

中華民國第 63 屆中小學科學展覽會

作品說明書

國中組 生活與應用科學科(一)

032810

AIOT 智能防護陪伴杖

學校名稱： 高雄市立林園高級中學

作者： 國二 劉靜璇 國二 陳星妘 國二 吳芊霈	指導老師： 葉建志 許莉雅
---	-----------------------------

關鍵詞： 人工智慧、影像辨識、機器學習

摘要

隨著醫療的進步，已開發國家快速高齡化問題日益被重視，根據聯合國研究人口統計，2019 年全球平均壽命為 72.8 歲，逐年增高中，內政部統計國人 2021 年的平均壽命為 80.86 歲，與聯合國公布 2019 年全球平均壽命比較，我國男、女性平均壽命分別高於全球平均水準 7.5 歲及 9.3 歲，2022 年 6 月底全國六十五歲以上的高齡者人口數為 3,983,640 人，占 17.18%，由此可見，未來高齡者的社會人口比例越來越高，食、衣、住、行、育、樂等相關產業發展會是極為重要的一環，伴隨著人口老化趨勢，老人照護相關產業備受重視，而因為近期家裡長輩發生跌倒事故，讓我們注意到高齡者伴隨行動不便的問題，希望能藉由本次作品研究的內容，協助事故發生後相關後續處理問題，降低行動不便者因跌倒受傷的嚴重性。

壹、研究動機

世界衛生組織(WHO)於 2002 年提出「活躍老化」(active ageing)政策，主張從「健康、參與及安全」三面向提升長者之生活品質，國民健康署 104 年中老年身心社會生活狀況長期追蹤調查成果 50 歲以上中老年人平常幾乎每天看電視的百分比為 89.2%；在心理健康部分，「有憂鬱」狀況之比例隨年齡層愈高而遞增，顯見高齡長者的身心健康狀況需要多關注。

根據美國麻省理工學院年齡實驗室創辦人 Joseph F. Coughlin 所著的《銀光經濟》一書當中提到，早期在美國，熟齡族的退休夢是在環境清幽的風景區、海灘購置度假別墅養老。然而時間一長就很乏味，為了解決度假別墅生活機能不佳的問題，到了 80 年代，美國開始流行名為「村莊」的退休社區。建商在景色優美的土地上蓋起一棟棟的獨棟住宅，村莊內有高爾夫球場、游泳池、健身中心、馬球場等公共設施，入住的都是經濟條件優渥的熟齡人士，與世隔絕是「村莊」最大特色之一，走進社區，幾乎看不見 55 歲以下的人。

「村莊」表面上看起來優渥舒適，卻陸續出現新的社會問題：附近的青少年會到「村莊」偷車、破壞物品。此外，住戶同質性過高，也讓「村莊」形成對其他年齡層訪客不友善的氛圍。連住戶在家帶孫子，都可能被附近的居民嚴正抗議：「小孩止步」。

Coughlin 認為，分齡社區加深了世代對立，年輕人把富裕的年長者當成社會資源的掠奪者；年長者則活在封閉的環境裡，感到寂寞、被孩子拋棄。他指出，適合熟齡者居住的社區，未來在住戶組成上應會以混齡為主流，讓年輕人與年長者可以互相支持。

回到熟習的台灣，青壯年平日大多都需工作，青少年們在學校，很多高齡退休的老人家，都在社區或鄰里間活動，而因自身家中發生阿嬤跌倒的經驗，讓我們注意到高齡者因身體老化或疾病，常面臨行動不便的問題，如果白天沒有人陪伴在身旁，當發生跌倒或緊急意外時該如何是好？

跌倒困擾的不只是當下軟組織的受損或骨折，嚴重的需要開刀甚至臥床，後續造成的肌少症問題與併發狀況，因此，「防跌」是我們高齡化社會中，很重要的課題，而該如何選擇適當的“步行輔具”，就是我們很需要注意的課題！

在國中科技課程裡面，聽著上課老師所教的開發版、感應器、程式積木、物聯網、人工智慧等概念，心想能否結合這些工具，製作出智慧化的行動輔具，能隨時注意到高齡者的行動狀態，並且及時防護監測的裝置，達到「青銀共好，守護陪伴」的理念，是我們這次作品實踐的目標。

貳、研究調查與目的

輔助器具 (Assiistive Devices) 泛指所有能用來輔助、協助人們完成各方面行動的輔助工具或輔助科技 (Assitive Technology)，特別是用來協助特定失能人士、身心障礙者或因高齡老化導致一般生理功能篩退者。

下列是台灣輔具需求：

- 一. 短期需求：運動傷害、疾病復健所需。
- 二. 身心障礙者與其家戶需求：全國身心障礙者總人數 1,125,632 人，佔總人口數 4.8%。
- 三. 銀髮族及其家戶需求：全國 65 歲以上高齡者人數佔 11.5%。(影響每家每戶) 預估 2020 進入「高齡社會」(超過總人口數 14%); 2036 進入「超高齡社會」(超過總人口數 21%)。
- 四. 長照服務相關法規將帶動輔具及衍生創新服務全面提升。

在我們調查市面上行動不便高齡者的防護用品中，助行器是常使用的工具，如下圖，常見的使用者有：

- 一. 銀髮族:65 歲以上之老人。
- 二. 骨科疾病患者:截肢、骨折、下肢關節炎、關節置換手術患者。
- 三. 神經系統疾病患者:不完全脊椎損傷、中風、帕金森氏症、失智症患者。
- 四. 廣泛性體能衰弱患者:心肺疾病患者。



圖一、四腳助行器



圖二、

四腳拐杖同時發現以下幾項很有特色的智慧項目：

- 一. WEWALK-AIoT 智慧拐杖，可望引導盲人「看」世界，這種智慧拐杖內建超音波感測器，可以感應到 160cm 高的障礙物，還帶有震動和聲音警示裝置，可以透過藍牙與智慧型手機連接。它還可以與 Google Maps 連接進行導航，電池續航達 5 小時，目前支援的語言包括英語、土耳其語、阿拉伯語和葡萄牙語。

WEWALK 拐杖採用哪一款微處理器作為運算處理主控晶片，目前還不得而知，據推測可能是 Arm Cortex-M 核心的微處理器。現在量產商用的 Cortex-M 處理器僅能提供有限的震動檢測、感測器融合及關鍵字檢測等功能，要想達到盲人導航所需要的要求，就需要處理器具有物體檢測和異常檢測功能，最好還有語音辨識、手勢檢測，甚至人體徵象認知等 AI 人工智慧功能。



圖三、WEWALK 拐杖



圖四、WEWALK 拐杖功能

二. 「智能盲人虛擬手杖」是一款手握遙控感測器，十字形紅外線雷射與接收，作為距離定位障礙物感測，感測路上障礙物的距離與大小。並加上鏡頭之影像辨識技術系統，配合盲人 RFID 感測系統，利用預先埋設 RFID 導盲磚式人行道+障礙物 RFID 貼紙，配合鏡頭的影像辨識系統，即可以讓盲人更清楚知道所碰觸物體的名稱與實際大小的狀態。加裝震動模擬器模擬訊號回饋之碰觸物體震動，來模擬辨識障礙物。系統也利用 GPS 衛星定位，提供所在地周遭生活資訊，大眾運輸資訊與智慧城市資訊網連線。虛擬手杖並提供自動 AI 學習該盲人生活習慣，例如個人使用習慣，身高與常用的感知長度，或生活環境區域之所有行為作出智慧校正；因此是一款專門為盲人生活打理設計的虛擬手杖。



圖五、盲人虛擬手杖



圖六、使用示意






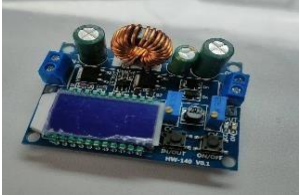

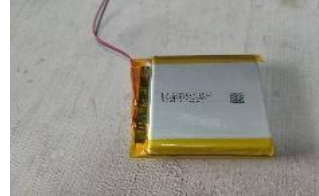




在經過文獻探討，瞭解目前市面上助行器與幾項智慧拐杖的功能後，以高齡行動不便者的行為模式來做思考，過重的裝置或過於複雜的操作模式都可能為長輩造成負擔，而裝置能智慧化結合平常助行器的使用習性，並能簡單操作是我們設計的方向，依照防摔、緊急通報與位置追蹤的三大要點，我們的 AIOT 防護陪伴杖也有了大致清楚的雛形，研究目的可以歸納成以下幾項：

- 一、以 AI 人工智慧開發套件，結合鏡頭依人體動作的影像辨識，做主動式偵測跌倒、痙攣等突發性狀況。
- 二、利用三軸感測器，做被動式偵測，當助行器瞬間傾倒時，發送求救警報，同時外部增加照明燈條，減少因天黑或陰暗處，視線燈光不明造成行走的危險性，並設置緊急呼叫按鈕。
- 三、搭配 IOT 物聯網，將上述的即時資訊做遠端行動裝置或醫療系統的通報。
- 四、利用 GPS 晶片做及時位置追蹤，如緊急情況、迷路走失的協尋定位，並結合 Linebot 做座標位置查詢。

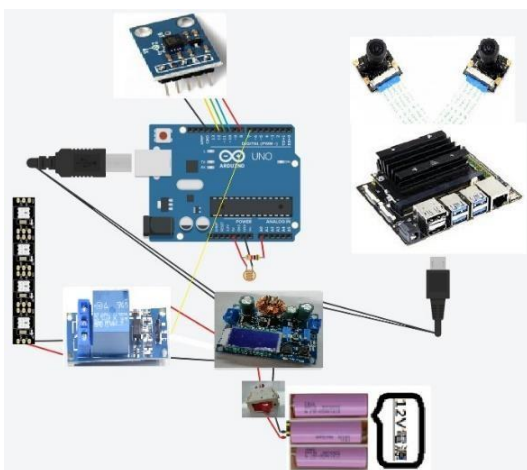
參、研究器材

			
雷射切割機	雷切軟體	Arduino IDE 編輯器	PuTTY 終端機
			
3D 列印機	3D 建模軟體	Python 編輯器	Circuit EDIT

圖七、使用機具與軟體

			
視訊鏡頭*2	jetson nano 人工智慧開發套件	Jetson Nano 4G GPS 擴充模組	三軸加速感測器
			
NodeMcu	電源供應模組	按壓開關	鋰電池
			
鋁燈條	啟動開關	光敏電阻	雷雕木板材

圖八、使用硬體材料



圖九、電路配置圖

肆、研究過程

一、以 AI 人工智慧開發套件，結合鏡頭依人體動作的影像辨識，做主動式偵測跌倒、痙攣等突發性狀況。

(一)首先建置 JetsonNano 環境工具，燒錄 Image，要下載 image 檔，下載網址：JetsonNano 2GB Developer Kit。其 JetPack SDK 預設版本為 4.4.1。

(二)接著燒錄 image 至 SD Card，建議 SD Card 為 64GB 以上，在燒錄前需要將 SDCard 格式化後，再做燒錄的動作，燒錄完後就可以將 JetsonNano2GB 開機，記得一定要使用 5V-3A 的插頭，不然會無法開機成功！

```
COM4 - PuTTY
Ubuntu 18.04.5 LTS aia ttyGS0
aia login: jetson
Password:
Last login: 一 十一 2 14:05:34 CST 2020 on ttyGS0
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.9.140-tegra aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'urminimize' command.

41 packages can be updated.
25 updates are security updates.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

jetson@aia:~$
```



(三)設定 VNC

可參考官方 Setup VNC server

```
# 更新一下套件
$ sudo apt-get update

$ sudo apt-get upgrade

# 設定
$ mkdir -p ~/.config/autostart

$ cp /usr/share/applications/vino-server.desktop
~/.config/autostart/.
```

(四)設置 VNC 登入密碼，更改 thepassword

```
$ gsettings set org.gnome.Vino authentication-methods "[ 'vnc' ]"

$ gsettings set org.gnome.Vino vnc-password $(echo -n
'thepassword' | base64)
```

(五)Install Pytorch

torch 安裝可參考：NVIDIA網站PyTorch for Jetson文章

```
# 安裝需要的依賴
$ sudo apt-get install python3-pip libopenblas-base libopenmpi-dev

$ pip3 install Cython

# 下載 torch 並安裝
$ wget
https://nvidia.box.com/shared/static/9eptse6jyly1ggt9axbj2yrmj6pbarc
.whl -O torch-1.6.0-cp36-cp36m-linux_aarch64.whl

$ pip3 install numpy torch-1.6.0-cp36-cp36m-linux_aarch64.whl
```

(六)安裝 torchvision

```
# 安裝需要的依賴
$ pip3 install setuptools

$ sudo apt-get install libjpeg-dev libfreetype6-dev pkg-config
libpng-dev libhdf5-serial-dev hdf5-tools libhdf5-dev

# 下載 torchvision 並安裝
$ git clone --branch release/0.7 https://github.com/pytorch/vision
torchvision

$ cd torchvision

$ sudo python3 setup.py install
```

(七)Install opencv-python package

```
$ pip3 install matplotlib

$ pip3 install opencv-python
```

若遇到 error:ModuleNotFoundError:Nomodulename 'skbuild'，則先更新 pip 後，再重安裝 opencv-python

```
joy@joy-desktop:~/torchvision$ pip3 install opencv-python
Collecting opencv-python
  Downloading https://files.pythonhosted.org/packages/bb/08/9dbc183a3ac6baa95fab
f749ddb531bd26256edfff5b6c2195eca26258e9/opencv-python-4.5.1.48.tar.gz (88.3MB)
100% |████████████████████████████████████████| 88.3MB 3.9kB/s
Complete output from command python setup.py egg_info:
Traceback (most recent call last):
  File "<string>", line 1, in <module>
  File "/tmp/pip-build-s720drbs/opencv-python/setup.py", line 10, in <module>
>
    import skbuild
ModuleNotFoundError: No module named 'skbuild'

-----
Command "python setup.py egg_info" failed with error code 1 in /tmp/pip-build-s7
20drbs/opencv-python/
```

(八)利用 NVIDIA DeepStream 建立跌倒偵測估計應用程式，可以預測人是否跌倒而需要就醫。使用 DeepStream 部署姿態估計模型，在 \$DEEPSTREAM_DIR\$/sources/sample_apps 上建構 DeepStream SDK 隨附的範例，以簡化流程。將工作流程分成六個主要步驟：

步驟一：編寫模型需要的後處理程式碼

NVIDIA-AI-IOT/deepstream_pose_estimation GitHub 儲存庫提供了此應用程式的原始碼，範例 C++ 應用程式附帶了以 DeepStream 啟用姿態估計需要的所有支援資料結構和後處理。

步驟二：下載人體姿態估計模型及轉換成 ONNX

下載 TRTPose 模型的 PyTorch 模型權重。該儲存庫是由兩個模型組成：一個在 ResNet 骨幹上，另一個在較密集的 DenseNet 骨幹上。

```
./export_for_isaac.py --input_checkpoint resnet18_baseline_att_224x224_A_epoch_249.pth
```

然後，複製 TRTPose 儲存庫，並前往容器內包含匯出指令碼的資料夾。將模型的 PyTorch 權重放入此目錄中，以匯出至 ONNX。

```
$ git clone https://github.com/NVIDIA-AI-IOT/trt_pose.git  
  
$ cd trt_pose/trt_pose/utils/
```

最後轉換模型，如下所示：

```
./export_for_isaac.py --input_checkpoint model_weights.pth
```

步驟三：取代 DeepStream 安裝目錄中的 OSD 函式庫

如應用程式的原始碼中所述，已針對螢幕顯示進行了一些變更，以支援此人體姿態估計應用程式。由於模型有時候會輸出視訊緩衝區框架以外的特徵點，因此我們已經在儲存庫中進行這些變更，為 Jetson 和 x86 平台提供已編譯 .so 檔案。根據您的平台，將 \$DEEPSTREAM_DIR\$/lib/ 中的現有 libnvds_osd.so OSD 函式庫更換為儲存庫中的版本。

步驟四：編輯 DeepStream 配置檔

DeepStream 配置檔為頂層配置檔，可以配置屬性，例如推論精度、分配給專案的工作空間大小、實際模型位置，在編譯之前，以參數形式，將 ONNX 檔案的位置提供給配置檔，以使應用程式知道從哪一個模型開始進行推論。

```
onnx-file=pose_estimation.onnx
```


在初次執行應用程式時，DeepStream 會產生 TensorRT 引擎檔案，可能需要幾分鐘的時間。在產生引擎檔案之後，即可提供 TensorRT 引擎檔案的路徑，以使 DeepStream 無須在後續執行時重新產生引擎檔案。

```
model-engine-file=pose_estimation.onnx_b1_gpu0_fp16.engine
```

步驟五：編輯 makefile 以加入平台專用的組建旗標

在編譯應用程式時，需要可以匯集所有相依性及編譯應用程式的 g++ makefile。某些相依性的位置是取決於編譯應用程式針對的目標系統。開啟 makefile，確保已正確設定

DeepStream 版本：

確保 makefile 指向正確的 lib 目錄。於此情況下，lib 是位於 \$DEEPSTREAM_DIR/lib/。

```
LIB_INSTALL_DIR?=/opt/nvidia/deepstream/deepstream-$(NVDS_VERSION)/lib/
```

相較於 x86 Linux 主機，以 L4T 為基礎之電腦在應用程式中建構 GStreamer 工作流程的方式稍微不同。在 DeepStream 應用程式中導入稱為 PLATFORM_TEGRA 的旗標，並使用簡單的 if 陳述式滿足平台的要求。makefile 在開始編譯流程之前是使用 shell 自動辨識目標平台，如下所示：

```
TARGET_DEVICE = $(shell gcc -dumpmachine | cut -f1 -d -)
```

步驟六：編譯和執行 DeepStream 應用程式

在正確設定 DeepStream 配置檔和 makefile 之後，即可編譯與測試應用程式。開啟新的終端機，並前往應用程式目錄：

```
$ cd $DEEPSTREAM_DIR/sources/apps /sample_apps/deepstream_pose_estimation
```

在 NVIDIA Jetson Xavier NX 上，編譯流程大約需要 1 分鐘的時間，應用程式的輸出，在中儲存為 Pose_Estimation.mp4。

(九)人體姿態估計模型的後處理－跌倒偵測估計

深入探討如何利用 DeepStream 為姿態估計模型進行後處理，以及建立視覺化成品，TRTPose 是採用由下而上的方法進行姿態估計，模型會先偵測畫格中每一個身體部位的關鍵點，然後判斷該畫格中的哪些部位是屬於哪個人。

步驟一：取得熱圖以從模型產生部位類同區域

這是應用程式的推論步驟。剖析模型輸出需要的參數，是在應用程式的

`parse_objects_from_tensor_meta` 方法中進行配置。此方法同時負責叫用姿態估計工作流程中的所有其他輔助方法，以及輸出最終結果。

```
void *cmap_data = tensor_meta->out_buf_ptrs_host[0];
NvDsInferDims &cmap_dims = tensor_meta->output_layers_info[0].inferDims;
void *paf_data = tensor_meta->out_buf_ptrs_host[1];
NvDsInferDims &paf_dims = tensor_meta->output_layers_info[1].inferDims;

parse_objects_from_tensor_meta (NvDsInferTensorMeta *tensor_meta)
```

模型輸出有兩個步驟。第一個步驟是針對在畫格中預測的各個身體部位，產生置信度圖。這是由下而上的方法，第二個步驟是預測各個身體部位的關聯度，以分配給特定的人。

步驟二：使用非最大值抑制

提供熱圖之原始輸出張量包含許多有用的資料，但是，必須使用熱圖中的最大置信度區域，並忽略所有其他不確定區域，才能從輸出中擷取身體部位。其可以透過使用非最大值抑制，尋找輸出中的所有局部最大值的方式完成。

在 `find_peaks` 函式中，定義 `window_size` 的值。這是表示在尋找最大值或「峰值」時，需要一次考量的像素數量。在找出峰值之後，於內迴圈中使用 `is_peak` 布林值進行標記，並將該最大值分配給視窗的中心像素。之後重複此操作，直至涵蓋整個畫格。

```
find_peaks (Vec1D < int >&counts_out, Vec3D < int >&peaks_out, void
            *cmap_data, NvDsInferDims & cmap_dims, float threshold, int
            window_size, int max_count)
```

最後，僅保留畫格中的峰值，並抑制所有其他非最大像素。此步驟是以 `refine_peaks` 方法處理。`refine_peaks` 方法同時負責考量與各個身體部位有關的置信度分數權重，將較早產生的峰值正規化。此流程之後剩下的唯一值，是表示任一身體部位的最大置信度區域。

步驟三：建立二部圖以將偵測到的身體部位分配給畫格中的個人

在找出與細化峰值之後，必須將這些最大置信度區域分配給畫格中的個人。此時開始尋找局部化身體部位之間的關聯，以及開始針對各個畫格建構每一個人的骨骼。

在畫格中有多個人的情況下，此步驟格外重要。假設輸入的影片中有兩個人。模型將會偵測兩個右肩候選峰值及兩個右臂候選峰值。之後可針對偵測到的兩個肩部，在偵測到的兩個右臂之間建立連接。建立圖形，使兩個肩部和兩個右臂分別代表圖形中的頂點。從兩個肩部到兩個右臂以及相反方向繪製邊線。

步驟四：為二部圖中的所有邊線分配權重

若想要在圖中之節點之間找出正確的連接，則必須先為圖中的每一條邊線分配權重。之後，可使用 Munkres 圖形配對方法解決問題。步驟 5 將詳細探討 Munkres 方法。

分配權重是以 `paf_score_graph` 方法處理。

```
Vec3D
paf_score_graph (void *paf_data, NvDsInferDims &paf_dims,
                 Vec2D < int >&topology, Vec1D < int >&counts,
                 Vec3D < float >&peaks, int num_integral_samples)
```

步驟五：使用匈牙利演算法解決分配問題

現在已具有表示所有偵測到之身體部位彼此間所有可能之連接的二部圖。接下來則是解決圖形配對問題。目標是將圖形總分最大化。使用新計算的邊線分數，解決此問題。

在程式碼中，是由 `munkres_algorithm` 方法負責建立對圖輔助資料結構，以表示問題。

```
void munkres_algorithm (Vec2D < float >&cost_graph, PairGraph &
                       star_graph, int nrows,int ncols)
```

最後，是由 `assignment` 方法負責實際分配偵測到之身體部位之間的連接。在此方法之主要迴圈的最後，將會忽略實際上不成對的連接。僅剩下可以建立有意義之連接的配對候選。

```
Vec3D
assignment (Vec3D < float >&score_graph,
            Vec2D < int >&topology, Vec1D < int >&counts,
            Float score_threshold, int max_count)
```

步驟六：連接所有身體部位及形成人體骨骼

此時，已偵測到所有的身體部位，以及找出他們之間的關係。之後將所有偵測到的部位連接，並形成 2D 人體姿態。這是使用 connect_parts 方法完成。

```
Vec2D  
connect_parts (Vec3D < int >&connections, Vec2D < int >&topology,  
              Vec1D < int >&counts, int max_count)
```

由於已知兩個元素對之間的關係，因此僅需要在兩對之間找出具有相同身體部位的元素。之後，即可推斷他們是屬於同一個人。重複此程序，直至無任何未分配的身體部位對。

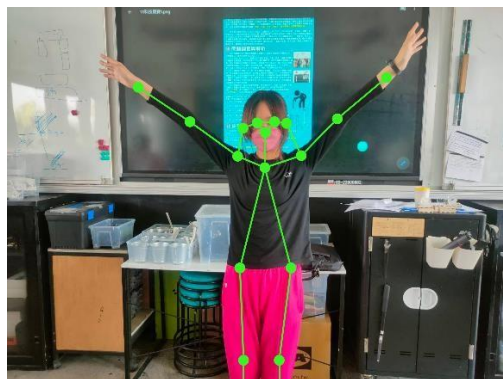
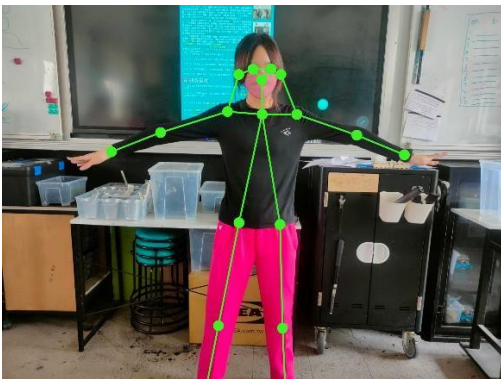
步驟七：設定 OSD 以繪製輸出

若想要將模型輸出視覺化，請設定 DeepStream 螢幕顯示 (on-screen-display, OSD)，將找到正規化峰值的區域繪製成圓圈，並繪製已分配身體部位的向量。

使用 nvds_acquire_display_meta_from_pool 方法取得目前的中繼資料。然後，使用 nvds_add_display_meta_to_frame 方法加入您的中繼資料。應用程式中的 create_display_meta 方法會示範如何執行此操作。

```
static void  
create_display_meta(Vec2D &objects, Vec3D &normalized_peaks,  
                  NvDsFrameMeta *frame_meta, int frame_width,  
                  int frame_height)
```

將會針對想要繪製在 OSD 上的每個正規化峰值，產生中繼資料。最後，由 OSD 元素負責將最終輸出實際繪製在原始影片上。



利用 DeepStream 部署姿態估計模型，有助於簡化整個工作流程。將 TensorRT 姿態估計模型與 DeepStream 結合，可以實現人體姿態估計的即時多資料流使用案例。無須擔心另外將系統資源最佳化，以進行解碼、推論、在影片上繪製或儲存輸出的問題。僅需要為模型編寫後處理程式碼、指定如何布置 GStreamer 工作流程，以及建立簡單的配置檔。

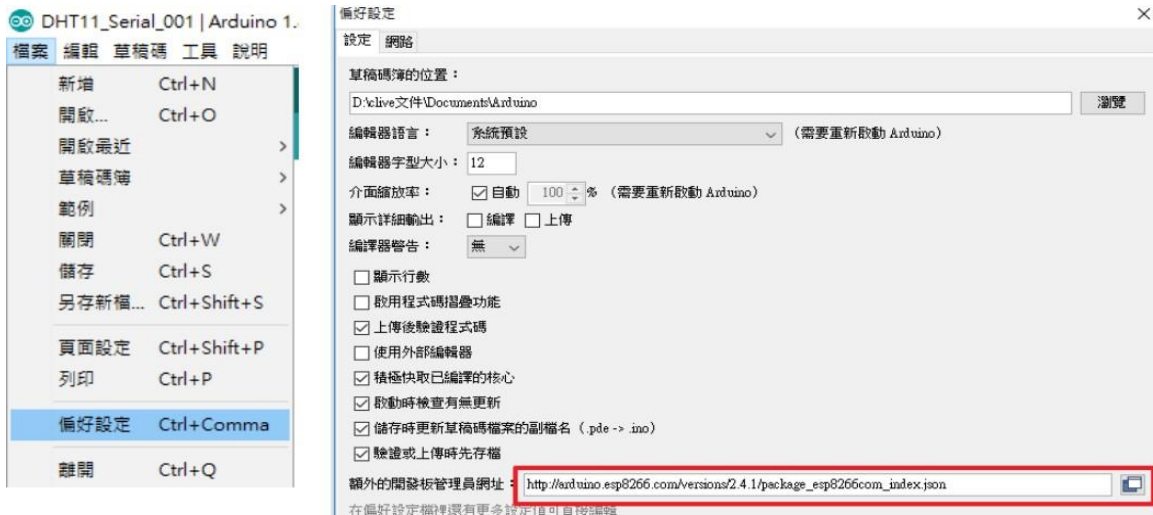
可以使用簡單的旗標和 if 陳述式，使應用程式在 x86 與裝置之間跨平台，並根據需求及可用平台，調整資料。

二、利用三軸感測器，做被動式偵測，當助行器瞬間傾倒時，發送求救警報，同時外部增加照明燈條，減少因天黑或陰暗處，視線燈光不明造成行走的危險性，並設置緊急呼叫按鈕。

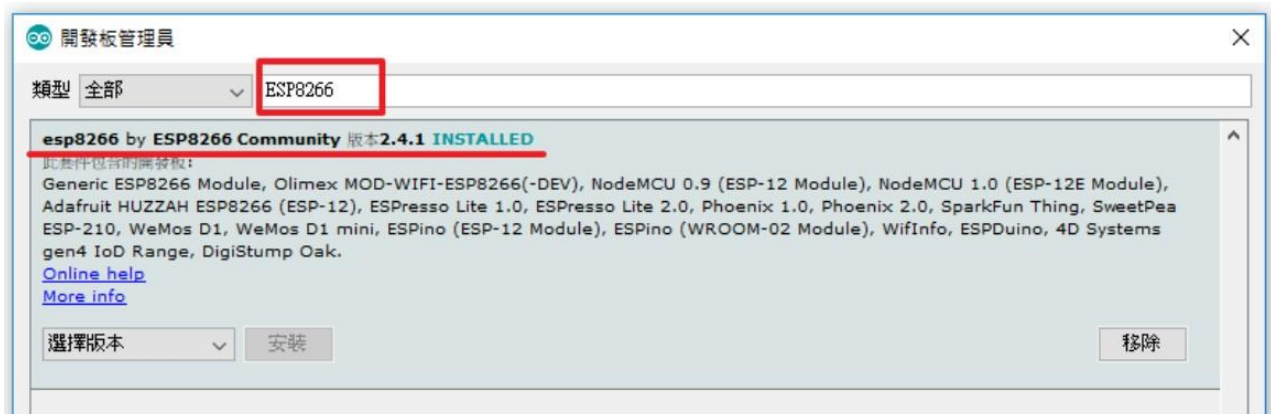
(一)在 Arduino IDE 設定 NodeMCU

打開 ARDUINO IDE 軟體，檔案==>偏好設定，在下方額外的開發板管理員

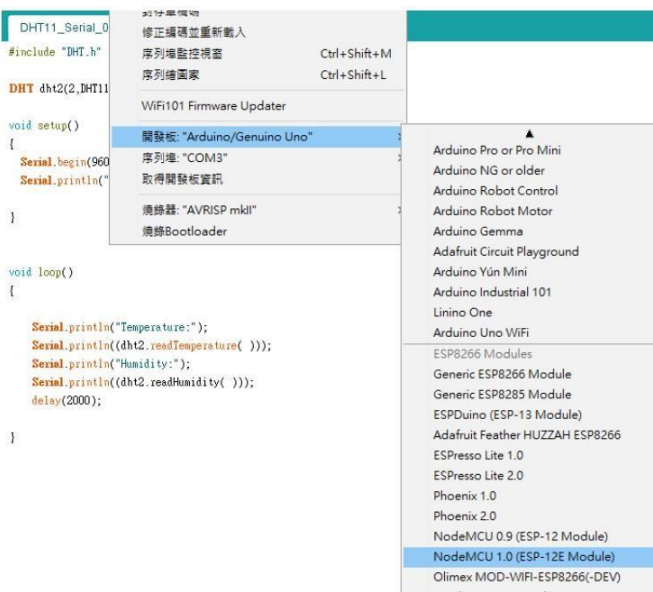
網址：輸入 http://arduino.esp8266.com/stable/package_esp8266com_index.json



接著到工具==>開發板==>開發板管理員，搜尋"ESP8266" 並安裝



選擇開發板：NodeMCU 1.0(ESP-12E Module)



(二) 以三軸感測器做被動式偵測，讀取三軸感測器數值，如果 1 秒內有做 Roll 或 Pitch 旋轉運動超過 60 度時，代表助行器可能傾倒，發送 2 到 GOOGLE 試算表欄位二。

(三) 光敏電阻輸入數值小於 100 時，啟動繼電器開啟照明燈條。

(四) 觸發紅色按鈕時發送 3 到 GOOGLE 試算表欄位三，觸發綠色按鈕時發送 4 到 GOOGLE 試算表欄位四。

```

PIR-IR | Arduino 1.8.13
檔案 編輯 草稿碼 工具 說明

PIR-IR

#include <Wire.h>
#include <Adafruit_AMG88xx.h>

Adafruit_AMG88xx amg;

#include <LiquidCrystal_I2C.h>
// 0x27是I2C模組的地址，16表示有16個位元，2表示總共有2個
LiquidCrystal_I2C lcd(0x27,16,2);

//人體紅外線移動偵測設定
const int led=9;
const int PIRout=8;
//偵測到物體移動時設定
const int START=13;

void setup() {
  //溫度感測器初始化
  Serial.begin(9600);
  Serial.println(F("AMG88xx test"));

  bool status;

  // default settings
  status = amg.begin();
  if (!status) {
    Serial.println("Could not find a valid AMG88xx sensor, check wiring!");
    while (1);
  }

  Serial.println("== Thermistor Test ==");

  Serial.println();

  delay(100); // let sensor boot up

  //LCD初始化
  lcd.begin();
  lcd.backlight();
  //關閉背光
  //設定第一行游標位置
  lcd.setCursor(0,0);
  lcd.clear();
}

void loop() {
  //人體紅外線移動
  int val=digitalRead(PIRout); //讀取 PIR 輸出
  if (val==HIGH) { //PIR 有偵測到物體移動時： 鋸齒燈亮3秒
    digitalWrite(led,HIGH);
    delay(3000);
    digitalWrite(led,LOW);
    delay(1000);
  }
  else { //PIR 沒有偵測到物體移動： 鋸齒燈暗
    digitalWrite(led,LOW);
  }

  int button=digitalRead(START); //讀取按鈕輸入
  if (val==HIGH) { //按鈕按下後啟動溫度測量
    //抓取溫度數值
    Serial.print("Thermistor Temperature = ");
    Serial.print(amg.readThermistor());
    Serial.println(" °C");

    Serial.println();

    //delay a second
    delay(1000);

    //將溫度顯示在LCD上
    if (amg.readThermistor()==HIGH) { //按鈕按下後啟動溫度測量
      lcd.clear();
      lcd.print("temperature:amg.readThermistor()");
      lcd.setCursor(0, 1); //設定游標位置 (字,行)
      lcd.print(" ok - welcome ");
      delay(3000);
    }

    //如果溫度低於37.5度顯示OK語彙，如果大於37.5度顯示請去身體檢查
    if (amg.readThermistor() < 37.5) {
      lcd.clear();
      lcd.setCursor(0, 1); //設定游標位置 (字,行)
      lcd.print(" ok - welcome ");
    } else {
      lcd.clear();
      lcd.setCursor(0, 1); //設定游標位置 (字,行)
      lcd.print("physical checkup");
    }
    delay(5000);
  }
}

```

儲存空間

某稿碼使用了 9302 bytes (32%) 的程序儲存空間，上限為 28672 bytes。
全球變數使用了 547 bytes (21%) 的動態記憶體，剩餘 2013 bytes 給區域變數，上限為 2560 bytes。

三、搭配 IOT 物聯網，將上述的即時資訊做遠端行動裝置或醫療系統的通報。

(一) 首先建立帳號登入 IoT : Temboo。

Code the Internet of Everything

Get more power and flexibility on any IoT topology with Temboo's code virtualization technology and software stack.

Sensors
Actuators

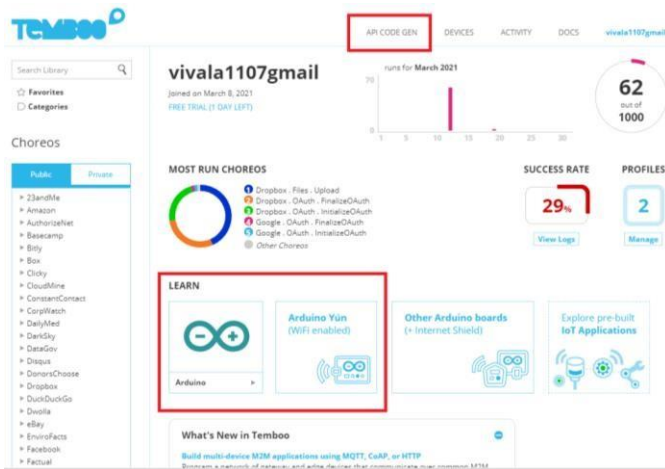
MCU
Temboo MCU Library

Temboo Cloud

Run the Internet of Things at Scale

Mesh networks with multiple combinations of MCUs, gateways, and devices can be programmed and updated in bulk with our REST API and run reliably at any scale with our enterprise class backend.

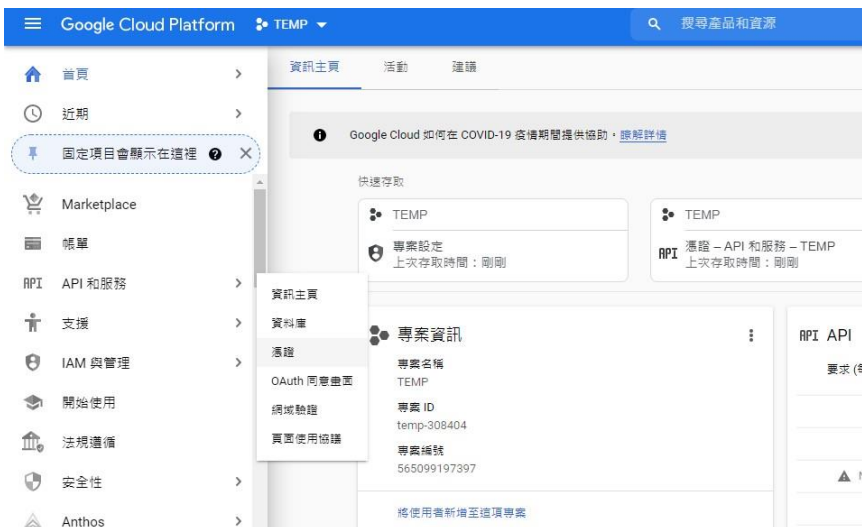
(二)點選 API CODE GEN，裝置選擇 MCU 後，於左側選擇 GOOGLE 的 Oauth 申請開放授權 API 。



(三)登入到 Google 的開發者網站申請 API 權限，先建立新專案。



點選左邊 API 服務的憑證。



點選建立憑證，先創建一組 API 金鑰。



(四)再點選一次建立憑證，創建 OAuth 用戶端 ID。



應用程式類型選「網頁應用程式」，設定名稱後，於重新導向 URI 中輸入。

https:// 自己 Temboo 的帳號.temboolive.com/callback/google



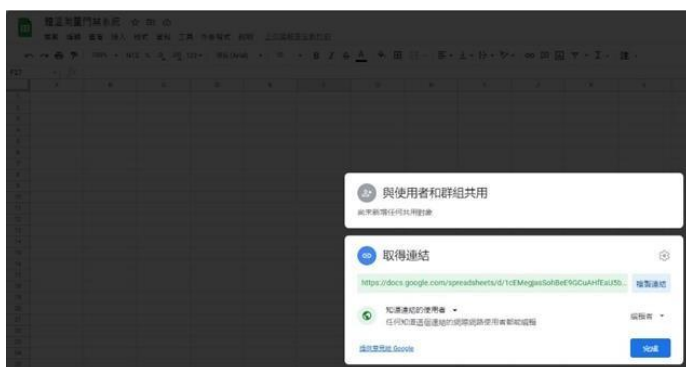
就會得到「用戶端 ID」、「用戶端密碼」，接下來會一直使用。



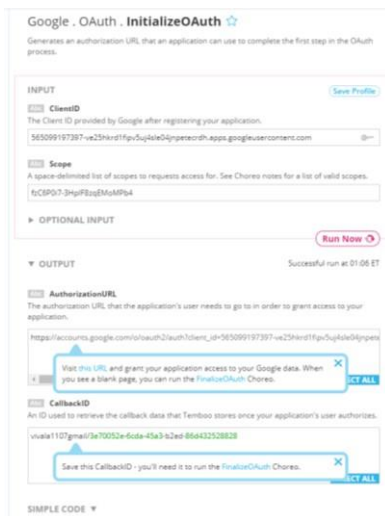
(五)返回到 Temboo 選擇 Google 中的試算表 Spreadsheets，點選 Google . OAuth . InitializeOAuth。



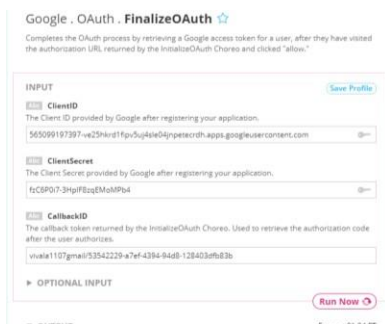
(六)建立一個儲存跌倒數據的 google 試算表並設定成知道連結即可編輯共用。



(七)在 InitializeOAuth 輸入 google 的「用戶端 ID」、「用戶端密碼」與儲存跌倒數據的 google 試算表(設定成編輯共用)網址後執行 RUN，得到 AuthorizationURL 與 CallbackID。



(八)再到 FinalizeOAuth 輸入「用戶端 ID」、「用戶端密碼」與 CallbackID 執行 RUN 後即可得到 AccessToken。

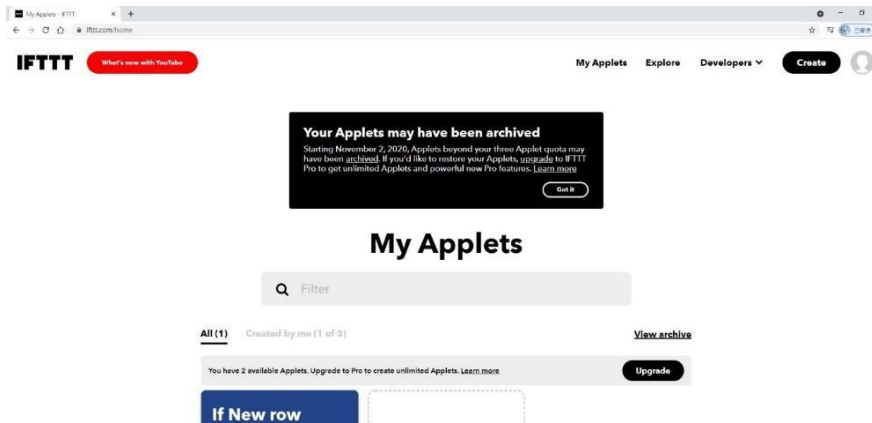


(九)於下方系統將產生的 Arduino 程式碼下載好，網頁已將 google 的「用戶端 ID」、「用戶端密碼」與 FinalizeOAuth 的 AccessToken 自動填入，再來只要用 Arduino IDE 打開下載的程式碼，並將原本寫好的硬體執行程式放入稍作修改後上傳到 MCU 板，這樣偵測的每筆數據就會依序添加入 google 試算表的欄位裡面，利用這個方式，依序將主動式跌倒偵測訊號紀錄為 1 發送到 GOOGLE 試算表欄位一、被動式跌倒偵測訊號紀錄為 2 發送到 google 試算表欄位二、觸發紅色按鈕時發送 3 到 GOOGLE 試算表欄位三、觸發綠色按鈕時發送 4 到 google 試算表欄位四。

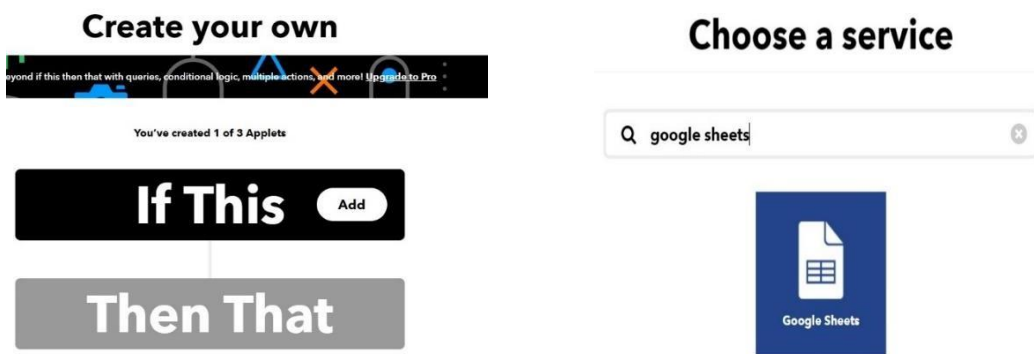
(十)將雲端資料即時回傳到緊急連絡人的行動裝置上顯示。

IFTTT 是 If this, then that 的縮寫，類似程式語言的“若 XX 進行 YY 行為，就執行 ZZ 動作”，可以將不同的 App、連網裝置和軟體服務整合在一起，然後讓支援 IFTTT 的某服務（或 App、連網裝置）去觸發另一個服務（或 App、連網裝置）。

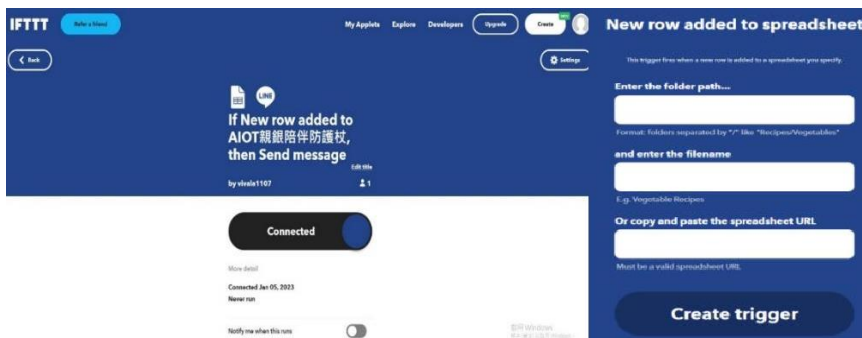
(十一)先創建帳號並登入 IFTTT 網頁後，在 My Applet 中點選 Create 建立一個新連動。



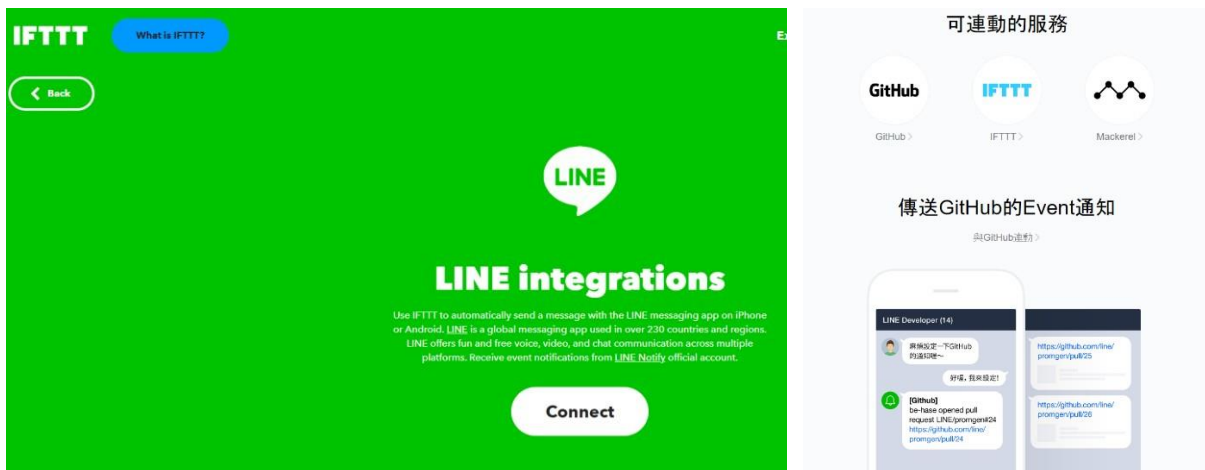
(十二)先增加 if this 建立觸發機制，以 google 試算表為觸發方式。



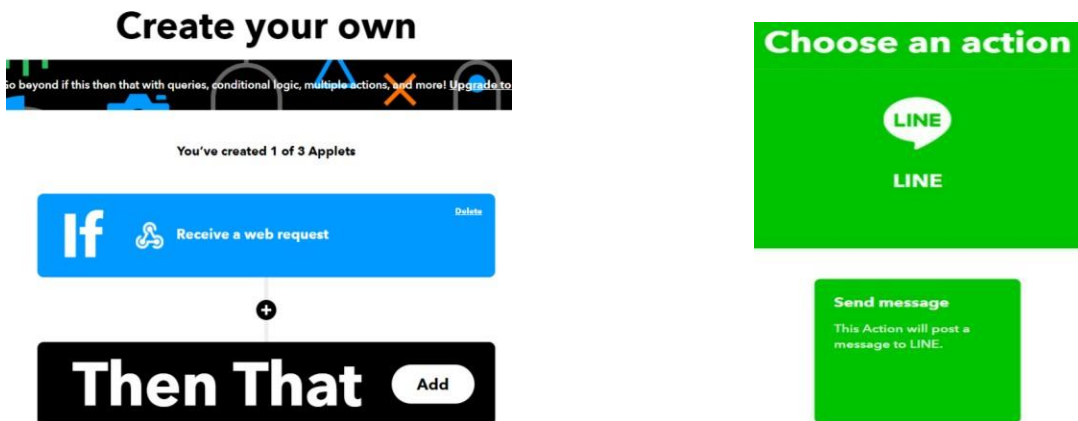
(十三)選擇增加新數據為觸發條件，輸入儲存 MCU 數據的試算表名稱與網址鏈結後，即可建立觸發機制。



(十四)接下來建立連動裝置，我們登入 line notify 做緊急通報的訊息通知。



(十五)選擇將 line notify 與 IFTTT 作連動



填入傳遞人或群組的名稱，訊息通知的文字內容，即可觸發動作，這樣每次有跌倒或緊急通知新數據存入 GOOGLE 試算表時，就會透過 IFTTT 的即時連動機制，傳遞到手機遠端的 LINE 的通知訊息裡面。



四、利用 GPS 晶片做及時位置追蹤，如緊急情況、迷路走失的協尋定位，並結合 Linebot 做座標位置查詢。

(一)利用 Jetson Nano 4G GPS 擴充模組 SIM7600G-H 4G 獲取經緯度等位置信息，由於 GPS 室內不穩定，將模組或者天線放到陽台或窗戶旁，或者直接在戶外進行實驗，插上 GPS 天線，在正常情況下上電需要等待約 1 分鐘。

(二)通過 minicom 打開 ttyUSB2 或 ttyTHS1

```
sudo minicom -D /dev/ttyUSB2
```

(三)在 minicom 發送以下指令打開 GPS：

```
AT+CGPS=1
```

(四)用以指令獲取經緯度等位置信息：

```
AT+CGPSINFO
```

(五)打開 NEMA COM 獲取詳細定位信息

```
sudo minicom -D /dev/ttyUSB1
```

(六)運行 python3 程式後獲取的經緯度複製到 GOOGLE 地圖 API

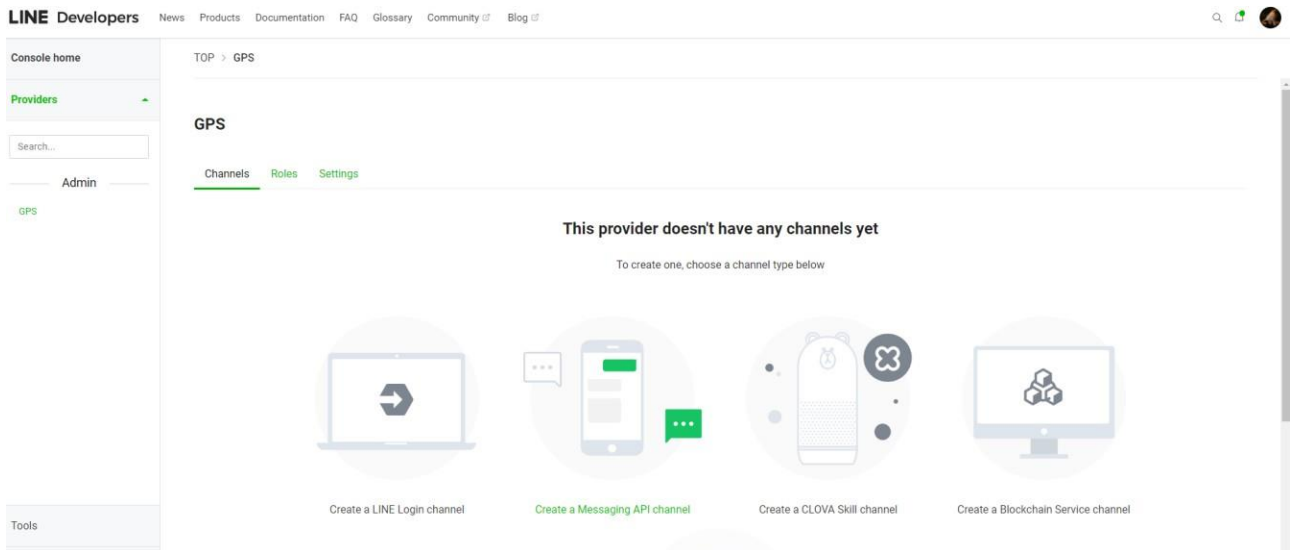
```
wget https://www.waveshare.net/w/upload/b/b2/SIM7600X\_GNSS.zip
sudo apt-get install unzip
unzip SIM7600X_GNSS.zip
sudo chmod 777 SIM7600X_GNSS.py
sudo python3 SIM7600X_GNSS.py
```

取得座標為：

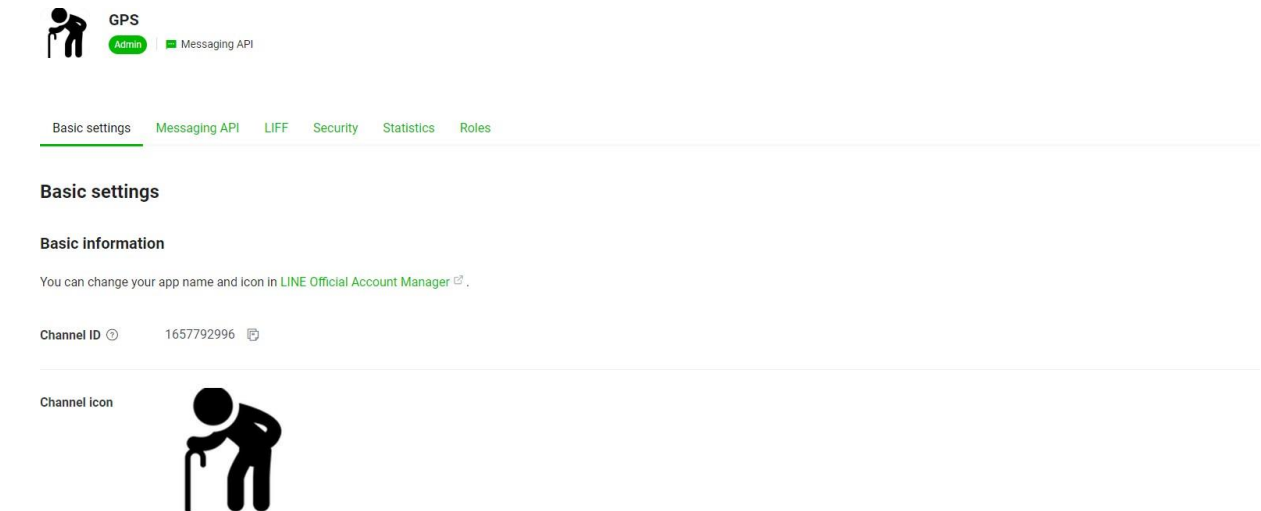
22.513559225899304, 120.39512893801079

(七) Line chatbot 座標經緯度查詢

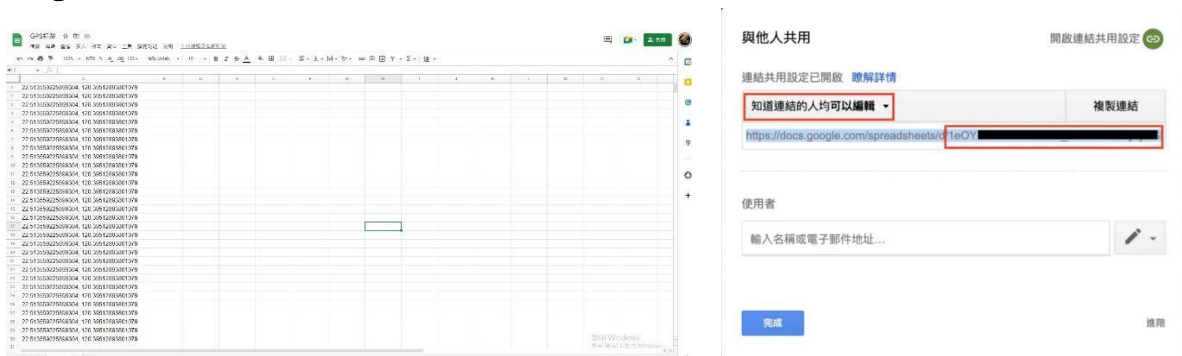
在開始之前，要去 Line Developers 申請一個 Line Messaging API 的帳號。



(八)建立完之後，要做一些簡單的基本設定，像是名稱，圖片，email ...等等，基本上只會回 default message 的 line chatbot 就已經建好了。



(九)Google Sheet API，在講 apps script 之前，先來準備我們需要儲存座標的資料庫 — Google 試算表。



金鑰 key 就會是網址中間那段。

https://docs.google.com/spreadsheets/d/{SHEET_ID}/edit#gid=0

(十)Google Apps Script

建立一個 Script 非常容易，只要開啟 Google drive 雲端硬碟，就可以直接新增建立了。使用剛剛從 Google 試算表連結中拿到的 key 來當作 SHEET_ID

```
var spreadsheet = SpreadsheetApp.openById(SHEET_ID);
```

(十一)接著使用 spreadsheet instance ，透過 SHEET_NAME 來存取內部的 sheets

```
var sheetInstance = spreadsheet.getSheetByName(SHEET_NAME);
```

(十二)使用 getSheetValues 可以從指定的 sheet 拿出資料的範圍

```
/**
 * @param {number} startRow 開始 row 的位置
 * @param {number} startColumn 開始 column 的位置
 * @param {number} numRows 目標要抓多少 row 數量
 * @param {number} numColumns 目標要抓多少 column 數量
 * @return {Object[][]}
 */
sheetInstance.getSheetValues(
  startRow,
  startColumn,
  numRows,
  numColumns
);
```

(十三)抓出有資料的最後一個 row

使用 getSheetValues 來抓出目前 sheet 中，最後一列 row 的 index

```
/**
 * @return {number}
 */
sheetInstance.getLastRow();
```

(十四)連接 Line Messaging API

最後一步就是要跟 chatbot 做連接了



(十五)Line Messaging API 設定

接著回到 Line Developers 你的 Line Messaging API 頁面，往下找到 Messaging settings 這個 section

Messaging settings

Channel access token (long-lived) ⓘ

[Redacted Token] Issue

Use webhooks ⓘ

Enabled Edit

Webhook URL Requires SSL ⓘ

[Redacted URL] Verify Edit

Allow bot to join group chats ⓘ

Enabled Set message ⓘ

(十六)在 Google Apps Script 與 Line API 溝通

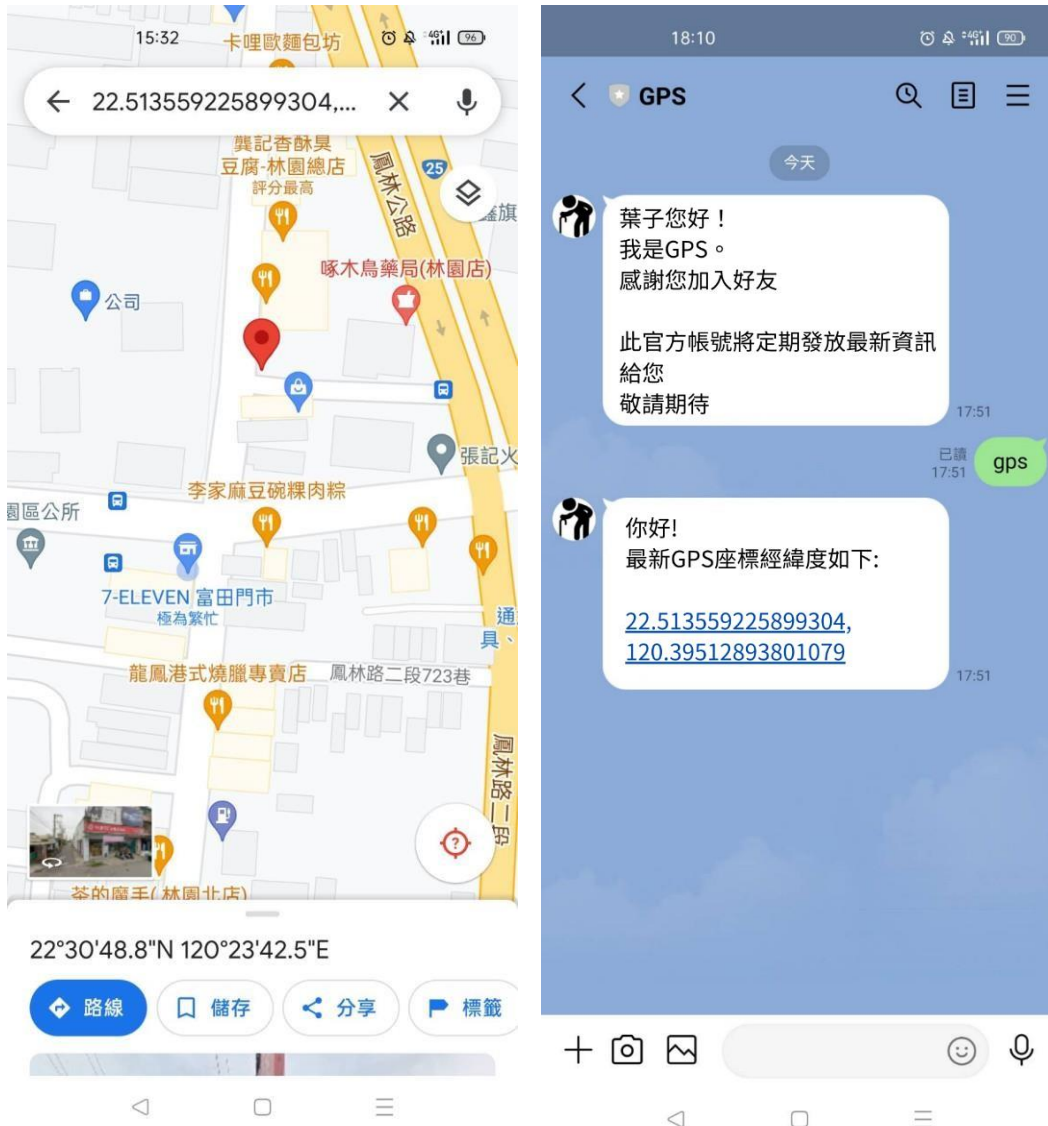
最後就是要在 Google Apps Script 中跟 Line API 做溝通了，以下是 Webhook event objects format，從這個 object 中，當你收到使用者訊息的時候，就可以相對應作出反應了，做完邏輯處理，決定好要回復內容之後，我們使用 Apps Script 內建的 Fetch API 來回覆訊息

```
var payload = {
  to: userId,
  messages: [{
    'type': 'text',
    'text': '要回覆的文字訊息'
  }]
};

var option = {
  'headers': {
    'Content-Type': 'application/json; charset=UTF-8',
    'Authorization': 'Bearer ' + CHANNEL_ACCESS_TOKEN
  },
  'method': 'post',
  'payload': JSON.stringify(payload)
};

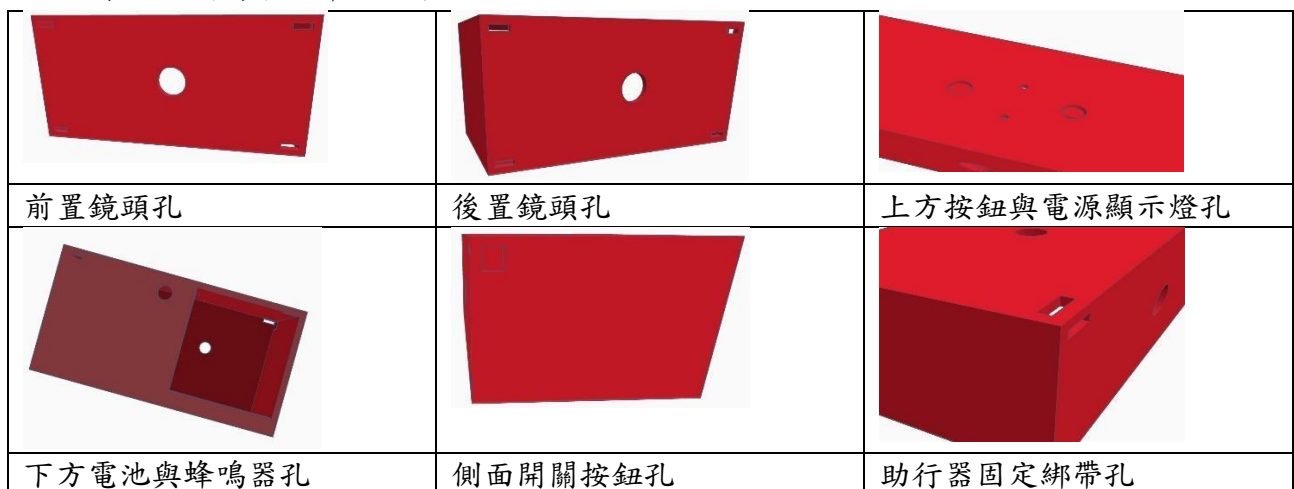
UrlFetchApp.fetch(
  'https://api.line.me/v2/bot/message/push',
  option
);
```


(十六)我們的設定是當在 Linebot 訊息裡輸入「GPS」關鍵字時，Linebot 就會回覆儲存座標 Google 試算表中的最新一筆資料



伍、研究結果

一、依裝置設備需求設計 3D 外殼：



圖十、裝置外殼 3D 設計圖

二、完成裝置 3D 設計圖後，以雷雕機切割外殼，依電路設計圖將各式設備組裝測試：

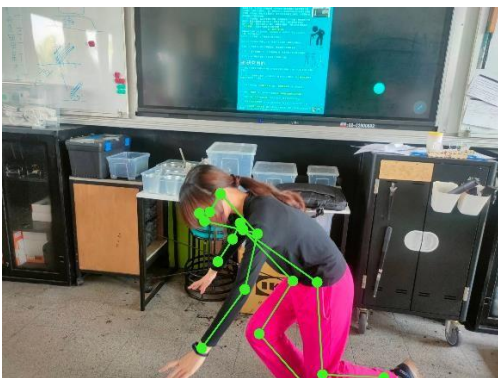


圖十一、AIOT 智能防護陪伴杖裝製成品

二、裝置使用與情境示意：



圖十二、與市面上助行輔具結合正面 圖十三、與市面上助行輔具結合背面

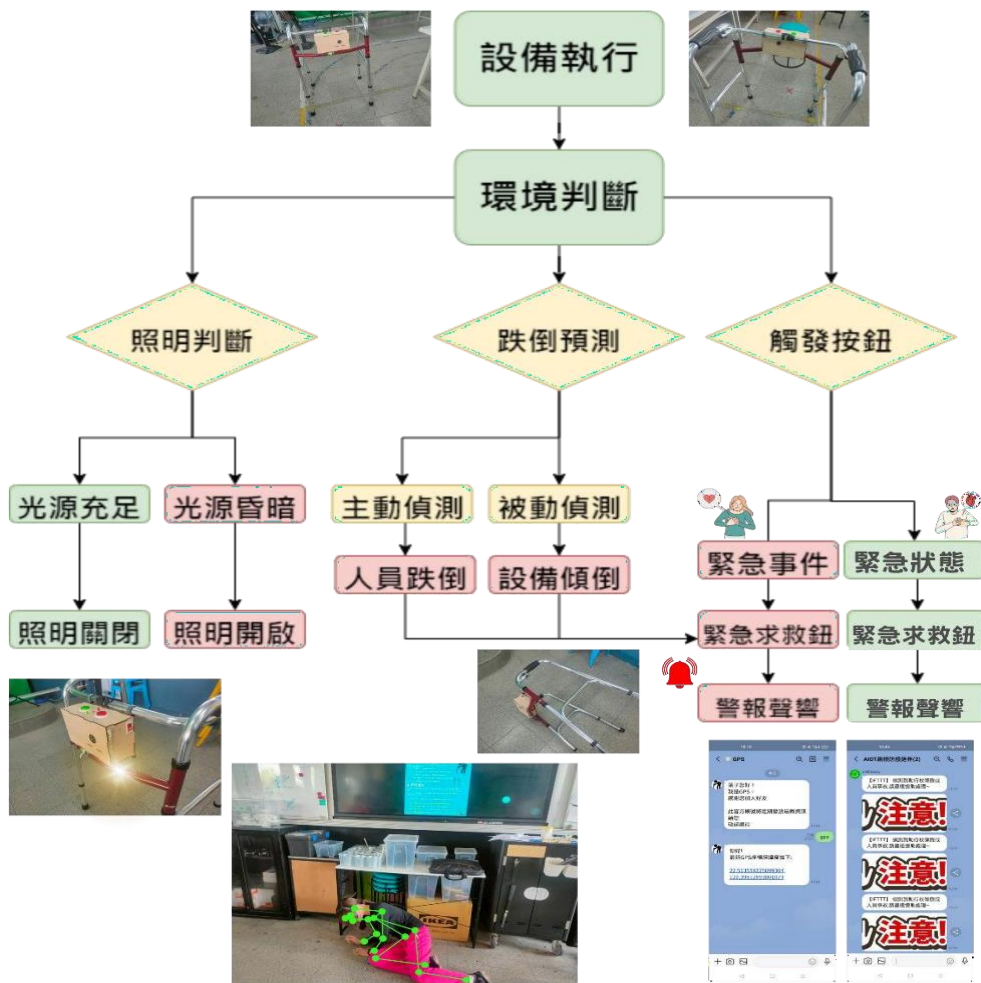


圖十四、人體跌倒動作偵測



圖十五、人體跌倒動作偵測

三、程式執行流程：



圖十六、裝置執行流程圖

陸、討論

我們完成作品後，進行實物功能測試，AI 鏡頭或三軸感應器偵測到跌倒狀態後，將資料上傳到設定的 google 試算表欄位，連同緊急按鈕的功能如下：

- 一、AI 前鏡頭與後鏡頭主動式跌倒偵測訊號紀錄為 1，並發送到 google 試算表欄位一。
- 二、三軸感應器被動式跌倒偵測訊號紀錄為 2，並發送到 google 試算表欄位二。
- 三、觸發紅色按鈕時，發送紀錄 3 到 GOOGLE 試算表欄位三。
- 四、觸發綠色按鈕時，發送紀錄 4 到 google 試算表欄位四。



圖十七、主動式跌倒偵測前鏡頭



圖十八、主動式跌倒偵測後鏡頭

紅色按鈕為醫療單位的緊急通報，綠色按鈕為遠端家屬的緊急通報，當四個欄位只要分別有新的資料匯入後，遠端的 Line 均能正常顯示通知如圖十六，再利用 Linebot 輸入關鍵字「GPS 或 gps」後，會回覆最新裝置所在的座標位置，以利後續人員搶救的找尋。



圖十九、Line 緊急通報



圖二十、Linebot 座標查詢

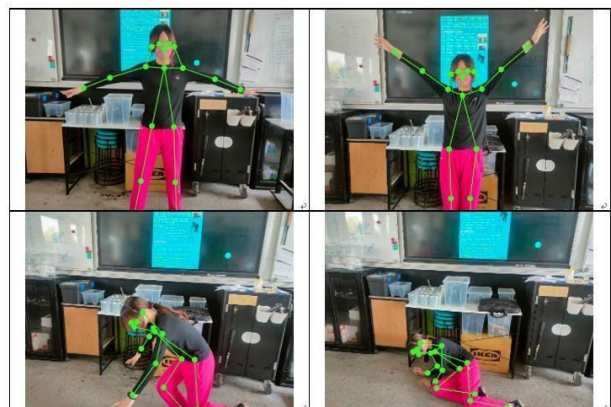


圖二十一、座標於 google maps

而在裝置測試當中，利用人工智慧做跌倒動作辨識時，會因為人與鏡頭的距離而影響準確率，故我們再做辨識準確度實驗的分析，依照人員距離辨識鏡頭每一公尺為區間，每區間做 20 次的跌倒動作偵測，做準確度的評估，距離鏡頭 1-8 公尺的辨識效果，準確度都能高於 80%。

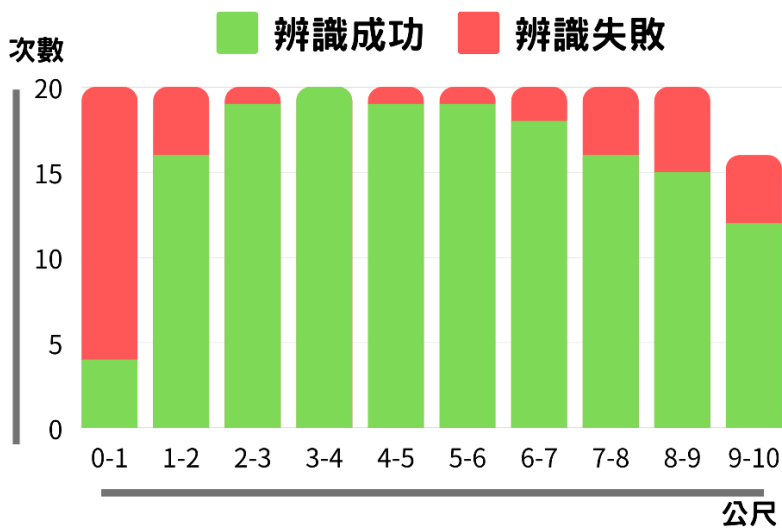


圖二十二、辨識鏡頭單位距離(公尺)



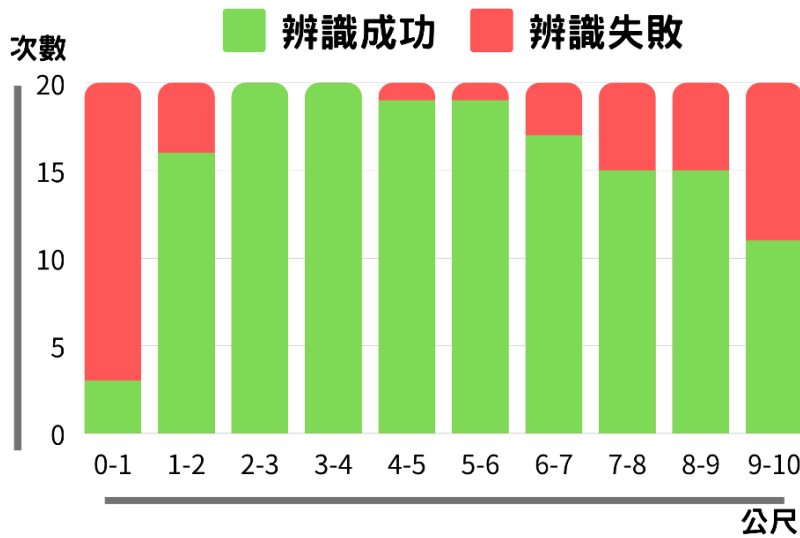
圖二十三、跌倒動作辨識特徵點

跌倒辨識準確實驗分析圖：



圖二十四、前置辨識鏡頭準確分析

	成功	失敗
0-1	4	16
1-2	16	4
2-3	19	1
3-4	20	0
4-5	19	1
5-6	19	1
6-7	18	2
7-8	16	4
8-9	15	5
9-10	12	4

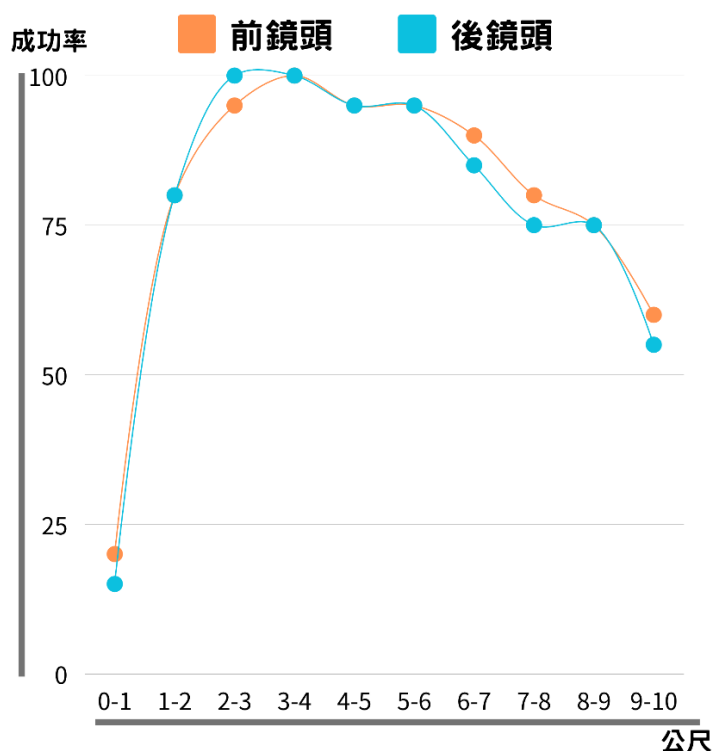


圖二十五、後置辨識鏡頭準確分析

	成功	失敗
0-1	3	17
1-2	16	4
2-3	20	0
3-4	20	0
4-5	19	1
5-6	19	1
6-7	17	3
7-8	15	5
8-9	15	5
9-10	11	9

柒、結論與未來展望

經過跌倒辨識準確度實驗分析後，我們可以發現，靠近鏡頭太近，距離小於 1 公尺內，鏡頭只能拍攝到腹部左右的範圍，無法有效辨識是否跌倒，而在距離大於 1.5 公尺之後，辨識準確度就大幅提升到 80%以上而在距離超過 7 公尺之後，因為人體模型過小，所以準確度降至 80%之下，而根據助行器使用者的行為分析，因為隨時都帶在身旁，活動範圍也都在助行器 5 公尺之內，但是因為前鏡頭與後鏡頭兩旁的區域為鏡頭死角，故會在兩旁活動時就無法做主動式偵測，是為辨識盲區，也是我們之後裝置改進的重點項目之一，本次作品物聯網緊急通報只有連結行動裝置的 line 訊息，來做遠端家屬的通知示意，未來如果將裝置改進的更為完善後，嘗試與醫療體系、救護車通報系統做連結，能讓醫護人員在意外發生第一時間前往事故現場，也許更能發揮本作品的預期效果，期許能降低高齡者事故後受傷風險的嚴重性。



	前鏡頭	後鏡頭
0-1	20	15
1-2	80	80
2-3	95	100
3-4	100	100
4-5	95	95
5-6	95	95
6-7	90	85
7-8	80	75
8-9	75	75
9-10	60	55

圖二十六、前置鏡頭與後置鏡頭跌倒辨識準確度



圖二十七、裝置兩旁側邊的辨識盲區



圖二十八、緊急通報措施

捌、參考文獻

- 一、約瑟夫·F·柯佛林，2018。銀光經濟：55 個案例，開拓銀髮產業新藍海。天下文化。
- 二、青銀跨代，共餐同居零距離——盤點歐、日 5 種「青銀共居」新型態
<https://npost.tw/archives/52257>
- 三、黃博偉，2017。2016 輔具產業全球市場與技術趨勢暨國內市場與學術研發量能。輔具之友 38 期。
- 四、曾吉弘，2021。初學 Jetson Nano 不說 No：CAVEDU 教你一次懂。翰吉文化。
- 五、林聖泉，2021。物聯網實戰(ESP32 篇)：使用樹莓派/NodeMCU32S/Python/MicroPython/Node-RED 打造安全監控系統。博碩。
- 六、曹永忠，許智誠，蔡英德，2023。ESP32 物聯網基礎 10 門課 The Ten Basic Courses to IoT Programming Based on ESP32。崧燁文化。

【評語】 032810

本研究受支援盲人的智能虛擬手杖啟發，希望探討可以在遠距離偵測到獨居老人跌倒時並傳送警報的 AI 跌倒辨識助行器。本作品結合多種功能製作 AIOT 智能拐杖，議題由社會問題出發，並能夠收集現有資料與分析各自優劣，值得肯定。

對於本研究，我們有以下之建議：

- (1) 若能不只是提供事後判斷跌倒並進行通知，而是加入協助老人防止跌倒功能，應有更佳效益。
- (2) 未來可考慮結合一般手環的功能會更有智慧。
- (3) 拐杖傾倒是否代表老人傾倒，可以討論。
- (4) 助行器的使用設定應該先討論。是老人家每一步行都要用？還是走長距離才需要？這會決定跌倒偵測的範圍和模式。
- (5) 使用助行器時，身體和辨識裝置的距離一定小於 1 公尺，設計尚可在思考如何改進。

作品海報



AIoT 智能防護陪伴杖



研究動機

根據聯合國研究人口統計，2019年全球平均壽命為72.8歲，逐年增高中，內政部統計國人2021年的平均壽命為80.86歲，與聯合國公布2019年全球平均壽命比較，我國男、女性平均壽命分別高於全球平均水準7.5歲及9.3歲，2022年6月底全國六十五歲以上的高齡者人口數為3,983,640人，占17.18%，由此可見，未來高齡者的社會人口比例越來越高，食、衣、住、行、育、樂等相關產業發展會是極為重要的一環，伴隨著人口老化趨勢，老人照護相關產業備受重視，而因為近期家裡長輩發生跌倒事故，讓我們注意到高齡者伴隨行動不便的問題，希望能藉由本次作品研究的內容，協助事故發生後相關後續處理問題，降低行動不便者因跌倒受傷的嚴重性。

跌倒困擾的不只是當下軟組織的受損或骨折，嚴重的需要開刀甚至臥床，後續造成的肌少症問題與併發狀況，因此，「防跌」是我們高齡化社會中，很重要的課題，而該如何選擇適當的“步行輔具”，就是我們很需要注意的課題！

在國中科技課程裡面，聽著上課老師所教的開發版、感應器、程式積木、物聯網、人工智慧等概念，心想能否結合這些工具，製作出智慧化的行動輔具，能隨時注意到高齡者的行動狀態，並且及時防護監測的裝置，達到「青銀共好，守護陪伴」的理念，是我們這次作品實踐的目標。

研究調查

輔助器具 (ASSISTIVE DEVICES) 泛指所有能用來輔助、協助人們完成各方面行動的輔助工具或輔助科技 (ASSISTIVE TECHNOLOGY)，特別是用來協助特定失能人士、身心障礙者或因高齡老化導致一般生理功能篩退者。

下列是台灣輔具需求：

- 一.短期需求：運動傷害、疾病復健所需。
- 二.身心障礙者與其家戶需求：全國身心障礙者總人數1,125,632人，佔總人口數4.8%。
- 三.銀髮族及其家戶需求：全國65歲以上高齡者人數佔11.5%。（影響每家每戶）預估2020進入「高齡社會」（超過總人口數14%）；2036進入「超高齡社會」（超過總人口數21%）。
- 四.長照服務相關法規將帶動輔具及衍生創新服務全面提升。

在我們調查市面上行動不便高齡者的防護用品中，助行器是常使用的工具，如右圖，常見的使用者有：

- 一.銀髮族:65 歲以上之老人。
- 二.骨科疾病患者:截肢、骨折、下肢關節炎、關節置換手術患者。
- 三.神經系統疾病患者:不完全脊椎損傷、中風、帕金森氏症、失智症患者。
- 四.廣泛性體能衰弱患者:心肺疾病患者。

同時發現以下幾項很有特色的智慧項目:

- 一.WEWALK—AIOT智慧拐杖，可望引導盲人「看」世界，這種智慧拐杖內建超音波感測器，可以感應到160CM高的障礙物，還帶有震動和聲音警示裝置，可以透過藍牙與智慧型手機連接。它還可以與GOOGLE MAPS連接進行導航，電池續航達5小時，目前支援的語言包括英語、土耳其語、阿拉伯語和葡萄牙語。
- 二.「智能盲人虛擬手杖」是一款手握遙控感測器，十字形紅外線雷射與接收，作為距離定位障礙物感測，感測路上障礙物的距離與大小。並加上鏡頭之影像辨識技術系統，配合盲人RFID感測系統，利用預先埋設RFID導盲磚式人行道+障礙物RFID貼紙，配合鏡頭的影像辨識系統，即可以讓盲人更清楚知道所碰觸物體的名稱與實際大小的狀態。加裝震動模擬器模擬訊號回饋之碰觸物體震動，來模擬辨識障礙物。系統也利用GPS衛星定位，提供所在地周遭生活資訊，大眾運輸資訊與智慧城市資訊網連線。虛擬手杖並提供自動AI學習該盲人生活習慣，例如個人使用習慣，身高與常用的感知長度，或生活環境區域之所有行為作出智慧校正；因此是一款專門為盲人生活打理設計的虛擬手杖。

研究目的

在經過文獻探討，瞭解目前市面上助行器與幾項智慧拐杖的功能後，以高齡行動不便者的行為模式來做思考，過重的裝置或過於複雜的操作模式都可能為長輩造成負擔，而裝置能智慧化結合平常助行器的使用習性，並能簡單操作是我們設計的方向，依照防摔、緊急通報與位置追蹤的三大要點，我們的AIOT防護陪伴杖也有了大致清楚的雛形，研究目的可以歸納成以下幾項：

- 一、以AI人工智慧開發套件，結合鏡頭依人體動作的影像辨識，做主動式偵測跌倒、痙攣等突發性狀況。
- 二、利用三軸感測器，做被動式偵測，當助行器瞬間傾倒時，發送求救警報，同時外部增加照明燈條，減少因天黑或陰暗處，視線燈光不明造成行走的危險性，並設置緊急呼叫按鈕。
- 三、搭配IOT物聯網，將上述的即時資訊做遠端行動裝置或醫療系統的通報。
- 四、利用GPS晶片做及時位置追蹤，如緊急情況、迷路走失的協尋定位，並結合LINEBOT做座標位置查詢。



圖一、四腳助行器



圖二、四腳拐杖

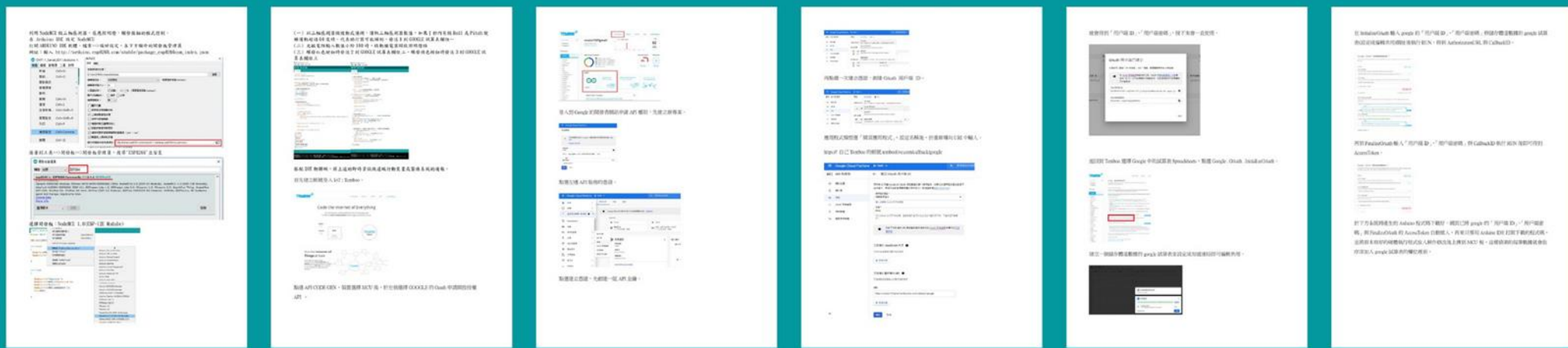


圖三、WEWALK拐杖



圖四、盲人虛擬手杖

(五)利用NODEMCU做三軸感測器，感應照明燈，觸發按鈕的程式控制



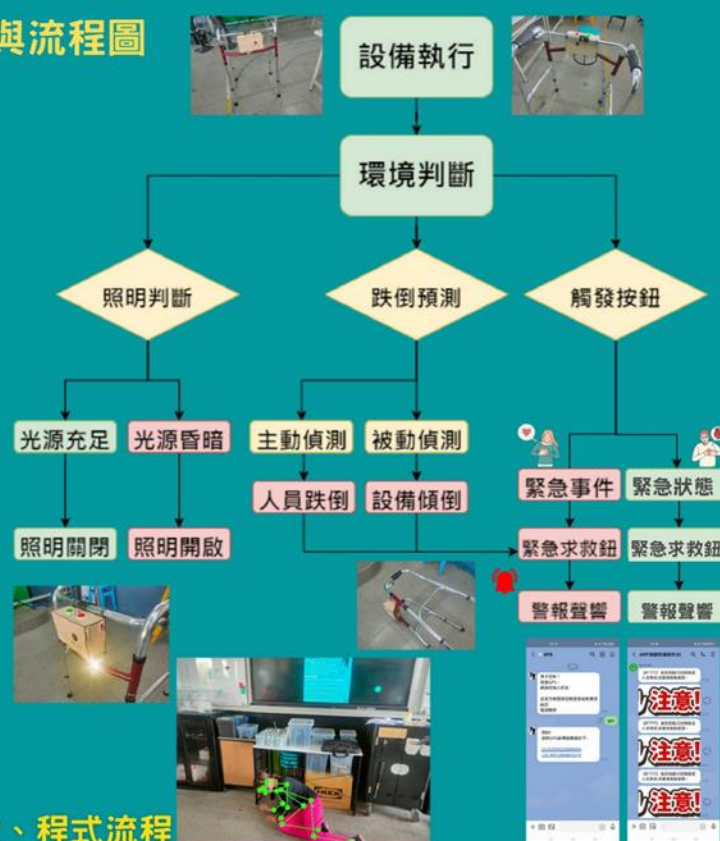
研究結果

(一)以雷雕機設計外殼後，依電路設計圖組裝測試



圖五、構造介紹

(二)使用情境與流程圖



圖六、程式流程

討論

我們完成作品後，進行實物功能測試，AI鏡頭或三軸感應器偵測到跌倒狀態後，將資料上傳到設定的GOOGLE試算表欄位，連同緊急按鈕的功能如下：

- 一、AI前鏡頭與後鏡頭主動式跌倒偵測訊號紀錄為1，並發送到GOOGLE試算表欄位一。
- 二、三軸感應器被動式跌倒偵測訊號紀錄為2，並發送到GOOGLE試算表欄位二。
- 三、觸發紅色按鈕時，發送紀錄3到GOOGLE試算表欄位三。
- 四、觸發綠色按鈕時，發送紀錄4到GOOGLE試算表欄位四。

紅色按鈕為醫療單位的緊急通報，綠色按鈕為遠端家屬的緊急通報，當四個欄位只要分別有新的資料匯入後，遠端的LINE均能正常顯示通知如圖十六，再利用LINEBOT輸入關鍵字「GPS或GPS」後，會回覆最新裝置所在的座標位置，以利後續人員搶救的找尋。

而在裝置測試當中，利用人工智慧做跌倒動作辨識時，會因為人與鏡頭的距離而影響準確率，故我們再做辨識準確度實驗的分析，依照人員距離辨識鏡頭每一公尺為區間，每區間做20次的跌倒動作偵測，做準確度的評估，距離鏡頭1-8公尺的辨識效果，準確度都能高於80%。



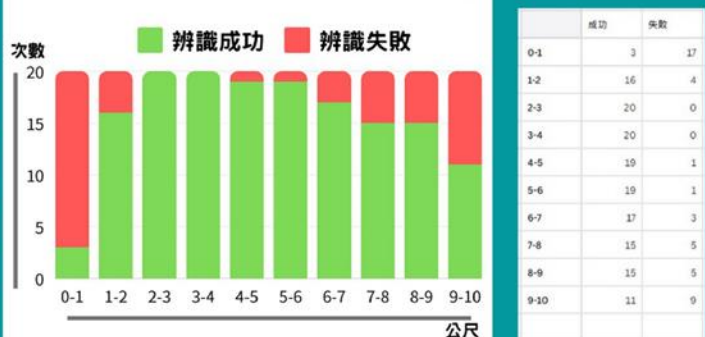
圖七、座標查詢



圖八、跌倒參數特徵值

結論與未來展望

經過跌倒辨識準確度實驗分析後，我們可以發現，靠近鏡頭太近，距離小於1公尺內，鏡頭只能拍攝到腹部左右的範圍，無法有效辨識是否跌倒，而在距離大於1.5公尺之後，辨識準確度就大幅提升到80%以上而在距離超過7公尺之後，因為人體模型過小，所以準確度降至80%之下，而根據助行器使用者的行為分析，因為隨時都帶在身旁，活動範圍也都在助行器5公尺之內，但是因為前鏡頭與後鏡頭兩旁的區域為鏡頭死角，故會在兩旁活動時就無法做主動式偵測，是為辨識盲區，也是我們之後裝置改進的重點項目之一，下階段可以結合麥克風做語音辨識的求救，或結合GPT做更多的應用，作品物聯網緊急通報只有連結行動裝置的LINE訊息，來做遠端家屬的通知示意，未來如果嘗試與醫療體系、救護車通報系統做連結，能讓醫護人員在意外發生第一時間前住事故現場，更能發揮本作品的預期效果，期許能降低高齡者事故後受傷風險的嚴重性。



圖九、前後鏡頭辨識統計



圖十、辨識統計分析

參考文獻資料

- 一、約瑟夫·F·柯佛林，2018。銀光經濟：55個案例，開拓銀髮產業新藍海。天下文化。
- 二、青銀跨代，共餐同居零距離——盤點歐、日5種「青銀共居」新型態
- 三、黃博偉，2017。2016輔具產業全球市場與技術趨勢暨市場與學術研發量能。輔具之友38期。
- 四、曾吉弘，2021。初學JETSON NANO不說NO：CAVEDU教你一次懂。翰吉文化。
- 五、林聖泉，2021。物聯網實戰(ESP32篇)：使用樹莓派打造安全監控系統。博碩。
- 六、曹永忠，許智誠，蔡英德，2023。ESP32物聯網基礎10門課。崧燁文化。



圖十一、通報機制