 6  | ARDUINO 圖表 6.....            | 9  |
| ▲圖 7  | TENSORFLOW 圖表 7.....         | 10 |
| ▲圖 8  | OPEN CV 圖表 8.....            | 10 |
| ▲圖 9  | KERAS 圖表 9.....              | 10 |
| ▲圖 10 | ANACONDA 圖表 10.....          | 11 |
| ▲圖 11 | LINE NOTIFY 圖表 11.....       | 11 |
| ▲圖 12 | FIREBASE 圖表 12.....          | 11 |
| ▲圖 13 | APP INVENTOR 圖表 13.....      | 12 |
| ▲圖 14 | TINKERCAD 圖表 14.....         | 12 |
| ▲圖 15 | ALTIVUM DESIGNER 圖表 15.....  | 12 |
| ▲圖 16 | AUTODESK INVENTOR 圖表 16..... | 12 |
| ▲圖 17 | TINKERCAD 繪製介面圖表 17.....     | 14 |

|                                   |    |
|-----------------------------------|----|
| ▲圖 18 TINKERCAD 編輯介面 圖表 18 .....  | 14 |
| ▲圖 19 輸出 STL 檔圖表 19 .....         | 15 |
| ▲圖 20 3D 列印(1) 圖表 20 .....        | 15 |
| ▲圖 22 3D 列印(3) 圖表 22 .....        | 15 |
| ▲圖 21 3D 列印(2) 圖表 21 .....        | 15 |
| ▲圖 24 按鈕 電路設計圖圖表 24 .....         | 16 |
| ▲圖 23 ESP32 CAM 電路設計圖 圖表 23 ..... | 16 |
| ▲圖 25 輸入額外開發板管理員網址圖表 25 .....     | 16 |
| ▲圖 26 安裝 ESP32 相容性套件圖表 26 .....   | 17 |
| ▲圖 27 選擇開發版設定圖表 27 .....          | 17 |
| ▲圖 28 INCLUDE 函式庫圖表 28 .....      | 17 |
| ▲圖 29 內部接腳設定圖表 29 .....           | 18 |
| ▲圖 30 WIFI 連線圖表 30 .....          | 18 |
| ▲圖 31 按鈕按下傳緊急通知圖表 31 .....        | 18 |
| ▲圖 32 LINE 傳照片通知(1) 圖表 32 .....   | 18 |
| ▲圖 33 LINE 傳照片通知(2) 圖表 33 .....   | 19 |
| ▲圖 34 LINE 傳照片通知(3) 圖表 34 .....   | 19 |
| ▲圖 35 網頁介面(1) 圖表 35 .....         | 19 |
| ▲圖 36 網頁介面(2) 圖表 36 .....         | 20 |
| ▲圖 37 網頁介面(3) 圖表 37 .....         | 20 |
| ▲圖 38 網頁介面(4) 圖表 38 .....         | 20 |
| ▲圖 39 網頁介面(5) 圖表 39 .....         | 21 |
| ▲圖 40 ESP32 CAM 介面圖表 40 .....     | 21 |
| ▲圖 42 緊急按鈕被按下時會收到提醒圖表 42 .....    | 21 |
| ▲圖 41 打開 APP 時要輸入病人名稱圖表 41 .....  | 21 |
| ▲圖 43 APP INVENTOR 介面 圖表 43 ..... | 22 |
| ▲圖 45 現場狀況的觀看及控制圖表 45 .....       | 22 |
| ▲圖 44 APP INVENTOR 介面圖表 44 .....  | 22 |
| ▲圖表 46 .....                      | 23 |
| ▲圖 47 設定變數圖表 47 .....             | 23 |
| ▲圖 48 初始化界面圖表 48 .....            | 24 |
| ▲圖表 49 .....                      | 24 |
| ▲圖 50 程式(2) 圖表 50 .....           | 25 |
| ▲圖 51 程式(3) 圖表 51 .....           | 26 |
| ▲圖 52 ANACONDA 介面圖表 52 .....      | 27 |
| ▲圖 53 打開終端機圖表 53 .....            | 27 |
| ▲圖 54 輸入一個名子和版本環境圖表 54 .....      | 27 |
| ▲圖 55 環境列表圖表 55 .....             | 28 |

|                                    |    |
|------------------------------------|----|
| ▲圖 56 進入自己所創建的 PYTHON 環境圖表 56..... | 28 |
| ▲圖 57 躺的影片圖表 57.....               | 28 |
| ▲圖 58 躺的图片圖表 58.....               | 29 |
| ▲圖 59 TXT 檔圖表 59.....              | 29 |
| ▲圖 60 測試圖表 60.....                 | 30 |
| ▲圖 61 ANACONDA 安裝套件圖表 61.....      | 30 |
| ▲圖 62 FIREBASE 規則圖表 62.....        | 30 |
| ▲圖 63 跌倒影片圖表 63.....               | 30 |
| ▲圖 64 PYTHON 判斷圖表 64.....          | 31 |
| ▲圖 65 FIREBASE 輸出數值圖表 65.....      | 31 |
| ▲圖 66 整體外觀圖表 66.....               | 31 |
| ▲圖 67 HC-SR501 圖表 67.....          | 32 |
| ▲圖 68 MYSQL 圖表 68.....             | 32 |

## 表目錄

|                      |    |
|----------------------|----|
| ▼表 1.....            | 2  |
| ▼表 2.....            | 2  |
| ▼表 3.....            | 2  |
| ▼表 4 流程圖 表格 4.....   | 13 |
| ▼表 5 預計甘特圖 表格 5..... | 13 |
| ▼表 6 實際甘特圖 表格 6..... | 14 |

## 第 1 章、前言

### 1-1 專題製作背景及目的

#### 1-1-1 背景

我們生活中時常會有需要照顧人的時候，然而我們卻很難時刻注視著同一個人，因為這可能需要耗費大量人力成本。

#### 1-1-2 目的

這時我們就可以交給機器幫助照顧，讓機器辨別人的動作，在被照顧的人有特殊動作時通知我們，而這個辨別動作的功能就是我們的專題題目——人體動態姿態辨識。

### 1-2 專題製作方法、步驟與進度

#### 1-2-1 方法

利用 3D 列印出來的機構結合馬達，使攝像頭轉動並判斷病人狀況，再用 APP Inventor 和 LINE Notify 傳送給使用者。

#### 1-2-2 步驟

Altium Designer、ESP32CAM、3D 列印與馬達轉動程式結合，都完成後，再與人體動態姿態辨識程式整合，接著回傳病人狀況。

#### 1-2-3 進度

可以判斷目前病人狀況，如有特殊狀況會通知使用者。

### 1-3 預計成果

製作一台能夠協助辨識病人姿態的辨識機，在病人出現變化時能夠查看現場狀況和看一日作息，當想要查看現場狀況可以讓 LINE 傳照片或是查看現場。

## 第二章、理論探討

### 2-1 硬體

#### ➤ ESP32CAM

ESP32 是一系列低成本，低功耗的單晶片微控制器，整合了 Wi-Fi 和藍牙。ESP32 系列採用 Tensilica Xtensa LX6 微處理器，包

括雙核心和單核變體，內建天線開關，RF 變換器，功率放大器，低雜訊接收放大器，濾波器和電源管理模組。



▲圖 1 ESP32CAM

### ➤ 攝像頭

傳送影像給電腦判斷



▲圖 2 攝像頭

### ➤ SG90 Servo 伺服馬達

伺服馬達裡含有直流馬達、齒輪箱、軸柄，以及控制電路，可透過訊號控制軸柄的停止角度，大概都是 0 到 180 度。一般伺服馬達有三條線，電源、接地、訊號線。



▲圖 3 SG90 Servo 伺服馬達

### ➤ FT232RL

USB 轉 TTL 並提供 5V 電壓。





▲圖 4 FT232RL

## 2-2 軟體

### ➤ Python

**Python**，是一種廣泛使用的直譯式、進階和通用的程式語言。Python 支援多種程式設計範式，包括函數式、指令式、反射式、結構化和物件導向程式設計。它擁有動態型別系統和垃圾回收功能，能夠自動管理記憶體使用，並且其本身擁有一個巨大而廣泛的標準庫。它的語言結構以及物件導向的方法旨在幫助程式設計師為小型的和大型的專案編寫清晰的、合乎邏輯的程式碼。



▲圖 5 Python

### ➤ Arduino

**Arduino** 是一個開源嵌入式硬體平台，用來供使用者製作可互動式的嵌入式專案。此外 **Arduino** 作為一個開源硬體和開源軟體的公司，同時兼有專案和使用者社群。



▲圖 6 Arduino

### ➤ TensorFlow

**TensorFlow** 充當機器學習的核心平台和函式庫。TensorFlow 的 API 使用 Keras 來允許使用者製作自己的機器學習模型。除了建造和訓練它們的模型之外，TensorFlow 還能幫助裝載資料來訓練模型，並使用 TensorFlow Serving 來部署它。



▲圖 7 Tensorflow

➤ **Open CV**

OpenCV 的全稱是 Open Source Computer Vision Library，是一個跨平台的電腦視覺庫。OpenCV 可用於開發即時的圖像處理、電腦視覺以及圖型識別程式。



▲圖 8 Open CV

➤ **Keras**

Keras 是一個用 Python 編寫的開源神經網路庫，從 2021 年 8 月的版本 2.6 開始，它是在 TensorFlow 2 上建立的高層 API；。Keras 旨在快速實現深度神經網路，專注於使用者友好、模組化和可延伸性，是 ONEIROS（開放式神經電子智慧機器人作業系統）專案研究工作的部分產物



▲圖 9 Keras

➤ **Anaconda**

Anaconda 是一個開源的 Python 和 R 語言的發行版本，用於計算科學（資料科學、機器學習、巨量資料處理和預測分析），Anaconda 致力於簡化軟體套件管理系統套件。



▲圖 10 Anaconda

### ➤ Line Notify

透過 LINE 接收其他網站服務通知與網站服務連動完成後，LINE 所提供的官方帳號「LINE Notify」將會傳送通知。不僅可與多個服務連動，也可透過 LINE 群組接收通知。



▲圖 11 Line Notify

### ➤ Firebase

提供了實時資料庫和後端即服務。該服務為應用程式開發人員提供了一個 API 允許應用程式數據能跨客戶端同步儲存在 Firebase 的雲端空間上。



▲圖 12 Firebase

### ➤ App Inventor

是一款卡通圖形界面的 Android 智慧型手機應用程式開發軟體。它使用圖形化界面，類似於 Scratch 語言。用戶可以拖放圖形對象來創造一個運行在安卓系統上的應用程式，它就可以在許多手機設備上運行。



▲圖 13 App Inventor

➤ **Tinkercad**

用來繪製 3D 圖，並輸出 STL 檔給 3D 列印機來影印。



▲圖 14 Tinkercad

➤ **Altium Designer**

用來設計電路板。



▲圖 15 Altium Designer

➤ **Autodesk Inventor**

輸入 STL 檔並繪製爆炸圖。

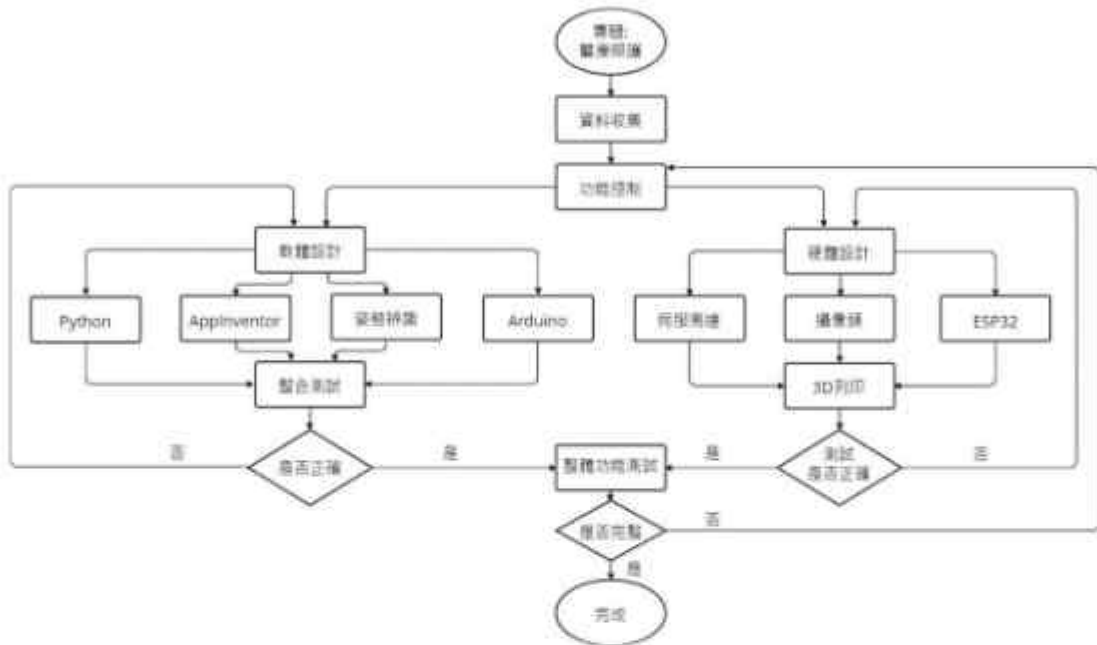


▲圖 16 Autodesk Inventor

### 第三章、專題設計

#### 3-1 流程圖

▼表 4 流程圖



#### 3-2 甘特圖

▼表 5 預計甘特圖

| 週次<br>工作項 | 1 | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18   |
|-----------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|
| 資料蒐集      | ■ | ■  | ■  | ■  |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| 理論探討      | ■ | ■  | ■  | ■  |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| 電路規劃      |   |    |    | ■  | ■  | ■  | ■  | ■  | ■  | ■  | ■  |    |    |    |    |    |    |      |
| 電路測試      |   |    |    |    |    |    |    |    |    |    | ■  | ■  | ■  | ■  | ■  |    |    |      |
| 軟體規劃      | ■ | ■  | ■  | ■  | ■  | ■  | ■  | ■  | ■  | ■  |    |    |    |    |    |    |    |      |
| 軟體測試      |   |    |    |    |    |    |    |    |    |    | ■  | ■  | ■  | ■  | ■  |    |    |      |
| 全體測試      |   |    |    |    |    |    |    |    |    |    | ■  | ■  | ■  | ■  | ■  |    |    | ■    |
| 外殼製作      |   |    |    |    |    |    |    |    |    |    |    | ■  | ■  | ■  | ■  | ■  | ■  |      |
| 報告撰寫      |   | ■  |    | ■  |    | ■  |    | ■  |    | ■  |    | ■  |    | ■  |    | ■  |    | ■    |
| 口頭報告      |   |    | ■  |    | ■  |    | ■  |    | ■  |    | ■  |    | ■  |    | ■  |    | ■  | ■    |
| 預定進度      | 5 | 10 | 25 | 30 | 35 | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 | 80 | 85 | 90 | 95 | 100% |

▼表 6 實際甘特圖

| 週次<br>工作項目 | 1 | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18   |
|------------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|
| 資料蒐集       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| 理論探討       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| 電路規劃       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| 電路測試       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| 軟體規劃       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| 軟體測試       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| 全體測試       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| 外殼製作       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| 報告撰寫       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| 口頭報告       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| 預定進度       | 5 | 10 | 25 | 30 | 35 | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 | 80 | 85 | 90 | 95 | 100% |

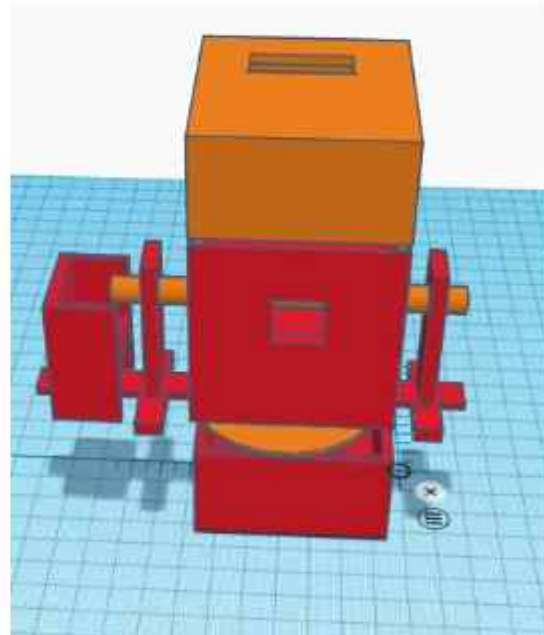
### 3-3 硬體部分

#### 3-4-1 3D 列印部分

使用 Tinkercad



▲圖 17 Tinkercad 繪製介面



▲圖 18 Tinkercad 編輯介面

轉成 STL 檔



▲圖 19 輸出 STL 檔

使用 3D 列印機列印

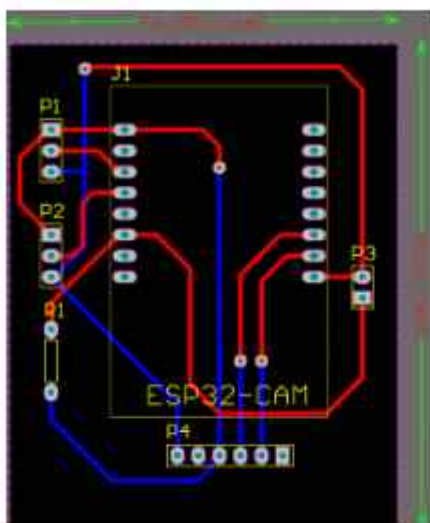


▲圖 20 3D 列印(1)

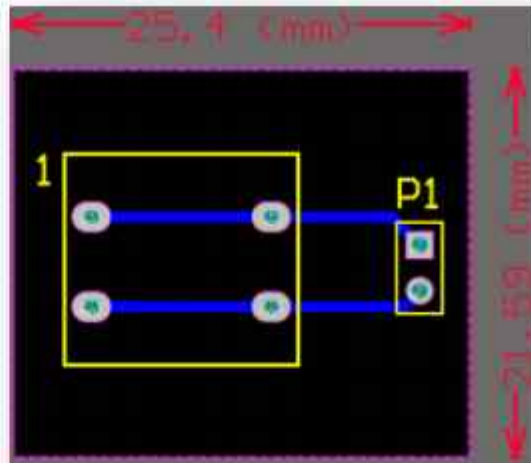
▲圖 21 3D 列印(2)

▲圖 22 3D 列印(3)

3-4-2 Altium Designer 部分



▲圖 23 ESP32 CAM 電路設計圖



▲圖 24 按鈕 電路設計圖

接著電路設計完成後我們將電路圖印出，並將電路板切成適當大小及進行曝光，然後鑽孔並焊接。

### 3-4 軟體部分

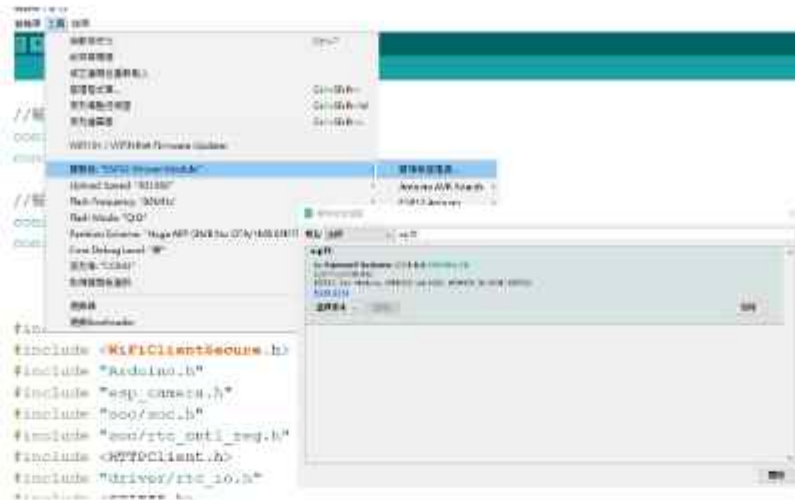
#### 3-4-1 ESP32-CAM 部分

ESP32-CAM 環境設定：



▲圖 25 輸入額外開發板管理員網址





▲圖 26 安裝 ESP32 相容性套件



▲圖 27 選擇開發版設定

程式部分：

```

#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "Arduino.h"
#include "esp_camera.h" // 照訊
#include "soc/soc.h" // 用於電源不穩不重開機
#include "soc/rtc_cntl_reg.h" // 用於電源不穩不重開機
#include <HTTPClient.h>
#include "driver/rtc_io.h"
#include <SPIFFS.h>
#include <FS.h>
#include <Firebase_ESP_Client.h>
//Provide the token generation process info.
#include <addons/TokenHelper.h>
#include "addons/TokenHelper.h"
//Provide the RTTB payload printing info and other helper functions.
#include "addons/RTTBHelper.h"

```

▲圖 28 include 函式庫

```

77 //安可信ESP32-CAM模組腳位設定
78 #define FWDN_GPIO_NUM    32
79 #define RESET_GPIO_NUM  -1
80 #define XCLK_GPIO_NUM    0
81 #define SIOD_GPIO_NUM    26
82 #define SIOC_GPIO_NUM    27
83
84 #define Y9_GPIO_NUM       38
85 #define Y8_GPIO_NUM       34
86 #define Y7_GPIO_NUM       39
87 #define Y6_GPIO_NUM       36
88 #define Y5_GPIO_NUM       21
89 #define Y4_GPIO_NUM       19
90 #define Y3_GPIO_NUM       18
91 #define Y2_GPIO_NUM        5
92 #define VSYNC_GPIO_NUM    25
93 #define HREF_GPIO_NUM     23
94 #define PCLK_GPIO_NUM     22

```

▲圖 29 內部接腳設定

```

177 for (int i=0;i<2;i++) {
178   WiFi.begin(ssid, password); //執行網路連線
179
180   delay(1000);
181   Serial.println("");
182   Serial.print("Connecting to ");
183   Serial.println(ssid);
184
185   long int StartTime=millis();
186   while (WiFi.status() != WL_CONNECTED) {
187     delay(500);
188     if ((StartTime+5000) < millis()) break; //等待10秒連線
189   }

```

▲圖 30 Wi-Fi 連線

```

275 PiQ = digitalRead(buttonPin);
276 //Line Notify (smile)
277 if (PiQ == HIGH or b==1) {
278   Serial.println("");
279   sendCapturedImageLineNotify(lineNotifyToken);
280   a=0;
281   b=0;
282   if (Firebase.getInstance().req(a,b, "/photo/camera-gomg-button", 1) != null) {
283     Serial.println("YES!!");
284     Serial.println("FBFB = " + Firebase.getInstance().data(a,b));
285     Serial.println("FBFB = " + Firebase.getInstance().data(a,b));
286   }
287   else {
288     Serial.println("FAILED");
289     Serial.println("REASON: " + Firebase.getInstance().errorReason());
290   }
291 }

```

▲圖 31 按鈕按下傳緊急通知

```

String sendCapturedImageLineNotify(String token) {
  camera_fb_t * fb = NULL;
  fb = esp_camera_fb_get();
  if(!fb) {
    Serial.println("Camera capture failed");
    delay(1000);
    ESP.restart();
    return "Camera capture failed";
  }
  WiFiClientSecure c1(cert, key);
  c1.setInsecure(); //for version 1.0.5 or above
  Serial.println("Connect to notify-api.line.me");
  if (c1.connect("notify-api.line.me", 443)) {
    Serial.println("Connection successful");

    String message = "病人有狀況，請檢查";
    String head = "--Tailwan%\nContent-Disposition: form-data";
    String tail = "\n\n--Tailwan%\n";

```

▲圖 32 Line 傳照片通知(1)



```

</section>

<div class="input-group" id="probability-group">
  <label for="probability">probability</label>
  <div class="range-min">0</div>
  <input type="range" id="probability" min="0" max="1" value="0" step="0.1" class="default-action">
  <div class="range-max">1</div>
</div>
<div class="input-group" id="flash-group">
  <label for="flash">flash</label>
  <div class="range-min">0</div>
  <input type="range" id="flash" min="0" max="200" value="0" class="default action">
  <div class="range-max">200</div>
</div>
<div class="input-group" id="resolution-group">
  <label for="resolution">resolution</label>
  <div class="input-group" class="multi-select">
    <option value="10">1000(1000x1000)</option>
    <option value="15">1500(1500x1024)</option>
    <option value="20">2000(1920x1080)</option>
    <option value="25">2500(2500x1600)</option>
    <option value="30">3000(3000x1600)</option>
    <option value="35">3500(3500x1600)</option>
    <option value="40">4000(4000x2240)</option>
    <option value="45">4500(4500x2240)</option>
    <option value="50">5000(5000x2240)</option>
  </div>

```

▲圖 36 網頁介面(2)

```

</div>
<div class="input-group" id="quality-group">
  <label for="quality">Quality</label>
  <div class="range-min">10</div>
  <input type="range" id="quality" min="10" max="25" value="10" class="default-action">
  <div class="range-max">25</div>
</div>
<div class="input-group" id="brightness-group">
  <label for="brightness">Brightness</label>
  <div class="range-min">-2</div>
  <input type="range" id="brightness" min="-2" max="2" value="0" class="default action">
  <div class="range-max">2</div>
</div>
<div class="input-group" id="contrast-group">
  <label for="contrast">contrast</label>
  <div class="range-min">-2</div>
  <input type="range" id="contrast" min="-2" max="2" value="0" class="default action">
  <div class="range-max">2</div>
</div>
<div class="input-group" id="saturation-group">
  <label for="saturation">Saturation</label>
  <div class="range-min">-2</div>
  <input type="range" id="saturation" min="-2" max="2" value="0" class="default action">
  <div class="range-max">2</div>
</div>
<div class="input-group" id="special_effect-group">
  <label for="special_effect">Special Effect</label>

```

▲圖 37 網頁介面(3)

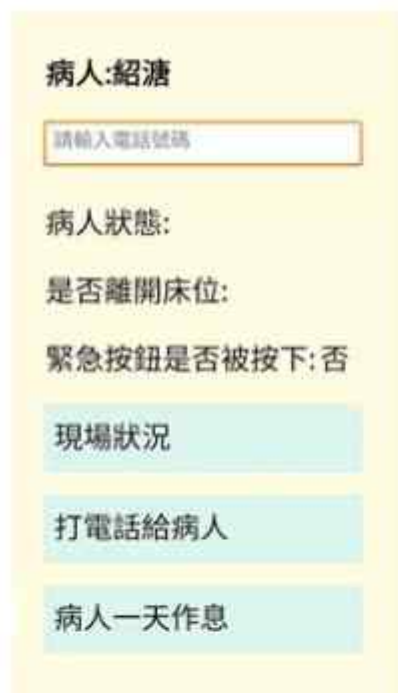
```

  <select id="special_effect" class="default-action">
    <option value="0" selected="selected">No Effect</option>
    <option value="1">Negative</option>
    <option value="2">Grayscale</option>
    <option value="3">Red Tint</option>
    <option value="4">Green Tint</option>
    <option value="5">Blue Tint</option>
    <option value="6">Sepia</option>
  </select>
</div>
<div class="input-group" id="mirror-group">
  <label for="mirror">Mirror</label>
  <div class="switch">
    <input id="mirror" type="checkbox" class="default-action" checked="checked">
    <label class="slider" for="mirror"></label>
  </div>
</div>
<div class="input-group" id="flip-group">
  <label for="flip">Flip</label>
  <div class="switch">
    <input id="flip" type="checkbox" class="default-action" checked="checked">
    <label class="slider" for="flip"></label>
  </div>
</div>
</div>

```

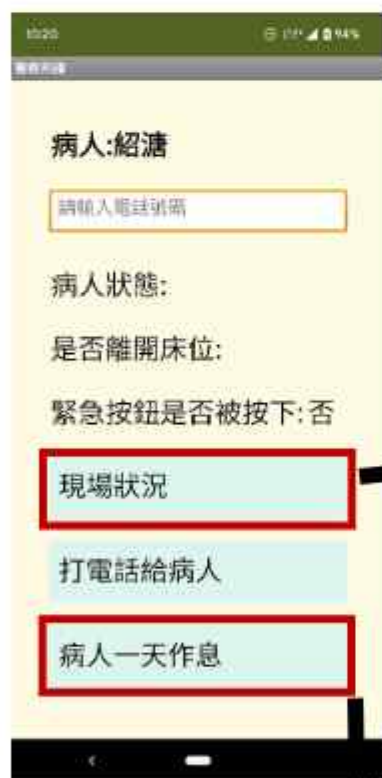
▲圖 38 網頁介面(4)



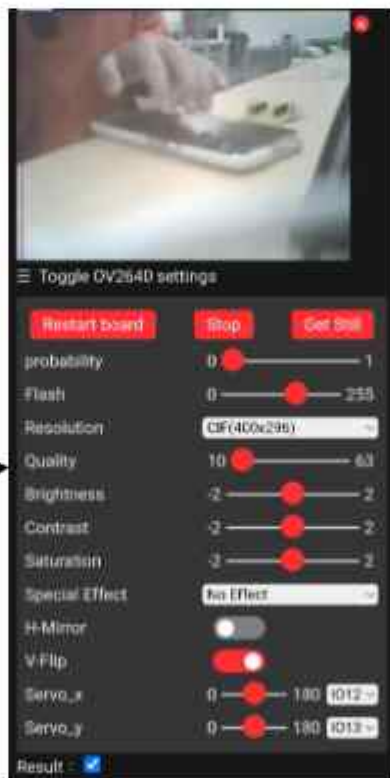


▲圖 43 App Inventor 介面

1. 輸入的病人名稱
2. 輸入電話號碼
3. 病人當前的狀態
4. 病人是否離開
5. 緊急按鈕是否被按下
6. 點擊看現場照片
7. 點擊打給病人
8. 點擊看病人作息



▲圖 44 App Inventor 介面



▲圖 45 現場狀況的觀看及控制



▲圖 46 病人當天作息的紀錄  
 程式部分：

```

initialize global state_in_chinese to " "
initialize global refresh to " "
initialize global state_of_time to " "
initialize global begin_h to " "
initialize global emergency_button to " "
initialize global begin_m to " "
initialize global leave_or_not to " "
initialize global color to " "
initialize global state to "123"
initialize global line_h to " "
initialize global get_up to " "
initialize global line_m to " "
initialize global get_up_time_common to " "
initialize global leave_count to " "
initialize global time to " "
initialize global leave_and_rest_h to " "
initialize global leave_and_rest_m to " "
initialize global back to " "
  
```

▲圖 47 設定變數

```

when Screen1 Initialize
do
  set clock TimerEnabled to true
  set photo Picture to join
    https://firebasestorage.googleapis.com/v0/b/
    project-8011899077907740159.appspot.com/
    /o/
    123
    %21
    photo.jpg
    ?alt=media&token=
    null
  call player Stop
  set clock TimerEnabled to true
  set work_or_rest_arrangement Visible to false
  set site_arrangement Visible to false
  set leave_and_refresh Visible to false
  call notifier ShowTextDialog
    message 提醒人請人事情
    title 放入名稱
    cancelable false
  call FirebaseDatabase1 StoreValue
    tag move
    valueToStore move

```

▲圖 48 初始化界面

```

when FirebaseDatabase1 DataChanged
  tag value
do
  set global time to join
    call clock Hour
    instant call clock Now
    .
    call clock Minute
    instant call clock Now
  if compare texts get tag = move
  then if and
    compare texts get value = get global state
    then
      set global leave_count to get global leave_count + 1
      if get global leave_count = 5
      then set global state_of_time to join
        get global state_of_time
        get global time
        .
        時間
        .
        in
      set global state_in_chinese to 時間
      set global leave_or_not to get global state
      if get global leave_or_not = false
      then set leave_or_not_input TextColor to
        set leave_or_not_input Text to
      else if get global leave_or_not = true
      then set leave_or_not_input TextColor to
        set leave_or_not_input Text to

```

▲圖 49 程式(1)



1. 判斷 Firebase 的資料是否改變
2. 獲取當下時間
3. 判斷病人是否離開

```

else if compare texts get (value) != get global state
then
  if compare texts get global state == 'lay'
  then set global get_up to 1
  else if compare texts get global state == 'in'
  then
    set global state_in_chinese to '異常'
    set global state_of_time to join get global state_of_time
    get global time
    get global state_in_chinese
    'in'
    set work_or_rest_input .Text to get global state_of_time
    set state_input .Text to get global state_in_chinese
    set global back to 1
  set global state to get value
  call state

```

4

```

else if compare texts get tag == 'emergency_button'
then
  set global emergency_button to get (value)
  if get global emergency_button == 0
  then
    set emergency_button_input .TextColor to black
    set emergency_button_input .Text to '閉'
  else if get global emergency_button == 1
  then
    set emergency_button_input .TextColor to red
    set emergency_button_input .Text to '開'
    call player .Start
    call notifier .ShowChooseDialog
    message '緊急按鈕已被按下'
    title '警告'
    button1Text '確定'
    button2Text '觀看現場狀況'
    cancelable false

```

5

▲圖 50 程式(2)

4. 判斷病人是否回來
5. 判斷緊急按鈕是否被按下

根據收到的資料輸出當前狀態：

```
to state
do
  if (compare texts (get global state) = "walk")
  then (set global state in chinese to "跑步")
  else if (compare texts (get global state) = "fall")
  then (set global state in chinese to "跌倒")
  else if (compare texts (get global state) = "jump")
  then (set global state in chinese to "跳")
  else if (compare texts (get global state) = "stand")
  then (set global state in chinese to "站")
  else if (compare texts (get global state) = "sit")
  then
    if (compare texts (get global get up) = "1")
    then
      set global get up to 1
      set global state in chinese to "起床"
    else
      set global state in chinese to "坐"
    else if (compare texts (get global state) = "lay")
    then (set global state in chinese to "休息")
    else if (compare texts (get global state) = " ")
    then (set global state in chinese to "離開")

  set global state of time to join (get global state of time)
  (get global time)
  " "
  (get global state in chinese)
  "\n"

  set work or rest input . Text to (get global state of time)
  set state input . Text to (get global state in chinese)
```

6

7

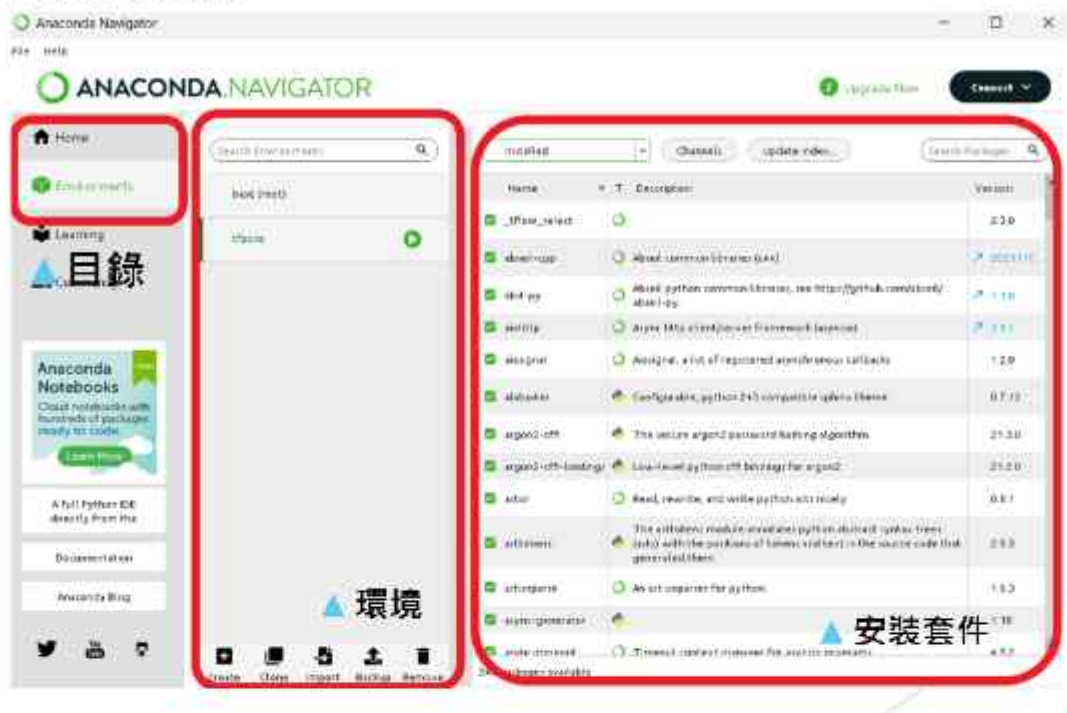
▲圖 51 程式(3)

6. 判斷病人當前狀態

7. 顯示狀態

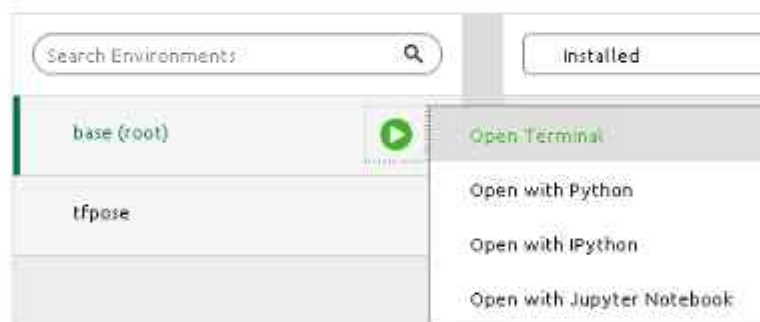
### 3-4-3 Python 部分

#### 1. Anaconda 安裝

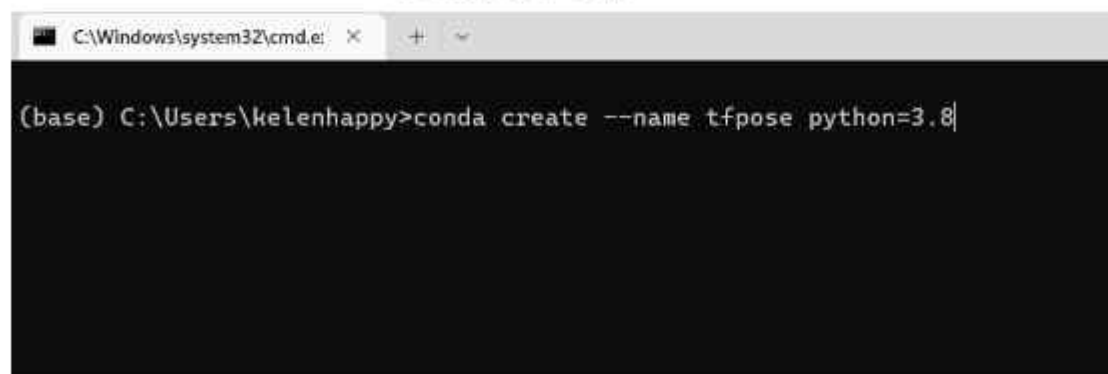


▲圖 52 Anaconda 介面

先使用 Anaconda 創建適用的 Python 版本



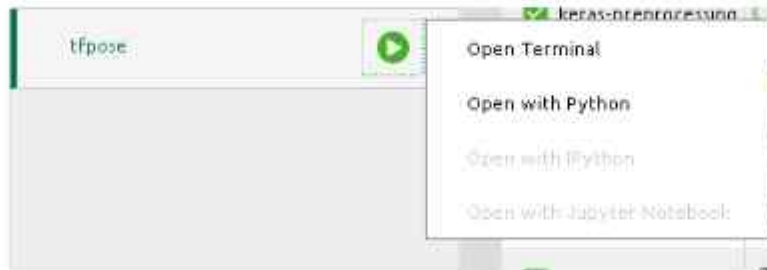
▲圖 53 打開終端機



▲圖 54 輸入一個名子和版本環境

```
C:\Windows\system32\cmd.e: x + v
(base) C:\Users\kelenhappy>conda env list
# conda environments:
#
base          * C:\Users\kelenhappy\anaconda3
tfpose       C:\Users\kelenhappy\anaconda3\envs\tfpose
```

▲圖 55 環境列表



▲圖 56 進入自己所創建的 Python 環境

## 2. 安裝 OpenCV

<https://www.lfd.uci.edu/~gohlke/pythonlibs/#opencv>

從網址中尋找適合的版本

## 3 下載 Realtime-Action-Recognition 到電腦

<https://github.com/felixchenfy/Realtime-Action-Recognition>

## 4. 安裝 TensorFlow

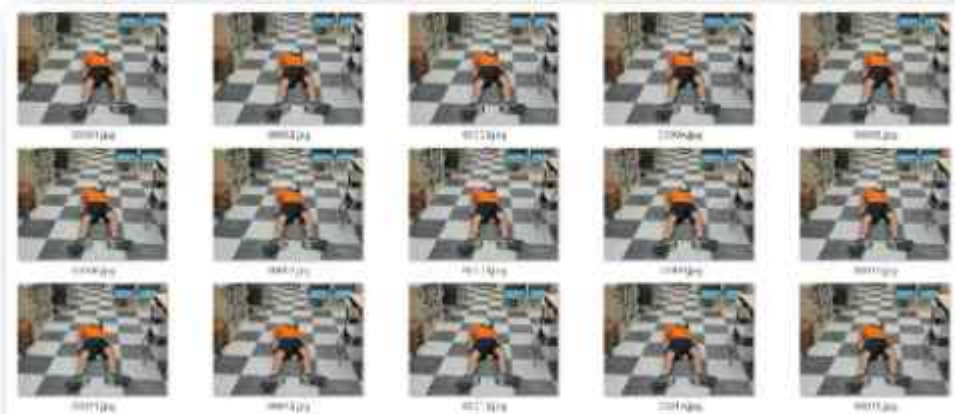
## 5. 安裝 swig 來使用 C++腳本

<http://www.swig.org/download.html>

## 6. 把影片轉成照片並用數命名



▲圖 57 躺的影片



▲圖 58 躺的圖片

7. 在 txt 檔中輸入關鍵影格

```
lay_0
1 310

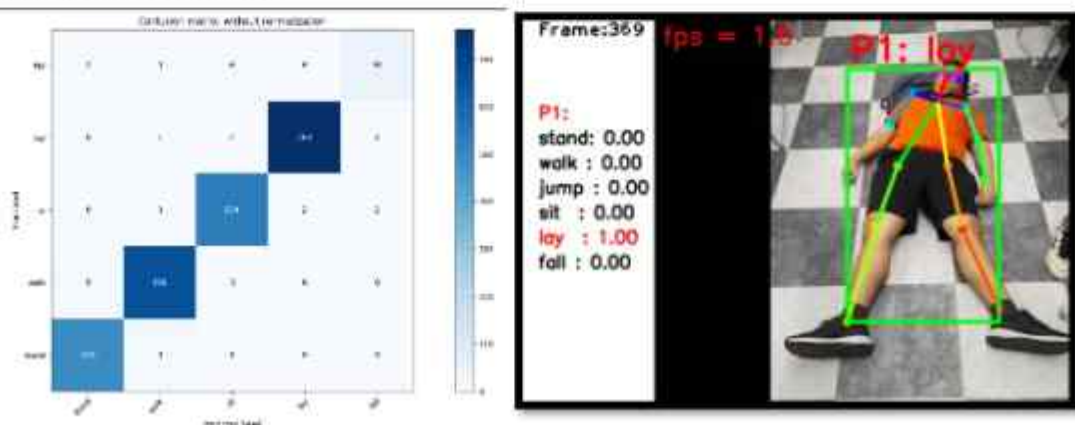
fall_0
64 129

fall_1
100 152
```

▲圖 59 txt 檔

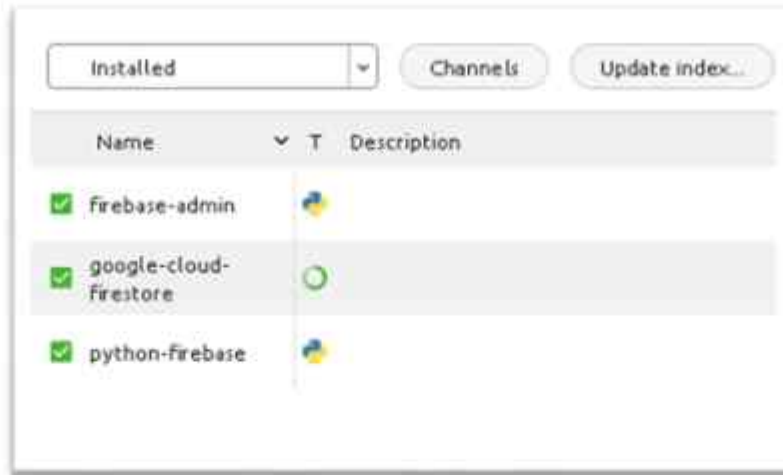
8. 依序輸入指令來訓練模型

```
python src/s1_get_skeletons_from_training_imgs.py
python src/s2_put_skeleton_txts_to_a_single_txt.py
python src/s3_preprocess_features.py
python src/s4_train.py
```



▲圖 60 測試

9. 在 Python 中安裝 Firebase 套件



▲圖 61 Anaconda 安裝套件

10. Firebase 修改規則讓使用者可讀取和寫入

```
{  
  "rules": {  
    ".read": true, // 2022-10-30  
    ".write": true, // 2022-10-30  
  }  
}
```

▲圖 62 Firebase 規則

11. 跌倒判斷何通知



▲圖 63 跌倒影片

```
if test=="fall":
    ref.update({
        'emergency_button': '1'
    })
```

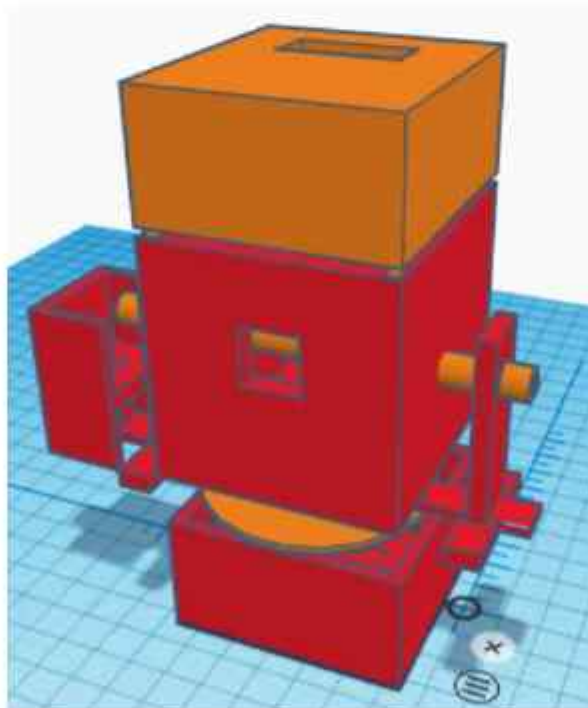
▲圖 64 Python 判斷

```
people
├── emergency_button: "1"
├── move: "fall"
└── name: 1
```

▲圖 65 Firebase 輸出數值

## 第四章、專題設計

### 4-1 整體外觀



▲圖 66 整體外觀

### 4-2 運作方式

用手機控制伺服馬達轉動外殼

## 4-3 問題與解決方法

### HC-SR501

- ◆ 與我們的功能需求不符



▲圖 67 HC-SR501

### MySQL

- ◆ App Inventor 連線時較為複雜



▲圖 68 MySQL

## 第五章、結論與建議

### 5-1 結論

動態姿態辨識需運用到許多影片資料來訓練人工智慧，如果要在姿勢判斷上做到更好的優化，就需要拍攝更多的照片和影片。我們也可以結合多個鏡頭，實現完整的系統。

### 5-2 建議

在這次的專題中，我們組員們有序的分工並各司其職，最終也如預期地完成了我們的長照輔助裝置，雖然它或許可以用來輔助長照，但若要實際投入生產，我們還是有多要改進的地方。

### 5-3 未來展望

- (1)在未來 5G 技術成熟時，可以再提升影像傳輸速率及品質，使影像辨識即時畫面的幀率提升。
- (2)目前只能在 Android 上使用，iOS 則不能，未來製作一個能在 Android 和 iOS 互通的 APP
- (3)因為人體動態姿態辨識判斷，是將影片轉換成數張照片進行判斷，在圖形處理上，GPU 的性能遠不及 TPU



## 第六章、參考資料

- OpenCV 維基百科。取自：  
<https://zh.wikipedia.org/zh-tw/OpenCV>
- CNN 與 RNN 得差異。取自：  
<https://blogs.nvidia.com.tw/2018/09/05/whats-the-difference-between-a-cnn-and-an-rnn/>
- Firebase 修改規則。取自：  
<https://community.kodular.io/t/error-from-companion-for-a-custom-firebase-host-you-must-first-set-your-authentication-server-before-using-authentication-features/125331/4>
- 在 Anaconda 底下創建 TensorFlow 和 Keras。取自：  
<https://medium.com/@virginiakm1988/%E5%9C%A8-anaconda-%E8%99%9B%E6%93%AC%E7%92%B0%E5%A2%83%E4%B8%8B%E5%AE%89%E8%A3%9D-tensorflow%E8%88%87-keras-c2c5aed98fef>
- 人體動態姿態辨識相關資料。取自：  
<https://ithelp.ithome.com.tw/users/20129150/ironman/2969>
- TensorFlow 和 Keras 的相關資料。取自：  
<https://medium.com/chiukevin0321/tensorflow%E8%88%87keras%E5%9F%BA%E6%9C%AC%E4%BB%8B%E7%B4%B9-621352fc7150>
- Anaconda 維基百科。取自：  
[https://zh.m.wikipedia.org/zh-tw/Anaconda\\_\(Python%E5%8F%91%E8%A1%8C%E7%89%88\)](https://zh.m.wikipedia.org/zh-tw/Anaconda_(Python%E5%8F%91%E8%A1%8C%E7%89%88))
- HC-SR501 為何無法達到預期功能。取自：  
<https://shop.mirotek.com.tw/iot/esp32-start-16/>
- App Inventor 接收和發送訊息到 Firebase。取自：  
<https://community.appinventor.mit.edu/t/esp32-firebase-db-send-receive-ide-arduino/10747>
- App Inventor 和 ESP32-CAM 結合到 Firebase。取自：  
[http://kio4.com/arduino/117\\_Wemos\\_Firebase.htm](http://kio4.com/arduino/117_Wemos_Firebase.htm)
- 深度學習和機器學習。取自：  
<https://levity.ai/blog/difference-machine-learning-deep-learning>

## 第七章、附件

### 設備清單

| 類別 | 設備、軟體名稱         | 應用說明   |
|----|-----------------|--|
| 軟體 | APP Inventor    | 用以製作手機 APP 來發送訊息以及獲得即時資訊。  |
| 軟體 | Tinkercad       | 用來繪畫 3D 圖，並輸出 STL 檔給 3D 列印機來影印。  |
| 軟體 | Python          | 動作辨識和讀出動作姿勢  |
| 軟體 | Arduino         | 控制 ESP32 CAM 的各個功能   |
| 軟體 | Anaconda        | 配合 TensorFlow 所支援的 Python 版本，可以利用 Anaconda 虛擬環境來產生全新的 Python 環境                  |
| 軟體 | TensorFlow      | TensorFlow 是一個機器學習框架，是一個用於機器學習的開源軟體庫，可以支援深度學習的各種演算法                              |
| 軟體 | Keras           | Keras 是一個開放原始碼，基於 Python 高階深度學習的程式庫。它已經將訓練模型的輸入層、隱藏層、輸出層，做好架構，使用者只需要加入並且填寫正確的參數。 |
| 軟體 | OpenCV          | OpenCV 是一個跨平台的電腦視覺庫，可用於開發即時的圖像處理、電腦視覺以及圖形識別程式                                    |
| 軟體 | Line Notify     | 發送警告訊息   |
| 軟體 | Firebase        | 用途:提供了實時資料庫和後端即服務。該服務為應用程式開發人員提供了一個 API 允許應用程式數據能跨客戶端同步儲存在 Firebase 的雲端空間上。      |
| 軟體 | Tinkercad       | 用來繪製 3D 圖，並輸出 STL 檔給 3D 列印機來影印。  |
| 軟體 | Altium Designer | 用來設計電路板。   |

|    |             |                               |
|----|-------------|-------------------------------|
| 軟體 | Tinkercad   | 輸入 STL 檔並繪製爆炸圖。               |
| 硬體 | ESP-32 CAM  | 用途:Wi-Fi 連接、傳送訊息到 Line Notify |
| 硬體 | 攝像頭         | 用途:拍攝姿勢                       |
| 硬體 | 伺服馬達(SG-90) | 使攝像頭能使用 APP 控制方向              |
| 硬體 | FT232RL     | USB 轉 TTL 並提供 5V 電壓。          |
| 硬體 | 電腦          | 提供軟體作業介面                      |

### 材料清單

| 類別名稱 | 材料名稱   | 單位 | 數量 | 應用說明  | 備註 |
|------|--------|----|----|-------|----|
| 硬體   | 印刷電路板  | 片  | 1  | 洗電路板  |    |
| 硬體   | 焊錫     | 卷  | 1  | 焊接    |    |
| 硬體   | ABS 樹脂 | 條  | 1  | 3D 列印 |    |

## 專題基本資料

|             |    |  |     |   |  |
|-------------|----|--|-----|---|--|
| 專 題 型 別     |    | <input type="checkbox"/> 個人型專題   |     | <input checked="" type="checkbox"/> 團隊型專題 |  |
| 專 題 性 質     |    |  |     |   |  |
| 學校名稱/科別     |    | 大安高工 電子科   |     |   |  |
| 專 題 名 稱     | 中文 | 長照輔助系統<br>人體動態姿態辨識   |     |   |  |
|             | 英文 | Long-Term Care Assistance Systems-<br>Recognition with Motion Information. |     |   |  |
| 指 導 老 師 姓 名 |    | 張顯盛  | 職 稱 | 老師  |  |
| 參 與 學 生 姓 名 |    | 組長：邱紹漣<br>組員：何冠韻<br>組員：黃奕智<br>組員：楊佳翰                                       |     |   |  |
| 專 題 執 行 日 期 |    | 自 民國 111 年 9 月 1 日 起 至 民國 112 年 1 月 25 日                                   |     |   |  |

## 參與學生簡介

|          |  |    |     |   |
|----------|--|----|-----|---|
| 姓名       | 邱紹澹  | 班級 | 子三乙 |    |
| 曾修習專業科目  | 1. 基本電學(含實習)<br>2. 電子學(含實習)<br>3. 數位邏輯(含實習)<br>4. 微處理機(含實習)<br>5. 程式設計實習 |    |     |   |
| 參與專題工作項目 | 人體動態姿態辨識、arduino、PPT 撰寫、Altium Designer                                  |    |     |   |
| 經歷簡介     | 工業電子丙級通過, 北區初賽   |    |     |   |
| 姓名       | 何冠頡  | 班級 | 子三乙 |  |
| 曾修習專業科目  | 1. 基本電學(含實習)<br>2. 電子學(含實習)<br>3. 數位邏輯(含實習)<br>4. 微處理機(含實習)<br>5. 程式設計實習 |    |     |   |
| 參與專題工作項目 | 3D 列印、書面撰寫、PPT 撰寫、爆炸圖  |    |     |   |
| 經歷簡介     | 工業電子丙級通過   |    |     |   |

|          |  |    |     |   |
|----------|--|----|-----|---|
| 姓名       | 黃奕智  | 班級 | 子三乙 |     |
| 曾修習專業科目  | 1. 基本電學(含實習)<br>2. 電子學(含實習)<br>3. 數位邏輯(含實習)<br>4. 微處理機(含實習) 5. 程式設計實習    |    |     |   |
| 參與專題工作項目 | Arduino、ESP32 CAM、Altium Designer  |    |     |   |
| 經歷簡介     | 工業電子丙級通過   |    |     |   |
| 姓名       | 楊佳翰  | 班級 | 子三乙 |  |
| 曾修習專業科目  | 1. 基本電學(含實習)<br>2. 電子學(含實習)<br>3. 數位邏輯(含實習)<br>4. 微處理機(含實習)<br>5. 程式設計實習 |    |     |   |
| 參與專題工作項目 | PPT 撰寫、App Inventor、採購、整合測試  |    |     |   |
| 經歷簡介     | 工業電子丙級通過   |    |     |   |