

2025年臺灣國際科學展覽會 優勝作品專輯

作品編號	100041
參展科別	工程學
作品名稱	影像辨識 在 智慧運輸系統 的應用 自動跟隨

就讀學校	國立臺灣師範大學附屬高級中學
指導教師	李啟龍
作者姓名	洪祐晞

關鍵詞	<u>影像辨識 、 物聯網 、 arduino</u>
-----	-----------------------------

作者簡介



我是師大附中高二的學生，我國中時開始接觸python程式，高中時透過專題研究課程開始認識人工智慧，並理解如何使用它，我對其中的影像辨識和強化學習有較多興趣。我結合YOLO影像辨識和arduino進行本次研究。

感謝在我的研究中提供指導的老師、捐贈設備的校友及支持我的家人，讓我能走到這一步。

2025 年臺灣國際科學展覽

研究報告

區別：

科別：工程學科

作品名稱：影像辨識在智慧運輸系統的應用—自動跟隨

關鍵詞：影像辨識、物聯網、arduino

編號：

（編號由國立臺灣科學教育館統一填列）

目錄

摘要	3
Abstract	3
一、前言	4
二、研究方法與過程	5
1. 研究設備與器材	5
2. 研究運用技術	6
物聯網	6
YOLO 影像辨識	8
Arduino 模擬運輸車	21
三、研究結果與討論	23
四、結論與應用	24
五、參考資料與附錄	24

摘要

本研究旨在開發一款能自動跟隨人的 AI 智慧運輸車，可應用在醫院、商場等環境，能減輕工作人員的負擔。本系統融合 YOLOv8 影像辨識技術和物聯網，使運輸車具備跟隨人體的能力。該系統通過物聯網，將車上攝影機接收的影像傳送至電腦，進行即時運算，不必在車上配備高階的微型電腦或 GPU 以降低成本。使用 YOLOv8 模型辨識人體與手勢，以 arduino ESP32 開發板作為主控單元控制減速馬達，使運輸車能自動跟隨人員。本研究不像傳統自動導引車（AGV），因為依賴固定路徑，而無法用在多變的場景，且不適合與人協同工作。將本系統應用於醫療院所或零售業商場等，經常出現多變障礙物的場所時，能與人員配合執行任務，例如當作病人的點滴架、輔助護理師工作的醫務車，以及賣場中協助工作人員上貨的籠車，能有效降低勞力負擔。

Abstract

This study aims to develop an AI-powered smart transport vehicle capable of automatically following human movement, designed for environments such as hospitals and shopping malls, to reduce the workload of staff. The system integrates YOLOv8 image recognition technology and the Internet of Things (IoT) to enable the transport vehicle to track and follow human subjects. By utilizing IoT, the system transmits real-time camera footage from the vehicle to a computer for processing, eliminating the need for expensive onboard microcomputers or GPUs, thereby reducing costs. The vehicle uses the YOLOv8 model to recognize human bodies and gestures, and an Arduino ESP32 development board controls the motors, allowing the vehicle to follow people automatically.

Unlike traditional Automated Guided Vehicles (AGVs) that rely on fixed paths and are unsuitable for dynamic environments or human collaboration, this system adapts to variable obstacles commonly found in hospitals and retail environments. It can collaborate with personnel

to perform tasks such as serving as a mobile IV stand for patients, assisting nurses with medical equipment, or helping staff with stocking in retail stores, significantly reducing the physical burden on workers.

一、前言

全球不少國家因為少子化，面臨缺工問題，若能應用 AI 人工智慧發展自動跟隨的輔助工具，能減輕工作人員負擔，並提高工作效率。例如在醫療院所，護理師需推動病床和醫療器材車，或是患者需拉著點滴架，前者非常耗費體力，後者若稍不注意便可能拉扯到點滴管和傷口。而現今在物流和製造業中常見的自動導引車（AGV）技術雖然成熟，但主要應用於重複且固定的路線上，且其避障能力有限，須事先規劃路線和設置標記，在需要靈活調度、變換任務的環境中顯得成本高且效率低，這些限制使其無法應用在多變的環境中，如人來人往的商場或醫院等複雜場合。因此許多超市員工仍需親自推著籠車在賣場補貨，大型商場內的貨物搬運仍仰賴工作人員駕駛手動油壓推車，顧客也依然使用手推車購物。若是使用自動跟隨人的運輸車將能改善上述困境。因此具備自動跟隨能力的智慧運輸車能在醫院和商場中扮演重要角色，不僅能減輕醫護人員的負擔，提升工作效率，解決人力短缺的問題。此外，若發展出自動跟隨病人的點滴架，也能在病人移動時，不必自己拉著點滴架，顯得更加便利與安全。

因此，本研究設計出一款能夠跟隨人員移動的智慧運輸車，未來能應用於醫療院所，作為智慧點滴架、醫務車和病床，或應用於賣場作為智慧推車。這樣的技術不僅能滿足複雜環境中的動態需求，還能有效降低勞力負擔，提升操作靈活性。

預計可運用在下列器材上



醫務車。截自「朝悅公司」網站，參考資料 1



工廠或賣場中常見的油壓推車。截自「恆智重機」網站，參考資料 2



超市補貨用的籠車。截自「奇渥科技」網站，參考資料 3



病床。截自「永樂屋」網站，參考資料 4



點滴架。截自「仁宇醫療器材」網站，參考資料 5

二、研究方法與過程

1. 研究設備與器材

◎硬體：

(1)ASUS vivobook i7，以筆記型電腦執行遠端運算。

- (2)自製 arduino 模擬運輸車，其上裝載 NodeMCU ESP32S 、減速馬達、L9110s 控制板、手機（當作攝影機鏡頭）、18650 電池。
- (3)學校的 A100 顯示卡，用於訓練 YOLO 模型。

◎軟體：

- (1)Visual Studio Code
- (2)Python 3.11.5
- (3)Arduino IDE 1.8.19
- (4)LabelImg
- (5)函式庫:
 - Ultralytics(YOLO)
 - pytorch
 - cv2
 - Pillow

2. 研究運用技術

物聯網

主要原因如下：

1. AI 運算必須快速，不然運輸車的動作會卡頓。若所有運算都要在車上進行的話，車上必須配備很好的微型電腦，甚至要裝 GPU，這必定讓運輸車的成本大幅上升。此外車上的電源供應也容易出問題，因為電腦運算時耗電量很高，像是微型電腦樹梅派 Pi 3B 閒置時的平均功耗就要 1.5W(自 icshop 網站，參考資料 6)，而 ESP32 在活動期間的功耗也大多在 100mW 以下(自 espressif 網站，參考資料 7)，可見用微型電腦運算比用 wifi 傳訊的 ESP32 開發板更耗電。而運輸車在運作時也不可能一直去充電，因此讓運輸車連線到主機，由主機運算能有效降

低成本，並增加續航力。

2. 方便統一遠端控制，且在醫院或是工廠大多都已裝有 wifi，因此無須再裝新的通訊設備，可降低執行的成本。

運作方式：

用手機裝 iVcam 當網路攝影機，透過 wifi 傳影像給電腦。電腦利用 YOLO 模型辨識影像中的手勢及人體位置，經由計算得到操作人相對於運輸車的位置和距離，決定運輸車的動作，再用 wifi 傳指令給車上的 ESP32 開發板控制運輸車。運輸車在物聯網中是一個伺服器，像是自己營運一個網站，接收電腦傳給它的訊號，做出對應的動作，完成後會再傳訊號給電腦，電腦發出訊號的那行程式才算執行完畢，才會



繼續執行下一行程式，所以能避免發生一個指令還沒做完，下一個指令就來。且這樣可以節省電腦運算的耗電量，因為車在移動的那幾秒，電腦不必辨識畫面。下面是電腦端傳訊息的程式。自行製作

```
ESP32_ip = '192.168.0.111'

url = f'http://{ESP32_ip}/data'

data = 'motor'+ ' '+'0'+ ' '+'500' #motor 代表是控制馬達，0 代表前進，500 是運轉毫秒數

response=requests.post(url, data=data,timeout=2)
```

wifi 熱點選擇：

ESP32 開發板發出 wifi 自己當熱點

優點：系統不論在哪運作都能快速連線，不必修改 ESP32 中的 arduino 程式（若是 ESP32 要去連其他 wifi 熱點，程式中要給該網路的 ssid 和 password）。

缺點：涵蓋範圍小，訊號不夠強，無法通過牆壁等障礙物。很耗電，若電池不是滿電時，可能會電壓不足，發不出訊號。

用 wifi 分享器當熱點

優點：範圍大，訊號強，不會耗費車上的電力。且若像是在醫院，都是同一個 wifi，但有多個 wifi 分享器，即使連不同個分享器，也能互相通訊，能讓訊號範圍非常廣泛。

缺點：在不同的場所，ESP32 被分配到的 IP 會不同，必須要先確認 IP 位置，才設定電腦連接。且因為每個 wifi 的 ssid 和 password 不同，要改 arduino 程式，再灌進 ESP32 開發板，前置準備工作較久。

在試過兩種方式後，我們是採用 **wifi 分享器當熱點**，因為需要遠距離傳輸訊號。且因為電腦只能連一個網路，而手機要當電腦的網路攝影機所以也要連和電腦同一個 wifi 熱點，因此如果電腦連 ESP32 發出的 wifi，就要用它來傳網路攝影機的視訊訊號，但 ESP32 的 wifi 只有 2.4G，無法傳送辨識需要的 720×1280 像素影像。因此我們在決定用手機當 wifi 攝影機後，就確定採用 wifi 分享器的 wifi 訊號，而 ESP32 就當作這個網域中的一個伺服器。

YOLO 影像辨識

YOLO (You Only Look Once) 是一個專門用於影像辨識的深度學習模型，基於 pytorch 開發，具有辨識速度快，準確性高等優勢。近期推出較新的版本，指令程式簡化很多，且能辨識不同大小的圖片，要進一步計算或搭配 cv2 視覺化辨識結果都很方便。YOLOv1 在 2015 年推出，到 2024 年九月已經推出 YOLOv11。現在我們採用的是 YOLOv8，它的功能包含 class、detect、segment、pose 等。每種模型有 n、s、m、l、x 大小分別，n 到 x 由小到大（大小是指模型中神經網絡的層數不是畫質）。性能如表一，數據為用 coco 資料集訓練和評估，大約分 80 個類別。

Model	size (pixels)	mAP ^{val} 50-95	Speed CPU ONNX (ms)	Speed T4 TensorRT10 (ms)	params (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

表一。截自 ultralytics 在 github 中的資料，參考資料 8

Speed 的兩欄位分別是用 CPU 和 nvidia T4 GPU 跑辨識的速度。

mAP50 是模型預測邊界框和真實邊界框重疊超過 50%算判斷正確，接著計算召回率(recall)召回率是透過下列公式算出：

$$\text{召回率(recall)} = \frac{\text{判斷正確次數}}{\text{判斷正確次數} + \text{判斷錯誤次數}}, \text{ 算出召回率就是 mAP50}$$

mAP95 則是要重疊超過 95%才算正確，算出的召回率。

由上表可看出雖然越小的模型準確度較差，但這個表的數據是模型訓練和辨識 coco dataset，模型要分 80 類的物件，而我們訓練的只要分 2 類，準確度就已經很夠了。考量到辨識速度，因為我們的電腦沒有 GPU，是用 CPU 運算，x 模型花費的時間超過 n 模型的 5 倍，在車上測試時也明顯變得卡頓。因此我們決定採用大小為 n 的模型。

以下的圖文是使用 YOLO 提供的預訓練模型(Pretrained Models)辨識的結果，介紹四種功能的 YOLO 模型，是我們選擇模型時的依據。

YOLO-classify 模型是以一張圖片為單位分類，如圖一。若像圖中不夠聚焦於人，它會認為這張圖片很多種可能的類別，圖中小字就是類別，右邊的數字是信心程度，經由 softmax 函數計算，越高代表越可能是那一類。而圖一中各類別的信心程度都很

低，且可明顯看出它判斷的類別是錯的。經過測試可發現 classify 模型需要聚焦目標物的圖片。訓練 YOLO-classify 模型時，需要給它要辨識的各個類別的圖片，分別放在以類別名稱命名的資料夾，這是最容易且快速處理方式。



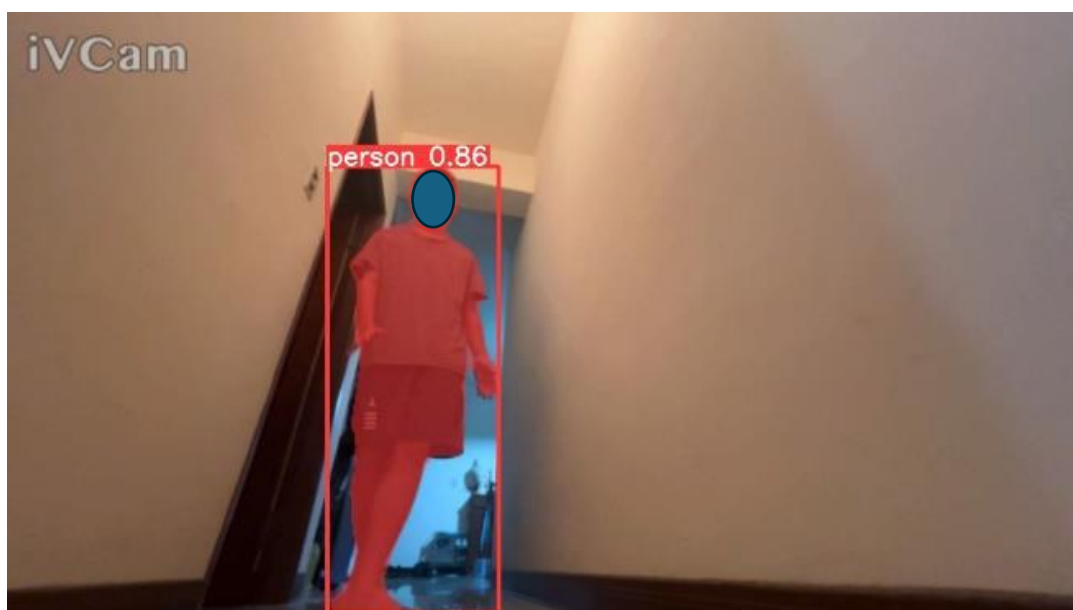
圖一，註:自行拍攝

YOLO-detect 模式，可偵測目標物的種類和在圖中的位置。如下圖二，模型會回報目標物種類和中心座標和邊界框的長寬，邊界框是一個框住目標物的長方形，它可自動顯示在圖片上，我們也可以用 cv2 自己修改，可連續辨識影片，並顯示在影片上。訓練 YOLO-detect 模型時，要為每張圖片做標籤，標籤是一個 txt 檔，包含類別名稱、邊界框中心的座標及邊界框的長寬。



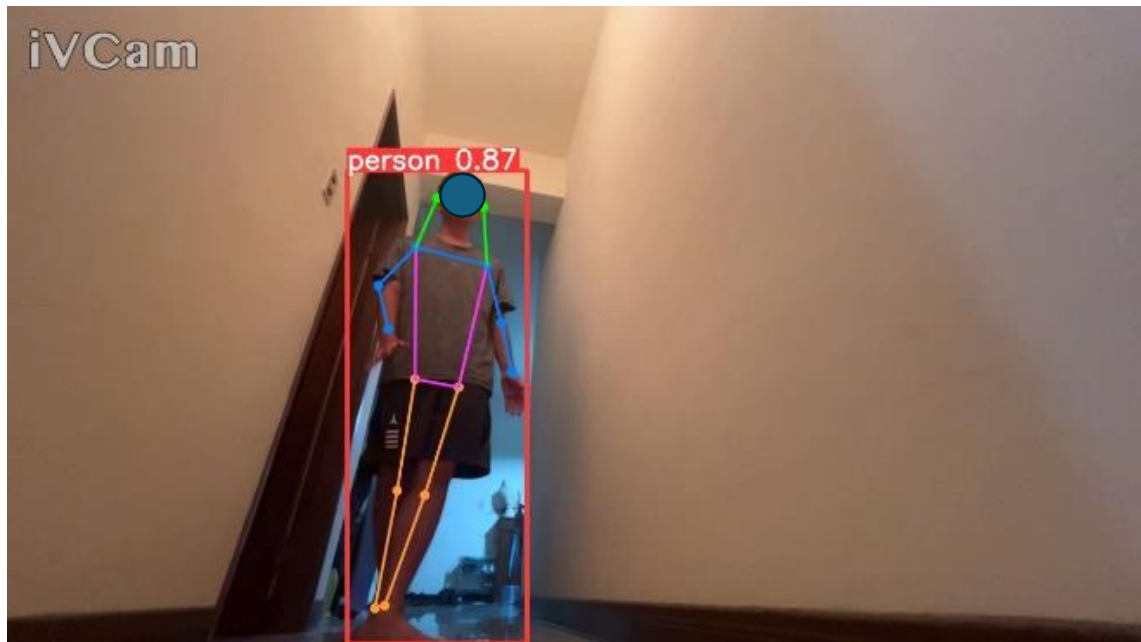
圖二，註:圖二及圖三自行拍攝

YOLO-segment 模式可偵測目標物種類和位置，還能擬合目標物的形狀，適合用在精準測量。模型回傳結果包含目標物種類、中心座標、邊界框的大小和分割掩碼 (Segmentation Masks)，分割掩能顯示物件的精確輪廓。辨識結果畫在圖上，如圖三中的紅色霧區就是目標物的輪廓。但是訓練 YOLO-segment 模型的訓練資料，除了要有目標框的位置和長寬，還要有物件在圖中的精確輪廓，需要花費很多時間準備。



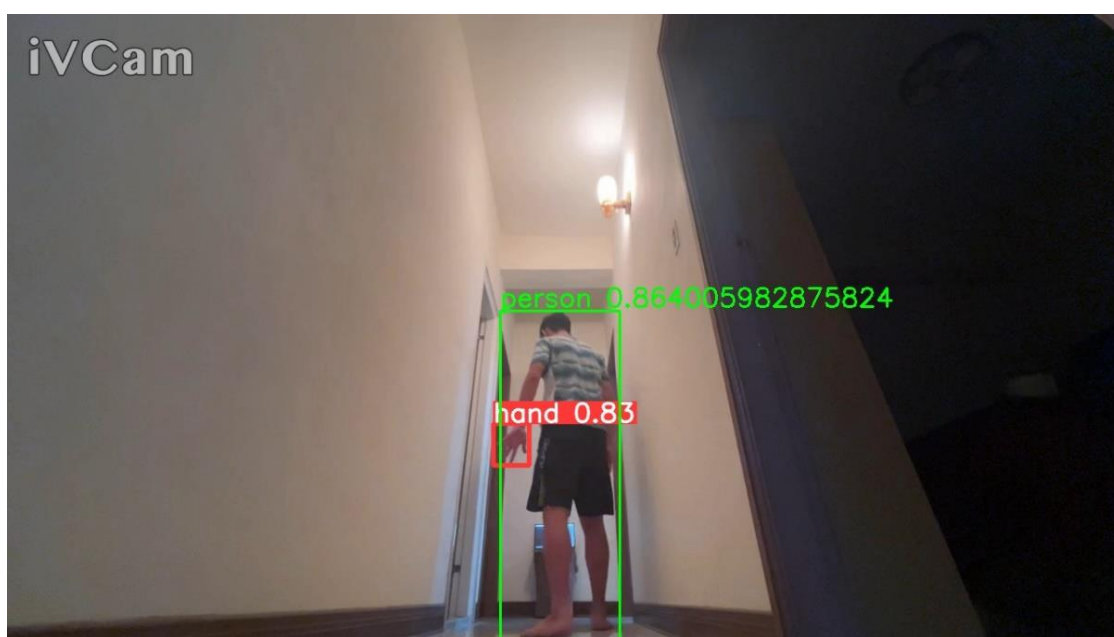
圖三

YOLO-pose 模型能辨識人的關鍵點位置，關鍵點如手肘、腳踝、眼睛等，可用來辨識人的姿勢，很適合用來視覺化人的動作，如圖四。YOLO-pose 的訓練資料，除了要有目標框的位置和長寬，還要包含關鍵點的位置。



圖四

考量到訓練資料的準備時間以及能夠跟隨操作人，我們最後選用 YOLO-detect，模型大小我們用 n，它可偵測目標物種類和在圖中的位置，能滿足辨識操作人的手勢和位置的需求。如下圖五，註:圖四及圖五自行拍攝



圖五

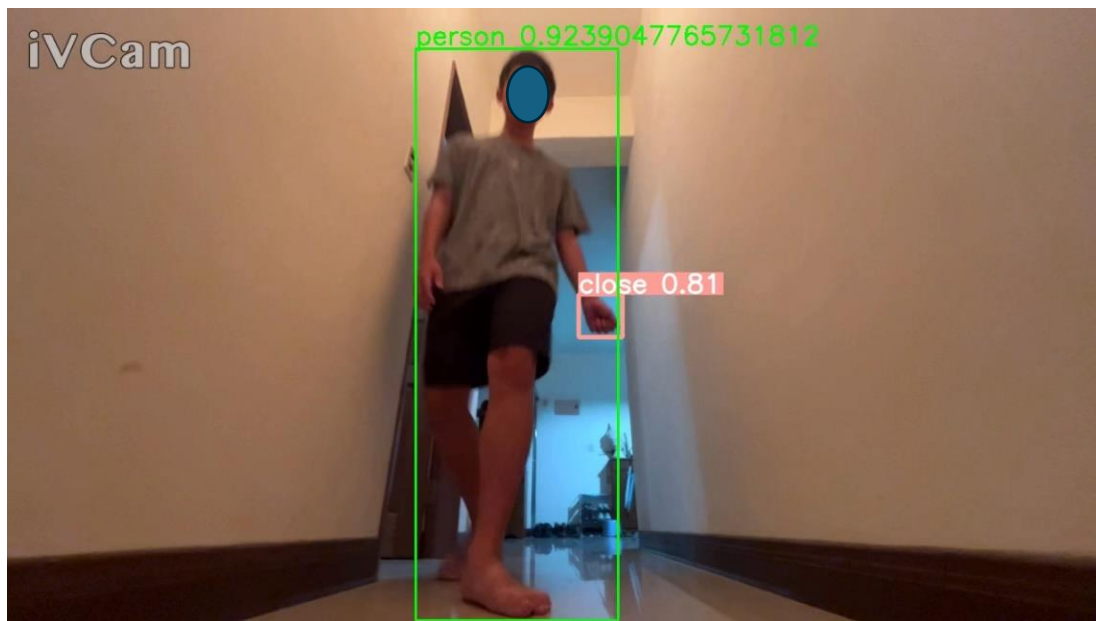
人體特徵的選用

遠距離：

運輸車在遠距離時是跟隨人體。因為距離遠時，人在畫面中較大且特徵明顯，適合應用在醫院的走廊或大廳，或是空間較大的廠房。若遠距離時伸出手掌，它也會優先跟隨手掌，因此如果操作人想要更精確的控制運輸車，那就可以伸出手掌。

近距離：

近距離時人體幾乎會站滿整個畫面甚至超出畫面，因此用人體當目標來控制就不太適合，此時操作人只要伸出手，就可以換成用手勢控制。能在狹小空間內靈活控制運輸車。而不論是用人體還是手作為目標物，只要操作人握拳就能使運輸車停止移動。如下圖六中的情況。

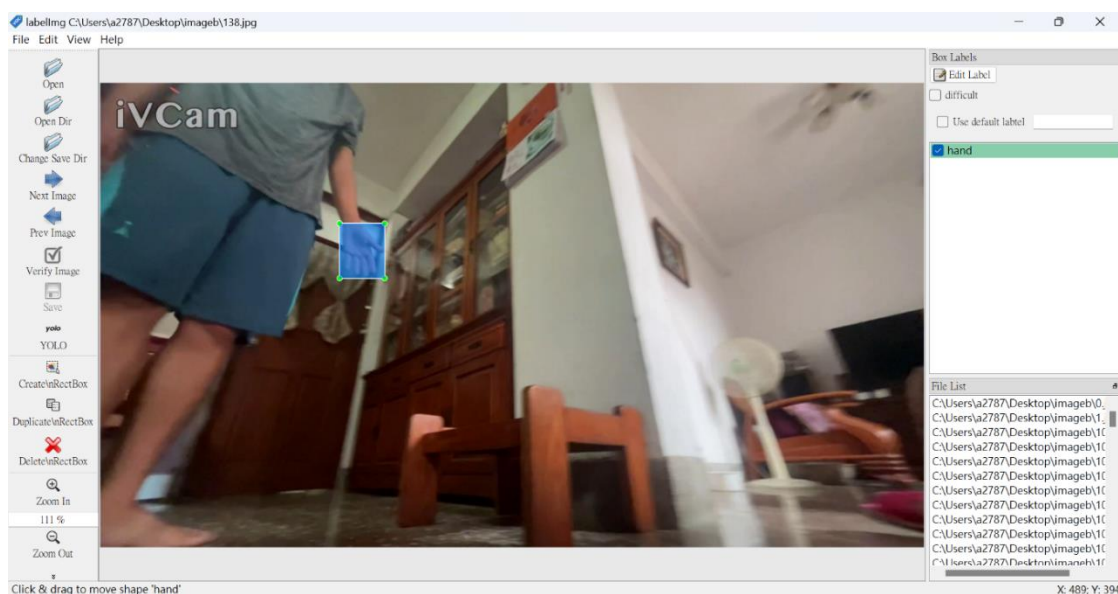


圖六，註:自行拍攝

我們達到上述功能的方法是分成兩個模型，一個模型辨識手的位置和開闔(類別名稱:開-hand，握拳-close)，另一個模型辨識人體。前者是我們自己訓練的，後者是用 YOLO 提供的預訓練模型。鏡頭回傳的每一幀照片都會經過兩個模型辨識，並優先採用辨識手的模型。

訓練辨識模型

我們大約拍了 4500 張相片，包含手張開和手握拳，接著用 labelImg 軟體幫圖片上標籤(如圖七)。而 YOLOv8 接受的標籤形式是 txt 檔，檔名要和它應對的照片檔名相同。

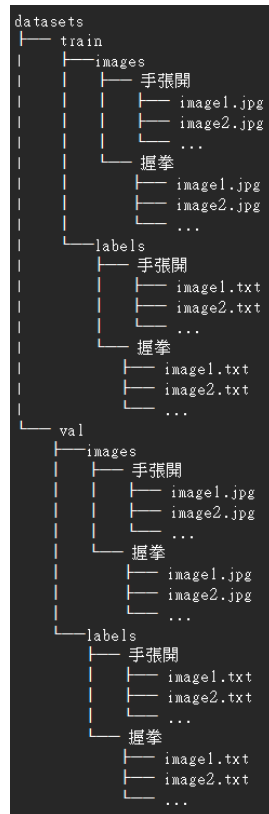


圖七，註:自行繪製

```
datasets5 > train > labels > 139.txt
1 0 0.457813 0.602083 0.034375 0.076389
2
```

圖八，標籤 txt 檔。註:自行繪製

<類別> <中心 X 座標> <中心 Y 座標> <框的 X 方向寬> <框的 Y 方向高>



圖九。註:自行繪製

在把圖片和標籤的 txt 檔分放入資料夾，並符合圖九的資料夾架構。

```
! hand.yaml
1  path: C:/Users/a2787/Desktop/hand_detect/datasets2/
2  train: train/images/
3  val: val/images/
4
5
6  nc: 2
7  # Classes
8  names:
9    0: hand
10   1: close
11
```

圖十。註:自行繪製

YOLO 訓練的程式需要一個 yaml 檔，如圖十告訴程式訓練資料的位置，以及有哪些類別等。

訓練程式很簡單，如圖十一。程式訓練完成後會自己建立名為 run 的資料夾，儲存訓練結果和模型。

```
1 from ultralytics import YOLO
2
3 model = YOLO('yolov8n.pt') # yolo提供的預訓練模型，會比完全自己訓練快且性能好，所需的訓練圖片也較少
4
5 results = model.train(data='hand.yaml', epochs=20, imgsz=640)#hand.yaml是中是放訓練資料路徑
6
```

圖十一。註:自行繪製

在訓練指令中可指定畫質、要用的模型大小(YOLO 有 n、s、m、l、x 可選)，我們用的畫質是 640*640；模型大小是 n。挑選這種畫質，是因為電腦的辨識速度才足夠，運輸車的動作才能流暢不會卡頓。

辨識結果轉換成運輸車的動作

辨識結果包含物件種類、在圖中的座標。如果只依據目標物在左邊就左轉固定度數，目標物在螢幕下半就退後，會一直對不到目標物。所以需要一些計算才能精確的控制運輸車。我們做的運輸車，因為成本考量，只能用馬達運轉的時間控制。而目前計算距離的單位是畫素，因為一畫素是一個正方形，但邊長不固定，是依據顯示器規格，所以用畫素只能計算出比例，不是實際距離。左右轉是依據角度，但也是用馬達運轉的時間控制，我們透過多次調整，才能對應到適合的轉動時間，在跟隨人體時是一畫素對應向前轉 1.016 毫秒，轉一度是 15 毫秒。iVCam 回傳的畫質是寬 1280、高 720，而 iphone14 的廣角鏡頭的是 120 度，我們根據這些數據計算。

以人體為目標物時，距離車超過 1100 畫素就是前進，小於 1000 畫素就後退，中間是緩衝，避免車子一直前後震動。

以手為目標物時，決定要前進還是後退，是透過手在圖中的座標決定，當手在圖片 y 座標大於的圖片高度(720)的四分之一是前進(如圖十二)；當 y 座標在高度的四分之一以內時就退後(如圖十三)。



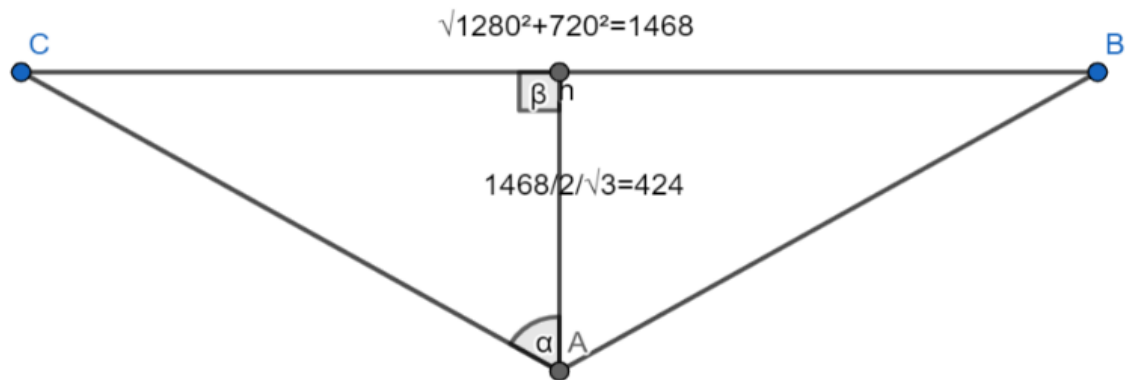
圖十二，註:自行拍攝



圖十三，註:自行拍攝

前進的時間控制：

前進以手或人當目標物都是用下述的方法

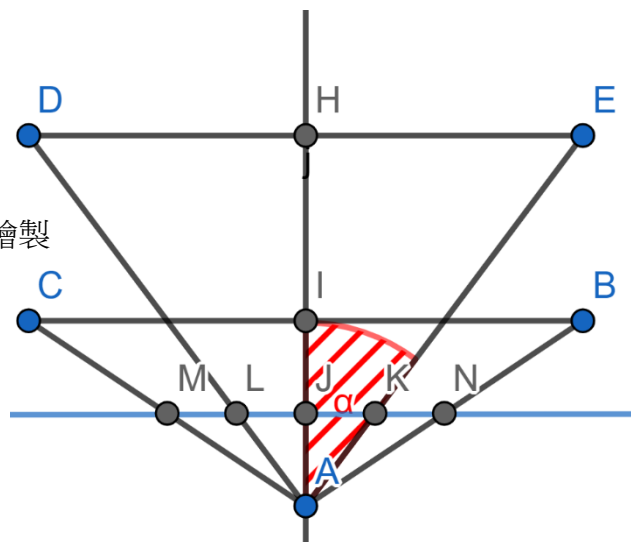


圖十四，註:自行繪製，A 為車子的鏡頭， $\overline{BC}$ 為 1280x720 的框之對角線，

$$\angle CAB = 120^\circ, \angle \alpha = 60^\circ, \overline{BC} = 1468 \text{ 畫素} \Rightarrow \text{距離車約 } 424 \text{ 畫素}$$

用手或人的邊界框在圖的大小根據比例換算距離，並以長寬 1280x720 的邊界框(其對角線約為 1468 畫素)當作操作人距離車最近時的邊界框，並經由圖十四算出距離車約是 424 畫素。

在下圖十中藍線是螢幕， $\overline{DE} = \overline{CB}$ 都是目標物的實際的對角線長度， $\overline{MN}$ 和 $\overline{LK}$ 是顯示在螢幕的對角線長度，也就是螢幕上看見的邊界框的對角線。對角線在螢幕上越長



圖十五。註:自行繪製

代表距離車就越近，而他們之間是反比關係，經由圖十五及下列數學式證明。

$$\overline{HE} = \overline{IB}$$
$$\tan \alpha = \frac{\overline{HE}}{\overline{AH}} = \frac{\overline{JK}}{\overline{AJ}} \Rightarrow \overline{HE} \times \overline{AJ} = \overline{AH} \times \overline{JK}$$

而 $\overline{HE}$ 是目標物的實際大小是定值， $\overline{AJ}$ 是從鏡頭中的透鏡到感光元件再顯示在螢幕上的對角線長度，即使會經過一些折射和處理，但用同一組設備時會是固定的。可知

$\overline{HE} \times \overline{AJ}$ 是定值。從而得到 $\overline{AH} \times \overline{JK} = \text{定值}$ 。也就是螢幕上邊界框的對角線長 $\times$ 離車的距離 = 定值，也就是他們成反比。

因為距離車最近的框是對角線長為 1468 畫素，離車的距離大約是 424 畫素，因此當出現其他對角線長度的框時就能用此算出它離車的距離，進而算出馬達要運轉的時間。如下：

y 是馬達要用轉的時間， x 是目標物的框之對角線長

$$y = \frac{1468}{x} \times 424 \times 1.016$$

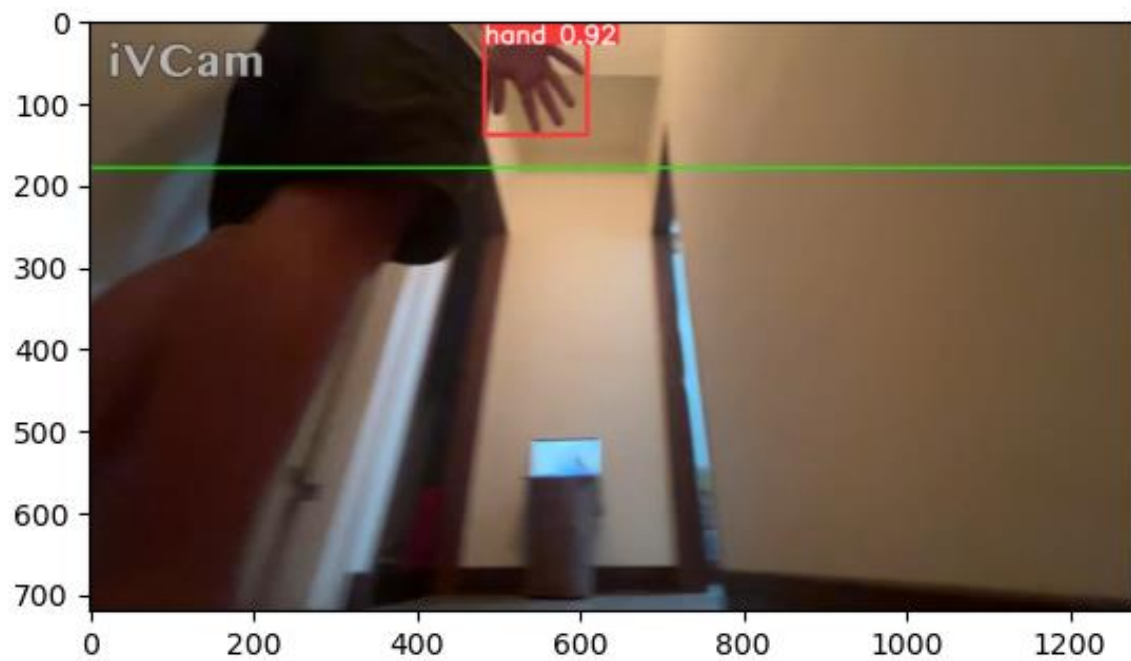
這套方法大致能準確計算出馬達適合的運轉時間，但因為能拍攝 120° 視野的鏡頭是廣角鏡頭，所以當目標物在邊緣時就會有較大的誤差。

後退的時間控制：

當用人體控制時，因為大約距離車 1.5 公尺就會開始後退，大多沒有目標物在邊緣的問題，所以可以用和前進相同的方法算出距離後退時馬達要運轉的時間。

不過用手勢控制後退時，因為目前做的運輸車很小，鏡頭的位置比人低很多，且需要用手控制後退時大多是人太靠近車，人車的水平距離很小，車和手掌的距離很多是垂直距離。如果用前進時的方法控制，會很不精準。所以我們依據手在畫面中的縱座標決定馬達運轉時間，越上面就代表越靠近人，就要退後越多。而移動多少時間的數學式是經過多次測試得出的結果。後來發現手在畫面中和圖片高度四分之一的線(圖十六

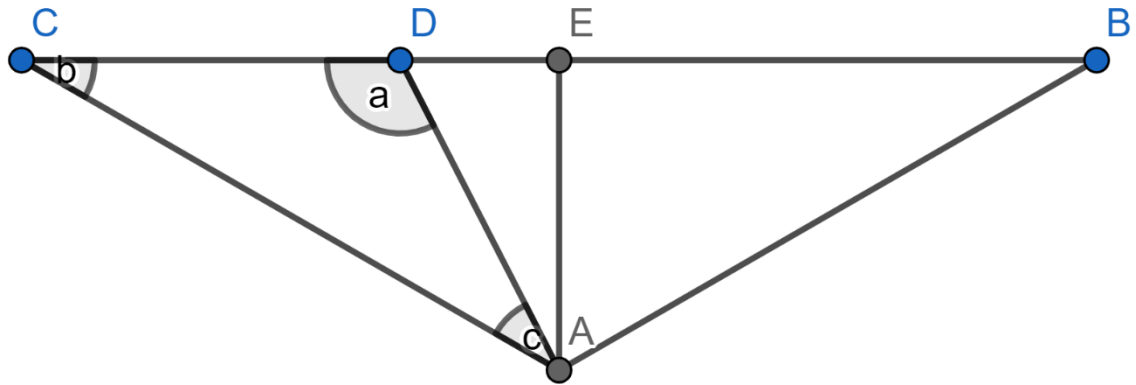
中的綠線)距離的平方，較適合用來當作後退時間的依據。



圖十六，註:自行拍攝

左右控制

不論是用手還是人體當目標物，左轉右轉的控制都是用下述方法算出目標是位在運輸車的左或右幾度角，再轉換成馬達轉動時間。圖十七的 A 是車上的鏡頭， D 是目標物的橫坐標。



圖十七，註:自行繪製

$$\angle BAC = 120^\circ \wedge \angle EAC = 60^\circ \wedge \angle b = 30^\circ \wedge \overline{BC} = 1280$$

$$1280 \div 2 \div \sqrt{3} = \overline{AE} = \frac{640}{\sqrt{3}}$$

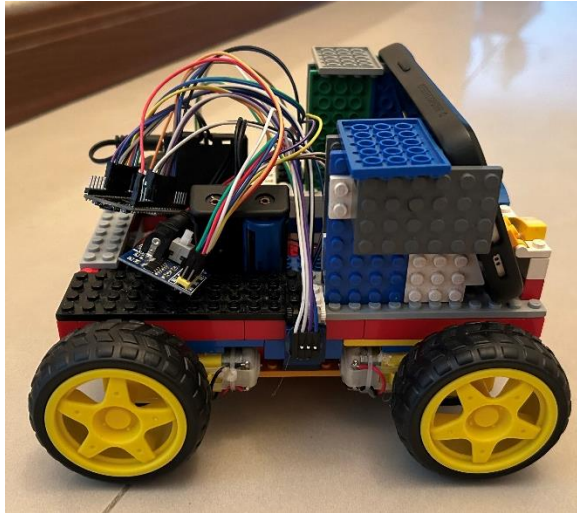
$$\tan \angle DAE = \frac{\overline{DE}}{\overline{AE}} = \frac{640}{\sqrt{3}} \Rightarrow \angle DAE = \tan^{-1}\left(\frac{\overline{DE}}{\overline{AE}}\right) = \tan^{-1}\left(\frac{\overline{DE}}{640}\right) = \tan^{-1}\left(\frac{\sqrt{3} \times \overline{DE}}{640}\right)$$

$$\text{結論 } \angle DAE = \tan^{-1}\left(\frac{\sqrt{3} \times \overline{DE}}{640}\right)$$

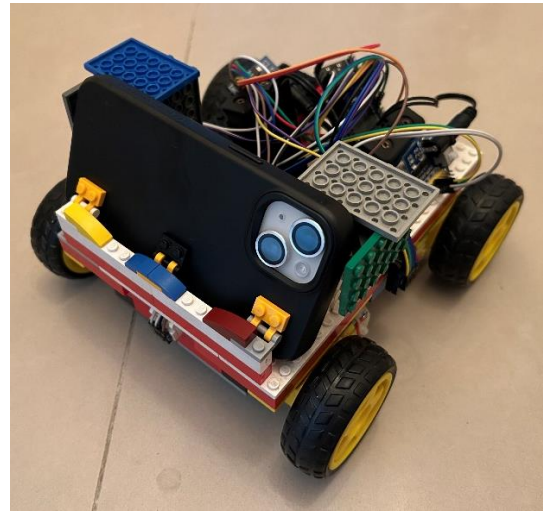
所以可由目標物距離左右邊界的距離算出 $\overline{CD}$ 得到 $\angle DAE$ ， $\angle DAE \times 15$ 就是馬達要運轉的時間(毫秒)。

Arduino 模擬運輸車

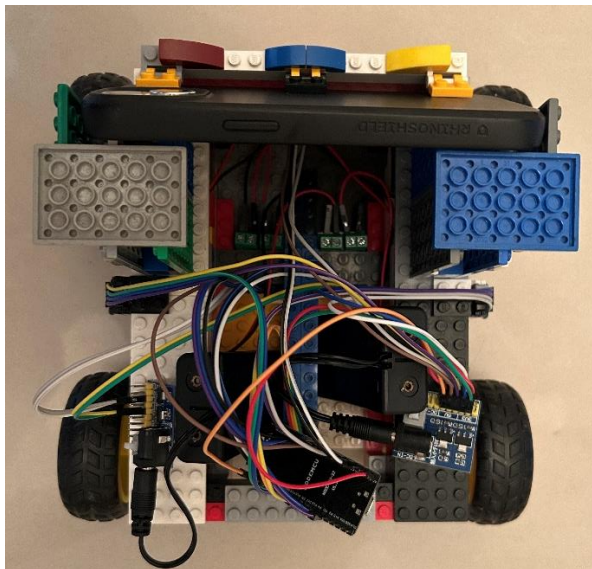
我們目前作出的是模擬運輸車，是縮小版。由 ESP32 控制板作為核心，控制 L9110s 馬達控制板帶動四個 5V 減速馬達，由四顆 18650 電池供電。車上載有手機當鏡頭，但視訊影像是用網路攝影機軟體 iVcam 回傳給電腦，不是用 ESP32 傳。車體如下。圖十八、十九、二十，圖二十一為車上的影像經過處理顯示在電腦上。



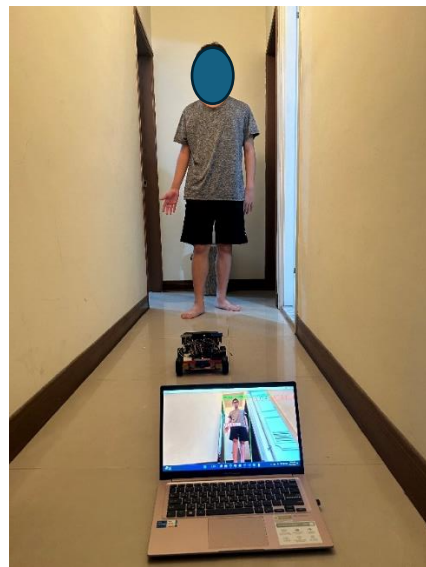
圖十八，註:自行拍攝



圖十九，註:自行拍攝



圖二十，註:自行拍攝



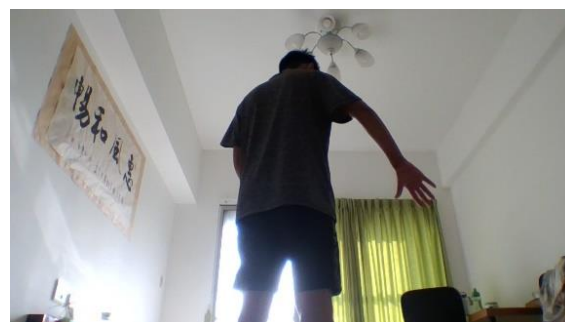
圖二十一，註:自行拍攝

三、研究結果與討論

1. 本研究的運輸車在小空間能做到靈活控制，像是在大約 1.2 公尺寬的走廊內，跟著人進行 180° 迴轉。這必須讓車子的鏡頭能在很小的水平距離拍到很高的目標物。因此我們必須採用手機的廣角鏡頭，而不是 arduino 的鏡頭模組或是 ESP32CAM。
2. 目前的運輸車因為採用 5V 減速馬達，所以速度不到人正常走的速度，我們只能透過計算人相對於車的角度，讓每次車轉向後能盡量對準人的方向，減少移動時間，並避免不必要的移動距離，或要轉兩次方向。若要做成符合實際場景和工具的運輸車，必須採用更好的馬達。
3. 研究過程曾發生 YOLO 模型不適應某些環境，可能因為影像辨識常有的問題-光線。其他可能原因包含訓練資料和實際影像的視角位置不同，以及環境雜亂。研究初期我們都是晚上在家裡拍訓練資料，也是在家中測試(圖二十二)，車子運作起來都很流暢。但環境改到學校之後，光線變得很亮（像圖二十三），運輸車常常會偵測不到目標物。於是研究後期我們廣泛蒐集訓練資料，包含像圖二十四中較雜亂或光線不同的環境。且蒐集的方式改成運輸車在跟隨人的同時拍照，因為這樣的視角才和真實運作時一樣。而且訓練資料才會有真實運轉時因為車子移動的晃動造成模糊(如圖二十五)，在真實運轉中才能辨識成功。



圖二十二，註:自行拍攝



圖二十三，註:自行拍攝



圖二十四，註:自行拍攝



圖二十五，註:自行拍攝

4. 未來如果我們再加上 YOLO-pose 模型，將能更精準的辨識人的動作，和人作更好的互動。

四、結論與應用

我們做出的運輸車已經能順暢跟隨人體，算是成功做出一款具備人體跟隨能力的智慧運輸車。且我們在兩個不同的環境測試過，都能順利運作。

就應用層面來說，智慧運輸車可作為醫院的智慧點滴架、醫務車或病床，幫助醫護人員減少推拉設備的體力消耗；也可用於零售場景，作為智慧推車或貨物籠車，協助員工在大賣場進行商品補貨等任務；也可用來彌補固定路線的搬運系統中的突發狀況。未來，此技術有潛力擴展至更多高需求的搬運任務，提升整體工作效率和協作能力。

五、參考資料與附錄

1. 醫務車照片 <https://www.conplus.com.tw/edcontent.php?lang=tw&tb=4&cid=6&id=39>
2. 油壓推車照片 <https://www.liftruck.com.tw/zh-tw/product-46386/%E6%B2%B9%E5%A3%93%E6%8B%96%E6%9D%BF%E8%BB%8A-HPT-20-30.html>
3. 籠車照片 <https://www.qiwotek.com.tw/product-detail-1060340.html>

4. 病床照片 <https://ulhouse.com.tw/product/bl-bhr61/>
5. 點滴架照片 https://renyumeco.com.tw/product_1579163.html
6. 樹梅派功耗 <https://www.icshop.com.tw/products/368030501074>
7. ESP32 功耗 <https://docs.ESPRESSIF.com/projects/ESP-idf/en/stable/ESP32s3/api-guides/current-consumption-measurement-modules.html>
8. YOLO 模型性能表 <https://github.com/ultralytics/ultralytics>

【評語】100041

1. 本計畫開發一個以影像處理與人工智慧方式開發的自動跟隨機器人，國內外相關之文獻與原型產品已有不少多，建議可加強文獻與產品回顧，並說明差異性，以確立本計畫的創新性與獨特性。
2. 系統之設計與實驗之說明尚屬完整，但有關機器人控制與自動跟隨機制的說明則略有不足，可更具體論述與分析。
3. 現階段測試多以人影正面為主，抓取臉部特徵，但大多數跟隨狀態為跟隨背影，應確認資料和 YOLO 演算法在此狀態下可正確跟隨，尤其在多人環境中。
4. 可考慮導入固定物和移動物的動態避障機制，這是跟隨機器人必然會碰到的情境。