

# 國立台北大學資訊工程學系專題報告

## Wearable Obstacle Detection for the Visually Impaired with Attention-Based Deep Learning

注意力為基礎之深度學習的視障者穿戴式障礙物檢測

專題編號:PRJ-NTPUCSIE-111-013

執行期間:2022 年 9 月 至 2023 年 5 月

### 摘要

在這個現代化的時代，人工智能的發展在快速增長。其中一個重要的 AI 發展是電腦視覺系統。這個系統使用深度學習演算法來分析視覺資料並識別臉孔、物體和其他特徵。因此，在這個專案中，我們決定製作一個針對視障者的可穿戴設備。雖然我們知道視障者通常有自己的白色手杖（棍子）來識別他們的周圍環境，但是透過這個設備，視障者可以提早得知他們即將穿越的物體。

這個設備使用 Raspberry Pi 4B 和一台攝影機，攝影機連接到 Raspberry Pi，用於捕捉環境的實時影像，並使用 YOLOv5 [1]來檢測和分類圖像中的物體。我們還添加了一個注意力機制（CBAM）到 YOLOv5，以增加物體的訓練準確性。用於此專題所偵測的物體包括階梯、地上坑洞、三角錐和花盆。在 Raspberry Pi 上，我們還添加了一個 Espeak 語音庫[2]，這個模組用於從文本生成語音。當偵測到物體時，它會將物體的名稱發送到 Raspberry Pi，並且這個模組會從接收到的文本（物體名稱）生成語音，以告知視障者他們前方將接近的是什麼。

### 1. 簡介

#### (一)研製背景

隨著時代演進，路上的公共建設不斷的增加，也代表著對用路人的障礙物就越來越多，很有可能會一個不注意就絆到跌倒，對於一般人來說已經要隨時注意路上的情況了，更不用說是對視障者來說了，走在路上就跟掃地雷沒兩樣，有位學者(Boyce)曾經發表了一個議題，10%的跌倒與視力異常是有直接相關[3]。

我們決定開發一個可穿戴裝置，以幫助視障者更快速、更精準地辨別在日常生活中需要小心的路況。這個裝置能夠實時檢測行走路線上的障礙物並立即向佩戴者回報。透過這個裝置，視障者能夠更自信地行走在道路上，並且更有效地避開可能造成危險的物品或地形。我們相信這樣的科技發明將會對視障者的生活帶來實質上的幫助和改善。

#### (二)研究目標

這個專題旨在利用 Raspberry pi 與 YOLOv5 的整合，讓使用者能夠穿戴裝置進行即時物件偵測。這個裝置將使用深度學習技術來辨識攝影機拍攝到的場景中的物體，並將結果透過耳機或小型音箱回饋給使用者。

YOLOv5 是一個目前應用廣泛的

物件偵測模型，它的精度和速度都非常優秀。樹莓派是一款小型而強大的計算機，能夠提供足夠的運算資源來執行這樣的模型。使用者可以將這個穿戴式裝置帶在身上，利用攝影機捕捉周圍的場景，並即時辨識出場景中的物體。當偵測到特定物體時，系統會通過耳機或小型音箱向使用者回饋偵測結果。例如，當使用者走在路上時，系統可以偵測到階梯、地上坑洞、三角錐、花盆等物體，並透過耳機告知使用者。

這裝置可以提供一種更加智慧化、便捷且安全的體驗，使使用者能夠更好地掌握周圍環境，並快速做出反應。透過深度學習技術的應用，這個穿戴式裝置有著無限的應用前景，可以為人們的生活帶來更多便利。

### (三)預期成果

這個穿戴式裝置的預期成果是能夠提供一種智慧化、便捷的體驗，讓使用者能夠更好地掌握周圍環境，並快速做出反應。透過深度學習技術的應用，這個穿戴式裝置可以幫助人們在日常生活中更輕鬆地識別物體，讓使用者能夠更加安全地進行各種活動。

## 2. 專題進行方式與技術

此專題使用了 YOLOv5 的影像識別技術再將其訓練模型用 CBAM 注意力機制改良後，再與樹莓派硬體進行結合，讓使用者能裝備在身上進行偵測，以達到能提前告知前方物件是什麼給使用者，讓使用者能提前做應對為目標。

### (一) YOLOv5

YOLOv5 是一種基於深度學習的

物件檢測算法，是影像辨識的一種方法，我們可以利用自己所蒐集的圖片，建成一個數據集，再使用 YOLOv5 的模型對此數據集進行訓練，訓練出一個用來偵測的偵測模型檔，使用此模型檔來對數據集蒐集的圖片種類進行偵測。

此專題所偵測的主要目標總共有四個，階梯、地上坑洞、三角錐、花盆，我們蒐集這四種類別的照片，組成一個有七千五百張左右的照片的數據集來進行訓練，再將數據集分為訓練集、驗證集和測試集來訓練，進而提升模型的泛化能力。

### (二) CBAM Attention Mechanism

CBAM Attention Mechanism 是一種用來提高卷積神經網路效能的注意力機制，CBAM 注意力機制可以提高模型的準確率和定位準確度，我們藉由更改原 YOLOv5 的模型，將 CBAM 添加進模型裡，且設定訓練時使用我們所更改的新模型，就能增加訓練出來的偵測模型檔的準確度上升。

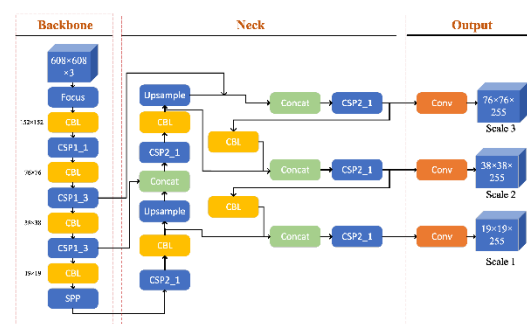


Fig.1 YOLOv5 原模型架構[3]

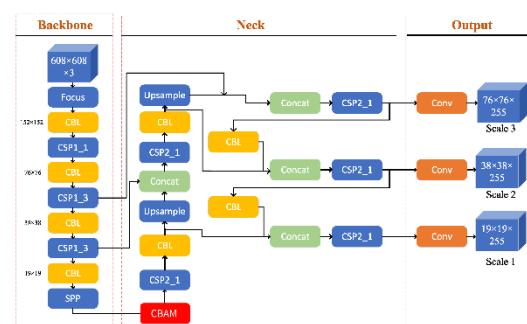


Fig.2 我們改良加入 CBAM 後的架構

以上是改良的簡單示意圖，經過此改良，我們可以在偵測時得到更好的準確率。

CBAM 注意力機制的運作包含兩個部分，通道注意力(Channel Attention)、空間注意力(Spatial Attention)，分別用於對通道和空間維度進行特徵提取和加權，這兩種注意力可以各自獨立或結合做使用，以下為 CBAM 注意力機制的運作圖。

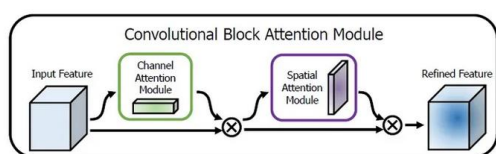


Fig.3 CBAM 運作圖[4]

### I. Channel Attention

通道注意力主要用於對不同通道的特徵進行加權，以提高模型的準確率，簡單來說，通道注意力是透過計算每個通道的重要性，以加權的方式將不同的通道的特徵融合在一起，使得模型可以更關注對目標有用的特徵，從而提高準確率。

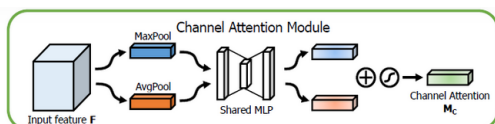


Fig.4 通道注意力運作示意圖[5]

### II. Spatial Attention

空間注意力主要是對不同的區域的特徵進行加權，以提高模型的定位準度，跟通道注意力類似，但空間注意力是透過計算每個區域的重要性，再用加權的方式將不同的區域的特徵結合，可以使得模型更關注於對目標有用的區域，從而提高定位準度。

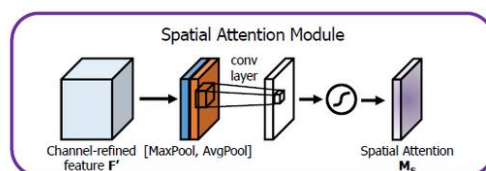


Fig.5 空間注意力運作示意圖[4]

### (三) 樹莓派與 Espeak 語音庫

為了將 YOLOv5 能從電腦上使用轉變成能攜帶在身上做使用，我們使用了樹莓派，樹莓派是一款微型計算機，也可以說他就是一台小型的電腦，只要做一些外部的設計，就能攜帶在身上做使用。

我們需要在電腦先訓練好模型，且把我們需要用到的 YOLOv5 程式寫好後，將檔案放進樹莓派上面執行，樹莓派上也要事先安裝 YOLOv5 所需要使用的一些函式庫，才能讓樹莓派完整的正常運作 YOLOv5。

為了能將偵測到的物件告知給使用者，我們也找到了樹莓派內的語音合成庫，Espeak 語音庫，利用此語音庫，搭配硬體，與我們程式的設定，就能在偵測到物件時，第一時間的告知給使用者，讓使用者能做應對。

## 3. 主要成果與評估

為了讓視障者能快速偵測到路面影像，我們製作了前背式的鏡頭(如



Fig.6)。

Fig.6 前背式鏡頭

我們先針對了 YOLOv5 的各個模型大小進行了訓練來找到最符合我們預期的偵測時間及準確度的模型，YOLOv5 原生的模型有 n、s、m、l、x 五種，我們針對前四種去做比對，訓練出來的偵測時間分別為 n 為 0.2 秒、s 為 0.35 秒、m 為 0.7 秒、l 為 2 秒(如 Fig.7)，因 l 的模型已經是 2 秒才能偵測一 Frame 的情況，x 的模型又比其更大所以判斷在時間上不利於我們做使用，n 跟 s 的模型雖然跑得快，但在我們測試完後，發現準確度並不高，會有很多的誤判產生，因此我們最後決定使用 YOLOv5m 的模型來做 CBAM 的添加。

| Class | Yolov5n | Yolov5s | Yolov5m | Yolov5l |
|-------|---------|---------|---------|---------|
| Time  | 0.2秒    | 0.35秒   | 0.7秒    | 2秒      |

Fig.7 YOLOv5 各個大小的模型進行訓練測試各模型的偵測時間

我們將使用添加了 CBAM 的 YOLOv5 用來檢測其性能，我們使用了七千五百張左右的照片作為數據集，且將其分為 7:2:1 的訓練集、驗證集、測試集來做檢測模型訓練，照片的輸入大小利用 YOLOv5 自動全部改為 640 \* 640 做訓練，如果用 320 \* 320 或是更小尺寸的輸入，可能會造成準確率降低，而照片解析度若要對較細節的小物件做標籤的話建議用高解析度的照片以便 YOLOv5 判斷，若物件較大的話可以盡量壓低解析度，讓 YOLOv5 能判斷到此物件就好，壓低解析度可以提升訓練出來的模型的偵測效率。

訓練出來的結果在使用 IoU 閾值為 0.5 的情況下測試後顯示(如 Fig.8 和 Fig.9)，對於階梯、地上坑洞、三

角錐以及花盆這些障礙物的偵測，使用 CBAM 訓練的準確率分別是階梯為 91%、地上坑洞為 83%、三角錐為 98%、花盆為 96%，整體準確率為 92%，相比原版的準確率 90%提升了約 2%左右的準確率。

再從線性圖形上來看(如 Fig.10 和 Fig.11)，原版 Yolov5 的準確率、召回率、和平均準確度較為凹凸不平，也就是它訓練出來的每一筆資料性能好壞差異較大，比較不平均。而加了 CBAM 後可以看到圖形變得較為平滑，每一筆的資料性能差異較小，比較少出現極端值的部分。從這裡可以發現，Yolov5 加了 CBAM 後性能有所提升。

但由於不同環境的光線及角度差異，實際測試的結果準確率可能會下降至七八成，主要問題發生在有誤判的情形。另外，目前我們所能辨識的障礙物種類相對於實際路徑上出現的各種大小障礙物來說還是較少。

| Class | Precision | Recall | F1-score | Accuracy |
|-------|-----------|--------|----------|----------|
| All   | 0.905     | 0.898  | 0.901    | 0.921    |
| 樓梯    | 0.869     | 0.866  | 0.867    | 0.897    |
| 坑洞    | 0.84      | 0.868  | 0.853    | 0.895    |
| 三角錐   | 0.98      | 0.94   | 0.959    | 0.945    |
| 花盆    | 0.919     | 0.859  | 0.888    | 0.92     |

Fig.8 原版訓練出的偵測模型

| Class | Precision | Recall | F1-score | Accuracy |
|-------|-----------|--------|----------|----------|
| All   | 0.927     | 0.903  | 0.913    | 0.938    |
| 樓梯    | 0.913     | 0.875  | 0.885    | 0.902    |
| 坑洞    | 0.836     | 0.866  | 0.849    | 0.896    |
| 三角錐   | 0.98      | 0.943  | 0.961    | 0.95     |
| 花盆    | 0.962     | 0.851  | 0.919    | 0.921    |

Fig.9 使用 CBAM 訓練出的偵測模型測試數據

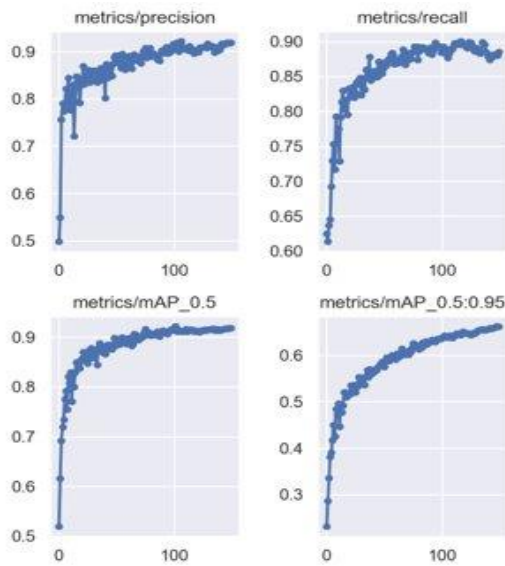


Fig.10 原版 YOLOv5

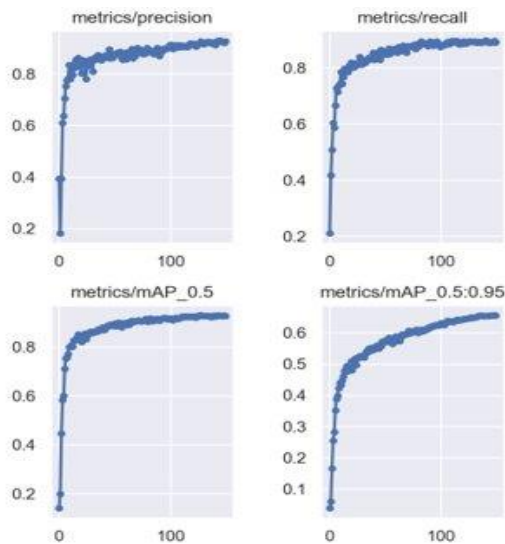


Fig.11 YOLOv5m+CBAM

## 4. 議題探討

### (一)樹莓派 32Bit OS 版本資源少

在上半學期的實作過程中，我們使用了樹莓派 32 位元的作業系統，但遇到了許多問題，特別是在安裝 YOLOv5 所需的函式庫。在網路上搜尋時，我們發現沒有這個版本的資源可以供我們安裝和使用，包括 torch、torchvision 和 numpy 等庫，不斷出現版本不相容的問題。嘗試了各種方法

和網路上的建議，但仍然無法解決問題。

最後，我們發現有些人使用 64 位元版本的樹莓派作業系統和與我們相同型號的硬體，他們成功地解決了這個問題。因此，我們決定重新安裝 64 位元版本的作業系統。這一次，我們輕鬆地安裝了所需的所有函式庫，包括 YOLOv5 原版的偵測程式，它可以順利運行。這次經驗讓我們認識到了選擇正確的作業系統和硬體配置對於順利完成任務的重要性。我們也學到了如何在網路上尋找解決問題的方法和資源，以及與他人交流經驗和知識的重要性。

### (二)YOLOv7 + CBAM 的偵測效率太差

在解決了樹莓派版本兼容問題後，我們也同時處理了 YOLOv7 偵測的問題。

原先，我們是打算使用 YOLOv7 作為我們專題所使用的物件檢測算法，隨後，我們將我們自己訓練出的 YOLOv7+CBAM 模型的訓練集和偵測檔放入了樹莓派中進行實時偵測。然而，我們發現偵測效率非常低落，每張照片需要約 5 秒的時間才能完成偵測，幀率十分之低。

為了改善這個問題，我們決定改用 YOLOv5 版本，因 YOLOv7 原先內建的模型過大，使得樹莓派的硬體條件不足以支撐在使用 YOLOv7 的模型情況下做實時偵測，YOLOv5 就能以較輕量化的模型，達到與 YOLOv7 版本相近的偵測準確度，又能提高偵測的影格速率。

我們同樣的在 YOLOv5 裡面加入了 CBAM 模型和自己的訓練集，同

時將偵測照片的解析度降低了一定程度。最後，我們成功地實現了約 0.7 秒的快速偵測反應速度。

這次的經驗讓我們認識到模型設計和參數設置對於偵測速度的影響。同時，我們學會了如何優化模型和調整參數以達到更好的效果。

## 5. 結語與展望

為了能夠更方便視障者偵測障礙物，我們需要改進操作方式和偵測技術。未來，我們希望能夠提高障礙物辨識的準確率和能夠辨識的障礙種類，同時增加 app 的操作功能以方便視障者使用。

## 6. 銘謝

感謝指導教授在我們設備上有需求時給予了幫助，在我們遇到不了解的問題時，能在開會中與老師討論及研究可行方法。

## 7. 參考文獻

[1] Fangbo Zhou; Huailin Zhao; Zhen Nie. Safety Helmet Detection Based on YOLOv5. Shenyang, China, 01 March 2021

[2] Dexter Industries; How to Make your Raspberry Pi Speak Out Loud: Give It a Voice!

[3] 詹廖明義著，“視力檢查對跌倒高危險群病人的重要性”，2018

[4] Zhu, L.; Geng, X.; Li, Z.; Liu, C. Improving YOLOv5 with Attention Mechanism for Detecting Boulders from

Planetary Images. Remote Sens. 2021, 13, 3776. [\\_](#)

[5] Woo, S., Park, J., Lee, JY., Kweon, I.S. (2018). CBAM: Convolutional Block Attention Module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science(), vol 11211. Springer, Cham.