2023 年臺灣國際科學展覽會
優勝作品專輯

作品編號　**190005**

參展科別　電腦科學與資訊工程

作品名稱　軌道安全，唯快不破-高效能 AI 軌道異物偵測系統設計之研究

得獎獎項

就讀學校　桃園市政府教育局高級中等教育科

指導教師　謝禎冏、羅堅秩

作者姓名　羅心樂

關鍵詞　軌道異物偵測、深度學習、YOLO

作者簡介

大家好，我是羅心樂，很高興可以再度參與 2023 年臺灣國際科展，再研發的過程中我 學習到許多全新的知識，感謝國際科展讓我們可以在這分享我們的研發結果與心得，同時也 感謝未來之星營隊以及聯發科給予許多上台發表的機會與建議，也感謝大同大學謝禎冏教授 與資訊工程系研究生的指導與協助，同時也非常感謝臺鐵、高鐵、北捷、輕軌在百忙中抽空 讓我有機會與相關主管人員進行交流與參訪，最後感謝家人一路上的支持，希望這次研發的 技術可達到為快不破高效能的目標，同時也可以運用於真實環境，為軌道安全盡一份心力！ 也藉由國際科展增加自己的視野。

摘要

臺鐵太魯閣號於 2021 年 4 月撞擊滑入軌道的工程車的事故，是 60 年最嚴重一場意外。北捷文湖線也曾有大型招牌掉落事件，顯示軌道安全的重要性。本研究參訪高鐵、臺鐵、北捷和新北捷-淡海輕軌，將四大軌道公司的異物偵測系統做探討。採用 Yolo 系列物件偵測演算法，進行模型訓練，設計一套「高效能 AI 軌道異物偵測系統」。將攝影機架設在車頭，並加裝望遠鏡頭，達到遠距離的預警。採用可見光攝影機與 AI 物件偵測的技術，並應用內嵌系統 Jetson TX2，讓列車提前確認是何異物，提升安全性，採取不同煞車措施，降低誤點率。以台北捷運文湖線為實驗場域，測試各種天候條件，如：晴天、雨天、傍晚等。也在不同場域實測如：臺鐵內灣線、淡海輕軌。本系統平均準確率 95% mAP 與運行的幀率達 40FPS，能縮短辨識時間，讓駕駛能立即反應和提前預警，達到保障人車安全的目的。

The accident in April 2021, when the Taiwan Taroko Express train collided with a construction truck that slid into the track, was the worst accident in 60 years. There was also a large signboard drop incident on the Taipei Metro Wenhu Line, showing the importance of rail safety. This research visited the Taiwan High Speed Rail, Taiwan Railway, Taipei Metro, and New Taipei Metro-Danhai Light Rail, and compared the track intrusion detection systems of the four major rail companies. Using the Yolo series object detection algorithm, model training is carried out, and a set of "high-performance deep learning based track intrusion detection systems" is designed. Install the camera on the front of the train and a telephoto lens to achieve long-distance early warning. Using optical camera, AI object detection technology, and using the embedded system Jetson TX2, the train can confirm in advance what foreign objects are, improve safety, and design different braking stragedy to reduce the delay rate. Take the Wenhu Line as the experimental site to test various weather conditions, such as: sunny, rainy, evening, etc. Taiwan Railway, Neiwan Line, and Danhai Light Rail are also involved in various experimental fields. This system has high performance and frame rate: 95% mAP and 40FPS, which can shorten the recognition time, allow drivers to respond immediately and give early warning, and achieve the purpose of ensuring the safety of passengers and trains.

壹、研究動機

臺鐵太魯閣號在 2021 年 4 月 2 日撞擊滑落邊坡工程車，這是 60 年最嚴重的意外，造成 49 人死亡和 213 人輕重傷，也是讓人難忘的傷心事故 (嚴文廷、曹馥年、林雨佑，2021)，如圖 1、圖 2。從 1911 年起到現在臺鐵總共有 80 多場事故，幾乎是每年發生一次。

圖 1 太魯閣號撞上工程車意外
(圖片來源：林銘鋒@阿美族的歌)

圖 2 受損太魯閣號
(圖片來源：聯合新聞網站)

繼太魯閣號事故後，又接連發生火車撞擊軌道異物的事故，如圖 3。2015 年北捷文湖線也曾發生大型招牌掉落事件 (郭逸，2015) 如圖 4。這些因素讓我意識到軌道安全的重要性，因此希望運用 AI 深度學習中物件偵測(Object Detection)的技術來做辨識，達到高效能預警的目標。

圖 3 平溪線列車撞擊巨石
(圖片來源：聯合報)

圖 4 捷運文湖線列車巨幅廣告招牌
(圖片來源：自由時報)

貳、研究目的及研究問題

一、研究目的:

1. 了解各軌道公司的異物偵測系統與措施。
2. 運用不同 YOLO 演算法來提升辨識的準確性與速度。
3. 採用可見光攝影機和內嵌式系統來辨識異物，降低成本。
4. 加裝望遠鏡頭，讓列車可以提前預警，達到保障人民、軌道安全的目標。

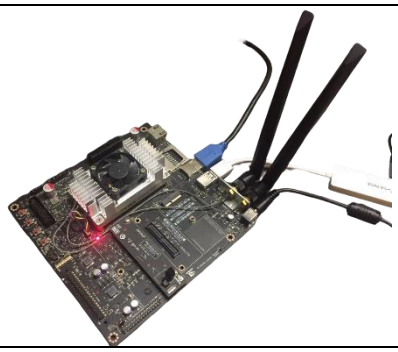
二、研究的問題:

1. 針對台鐵、高鐵、北捷、新北捷-淡海輕軌異物偵測系統做分析與比較。
2. 研究資料集(Datasets)對訓練模型效能的影響
3. 研究不同平台對訓練模型的影響
4. 研究物件偵測(Object detection)演算法的效能，如：YOLO(You Only Look Once)系列中的 YOLOv4、YOLOv4-Tiny、YOLOv6。
5. 分析內嵌式系統 Nvidia Jetson TX2 的在異物偵測的效能。
6. 針對北捷文湖線場域條件進行研究。
7. 研究不同大小物體、不同位置、與攝影機間距離的辨識效果。
8. 在不同天候，如傍晚雨、陰天實測。
9. 實測在其他不同場域效果，如：臺鐵內灣支線、淡海輕軌。
10. 研究加裝望遠鏡頭的辨識效果，讓列車有足夠煞車距離。

參、研究設備及器材

一、實驗器材

本研究為達高效能異物偵測目的，在列車高速行駛下，能夠取得清晰的影像畫面，採用高速攝影機。因 GoPro Hero7 Black 運動型攝影機以拍攝極限運動為主，有手持式防手震的技術，故為本研究採用。本次實驗器材族繁不及被載，僅列重要元件，如下：

		
圖 5 GoPro Hero7 Black (圖片來源：研究者自行拍攝)	圖 6 Jetson TX2 (圖片來源：研究者自行拍攝)	圖 7 伺服器 GPU 2080Ti RAM 11G (圖片來源：研究者自行拍攝)
		
圖 8 iphone 6 (圖片來源：研究者自行拍攝)	圖 9 筆電 (圖片來源：研究者自行拍攝)	圖 10 羅技網路攝影機 (圖片來源：研究者自行拍攝)
		
圖 11 螢幕 13 吋 (圖片來源：研究者自行拍攝)	圖 12 望遠鏡頭 10 倍 (圖片來源：研究者自行拍攝)	圖 13 Nvidia Xavier NX (圖片來源：研究者自行拍攝)

二、筆電硬體：

CPU: i7-8565U @1.8GHz

記憶體:8GB

GPU: NVIDA GeForce MX150 2G

三、伺服器：

CPU AMD Ryzen Threadripper 2950X

GPU 2080ti RAM 11G

四、使用軟體：

系統:：Windows 10

程式語言：python 3.8

機器學習庫：Pytorch 1.21.1

深度學習框架 ：tensorflow 2.4.1

深度學習神經網路庫： CUDA11.3

GPU 加速神經網路基元庫：cuDNN8.2

圖表生成： tensorboard、matplotlib

其餘擴充程式庫：numpy 1.1915

標註資料集：labelimg 1.8.6

肆、研究方法及進行步驟

一、研究流程概述

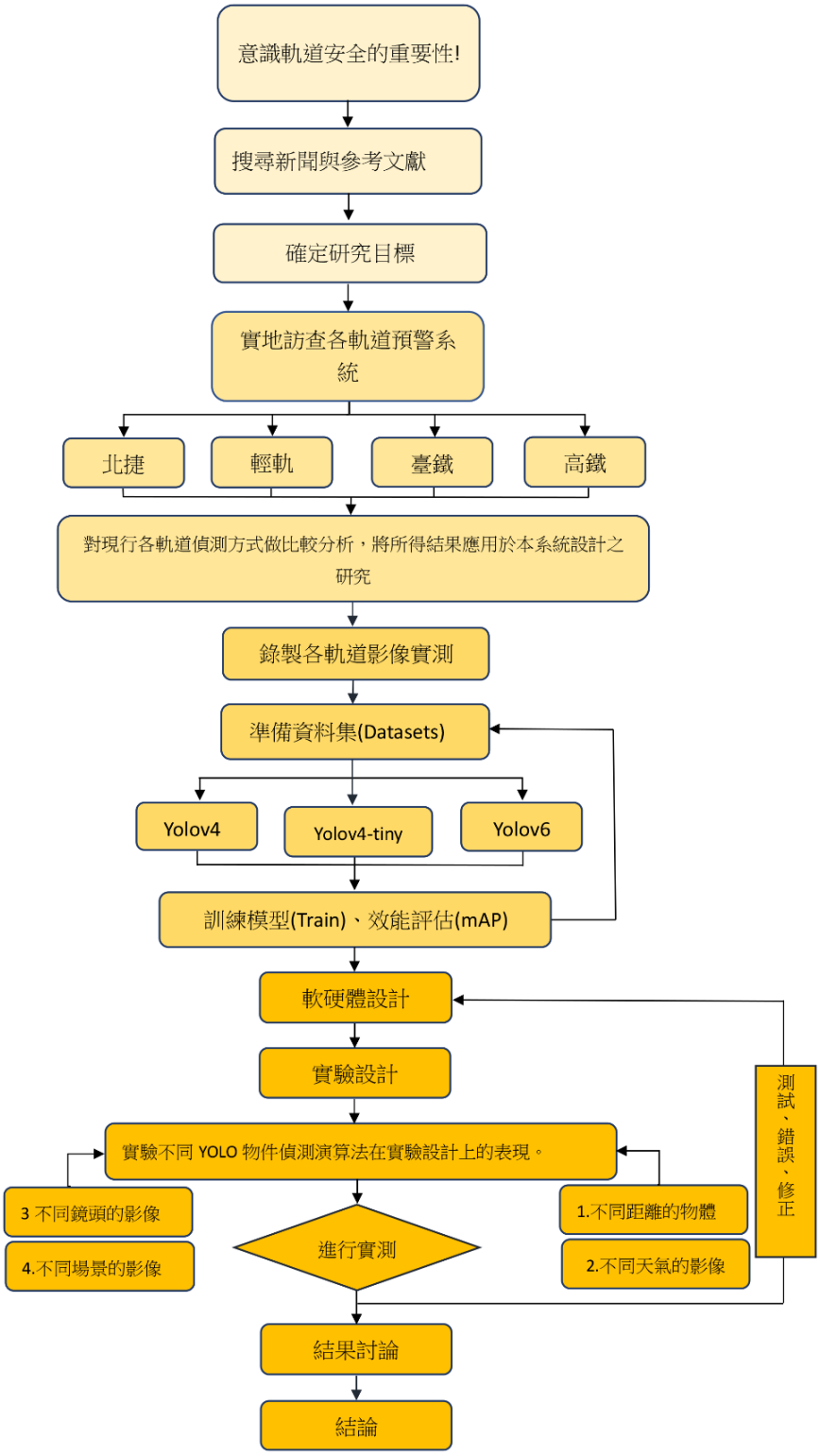
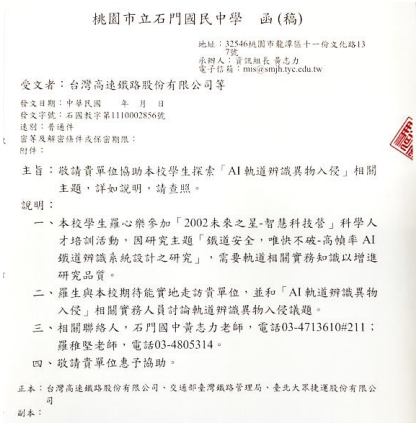


圖 14 研究架構圖
(圖片來源：研究者自行繪製)

(一)、實際訪查-透過電話、Email 和公文

本研究為能貼近現實，改善現行異物偵測方式的缺點，解決真實問題，對各軌道公司做電話，及 Email 訪查。但因資料多涉及各公司業務機密，也請學校多次行文，以利訪談順利，也為資料提供佐證，將公文摘要整理如圖 15、圖 16。



桃園市立石門國民中學　函(稿)

地址：32546桃園市龍潭區十一份文化路137號
承辦人：資訊組長 黃志力
電子信箱：mis@smjh.tyc.edu.tw

受文者：台灣高速鐵路股份有限公司等

發文日期：中華民國　年　月　日
發文字號：石國教字第1110002856號
速別：普通件
密等及解密條件或保密期限：
附件：

主旨：敬請貴單位協助本校學生探索「AI 軌道辨識異物入侵」相關主題，詳如說明，請查照。

說明：

一、本校學生羅心樂參加「2002未來之星-智慧科技營」科學人才培訓活動，因研究主題「鐵道安全，唯快不破-高幀率 AI 鐵道辨識系統設計之研究」，需要軌道相關實務知識以增進研究品質。

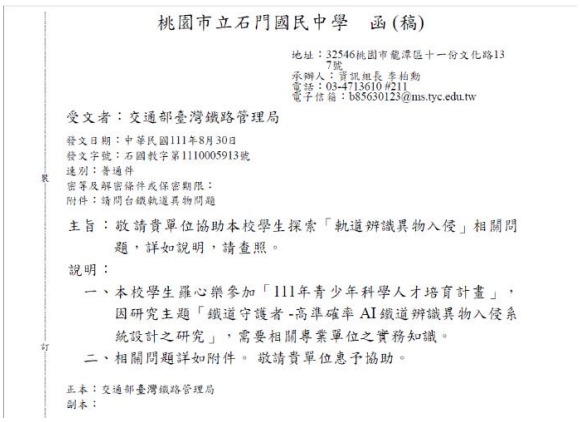
二、羅生與本校期待能實地走訪貴單位，並和「AI 軌道辨識異物入侵」相關實務人員討論軌道辨識異物入侵議題。

三、相關聯絡人，石門國中黃志力老師，電話03-4713610#211；羅雅堅老師，電話03-4805314。

四、敬請貴單位惠予協助。

正本：台灣高速鐵路股份有限公司、交通部臺灣鐵路管理局、臺北大眾捷運股份有限公司
副本：

圖 15 行文各軌道公司，以利訪查
(圖片來源：研究者自行拍攝)



桃園市立石門國民中學　函(稿)

地址：32546桃園市龍潭區十一份文化路137號
承辦人：資訊組長 李柏勳
電話：03-4713610 #211
電子信箱：b85630123@ms.tyc.edu.tw

受文者：交通部臺灣鐵路管理局

發文日期：中華民國111年8月30日
發文字號：石國教字第1110005913號
速別：普通件
密等及解密條件或保密期限：
附件：請問台鐵軌道異物問題

主旨：敬請貴單位協助本校學生探索「軌道辨識異物入侵」相關問題，詳如說明，請查照。

說明：

一、本校學生羅心樂參加「111年青少年科學人才培育計畫」，因研究主題「鐵道守護者 -高準確率 AI 鐵道辨識異物入侵系統設計之研究」，需要相關專業單位之實務知識。

二、相關問題詳如附件。敬請貴單位惠予協助。

正本：交通部臺灣鐵路管理局
副本：

圖 16 發文詢問軌道辨識異物問題
(圖片來源：研究者自行拍攝)

(二)、實地參訪-參訪臺鐵、高鐵、新北捷-淡海輕軌各軌道公司

為了解現行各軌道運輸公司的軌道異物偵測系統，先後訪談臺鐵、高鐵、北捷和新北捷運-淡海輕軌四大軌道運輸公司。在 4 月 28 日獲邀到高鐵的「台灣高鐵探索館」進行參訪與討論，見圖 17。新北捷運則在 9 月 13 日到淡海輕軌參訪，見圖 18。10 月 4 日到臺鐵的參訪「桃園平鎮區德育路口平交道障礙物偵測系統」，見圖 19，並獲各公司相關主管人員接待並交流。

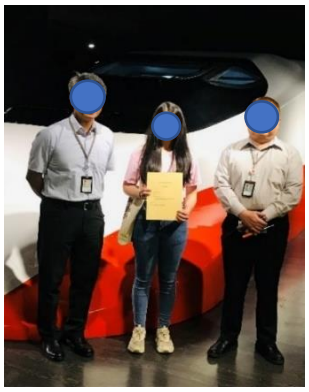


圖 17 與高鐵賴副理、吳長官合影(圖片來源：研究者自行拍攝)

圖 18 與新北捷運阮經理合影
(圖片來源：研究者自行拍攝)

圖 19 臺鐵戴主任講解
(圖片來源：研究者自行拍攝)

二、文獻探討

(一)、物件偵測(Object Detection)介紹

物件偵測演算法有 RCNN、Fast-RCNN、YOLO 和 SSD 等，以 YOLO 為目前影像物件偵測的主流方法，可運用於各領域的人工智慧視覺辨識中。

物件偵測分為兩大類：Two stage 和 One stage

Two stage: 先透過特殊方式選出物件如：Selective Search，選出過程稱為 Region Proposal，在進行影像辨識，但需要強大 GPU 平行運算，精確率高。

One stage: 物件的類別和位置偵測及物件辨識一步到位！這樣的做法速度很快，精確度就比 Two stage 略低了一些。

但整體辨識率仍在可接受範圍內，因此 One stage 的方法是目前比較多人研發用在行動裝置上的方法(Tommy Huang, 2018)。本計畫主要研究 One stage 架構的 YOLO 演算法，已達到即時辨識的目的。

YOLO(You only look once, 簡稱 YOLO)是關於物件偵測的類神經網路演算法 ，可框出影像中的每個物件，進行物件偵測、追蹤及判斷，見圖 20。目標檢測算法的思想首先就分為兩個步驟，第一個任務是定位、第二個任務則是分類。目標檢測就是要找出圖片中的物體，並且使用 Bounding box 將物體框出來。而 Bounding box 可以使用左上角的坐標(x,y)與矩形的寬和高(h,w)表示(李謦伊，2020)。

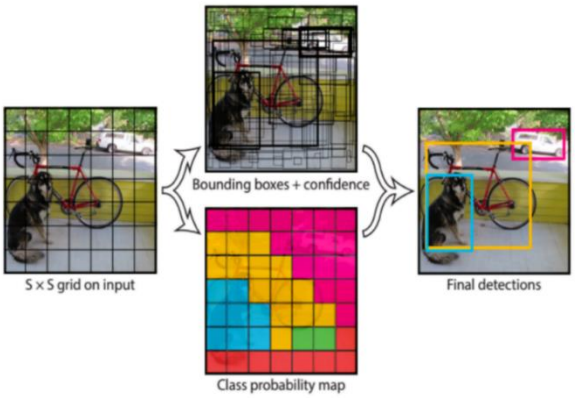


圖 20 Yolo 架構
(圖片來源：謦伊的閱讀筆記網站)

本研究也將對 YOLO 系列的演算法作探討，一般物件偵測主要結構分為 Input、Backbone、Neck 和 Head 等 (Alexey Bochkovskiy and Chien-Yao Wang and Hong-Yuan Mark Liao, 2020)，見圖 21，分別功能如下：

1. Input: 指圖片的輸入

2. BackBone: 在 ImageNet 預訓練的骨架

3. Neck: 通常用來提取不同層級的特徵圖

4. Head: 預測對象類別和 Bounding Box 的檢測器，通常分兩類 Dense Prediction (one stage), Sparse Prediction (two stage)

YOLO 演算法系列有 YOLOv4-Tiny、YOLOv4、YOLOv6 等，本研究希望能找到適合軌道辨識精準又快速的演算法。

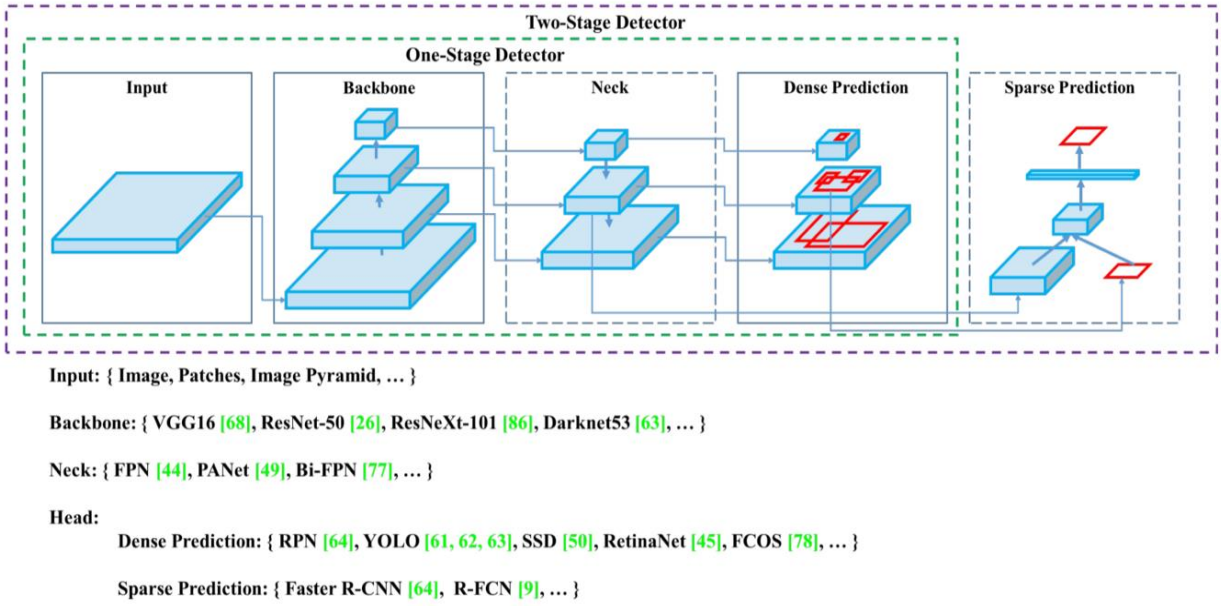


圖 21 YOLO 架構圖

(圖片來源：Alexey Bochkovskiy, Chien-Yao Wang and Hong-Yuan Mark Liao ，2020)

(二)、YOLOv4

YOLOv4 做了許多改進和創新方法，讓其能保證速度的，同時大幅提高模型的檢測精度，並降低硬體使用的要求(李謦伊，2021)。

1. YOLOv4 的架構為:

(1). Backbone: CSPDarknet53

(2). Neck: SPP+PAN

(3). Head: YOLO HEAD

2. BackBone 使用 CPSDarknet53，讓其參數量減少，運算量減少，進而提高準確率(張家銘，2021)。

3. Neck 使用 SPP+PAN，讓其提升局部特徵和全局特徵的融合，進而豐富最終特徵圖的表達能力(張家銘，2021)。

YOLOv4 提出許多創新的方法如：Mosaic 採用隨機縮放、裁剪的方式混合拼接 4 種圖片，可以豐富資料集，增加許多小尺寸的目標，使模型更穩健。也驗證了 SOTA (State of the Art) 的 Bag-of-Freebies 和 Bag-of-Specials 目標檢測方法，使其能保有一定速度，又有高的精度。YOLOv4 在 MS COCO 數據集上的 AP 值為 43.5% (65.7% AP50)。與 YOLOv3 相比，YOLOv4 的 AP 和 FPS 分別提高了 10% 和 12%。見圖 22。

YOLOv4-Tiny 比較輕量，只有兩層 yolo layer，參數量為 600 萬，YOLOv4 有三層 yolo layer，YOLOv4 約有 6000 萬。YOLOv4-Tiny 相對運算量少，推理速度較快。

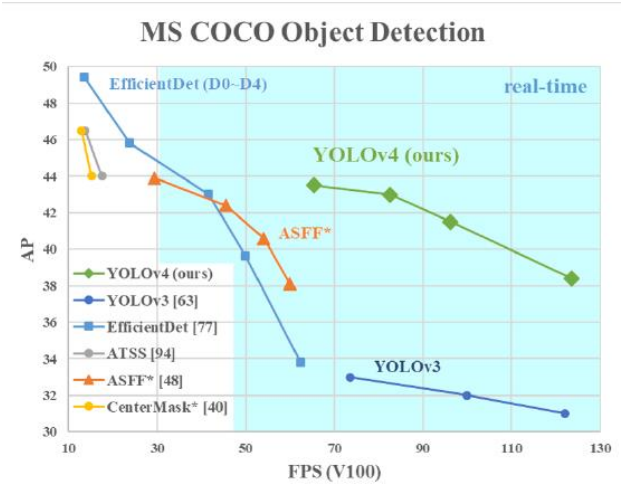


圖 22 YOLOv4 幀率與 AP 的比較圖

(圖片來源：Alexey Bochkovskiy et al., 2020)

(三)、YOLOv6

YOLOv6 在速度和準確率上有高的效能，在 COCO 資料集中，YOLOv6-N 版的平均準確率(mAP)有 35.9%，用 T4 幀率有 1234FPS；YOLOv6-S 的 mAP 為 43.5%，幀率為 495FPS，見圖 23 (Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, Xiaolin Wei, 2022)。

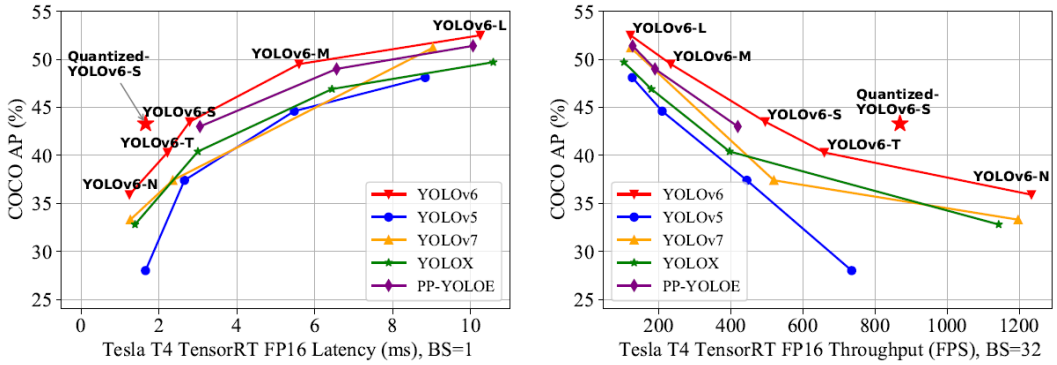


圖 23、YOLOv6 速度比較圖

(圖片來源：Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, Xiaolin Wei, 2022)

YOLOv6 基於硬體感知神經網路的觀點設計，引進 RepVGG 的觀念，特點如下：

1. YOLOv6 的網路架構，見圖 24:

   (1). Backbone: EfficientRep 能够高效利用硬體（如 GPU）運算力，同時具有較强的表徵能力。

   (2). Neck: Rep-PAN 更有效的特徵融合的網路結構。

   (3). Head: Efficient Decoupled Head 採用 Hybrid Channels 策略的解耦頭結構，可以维持精度，同時降低了延時。

2. 在訓練上，採用 Anchor-free 無錨方法，並以 SimOTA 標籤分配策略和 SIoU 邊界框迴歸損失，來強化偵測精確度。

YOLOv6 以硬體加速架構設計，考量 GPU 的運算能力，使其精度和速度上有高的表現，優於 YOLO 前幾代演算法。

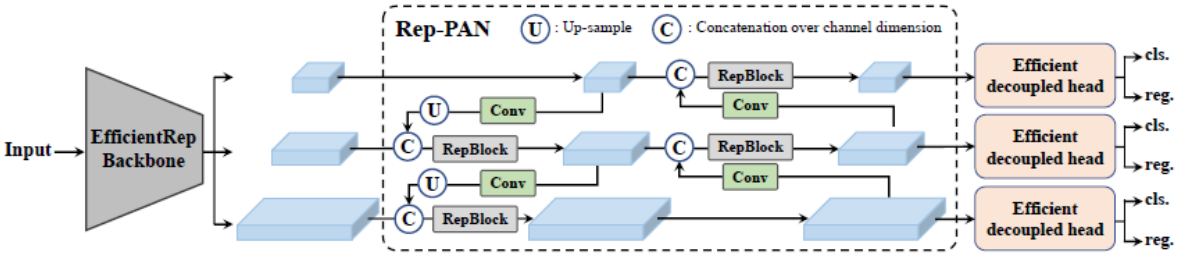


圖 24 YOLOv6 網路架構圖 (N/S 版; M/L 版的 RepBlock，以 CSPStackRep 代替)
(圖片來源：Chuyi Li et al, 2022)

(四)、IoU (Intersection over Union)

預測物體的 Bounding Box 與 Ground Truth 交集面積除以聯集面積， 如下式 (林雅雯、謝禎冏、黃維信、謝尚琳、洪瑋宏、李明德 ，2020)。

$$\mathrm{IoU}(A,B) = \frac{A \cap B}{A \cup B} \qquad \text{(式一)}$$

亦即是預測 Bounding Box 與物體實際位置的框重疊的比例。最常用的指標為 0.5IoU，表示一次性預測 Bounding Box 任務時，當預測 Bounding Box 的 IoU > 0.5 代表預測成功。

(五)、mAP (Mean Average Precision)

是衡量 AI 模型效能的重要指標，計算每個類別(Class)的精確率再加以平均，為各類別的平均準確率。通常使用 IoU0.5 作為判斷基準 (林雅雯等 ，2020)。

TP(c): True Positive in Class c 預測的類別是正確的而且重疊度高，亦即 Proposal 和 Ground Truth 相吻合。

FP(c): False Positive in Class c 預測的類別是錯誤的而且重疊度低，亦即 Proposal 和 Ground Truth 不吻合。

FN(c): False Negative in Class c 沒有被預測出來的類別數量。

在 Class c 的準確率(Precision)表示成下式：

$$\mathrm{P(c)} = \frac{\mathrm{TP(c)}}{\mathrm{TP(c)+FP(c)}} \quad \text{(式二)}$$

在 Class c 的召回率(Recall)表示成下式：

$$\mathrm{R(c)} = \frac{\mathrm{TP(c)}}{\mathrm{TP(c)+FN(c)}} \quad \text{(式三)}$$

AP 是 PR Curve 下的面積(李聲伊，2020)，P 為 Y 軸，R 為 X 軸所畫的曲線圖，一般式如下：

$$\mathrm{AP} = \int \mathrm{Pdr} \quad \text{(式四)}$$

mAP 為各類別的平均準確度表示如下式：

$$\mathrm{mAP} = \frac{1}{\mathrm{classes}} \sum \mathrm{c} \in \mathrm{classes}\ \ \mathrm{AP} \quad \text{(式五)}$$

(六)、混淆矩陣 (Confusion Matrix)

表 1、混淆矩陣

		True Condition	
		Positive	Negative
Predicted Outcome	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

混淆矩陣的四個指標 TP、FP、FN 和 TN，分別描述如下：

TP (True Positve)：預測「目標物件」正確，實際是「目標物件」，例如：預測是人，實際是人，預測正確。

FP(False Positive)：預測「目標物件」錯誤，實際是「非目標物件」，例如：預測是人，實際不是人，預測錯誤。

FN(False Negative)：預測「非目標物件」錯誤，實際是「目標物件」，例如：預測不是人，實際是人，預測錯誤。

TN(True Negative)：預測「非目標物件」正確，實際是「非目標物件」例如：預測不是人，實際不是人，預測正確。

在物件偵測中的意義分別為：

TP：IoU > 0.5 且同一個 Ground Truth 只能計算一次。

FP：IoU < 0.5 或檢測到同一個 Ground Truth 多餘的數量。

FN：沒有檢測到 Ground Truth.的數量。

混淆矩陣在衡量 AI 模型好壞的指標之一，通常在 mAP 不高情況下，我們會個別分析每個類別的混淆矩陣，作為調整修正資料集的參考。

(七)、幀率 (Frame Rate per Second)

Frames per second，簡稱 FPS（也有文獻稱之為 Frame rate, 幀率），是每秒幀數的意思 (王智偉，2019)。一般指影像播放的速度，越高的 FPS，畫面的播放則越流暢。人眼在 16-24FPS，就解讀為連續的畫面，一般短片在 24FPS 左右，電視畫面主流為 30FPS，遊戲為求流場會在 60FPS。在物件偵測則為一個影格處理的速度，亦即是一個畫面辨識的速度，與模型推論(Inference)速度有關。本研究所需的目標幀率 (Target FPS)與列車行駛速度有關，必須大於列車時速，通常換算成每秒列車走的距離，以公尺(m)為單位(林雅雯等 ，2020)，其關係式如下：

$$\text{Target FPS} > \frac{\text{列車時速 x1000}}{3600} \qquad \text{(式六)}$$

我們以此方式評斷 YOLO 演算法的速度，以符合系統異物偵測的需求。

(八)、MSCOCO (Common Objects in Context)

MSCOCO 是由 Microsoft、Facebook、CVDF 及 Mighty Ai 等組織所提供的一個大型開源圖片數據集，見圖 25 (COCO 官網)。包含超過了 33 萬張影像（其中有超過 20 萬張已標記），包含 150 萬個物件、並分成 80 個類別（for object detection）以及 91 類的 stuff（for semantic scene labeling）。相較 PASCAL VOC 和 ImageNet，影像內容較複雜，但更貼近日常生活的場景，見圖 26。COCO 對於 Person class 的標示相當準確，只要有人的部份皆會標示為 Person。我們只需花一點時間，另外加入一些針對專案需要的特殊場景等照片 ，就能快速建立一個數量龐大且完善的資料庫了(Jason Chen , 2020)。

本研究除了自行拍攝和網站搜尋的照片，也增加 COCO 資料集，建構自己的資料集來增加資料集的豐富性和多樣性以提升 AI 模型的 mAP。

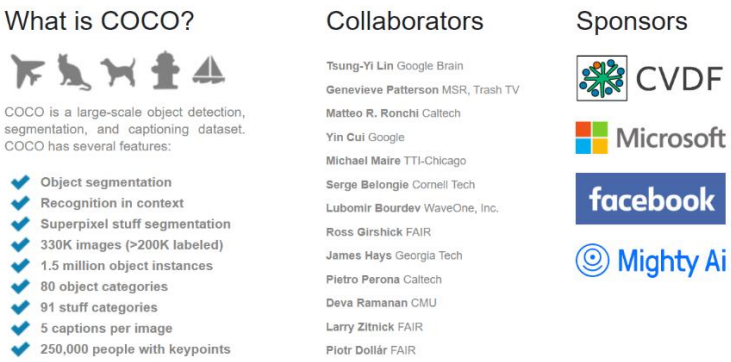


圖 25 COCO 官網
(圖片來源：COCO 網站)

Dataset	Categroy	Image quantity	BBox quantity
PASCAL VOC (07++12)	20	21,503	62,199
MSCOCO (2014trainval)	80	123,287	886,266
ImageNet Det (2017train)	200	349,319	478,806

Dataset	Total	Small	Middle	Large
PASCAL VOC (07++12)	62,199	6,983(11.2%)	19,677(31.6%)	35,539(67.2%)
MSCOCO (2014trainval)	886,266	278,651(31.4%)	311,999(35.2%)	295,616(33.4%)
ImageNet Det (2017train)	478,806	22,677(4.7%)	86,439(18.1%)	369,690(77.2%)

圖 26 三大開源影像 Dataset 的比較：PASCAL VOC、MSCOCO、ImageNet Det
(圖片來源：Jason Chen , 2020)

(九)、現行有關軌道異物入侵系統的論文探討

作者 Wang, Y.和 Yu, P.在 2021 年提出的論文 A Fast Intrusion Detection Method for High-Speed Railway Clearance Based on Low-Cost Embedded GPUs 中提到現今軌道異物入侵的系統有高成本和低的覆蓋率的問題，其提出低成本和快速辨識的方法應用在高鐵上。研究中其應用物件偵測演算法 SSD 和內嵌系統 Nvidia Jeson TX2，也使用 TensorRT 優化模型達到 89% mAP 和 25.9FPS。本研究將實驗各 YOLO 演算結果與其比較，驗證實驗成果。

三、以北捷文湖線為實驗場域

根據場域和幀率分析，我們選擇北捷文湖線為實驗的場域，其描述如下：

(一)、場域分析:

文湖線場域有以下特點：

1. 北捷文湖線為自動化無人駕駛，須要被監控。
2. 車速每小時為 70 公里，幀率易於達成。
3. 第一車廂的車窗玻璃透明，便於進行錄製測試時運用的。
4. 為開放場域，行經住商混合路段和山林間，異物易於入侵。
5. 2015 年有在萬芳醫院和辛亥間大型招牌掉落事件。

(二)、幀率分析：

文湖線列車最高車速為每小時 70 公里，煞車距離為 65 公尺 (國家運輸安全調查委員會，2021)，亦即車子每秒行進 19.4 公尺，若以每秒 1 公尺(m)為監測單位，約需目標幀率(Target FPS)為 20FPS，計算如式七，這目標幀率易於達成。也能推估臺鐵太魯閣號最高時速為每小時 130 公里，所需幀率約為 40FPS，但煞車距離需要 600-650 公尺。

$$\text{Target FPS} > \frac{70000}{3600} \cong 19.4 \qquad \text{(式七)}$$

(三)、準備資料集

在訪問過程中得知：北捷在麟光和辛亥間的福州山隧道，常有果子狸闖入遭列車撞擊。而貓跟果子狸有些相似，可以增加辨識的難度。而在 2015 年 6 月北捷列車從萬芳醫院站往辛亥站行駛約 100 公尺，撞上廣告招牌出現警訊急停(郭逸，2015)，因此選擇辨識的類別(class)為：果子狸(Paguma)、貓(Cat)、人(Person)、招牌(Shopsign)，用來訓練模型。初期先以每個類別各 100 張圖片準備，這些圖片分別為從網路上、或自行拍攝，圖片甚多僅挑幾張代表，見圖 27。




圖 27 訓練資料集分別為果子狸、貓、人、招牌
(圖片來源：網路、COCO 資料集或研究者自行拍攝)

(四)、錄製軌道場景

搭乘臺北捷運文湖環狀線列車往返「動物園」與「南港展覽館」間，見圖 28，於第一節車廂以 GoPro 或手機貼近玻璃方式錄製前方軌道場景，見圖 29、30、31、32。

圖 28 文湖線站別
(圖片資料來源：文湖線-台北捷運出口 TPMRTEXIT 網站)

圖 29 出發錄製
(圖片來源：研究者自行拍攝)

圖 30 錄製軌道中
(圖片來源：研究者自行拍攝)

圖 31 錄製軌道中
(圖片來源：研究者自行拍攝)

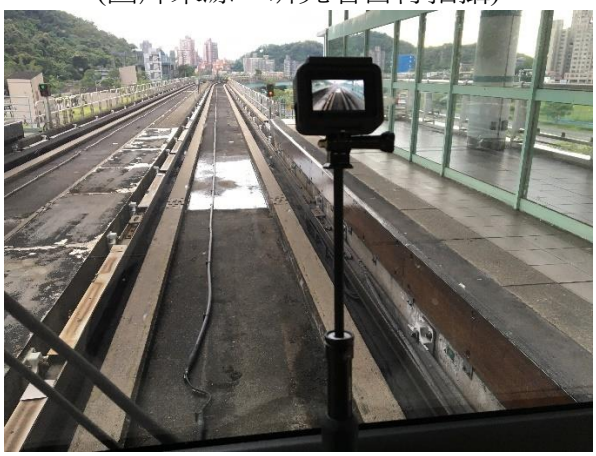


圖 32 GoPro 特寫
(圖片來源：研究者自行拍攝)

於第一節車廂拍攝前方軌道場景，模擬攝影機裝置在列車頭情形示意如下：

圖 33 模擬列車頭裝設攝影機
(圖片來源：研究者自行繪製、內湖線-維基百科)

影像解析初期設為 1080p，60FPS，但因為幀率高 GoPro 很快沒電，因此改為 30FPS 錄影拍攝。期間因列車沿途晃動，必須全程以手輔助，腳架可以提供適度的穩定性，見圖 30。

錄製期間將量得各數據紀錄如下：

1. 北捷車前景窗框為高 100cm、寬 180cm。

2. 手機拍攝架設位置：高 26.5cm、寬 70cm、與玻璃間距離為 25cm。

3. GoPro 拍攝架設位置：高 40cm、寬 60cm、與玻璃距離為 10cm。

期間行經麟光-辛亥隧道、大直-松山隧道，從文湖線動物園-南港展覽館全程約 50 分鐘，拍攝影片將作為我們「AI 軌道異物偵測系統」的測試影像，見下圖：

圖 34 文湖線一小段擷取照片，系統辨識使用
(圖片來源：研究者自行拍攝)

再將所得場景影片以影像軟體將異物 P 圖，供我們的系統測試和驗證使用，如下圖所示：

圖 35 將異物 P 圖入軌道
(圖片來源：研究者自行拍攝)

四、系統硬體設計

現行異物偵測系統多為感應器式，僅能知道物體入侵，無法得知何種物體。僅臺鐵的落石邊坡系統有 AI 影像辨識的功能，但其架設在道旁，僅限於某危險無防護路段，若要全線施作，則成本過高。

為達到省成本又有能即時預警，本研究將攝影機裝置在列車車頭上，讓列車在行駛中，使用內嵌系統 Nvidia JetsonTX2 做為影像處理核心，將訓練好的模型放入 TX2 中進行辨識，將辨識結果藉由 WIFI 回傳至行控中心，系統硬體架構見圖 36。在筆電上以 Python 的 Flask 函式庫撰寫程式架設伺服器，模擬的行控中心，這裡我們以筆電模擬，針對不同物體不同狀況對列車下達控制指令。實際情況列車會以無線訊號傳至軌道沿線的基地台，再經由光纖網路傳回行控中心，再下達指令經由光纖和無線訊號回傳至列車。

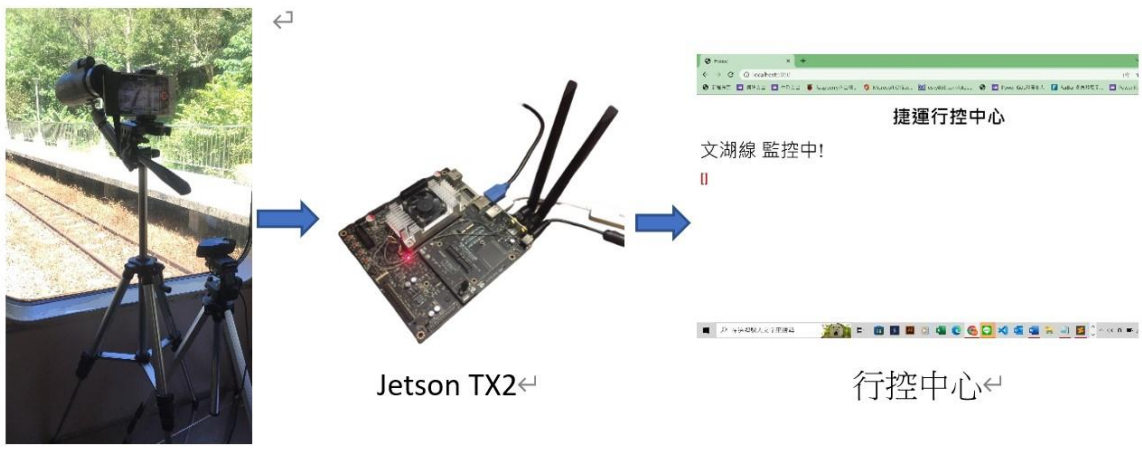


圖 36 系統硬體架構
(圖片來源：研究者自行拍攝)

硬體組成單元如下：

1. 攝影機

架設在列車車頭，因現行攝影機可拍攝的距離為 20-30 公尺為極限，而每小時 70 公里的列車煞車距離為 65 公尺左右，因此建議可加裝望遠鏡頭，達到 70 公

尺外的預警。當攝影機看到物體時，由 TX2 做辨識，並將結果送回行控中心，控制訊號送至列車，並根據不同異物採取不同煞車措施。

2. 內嵌系統 Nvidia Jeson TX2

使用 Nvidia 的 Jetson TX2，因其為 GPU 架構，此 GPU 具有高達 8 GB 記憶體、59.7 GB/s 記憶體頻寬，以及為邊緣端提供真正的人工智慧(取自 Nvidia 官網)。TX2 為運算核心，讓列車可以 Edge 運算直接辨識軌道上的異物，不需要大頻寬將訊號回傳至後端行控中心的伺服器做運算，能即時預警，也能減少通訊硬體的成本。

3. 以筆電模擬行控中心

以 Python 的 Flask 函式庫撰寫程式架設伺服器，接收回傳的告警訊號，再由行控中心做雙重確認，針對不同物體不同狀況對列車下達控制指令，見下圖：

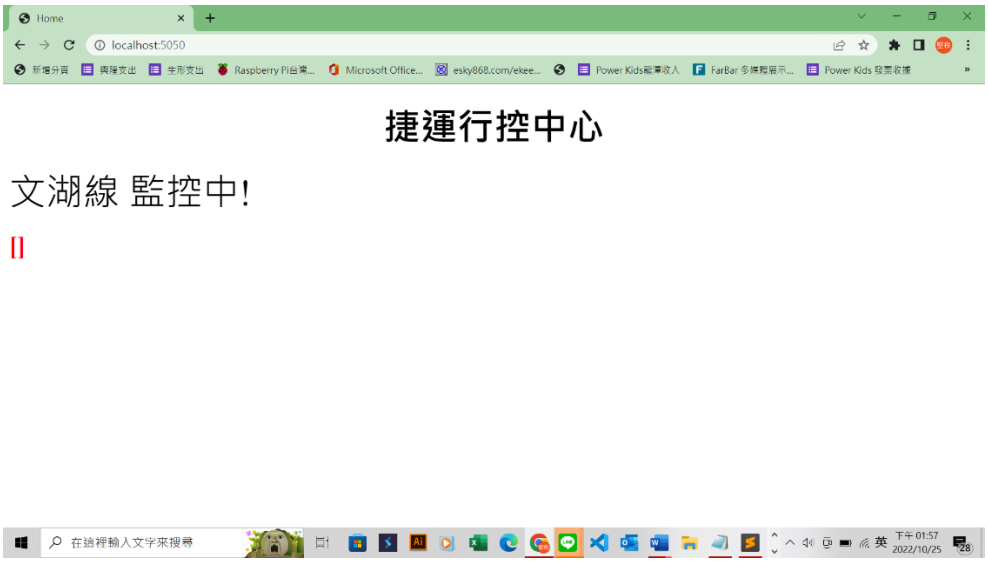


圖 37 行控中心介面
(圖片來源：研究者自行拍攝)

將系統攝影機和 TX2 在列車上實驗的情形如下：

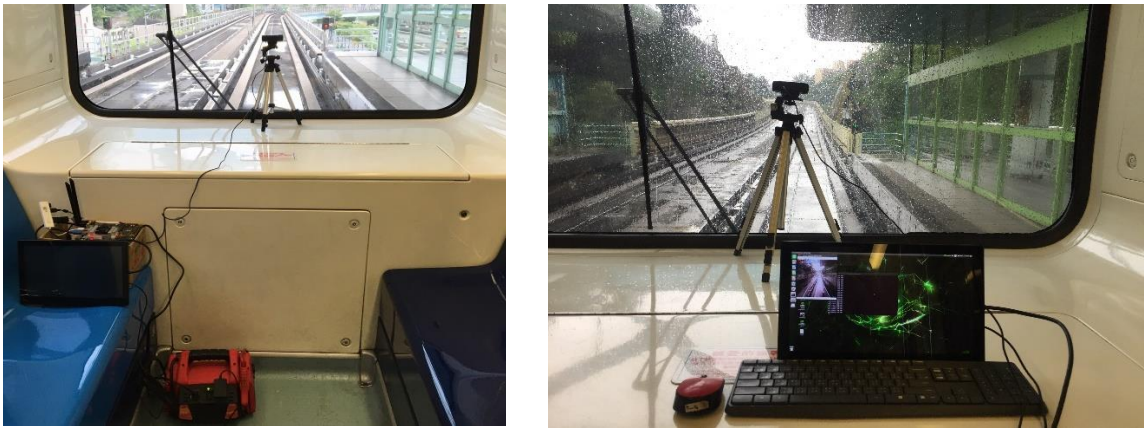


圖 38 在列車上的情形
(圖片來源：研究者自行拍攝)

五、系統軟體設計

在網路上可以找到 YOLOv4、和 YOLO6 的 Python 的開源碼，加上 Pyhton 在編譯 AI 類神經演算法相對容易和廣泛應用，所以使用 Python 作為程式語言。程式開發初期因會比較不同 YOLO 系列演算法，不同的演算法有許多相對應版本的函式庫，所以使用 Anaconda 做為虛擬環境，方便管理不同 YOLO 系列研究。

本研究將程式分為訓練(Train)、測試(Test)和控制(Control)三大部分。YOLOv4、YOLOv4-Tiny、YOLOv6 訓練準備檔案架構如下：

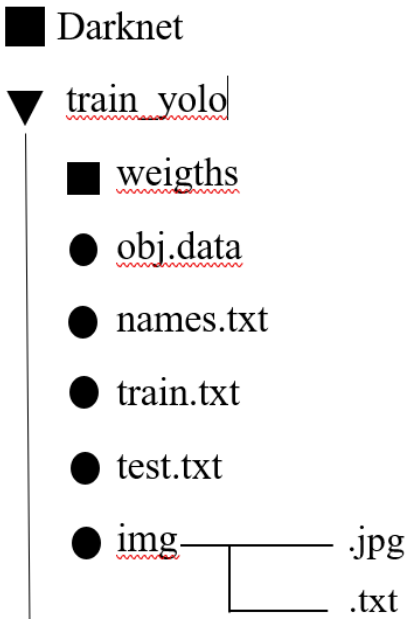


圖 39 YOLOv4 檔案架構
(圖片來源：修改聲伊的閱讀筆記網站)

```
custom_dataset
├── images
│   ├── train
│   │   ├── train0.jpg
│   │   └── train1.jpg
│   ├── val
│   │   ├── val0.jpg
│   │   └── val1.jpg
│   └── test
│       ├── test0.jpg
│       └── test1.jpg
└── labels
    ├── train
    │   ├── train0.txt
    │   └── train1.txt
    ├── val
    │   ├── val0.txt
    │   └── val1.txt
    └── test
        ├── test0.txt
        └── test1.txt
```

圖 40 YOLOv6 檔案架構
(圖片來源：meituan/YOLOv6, 2022)

訓練完的權重會存入 weigths 資料夾中，names .txt 檔案寫入要預測的 label(要辨識的物體名稱)，img 照片檔則是所有照片，讓 train.txt、test.txt 拿取照片。

train.txt 文字檔為照片中 80%的照片名稱，做為訓練。

test.txt 文字檔為照片中 20%的照片名稱，做為測試。

obj.data 檔案則是寫入 classes 類別數量、weigths、naems.txt 參數檔、train.txt 文字檔、test.txt 文字檔的位置路徑。

訓練(Train)程式碼則是應用網路上的開源碼。首先須編寫整理資料集(dataset)的程式，並編寫整理各照片路徑的程式，產生訓練文字檔(train.txt、test.txt)以告知編譯器相對應照片的路徑，並執行 Darknet 編譯器(darknet.exe)，同時在伺服器訓練模型。本研究使用 labelimg 軟體來標定物體的邊界框(Bounding Box)，如下：

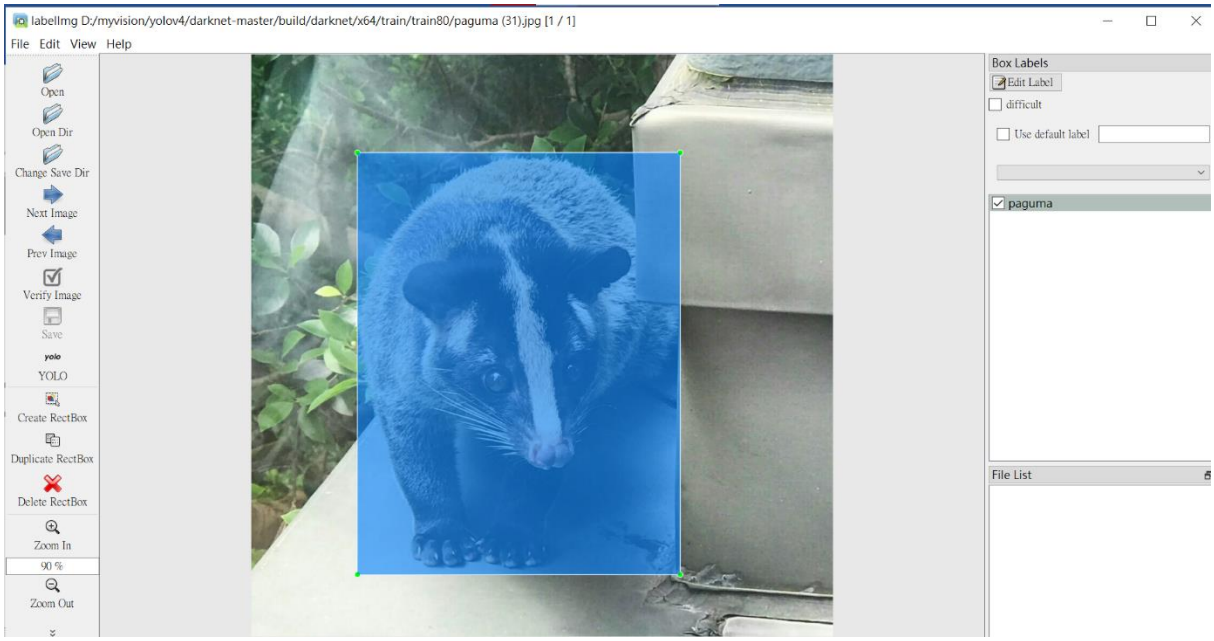


圖 41 使用 labelimg 軟體來標定物體的邊界框
(圖片來源：研究者自行拍攝)

產生 YOLO 文字檔格式，分別表示物體的類別(Class)、中間座標(cx, cy)、寬(width)和高(height)，如下：

```
# class_id center_x center_y bbox_width bbox_height
0 0.300926 0.617063 0.601852 0.765873
1 0.575 0.319531 0.4 0.551562
```

圖 42 YOLO 文字檔格式
(圖片來源：meituan/YOLOv6, 2022)

設計測試(Test)程式，修改網路上的開源碼符合系統需求，所得訓練模型載入做辨識，將辨識所得結果，載入控制函式中，依據不同物體制定不同煞車措施作為停車、減速或通過之判斷，同時將辨識結果送至行控中心作確認，再將訊號送至列車自動控制系統(ATC)執行相對應動作。

本系統煞車措施為辨識到車、人則告警並停止；辨識到動物如果子狸(Paguma)、貓(Cat)則告警並慢速通過，其流程如下：

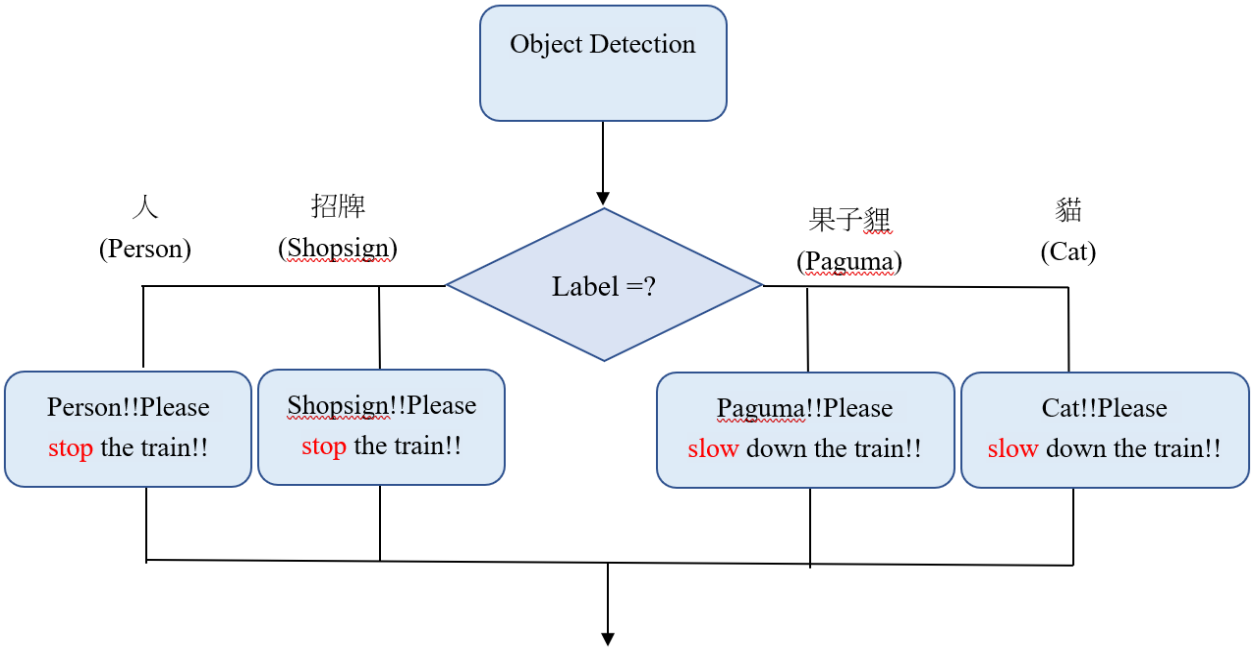


圖 43 煞車措施
(圖片來源：研究者自行拍攝)

本研究在影像畫面中畫定一個「辨識區」，以軌道為中心，包圍軌道的梯型區域或三角形區，入侵此區域的異物將會被視為對列車有危害，須要被辨識，才會告警。因為畫面中異物可能出現非列車行駛的區域，並未對行車造成危險，則無須辨識，如：文湖線忠孝復興站至大安站間就有許多招牌林立的大樓，若未區分，將辨識為招牌入侵，導致誤告警頻繁。辨識區限定異物偵測的區域，避免誤報警的發生。

在影像中畫面是以像素(pixel)表示，可以像素座標(x, y)來表示物體的位置，原點(0, 0)位置為螢幕畫面右上角，x 軸往右遞增，y 軸往下遞增。本研究使用一元一次不等式來劃分「非辨識區域」，方式如下：

1. 將以三條直線一元一次不等式區域的聯集，因 y 軸往下遞增，故以小於(<)表示，將畫面區分為「非辨識區」如下：

$$y < abs(e)$$

$$y < abs(ax + b) \qquad \text{(式八)}$$

$$y < abs(-cx + d)$$

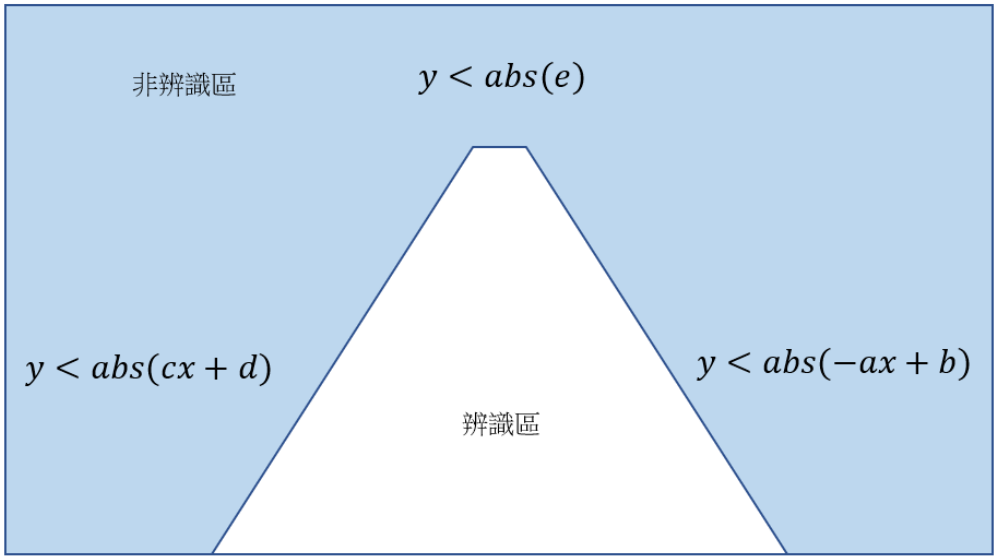


圖 44 辨識區與非辨識區
(圖片來源：研究者自行拍攝)

2. 視覺辨識產生的 Bounding Box 是一框住物體的矩形框，其座標表示一般為左上角座標右下角座標(xmin, ymin)和(xmax, ymax)，本研究以 (Left, Top)和 (Right, Bottom)，表示如下圖：

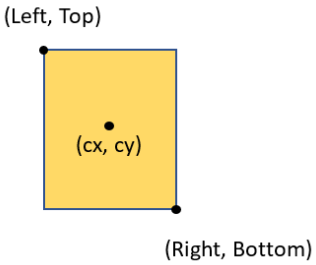


圖 45 異物 Bounding Box 的座標
(圖片來源：研究者自行拍攝)差

3. 異物判定條件：當異物右角的 bottom 對的是$y < ax + b$，亦即是下 bottom 必須小於 abs($a\ right + d$)；或是左下角的 bottom 對的是$y < -cx + d$，亦即是 bottom 必須小於 abs($-c\ left + b$)；或物體 Bottom < e 值，符合以上條件就判定異物落在為非辨識區，無須辨識，反之小於落在「辨識區」，須要被偵測，如下圖：

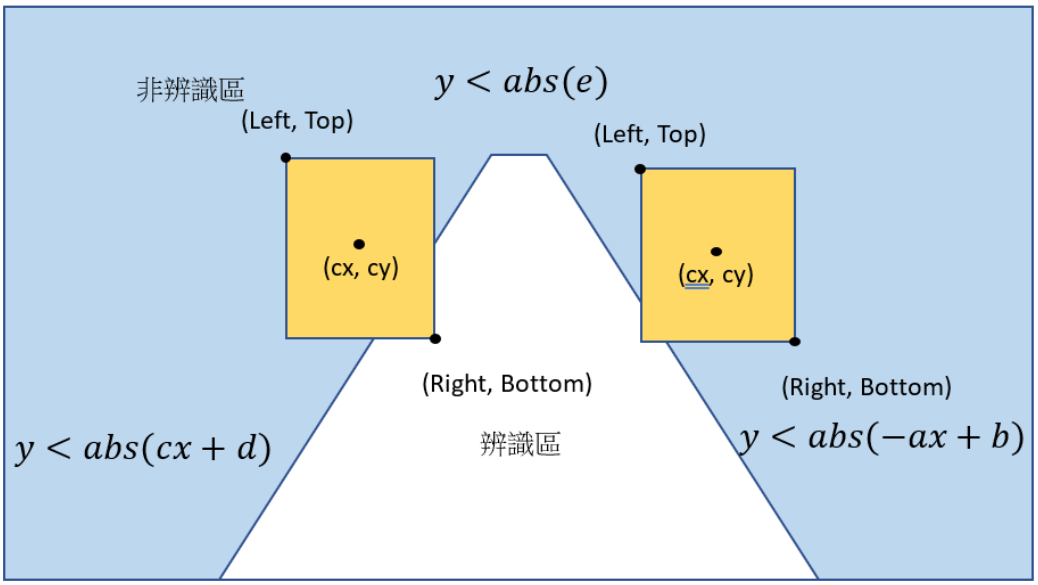


圖 46 異物入侵判定條件
(圖片來源：研究者自行拍攝)

程式碼在 YOLO4 畫面尺寸(416x416)，文湖線軌道為背景如下所示：

```
if (bottom < abs(3.152*right-624)) or (bottom < abs(-3.152*left+687.03)) or (bottom < 10):
        continue
```

圖 47 異物入侵判定條件程式碼
(圖片來源：研究者自行拍攝)

以 Python Flask 架設伺服器，模擬行控中心的，收到告警訊息，分別碰到人(Person)、招牌(Shopsign)、果子狸(Paguma)、貓(Cat)，告警訊息會在網頁顯示，運作方式如下：

圖 47 行控中心告警畫面
(圖片來源：研究者自行拍攝)

神經網路推論運算是最耗時間的，程式對辨識結果，做相對應煞車措施時應盡量減少會增加時間的消耗。建議以平行運算(multiprocessing)的方式，加快程式運行的速度，為避免傳遞告警訊息時延遲。平行運算的流程如下圖所示：

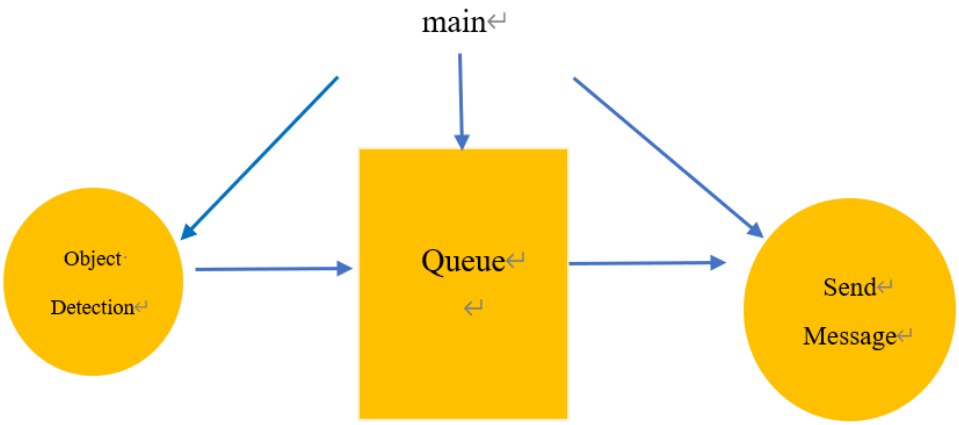


圖 48 平行運算流程圖
(圖片來源：研究者自行拍攝)

六、實驗設計

實驗分為整理資料集、訓練模型和測試三階段，將實驗變因整理如下：

(一)、整理資料集

變因 1. 資料集的型式

變因 2. 資料集的數量

實驗分析:

1. 探討資料集的形式:如清晰度、匡列物體方式等，對訓練模型的 mAP 的提升效果
2. 資料集的數量多寡對訓練模型的 mAP 提升效果。

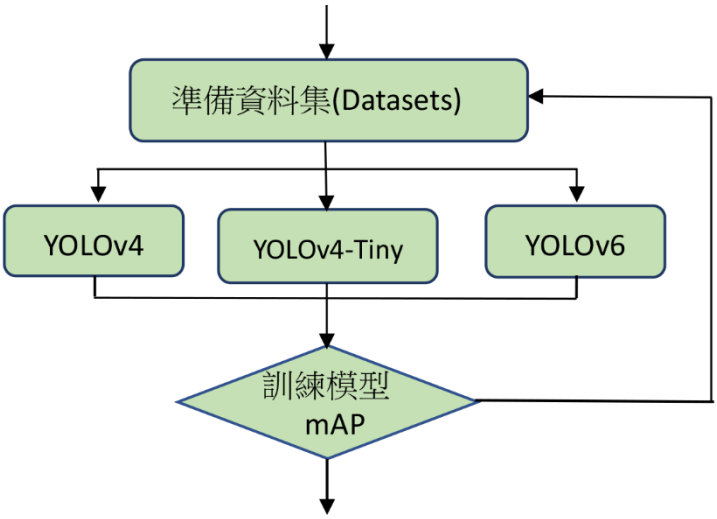


圖 49 資料集與 mAP 的實驗流程
(圖片來源：研究者自行設計)

(二)、訓練模型

變因 1. 訓練平台的不同對模型 mAP，及時間的影響

變因 2.不同 YOLO 系列-YOLOv4-Tiny、YOLOv4、YOLOv6 演算法的 mAP

實驗分析:

1. 探討不同平台:筆電(GPU-MX150)與伺服器的幀率訓練模型的 mAP 和花的訓練時間長短。
2. 探討比較 YOLO 系列演算訓練模型的 mAP。

(三)、測試階段

變因 1. 不同演算法的幀率(FPS)

變因 2. 不同測試平台幀率(FPS)

變因 3. 物體的大小

變因 4. 物體在影像中的位置

變因 5. 物體的與攝影機的距離

變因 6. 不同實驗場域：北捷文湖線、臺鐵內湖線、新北捷淡海輕軌

變因 7. 加裝望遠鏡頭

實驗分析:

1. 探討不同平台:筆電(GPU-MX150)與內嵌系統 TX2 的幀率(FPS)和辨識效果
2. 探討系統對物體辨識的能力，多大多小的物體可以辨識，物體在影像中的哪個位置需要告警。
3. 探討系統在各場域的幀率(FPS)和辨識效果。
4. 加裝望遠鏡頭的辨識效果。

(四)、異物設計：因無法將真實物體放入軌道上，本研究以 P 圖(retouched image)的技術，將要辨識的物體加入軌道中，如下圖：

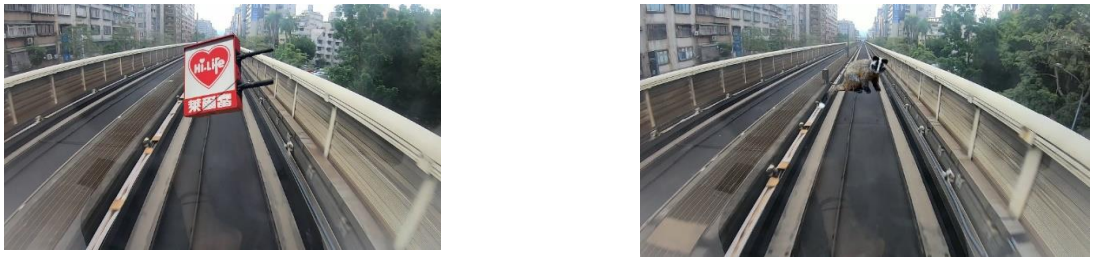


圖 50 P 圖情形 (圖片來源：研究者自行拍攝)

伍、研究結果與討論

一、臺鐵、高鐵、北捷、新北捷運的異物偵測系統之比較。

為了解現行各軌道運輸公司的軌道異物偵測系統的優缺點，使我們的能設計出更有效的系統，先後訪談臺鐵、高鐵、北捷和新北捷運-淡海輕軌四大軌道運輸公司，將結果整理如下：

1.臺鐵：

臺鐵相關人員表示：邊坡告警系統使用視覺辨識、AI 深度學習的技術預警。台鐵煞車距離為 600 公尺至 650 公尺之間，目前是將攝影機架設在危險邊坡軌道旁，約 20-30 公尺的距離架設一處，類似電線杆的距離，如圖 51、圖 52。

圖 51 攝影機立柱日間
(圖片來源：聯合新聞網)

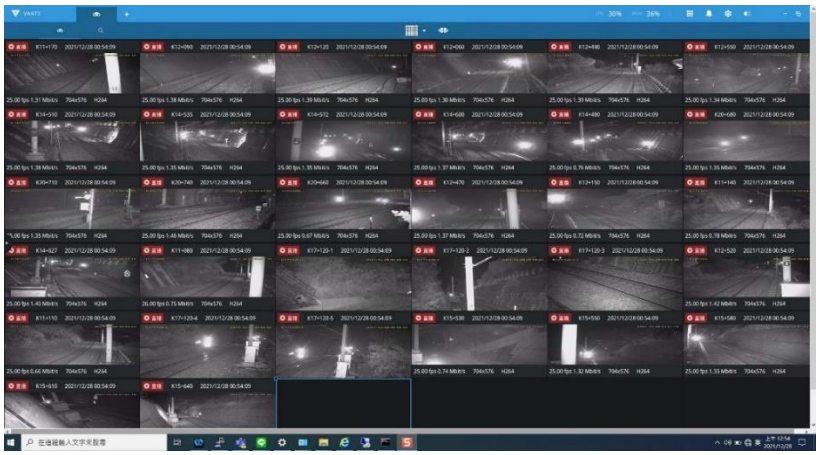


圖 52 邊坡落石告警系統
(圖片來源：聯合新聞網)

並透過 AI 深度學習進行軌道異物辨識，有異物侵入會立即將告警訊息即時通知，並透過台鐵的環島光纖將即時訊息通報列車司機員、前後端車站、發送告警訊號，通知鄰近列車司機進行停車程序 。為確保人工智慧辨識能力，臺鐵局表示：他們邀請外部委員協助，進行辨識功能測試計畫，因 AI 人工智慧辨識系統規畫涉及硬體規格選擇、氣候狀況模擬及辨識成功率等較複雜因子。

台鐵工程處橋隧科人員說：臺鐵邊坡告警系統是用攝影機可見光；平交道則是使用紅外線攝影機與毫米波，如圖 53。邊坡落石告警系統裝置在北迴線 2 處(K48、K51)為太魯閣號出事的路段和南迴線 9 處，共 11 處已完成建置。

圖 53 平交道障礙物偵測系統 左：紅外線攝影機，右：毫米波雷達
(圖片來源：研究者自行拍攝)

2. 高鐵：

高鐵的異物入侵偵測系統名為：「天然災害告警系統」（DWS），此系統包括地震偵測器、氣象偵測器、闖入偵測器等(王嘉慶，2021)。高鐵賴副理表示：他們是使用機械、電子設備的方式，因為其安全考靠。以下是不同物體的預警方式：

車、牛：使用感應線或纜線，若車子、動物闖入撞斷了感應線、纜線就會告警。

落石、邊坡：使用傾斜桿，在邊坡或落石滑落時，位移或傾斜超過一定的角度就會告警，如圖 54。

圖 54 高鐵的邊坡系統 (圖片來源：中時新聞網)

此技術是從日本引進，所以穩定度與精確性等技術相對成熟。高鐵吳長官表示：高鐵

有自己內線封閉資訊傳輸系統，使用光纖串連所有的設備與感應器加速準確判。

因參訪了高鐵，了解不是辨識到任何物體就馬上停車，這會造成很大運輸影響。高鐵吳長官說：像是誤點率(沒有在正常時間內進站)或是車距問題等。

(1). 一定停車：落石、地震、邊坡

(2). 不停車：大中小型動物不會影響營運的，會以按喇叭、慢速通過。

如果高鐵撞到牛，車身不會有太大損傷，僅車頭凹一洞，但因為會牽扯到保護動物的議題，所以還是以按喇叭、慢速通過。如圖 55。

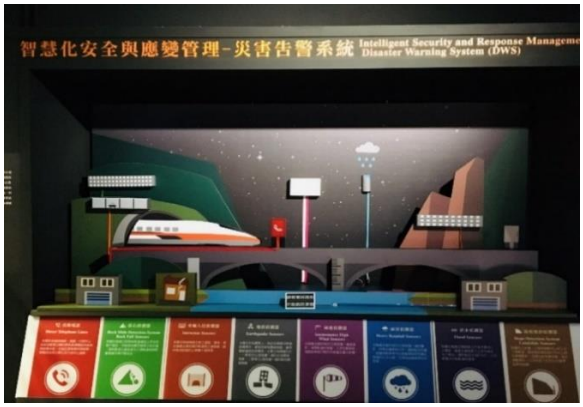


圖 55、災害告警系統（DWS）：邊坡、洪水、雨天、地震、車輛、落石等告警系統。
(圖片來源：研究者自行拍攝)

3.北捷

從內部人員得知，北捷異物入侵系統以文湖線為主要設置，因文湖線為無人駕駛控制，須更嚴謹可靠的系統提供告警。加上文湖線為開放場域，曾發生招牌或路樹掉落軌道事件。麟光和辛亥間的福州山隧道生態豐富近動物園，常有果子狸闖入遭列車撞擊如圖 56，導致列車必須停下來檢查，影響列車準點率，亦即列車準時到站的時間。北捷文湖線的軌道為高架軌道，有絕對的路權，所以無異物入侵的系統。目前研發的異物偵測系統是使用光達和 AI 影像辨識作為預警系統，原先架設在列車車頭上，但因成本過高因此將此系統架設在危險路段。因光達速度快，不受天候影響，又有能偵測物體距離的優點，但成本高，目前正在開發測試階段。目前異物入侵系統裝置在車頭，以光達(Lidar)和 AI 辨識為主。訪談得知因過高成本不符經濟效益，改將設備裝置在道旁，正試驗中。

圖 56 果子狸
(取自綠色保育標章雲林「諸樹柚子園」之生物多樣性指南)

4.新北捷運- 淡海輕軌

參訪過程中得知輕軌還是屬於較人工化，由駕駛員預警。目前淡海輕軌係兼具 A 型/B 型路權，因此為避免行人或車輛誤闖 A 型路權區段，於特定區段(如藍海橋兩端)架設紅外線偵測系統；一旦發生入侵事件，相關告警訊息將立即傳送至行控中心，系統 CCTV 亦立即轉播場狀況，續依相關運轉作業程序因應作為處置，以維行車安全。目前異物入侵於較多為塑膠袋或紙板等廢棄物品飄落至軌道區，若列車於行駛中發現前方有異物影響列車通行時，將立即停車並向行控中心回報申請授權下車排除異物。輕軌時速為：有架空線路段 70 km/h、 無架空線路段 50 km/h，速度較緩慢，而輕軌與汽車、人共用馬路，因此最常發生的情況為汽車誤闖軌道，如圖 57。輕軌唯一預警系統為電子圍籬，電子圍籬為條狀式，分別有 7 個紅外線，在軌道的兩邊各裝設一支電子圍籬並透過紅外線去感應，當 7 個紅外線都被擋住時代表有列車經過，若只有 1-6 個紅外線被擋住將代表有其餘異物闖入如圖 58。

圖 57 常有汽車誤闖軌道，因與轉彎口相近。並有黃色交通錐引導。
(圖片來源：研究者自行拍攝)

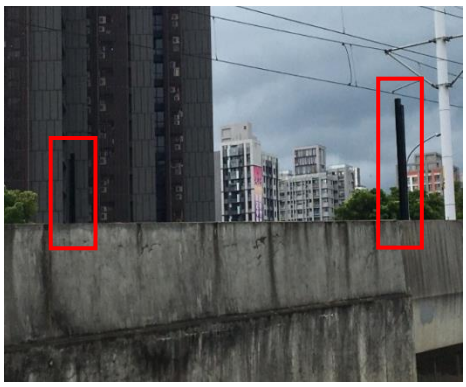


圖 58 電子圍籬 (黑色條狀物體)
(圖片來源：研究者自行拍攝)

高鐵、臺鐵、北捷、輕軌的異物入侵偵測系統各有各的系統，也各自發展，現將其整理與比較，如表 2。

表 2 臺鐵、高鐵、北捷和輕軌異物偵測系統比較表

名稱 / 項目	臺鐵		高鐵	北捷		輕軌
系統名稱	邊坡落石告警系統	平交道障礙物偵測系統	天然災害告警系統（DWS）	文湖線列車異物偵測系統		紅外線偵測系統
設備	攝影機	紅外線攝影機+毫米波	感應器	光達+攝影機	光達+	感應器+交通錐
安裝位置	道旁	道旁	道旁	列車頭	道旁	道旁
運作方式	AI 視覺辨識	紅外線攝影機+毫米波	機械式感應器	光達+AI 辨識	光達	電子圍籬
異物辨識	可	感知	感知	可	感知	感知
深度學習	資料學習	不用學習	不用學習	資料學習	不用學習	不用學習
抗天候	低	高	高	高		中
成本	-	-	-	相對高		-
完成狀況	部分建置	已完成	已完成	技術可行，成本高未達效益	試驗中	已完成

總結：本研究訪談各四大鐵路公司的後，得知列車的「誤點率」是各軌道公司關心的議題，如果能得知闖入異物為何?對不同異物採取不同煞車策略，不是一律都停車，有些是可以鳴按喇叭，慢速通過，如動物等，勢必能改善「誤點率」問題。但機械式感

應器或電子圍籬無法得知異物為何?而光達(LiDar)成本過高未達經濟效益；臺鐵的邊坡落石告警系統雖然有 AI 識別功能，但若沿線布置，臺鐵營運鐵路總長 1085.3 公里(維基百科)勢必是一大成本。因此本研究採用可見光攝影機，裝置在車頭方式，並加裝望遠鏡頭，結合 AI 深度學習和 YOLO 物件偵測演算法達到遠距離的辨識和預警。兼具可見光攝影機成本低的優點，達到視覺辨識，針對不同異物採取不同煞車措施，也讓列車可以第一時間針對軌道上的異物做反應，提高安全性，降低誤點率。

二、研究資料集(Datasets)對訓練模型的平均準確率(mAP)影響

(一).資料集的形式對訓練模型的平均準確率(mAP)影響

1. 實驗目的：探討資料集準備對訓練模型的重要性，先準備四個類別各果子狸(Pauma)、貓(Cat)、人(Person)、招牌(Shopsign)100 張，圖片來源 Google 網站擷取，進行訓練。
2. 參數設定：500500 iterlations、YOLOv4、Batch Size: 32、Train/Test 資料為 80%、20%、GPU 2080Ti。
3. 實驗結果: 第一次訓練的在 YOLOv4，準確度 mAP 很不理想，僅維持在 61%-68%，如下圖：

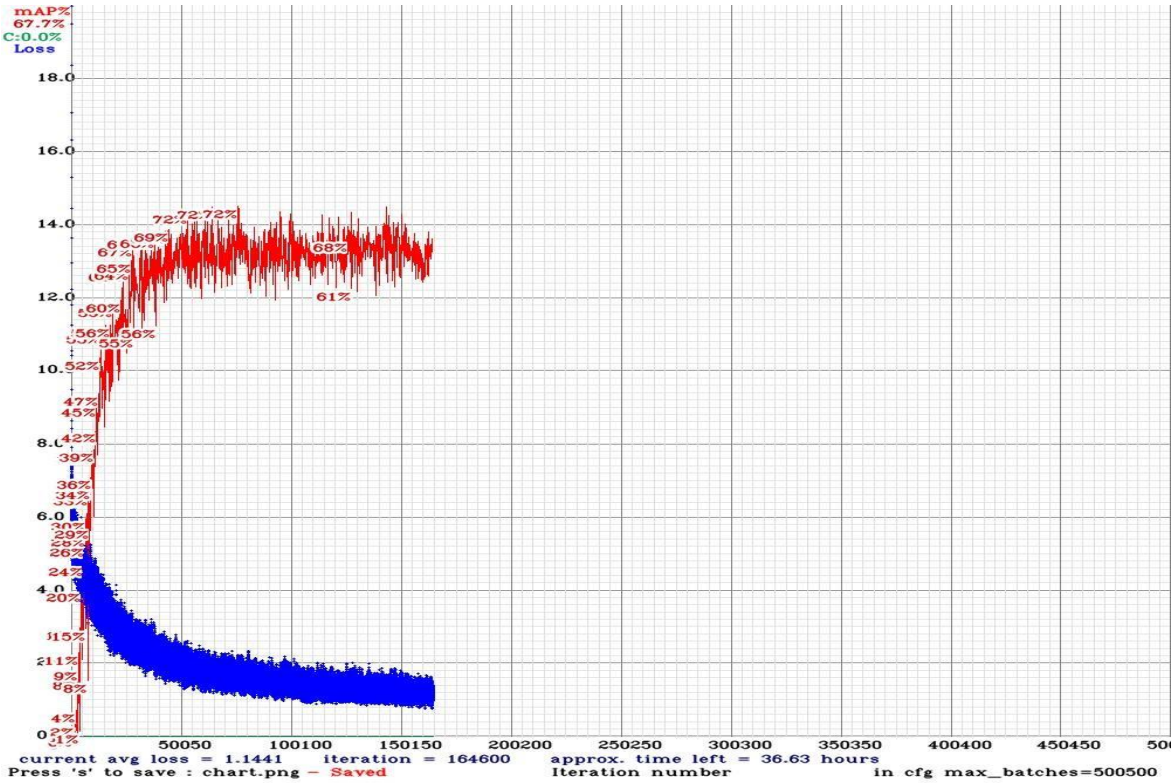


圖 59 第一次訓練的準確度很不理想，只維持在 61%-68%。
(圖片來源：研究者自行拍攝)

重新調整資料集，運用以下這四個方法，準確度也有相對的提升。

1. 調整框取不佳的圖片 ，如圖 60-圖 62。
2. 將資料集中模糊不情楚的圖刪除，如圖 63、圖 64。
3. 將 COCO 資料集的圖片加入資料集，彌補網路上找不到物體各角度的問題。
4. 將自行拍攝的圖片加入資料集，因招牌在前幾次訓練結果為 60%，也發現招牌中的照片較模糊、背景雜亂，而網路上的圖不多，因此自行在街上拍攝需要的招牌。如圖 66。

圖 60 正確框法
(圖片來源：COCO 資料集)

圖 61 框的不明確(特徵也沒有框到)
(圖片來源：COCO 資料集)

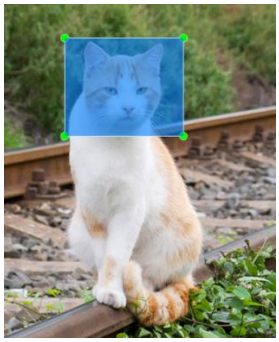


圖 62 框的不好(訓練出來，機器會只框貓的頭)
(圖片來源：COCO 資料集)

圖 63 人的照片模糊
(圖片來源：COCO 資料集)

圖 64 人的照片背景光太暗
(圖片來源：COCO 資料集)

圖 65 網路上招牌較凌亂模糊
(圖片來源：Google 網站)

圖 66 自行拍攝招牌
(圖片來源：研究者自行拍攝)

重新調整資料集後，實驗結果：在 YOLOv4 下 mAP 提高至 90%以上，高達 93%；YOLOv4-Tiny 也有 88%，如下圖：

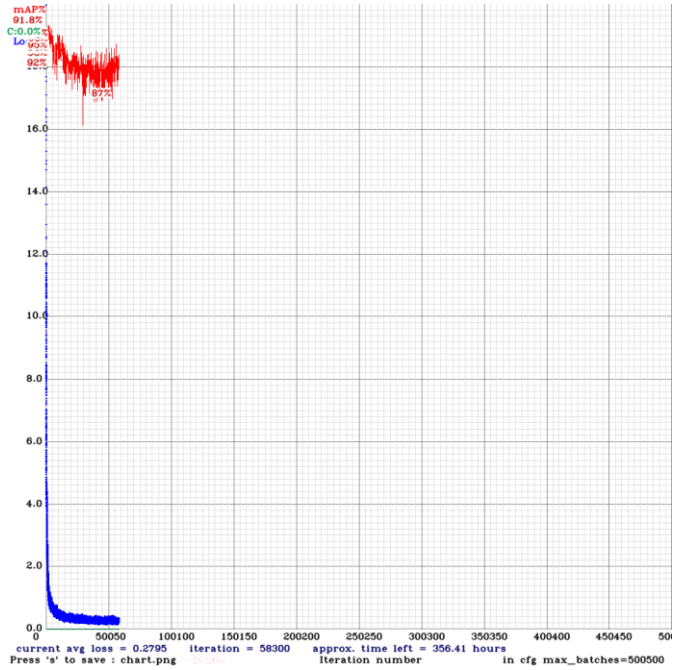


圖 67 YOLOv4 重新調整資料集的 mAP，各 100 張

(圖片來源：研究者自行拍攝)

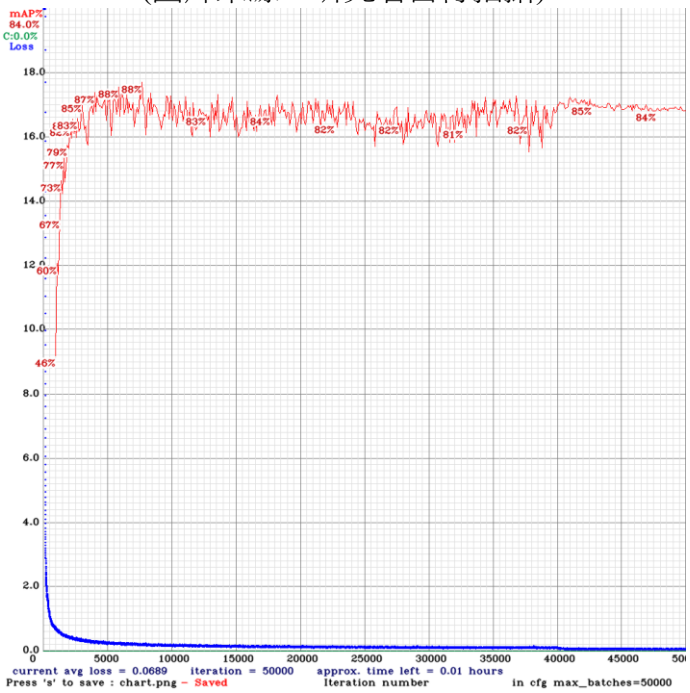


圖 68 YOLOv4-Tiny 重新調整資料集的 mAP，各 100 張

(圖片來源：研究者自行拍攝)

(二)、資料集的數量對訓練模型的平均準確率(mAP)影響

1. 實驗目的：探討資料集數量對訓練模型的影響，增加四個類別各果子狸(Pauma)、貓(Cat)、人(Person)、招牌(Shopsign)為 200 張，圖片來源：調整後新的資料集，進行訓練。
2. 參數設定：50000 iterlations、YOLOv4、Batch Size: 32、Train/Test 資料為 80%/20%、GPU 2080Ti。
3. 實驗結果: 增加張數 2 倍，準確率 mAP 在各演算法都提高，Yolov4 準確度高達 99%，見圖 69；Yolov4-Tiny 準確度為 89%，見圖 70；YOLOv6 S 版、N 版有 95%，見圖 71、圖 72。YOLOv4 的混淆矩陣顯示張數 200 張的資料集各類別的 AP 均在 90%以上，見圖 73；YOLOv4-Tiny 的混淆矩陣顯示張數 200 張的資料集各類別的 AP 均在 80%以上，見圖 74。

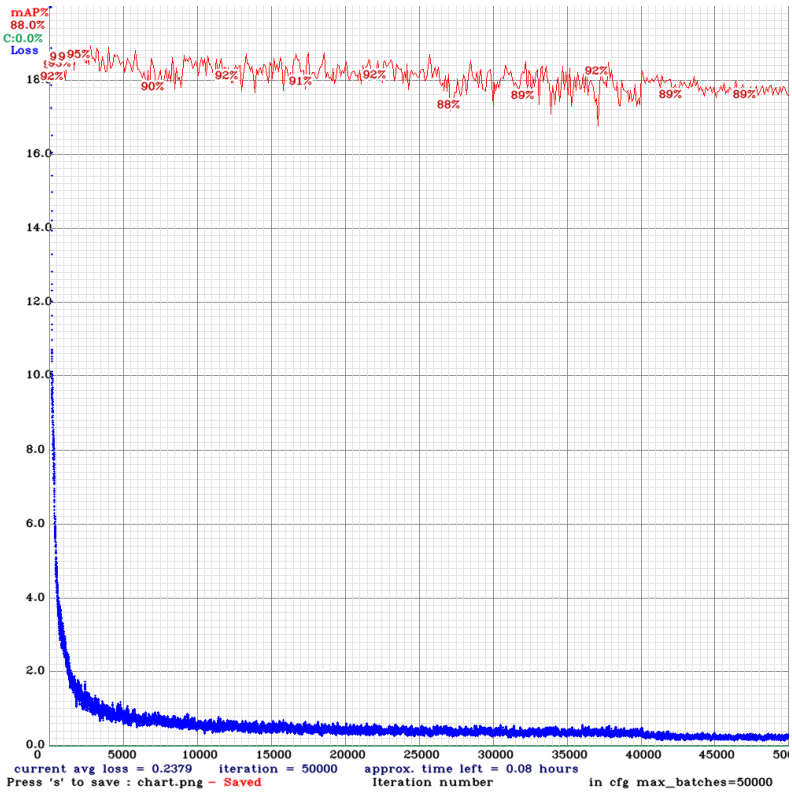


圖 69 YOLOv4 調整資料集的 mAP 圖，各 200 張
(圖片來源：研究者自行拍攝)

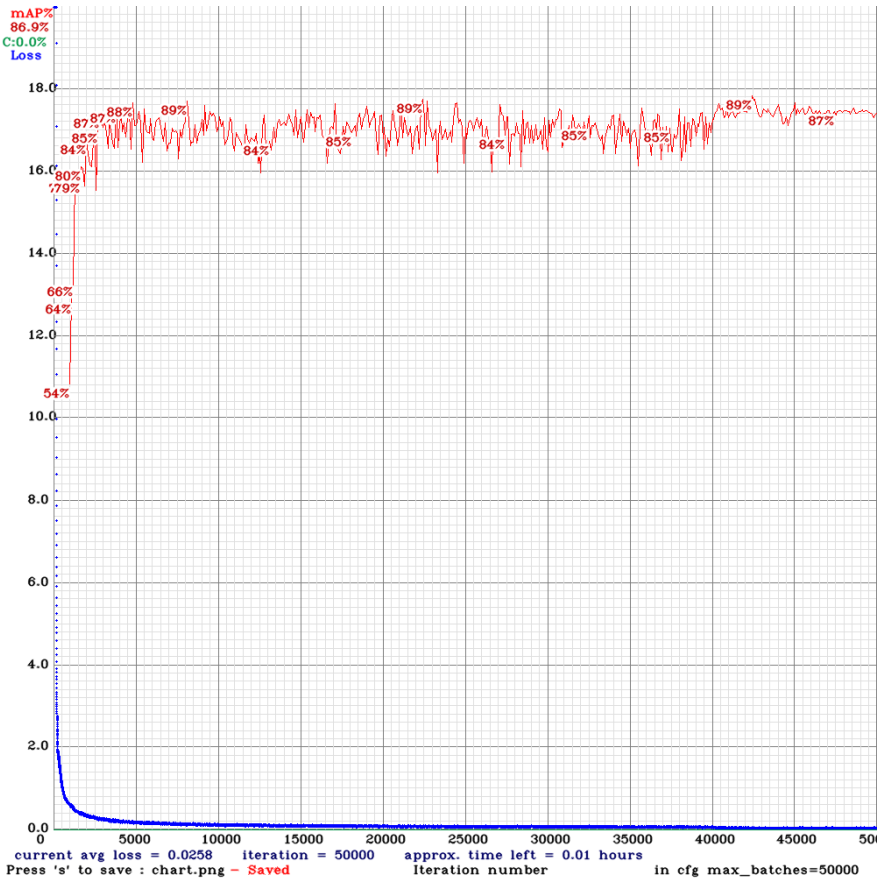


圖 70 YOLOv4-Tiny 調整資料集的 mAP 圖，各 200 張
(圖片來源：研究者自行拍攝)

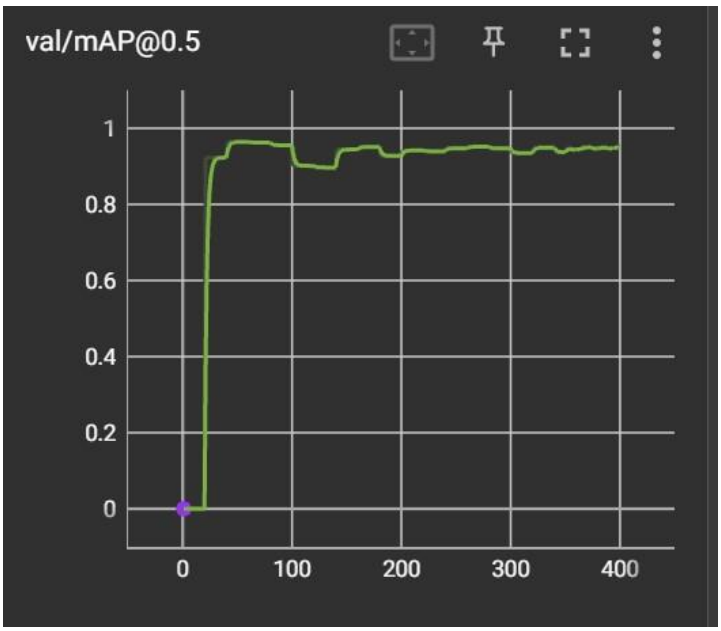


圖 71 YOLOv6 S 版調整資料集的 mAP 圖，各 200 張 400 epochs
(圖片來源：研究者自行拍攝)

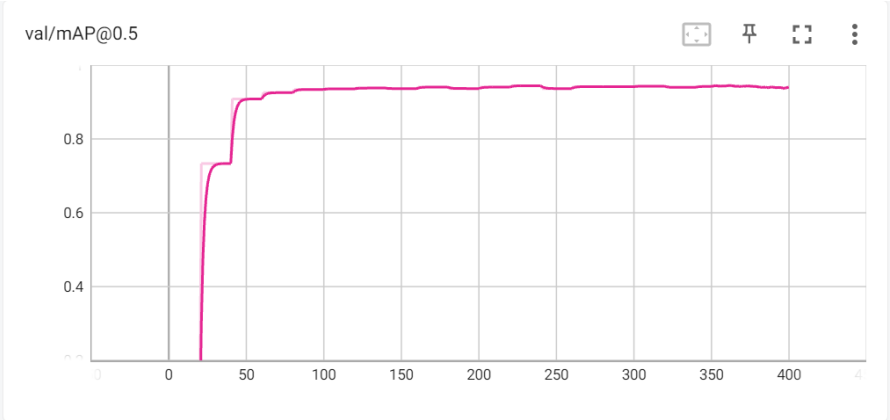


圖 72 YOLOv6 N 版調整資料集的 mAP 圖，各 200 張 400 epochs
(圖片來源：研究者自行拍攝)

```
 detections_count = 275, unique_truth_count = 114
class_id = 0, name = paguma, ap = 98.81%          (TP = 19, FP = 1)
class_id = 1, name = cat, ap = 100.00%            (TP = 20, FP = 0)
class_id = 2, name = person, ap = 95.71%          (TP = 37, FP = 8)
class_id = 3, name = ad, ap = 91.83%      (TP = 30, FP = 3)
```

四個類別各 200 張

```
 detections_count = 478, unique_truth_count = 215
class_id = 0, name = paguma, ap = 96.83%          (TP = 39, FP = 1)
class_id = 1, name = cat, ap = 99.94%             (TP = 40, FP = 2)
class_id = 2, name = person, ap = 94.32%          (TP = 63, FP = 6)
class_id = 3, name = ad, ap = 87.15%      (TP = 57, FP = 14)
```

四個類別各 100 張

圖 73 比較 YOLOv4 不同張數的資料集的混肴矩陣 (Confuse Matrix)
(圖片來源：研究者自行拍攝)

```
 detections_count = 689, unique_truth_count = 215
class_id = 0, name = paguma, ap = 81.66%          (TP = 31, FP = 9)
class_id = 1, name = cat, ap = 91.38%             (TP = 36, FP = 9)
class_id = 2, name = person, ap = 81.20%          (TP = 49, FP = 9)
class_id = 3, name = ad, ap = 80.11%      (TP = 53, FP = 20)
```

四個類別各 200 張

```
 detections_count = 478, unique_truth_count = 215
class_id = 0, name = paguma, ap = 77.79%          (TP = 29, FP = 8)
class_id = 1, name = cat, ap = 94.17%             (TP = 33, FP = 6)
class_id = 2, name = person, ap = 81.43%          (TP = 52, FP = 6)
class_id = 3, name = ad, ap = 79.06%      (TP = 52, FP = 11)
```

四個類別各 100 張

圖 74 比較 YOLOv4-tiny 不同張數的資料集的混肴矩陣 (Confuse Matrix)
(圖片來源：研究者自行拍攝)

結果與討論：200張資料集有比較高的準確率。在實測過程中辨識到物體的效果佳，原先 100 張辨識不到物體，增加張數後可以辨識到，見圖 75。招牌在視覺辨識中困難度算高的，因其為大量圖和文字的組合，易於混淆(Yohannes, Ervin and Lin, Chih-Yang and Shih, Timothy K. and Hong, Chen-Ya and Enkhbat, Avirmed and Utaminingrum, Fitri , 2021)。

建議：本研究僅對四個類別各 200 張的資料集，就有高的準確率。實務上各軌道公司每日在軌道上可以蒐集到大量的資料集，適當剔除模糊照片，增加資料集數量，將可大大增加模型的準確性和辨識的效果。

YOLOv4，左圖 100 張照片 、右圖 200 張

YOLOv4-Tiny，左圖 100 張照片 、右圖 200 張

圖 75 比較 100 張照片 ，200 張照片在 YOLOv4、YOLOv4-tiny 模型下的辨識效果

(圖片來源：研究者自行拍攝)

三、 研究不同平台對訓練模型的影響

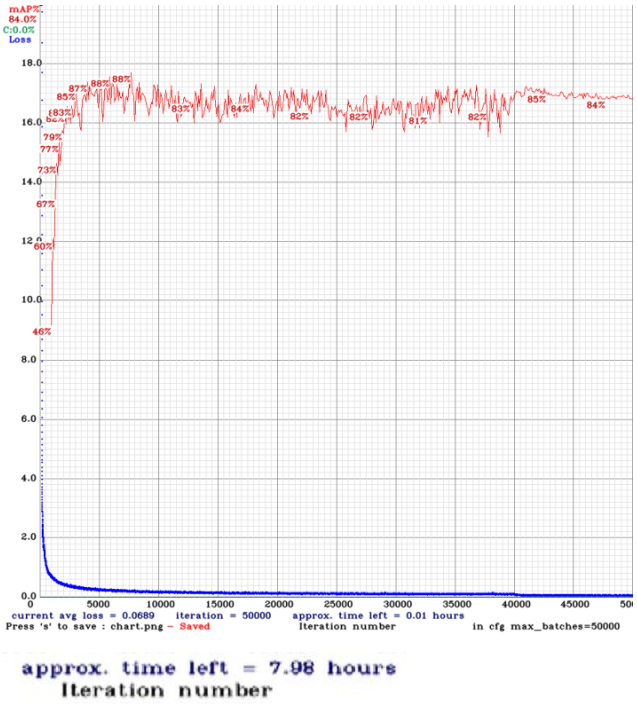
(一)、研究不同平台：伺服器 GPU 2080Ti RAM 11G、

筆電 GPU NVIDA GeForce MX150 對訓練模型的影響。

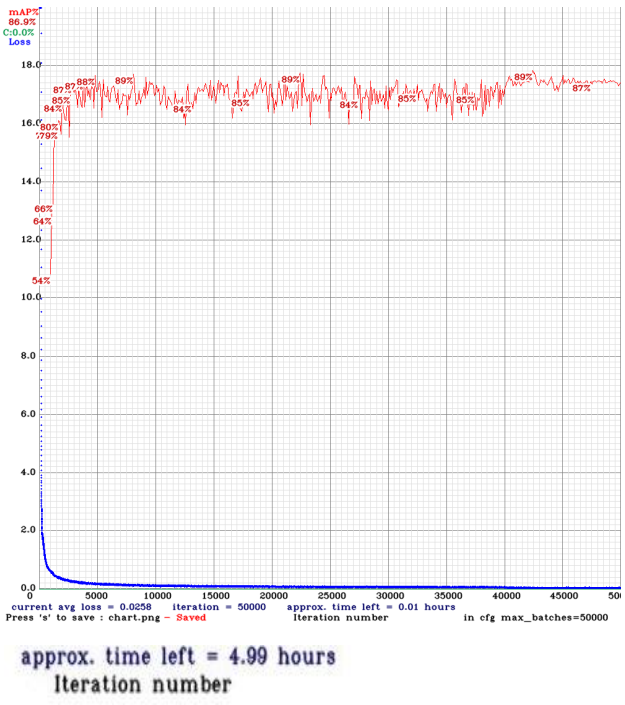
1. 實驗目的：探討不同平台-伺服器和筆電在訓練模型的速度與準確率(mAP)。

2. 參數設定：YOLOv4batch=64、subdivisions = 16、訓練次數為：50000 iterlations。

3. 實驗結果：伺服器與筆電訓練結果準確度只相差 2%，而訓練 50000 次時筆電的訓練時長為 7.90 小時，伺服器訓練時長為 4.99 小時，如下圖所示：



筆電訓練 mAP 為 84-88%，時長為 7.98 小時



伺服器訓練 mAP 為 86-89%，時長為 4.99 小時

圖 76 YOLOv4-tiny 筆電和伺服器訓練模型比對。(左圖為筆電訓練、右圖為伺服器訓練)
(圖片來源：研究者自行繪製)

結果與討論： 證實伺服器 GPU 訓練 AI 模型上有準確率高與訓練時間快的優點，而筆電腦雖準確率低一點，但訓練時間長，如果在成本考量上，沒有伺服器的條件下，仍可作為訓練平台。本研究的模型為達高的準確率和較少的訓練時間，均在伺服器上訓練。

四、研究物件偵測(Object detection)演算法的效能。

(一)、實驗 Yolo 系列中的 Yolov4、Yolo4-Tiny、Yolov6 幀率和辨識的效果。

1. 實驗目的：探討不同 YOLO 架構訓練後辨識的幀率 FPS。

2. 控制變因：200 張數的資料集，YOLOv4、YOLOv4-Tiny：訓練次數為：50000 iterlations；YOLOv6：S 版、N 版訓練次數為 400 epochs；測試平台:筆電(Nvidia Geforce MX150)。

3. 實驗結果：如下圖所示：

YOLOv4 為 7 FPS

YOLOv4-Tiny 為 40 FPS

YOLOv6 S 版為 20.9 FPS

YOLOv6 N 版為 42 FPS

圖 77 比較 YOLO 系列的效能，辨識招牌
(圖片來源：研究者自行拍攝)

YOLOv4 為 6 FPS

YOLOv4-Tiny 為 40 FPS

YOLOv6 S 版為 20.9 FPS

YOLOv6 N 版為 42.4 FPS

圖 78 比較 YOLO 系列的效能，辨識人
(圖片來源：研究者自行拍攝)

結果與討論：Yolov4 的準確率高，但 FPS 只有 6-8，影像延遲較嚴重，若要即時辨識會比較不易，但物件可準確辨識。Yolov4-Tiny 的幀率為 40 FPS；Yolov6 的 S 版實測時 FPS 為 20 幀率；而 N 版高於 40FPS，均符合本系統的目標幀率要求。

五、 分析內嵌式系統 Nvidia Jetson TX2 的在異物偵測的效能。

(一)、實驗 Yolov4、Yolo4-Tiny、Yolov6 在 Nvidia Jetson TX2 幀率和辨識的效果。

1. 實驗目的：探討不同 YOLO 架構在 Nvidia Jetson TX2 幀率和辨識的效果。
2. 控制變因：200 張數的資料集，YOLOv4、YOLOv4-Tiny：訓練次數為：50000 iterlations。YOLOv6：S 版、N 版訓練次數為 400 epochs。
3. 實驗結果：YOLOv4 約在 4-5 FPS、YOLOv4-Tiny 約在 37 FPS、YOLOv6 S 版約在 15 FPS、YOLOv6 N 版則在 26 FPS。辨識效果如下圖所示：

Nvidia Jetson TX2

YOLOv4 為 5 FPS

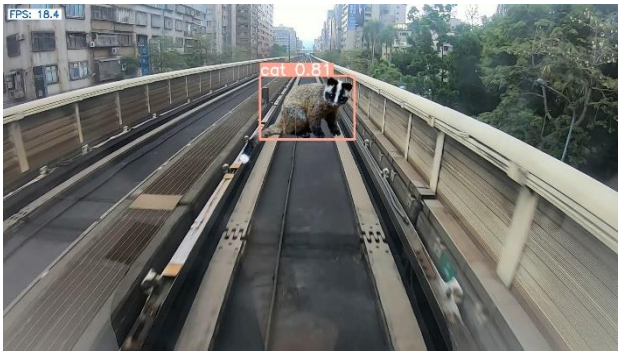
筆電(GPU：Gforce MX150)

YOLOv4 為 6 FPS

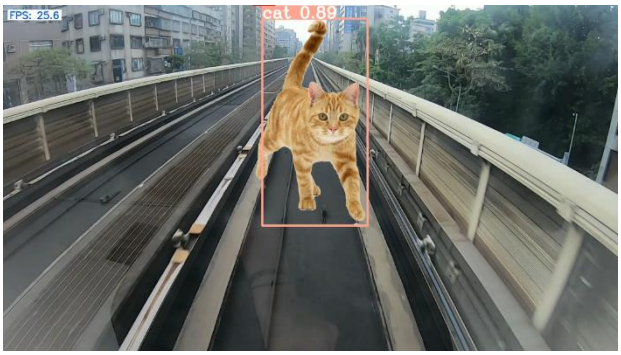
YOLOv4-Tiny 為 37 FPS

YOLOv4-Tiny 為 40 FPS

Nvidia Jetson TX2　　筆電(GPU：Gforce MX150)



YOLOv6 S 版為 14.9 FPS　　YOLOv6 S 版為 18.4 FPS



YOLOv6 N 版為 25.6FPS　　YOLOv6 N 版為 42 FPS

圖 79 比較 TX2 辨識的 FPS
(圖片來源：研究者自行拍攝)

結果與討論：(1). Jeson TX2 運行 YOLOv4、YOLOv4-Tiny、YOLOv6 S 版的 FPS 比在筆電(GPU：Gforce MX150)運行平均略降 1-3FPS，差異不大。

(2). YOLOv6 N 版在 Jetson TX2 運行速度下降很多，約為 16FPS，但仍符合本系統的目標幀率要求。

(3). YOLOv4-Tiny 在 Jetson TX2 運行的速度仍維持在接近 40FPS 的表現，約為 2 倍左右。

六、針對北捷文湖線場域條件進行研究。

(一)、實驗 Yolov4、Yolo4-Tiny、Yolov6 在北捷文湖線場域幀率和辨識的效果。

1. 實驗目的：探討不同 YOLO 架構在北捷文湖線場域幀率和辨識並做比較。

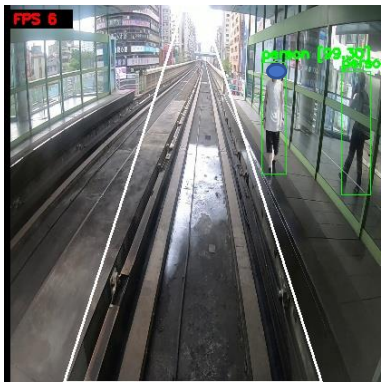
2. 控制變因：200 張數的資料集，YOLOv4、YOLOv4-Tiny：訓練次數為：50000 iterlations。YOLOv6：S 版、N 版訓練次數為 400 epochs。

3. 實驗結果：(1). YOLOv4 的信心值(confidence)均在 90%以上，幀率約 6-7 FPS，如下圖所示：

招牌辨識的信心值為 98%，7 FPS

貓辨識的信心值為 99%，6 FPS

果子狸辨識的信心值為 97%，6 FPS



人辨識的信心值為 99%，6 FPS

圖 80 比較 YOLOv4 在北捷文湖線辨識的各效能
(圖片來源：研究者自行拍攝)

(2). YOLOv4-Tiny 的信心值均在 74%，幀率約 40 FPS，辨識效果如下圖所示：

招牌辨識的信心值為 87%，41 FPS

果子狸辨識的信心值為 74%，43 FPS

貓辨識的信心值為 76%，40 FPS

人辨識的信心值為 85%，41 FPS

圖 81 比較 YOLOv4-Tiny 在北捷文湖線辨識的效能
(圖片來源：研究者自行拍攝)

(3). YOLOv6 S 版信心值在人的辨識最好，信心值在 93%、招牌有 80%、貓為 70%，在果子狸辨識效果則偏低為 49%，容易誤判，如下圖：

招牌辨識的信心值為 81%，21 FPS

果子狸辨識的信心值為 49%，18 FPS

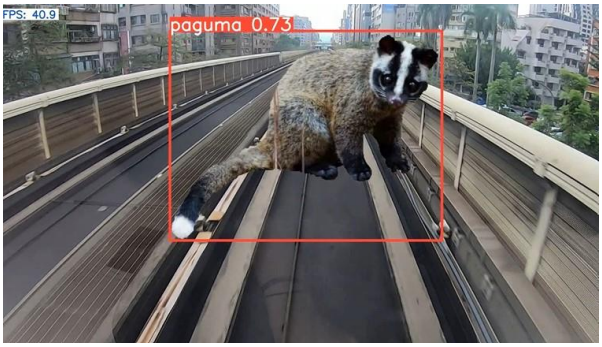
貓辨識的信心值為 70%，20 FPS

人辨識的信心值為 93%，19 FPS

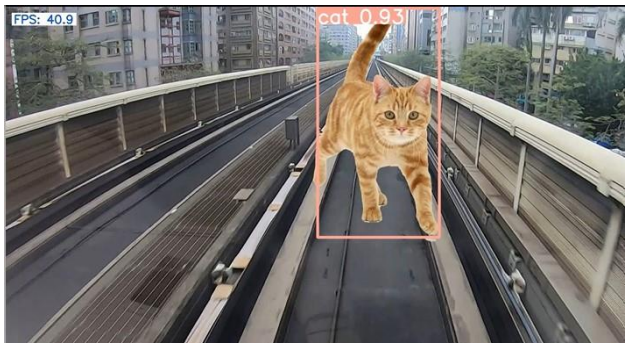
圖 82 比較 YOLOv6 S 版在北捷文湖線辨識的效能
(圖片來源：研究者自行拍攝)

(4). YOLOv6 N 版信心值在招牌、人和貓的辨識最好，信心值均在 89%以上，在果子狸辨識效果也有 73%，如下圖：

招牌辨識的信心值為 90%，41 FPS



果子狸辨識的信心值為 73%，41 FPS



貓辨識的信心值為 93%，41 FPS

人辨識的信心值為 89%，41 FPS

圖 83 比較 YOLOv6 N 版在北捷文湖線辨識的效能
(圖片來源：研究者自行拍攝)

結果與討論：YOLOv4、YOLOv4-Tiny、YOLOv6 在北捷文湖線場域辨識情況似各類別均能辨識，尤以 YOLOv4 的信心值高，但幀率低(FPS)；而果子狸時在辨識情形稍差，對 YOLOv6 S 版在辨識果子狸時信心值低，易誤判為貓。應為果子狸圖片甚難取得，在網路上均為特寫角度，若能取得多的樣本，勢必能改善，增加模型的準確性。比較過後由上圖所示：YOLOv4-Tiny、YOLOv6 N 版的信心值和 FPS 較高，因此 YOLOv4-Tiny、YOLOv6 N 版是符合本系統需要的高幀率與高準確辨識的要求。

七、 研究不同大小物體、不同位置、與攝影機間距離的辨識效果。

(一)、 探討系統對不同大小物體，距離相同的辨識效果。

以北捷文湖線隔音牆的尺寸估算物體的比例和位置。

一個隔音牆約 40 cm x 3=120 cm 高、2m 寬。

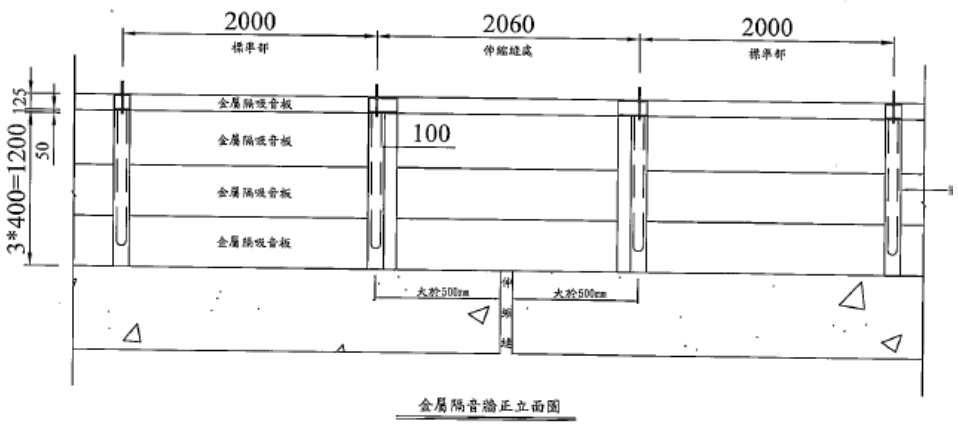


圖 84 北捷文湖線隔音牆的尺寸
(圖片來源：訪談得知)

1. 實驗目的：探討系統可以辨識物體的大小。

2. 實驗變因：測試不同物體高度為分別為約 100cm、80cm、40cm。

3. 控制變因：與攝影機固定約 4m 距離內。

4. 實驗結果：Yolov4：在 200 張照片的資料集模型，物體的大小從高度 20cm 以上均能辨識，僅果子狸辨識不到，如下圖。

招牌辨識結果，100cm 信心值為 99%、80cm 信心值為 99%、40cm 信心值為 95%。

貓辨識結果，100cm 信心值為 99%、80cm 信心值為 94%、40cm 信心值為 72%。

人辨識結果，100cm 信心值為 90%、80cm 信心值為 99%、40cm 信心值為 99%。

果子狸辨識結果，100cm 辨識不到，誤辨識成貓。。

圖 85 比較 YOLOv4 在辨識不同大小的物體。

(圖片來源：研究者自行拍攝)

招牌辨識結果，100cm 信心值為 99%、80cm 信心值為 96%、40cm 沒有辨識到。

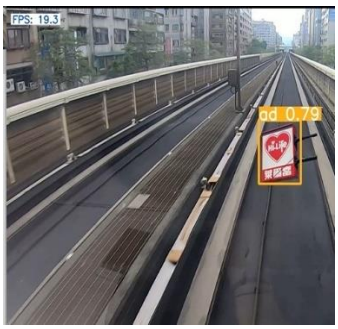
貓辨識結果，100cm 信心值為 89%、80cm 信心值為 77%、40cm 信心值為 45%。

人辨識結果，100cm 信心值為 97%、80cm 信心值為 85%、40cm 信心值為 96%。

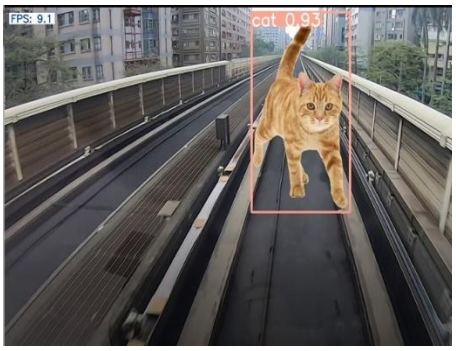
果子狸辨識結果，沒有辨識到。

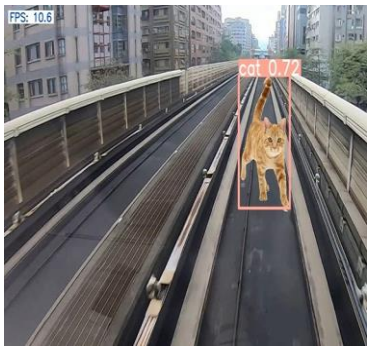
圖 86 比較 YOLOv4-Tiny 在辨識不同大小的物體。

(圖片來源：研究者自行拍攝)

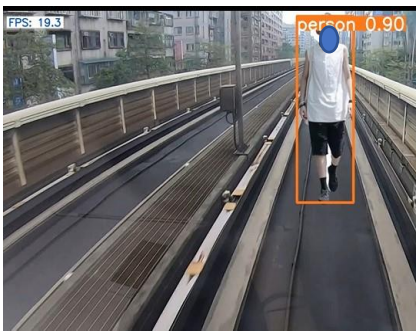


招牌辨識結果，100cm 信心值為 85%、80cm 信心值為 83%、40cm 信心值為 79%。

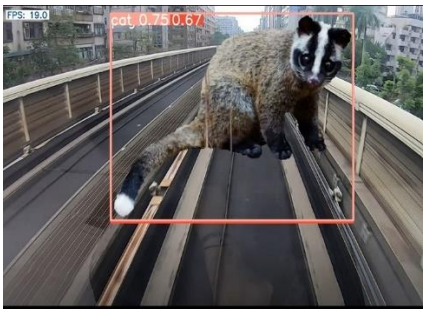


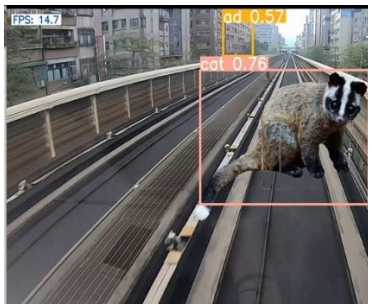


貓辨識結果，100cm 信心值 93%、80cm 信心值為 72%、20cm 信心值為 71%。



人辨識結果，100cm 信心值 90%、80cm 信心值為 93%、40cm 信心值為 90%。



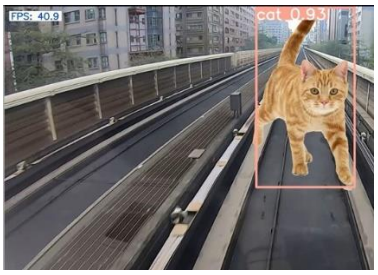


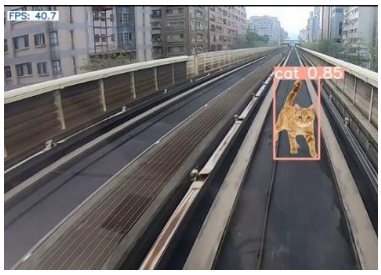
果子狸辨識結果，誤辨識成貓。

圖 87 比較 YOLOv6 S 版在辨識不同大小的物體。

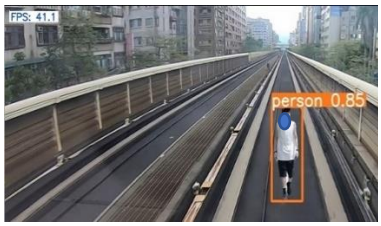
(圖片來源：研究者自行拍攝)

招牌辨識結果，100cm 信心值 95%、80cm 信心值為 90%、40cm 信心值為 80%。

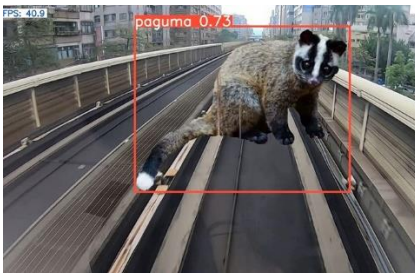




貓辨識結果，100cm 信心值 93%、80cm 信心值為 88%、40cm 信心值為 85%。



人辨識結果，100cm 信心值 92%、80cm 信心值為 89%、40cm 信心值為 85%。



果子狸辨識結果，100cm 信心值 73%、80cm 信心值為 59%、40cm 誤辨識成貓。

圖 88 比較 YOLOv6 N 版在辨識不同大小的類別。

(圖片來源：研究者自行拍攝)

結果與討論： 從圖中可以發現 YOLOv4 除了 FPS 較低，而 100cm 的果子狸未辨識到，實際也沒有 100cm 高的果子狸，其他類別在辨識上都達到 70%以上，表現良好。

YOLOv4-Tiny 的 FPS 均維持在 30-40 之間，在辨識上除了果子狸以外其餘類別辨識情況是可採用的，但偶而會辨識不穩定，尤其是辨識 40cm 的類別。

YOLOv6 S 版的 FPS 維持在 9-20 之間，而辨識果子狸時容易誤判為貓，因體型較相近，且都有尾巴、尖耳朵。可提高果子狸的資料集，讓果子狸與貓有更多可區分的點。YOLOv6 N 版的 FPS 維持在 40。辨識上相比其他演算法更正確，僅 40cm 果子狸未辨識到。較像 YOLOv4 與 YOLOv4-Tiny 的綜合版，因 YOLOv6 N 版可保證 FPS 的速度，同時可兼具辨識的準確度。

(二)、 探討系統對不同位置物體的辨識效果

1. 實驗目的：探討異物在何位置?才需告警。
2. 實驗變因：將畫面分為無分區、矩形區和梯形區。
3. 控制變因：YOLOv4、四個類別各 200 張資料集。
4. 實驗結果：初期發現無劃分區域時，會辨識到道旁招牌等物體，矩形區仍無法達到效果，最後修改區梯形區，就能準確辨識，避免誤判，如下圖：

圖 89 無分區：會辨識到軌道外的招牌(圖片來源：研究者自行拍攝)

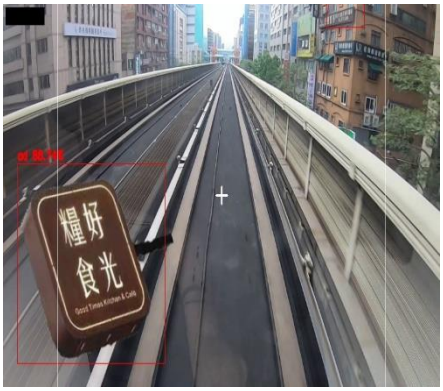


圖 90 矩形區：仍會辨識到(圖片來源：研究者自行拍攝)

圖 91 梯形區：可準確辨識。(圖片來源：研究者自行拍攝)

結果與討論：當物體侵入「辨識區」才須告警，此區域以梯形區，包圍軌道，最能發揮效果，避免誤報警發生。

(三)、 探討系統對不同距離物體的辨識效果

1. 實驗目的：探討系統可以辨識多遠距離的物體。

2. 實驗變因：以隔音牆的尺寸推估物體與攝影機距離約為 4m、6m、10m。

3. 控制變因：相同演算法，相同路段做比較。

4. 實驗結果：(1). Yolov4 在 200 張照片的資料集模型，在辨識物體時 4-10m 距離均能辨識，僅果子狸易誤判成貓。

招牌辨識結果，4m 信心值為 99%、6m 信心值為 100%、10m 信心值為 81%。

貓辨識結果，4m 信心值 99%、6m 信心值為 99%、10m 信心值為 96%。

人辨識結果，4m 信心值 97%、6m 信心值為 98%、10m 信心值為 97%。

果子狸辨識結果，誤辨識成貓。

圖 92 比較 YOLOv4 在不同距離的辨識效果

(圖片來源：研究者自行拍攝)

(2). Yolov4-Tiny 在 200 張照片的資料集模型，在辨識物體時 4-10m 距離均能辨識，信心值(confidence)比 YOLOv4 低，果子狸也易誤判成貓。

招牌辨識結果，約 4m 信心值為 84%、6m 信心值為 98%、10m 信心值為 93%。

貓辨識結果，4m 信心值 77%、6m 信心值為 45%、10m 信心值為 76%。

人辨識結果，4m 信心值 85%、6m 信心值為 99%、10m 信心值為 45%。

果子狸辨識結果，誤辨識成貓。

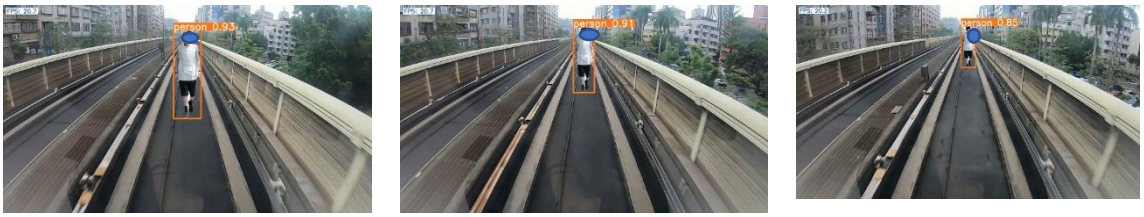
圖 93 比較 YOLOv4-Tiny 物體在不同距離的辨識效果

(圖片來源：研究者自行拍攝)

(3). Yolov6 S 版在 200 張照片的資料集模型，在辨識物體時 4-10m 距離均能辨識，信心值(confidence)比 YOLOv4 低，果子狸距離 4m 可以辨識出(Paguma)，4-6m 也易誤判成貓。

招牌辨識結果，約 4m 信心值為 85%、6m 信心值為 83%、10m 信心值為 80%。

貓辨識結果，4m 信心值 84%、6m 信心值為 77%、10m 信心值為 70%。



人辨識結果，4m 信心值 93%、6m 信心值為 91%、10m 信心值為 85%。

果子狸辨識結果，4m 信心值 49%， 6-10m 誤判成貓。

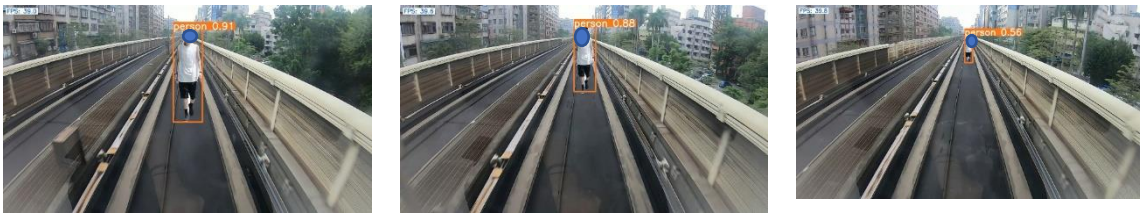
圖 94 比較 YOLOv6 S 版 物體在不同距離的辨識效果

(圖片來源：研究者自行拍攝)

(4). Yolov6 N 版在 200 張照片的資料集模型，在辨識物體時 4-10m 距離均能辨識，信心值(confidence)比 YOLOv4 低，果子狸距離 4m 可以辨識出(Paguma)，4-6m 也易誤判成貓。

招牌辨識結果，約 4m 信心值為 94%、6m 信心值為 93%、10m 信心值為 44%。

貓辨識結果，4m 信心值 89%、6m 信心值為 72%、10m 信心值為 44%。



人辨識結果，4m 信心值 91%、6m 信心值為 88%、10m 信心值為 56%。

果子狸辨識結果，4m 信心值 73%， 6-10m 誤判成貓。

圖 95 比較 YOLOv6 N 版 物體在不同距離的辨識效果
(圖片來源：研究者自行拍攝)

結果與討論：YOLOv4、YOLOv4-Tiny 在物體距離攝影機 4m 外均都可以辨識，以 YOLOv4 信心值較高，準確率高，但果子狸則比較難辨識，易判斷成貓。YOLOv6 系列也在物體距離攝影機 4m 外均都可以辨識，對甚難辨識的果子狸 4m 也能辨識， 尤以 YOLOv6 N 版幀率為目標幀率 20FPS 的二倍，可為本系統採用。果子狸的資料集多來自網站圖片 ，可以在適當調整，取得更多樣本數，來解決辨識度差的問題。

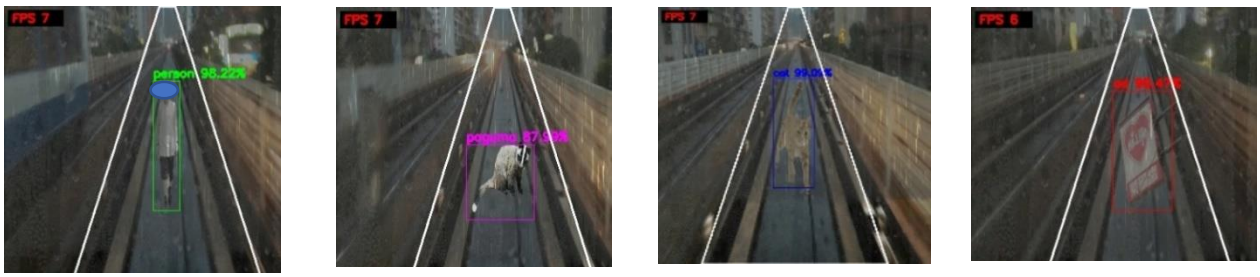
八. 在不同天候，如雨、陰天實測。

1.實驗目的：探討系統在不同天氣條件辨識的效果。

2.實驗變因：天氣的影像，在傍晚陰天、傍晚雨天實測。

3.控制變因：相同 Yolo 架構，辨識物體。

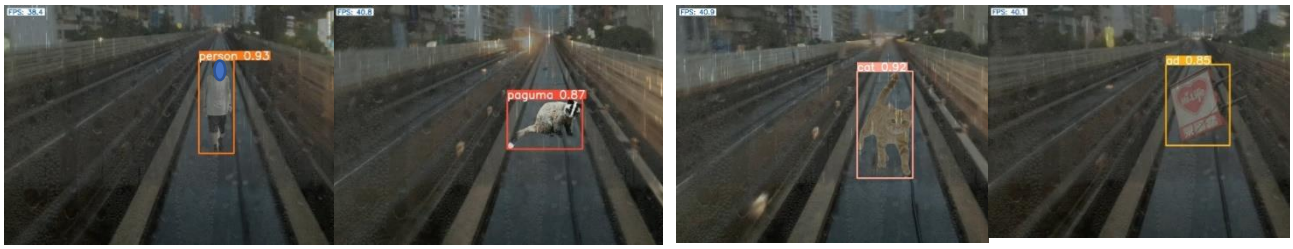
4.實驗結果：本次實驗在傍晚時段陰雨天，測試各 YOLO 演算法的辨識情況，各類別除果子狸辨識易誤判或信心值偏低，其他如人、貓、招牌均能辨識如下：



YOLOv4 雨天辨識效果

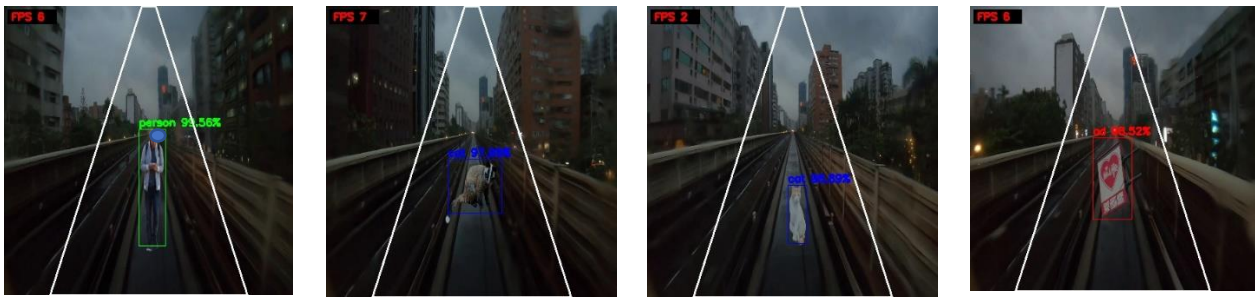
YOLOv4-Tiny 雨天辨識效果

YOLOv6 S 版雨天辨識效果

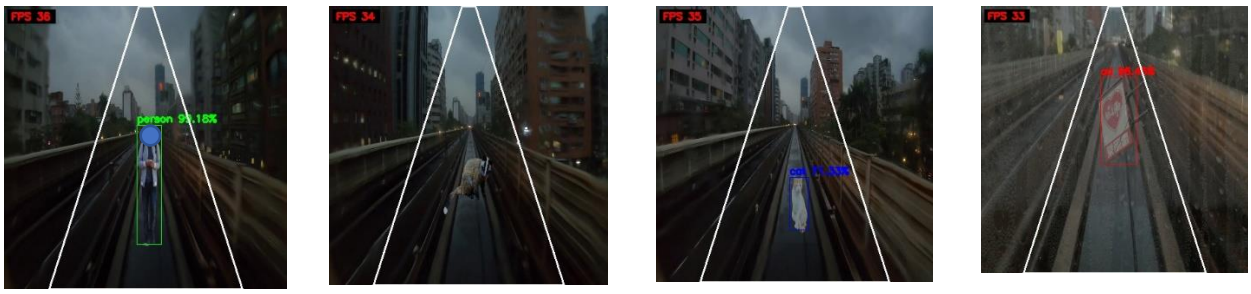


YOLOv6 N 版雨天辨識效果

圖 96 比較 YOLOv6 N 版 物體在雨天辨識效果 (圖片來源：研究者自行拍攝)



YOLOv4 陰天辨識效果



YOLOv4-Tiny 陰天辨識效果

YOLOv6 S 版陰天辨識效果

YOLOv6 N 版陰天辨識效果

圖 97 比較在傍晚雨天、陰辨識效果 (圖片來源：研究者自行拍攝)

結果與討論：如同上方圖所示，本次實驗採用傍晚光線不佳嚴苛環境測試系統能力。比較 YOLOv4 在雨天的環境的影像都可以辨識的到。YOLOv4-Tiny 在辨識陰天時較不穩定，人和果子狸時有，時沒有。YOLOv6 S 版在陰天辨識時，除了果子狸辨識成貓以外，其他類別辨識結果良好。YOLOv6 N 版在所有天氣的影像都可以辨識的到。

九．實驗不同場域辨識效果，如：北捷文湖線、臺鐵內灣支線、淡海輕軌。

1. 實驗目的：分析本系統使用 Jetson TX2 在不同場域運作情況

2. 實驗變因：(1). 不同場景辨識個物體

(2). 不同天候環境- 晴、雨、陰天(依實測當天天候而定)

3. 控制變因：使用 4 類別各 200 張的模型

演算法 YOLOv4-Tiny、YOLOv4

4. 實驗結果： 本次實驗在不同場域天候路段，以 TX2 實測，無論 YOLOv4 或 Tiny 均少誤判情形發生，也 Tiny 維持高的 FPS 約 35-40FPS，如下圖。

北捷-文湖線場域

天氣條件：雨天

圖 98 YOLOv4-Tiny 在各種路段情形 (圖片來源：研究者自行拍攝)

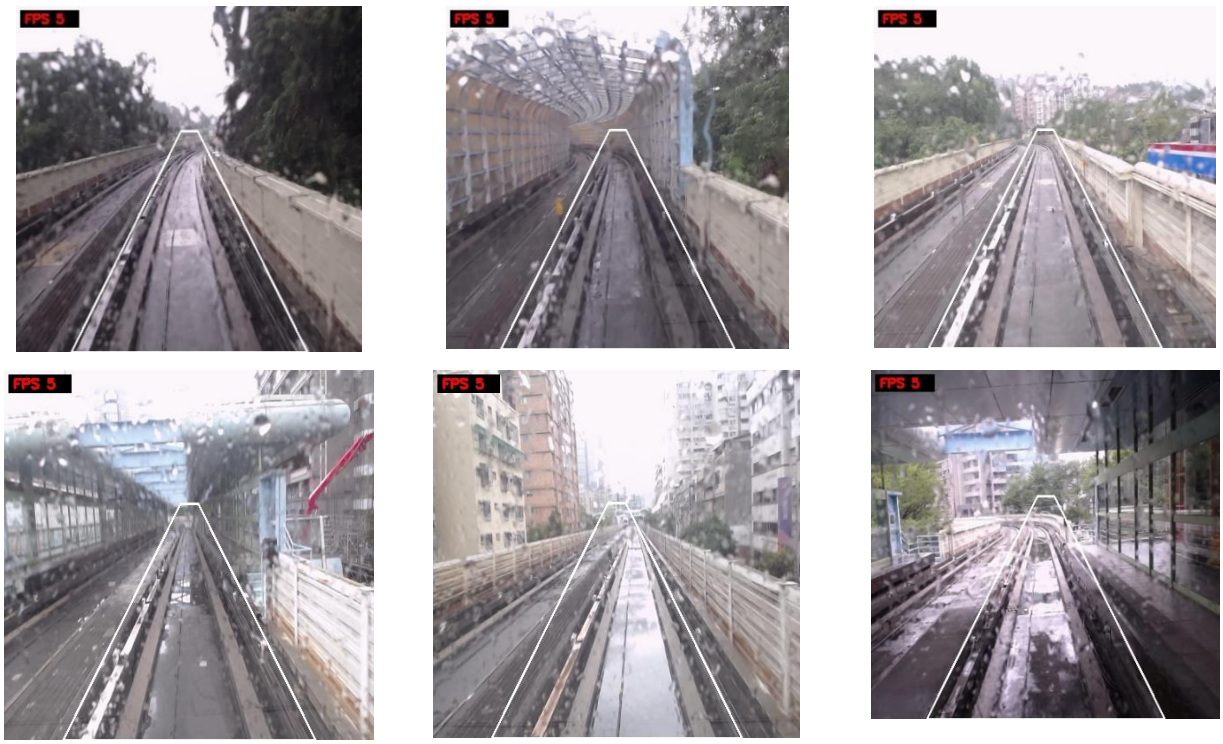


圖 99 YOLOv4-Tiny 在各種路段情形 (圖片來源：研究者自行拍攝)

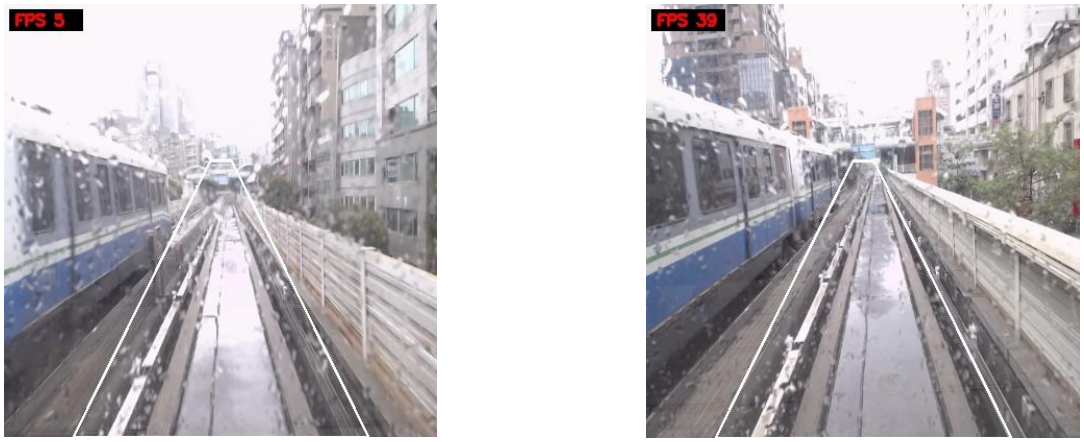


圖 100 會車情形 (圖片來源：研究者自行拍攝)

臺鐵-內灣線場域(內灣-新竹)

天氣條件：晴朗

圖 101 YOLOv4-Tiny 在各種路段情形 (圖片來源：研究者自行拍攝)

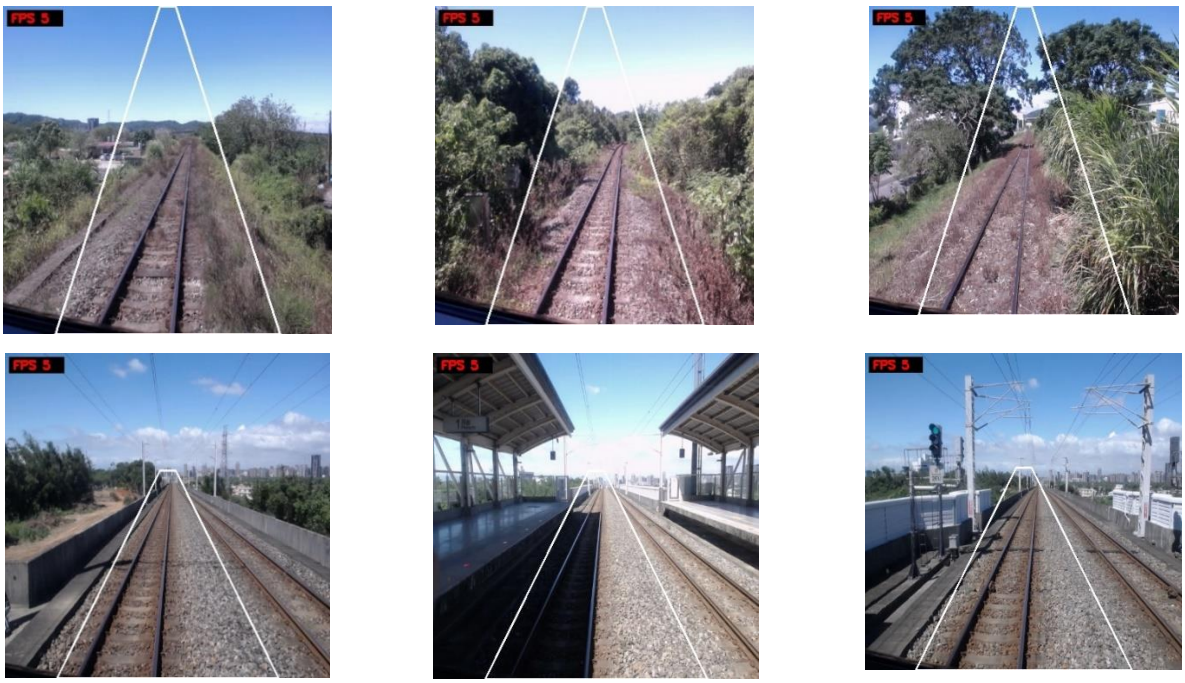


圖 102 YOLOv4 在各種路段情形 (圖片來源：研究者自行拍攝)

圖 103 YOLOv4-Tiny 40FPS 情形 (圖片來源：研究者自行拍攝)

特殊路段

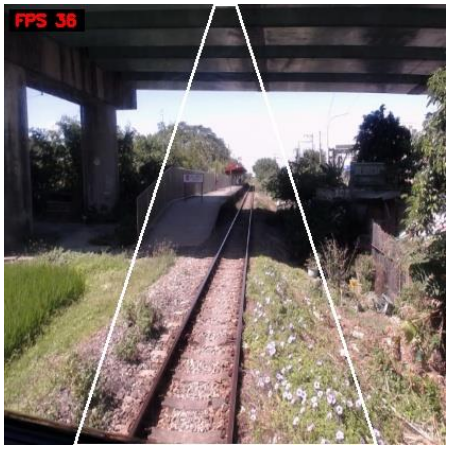


圖 104 YOLOv4-Tiny 橋下情形
(圖片來源：研究者自行拍攝)

圖 105 YOLOv4 未限制辨識區域時，
可辨識到月台上的人
(圖片來源：研究者自行拍攝)

新北捷-淡海輕軌場域

天氣條件：陰、雨天

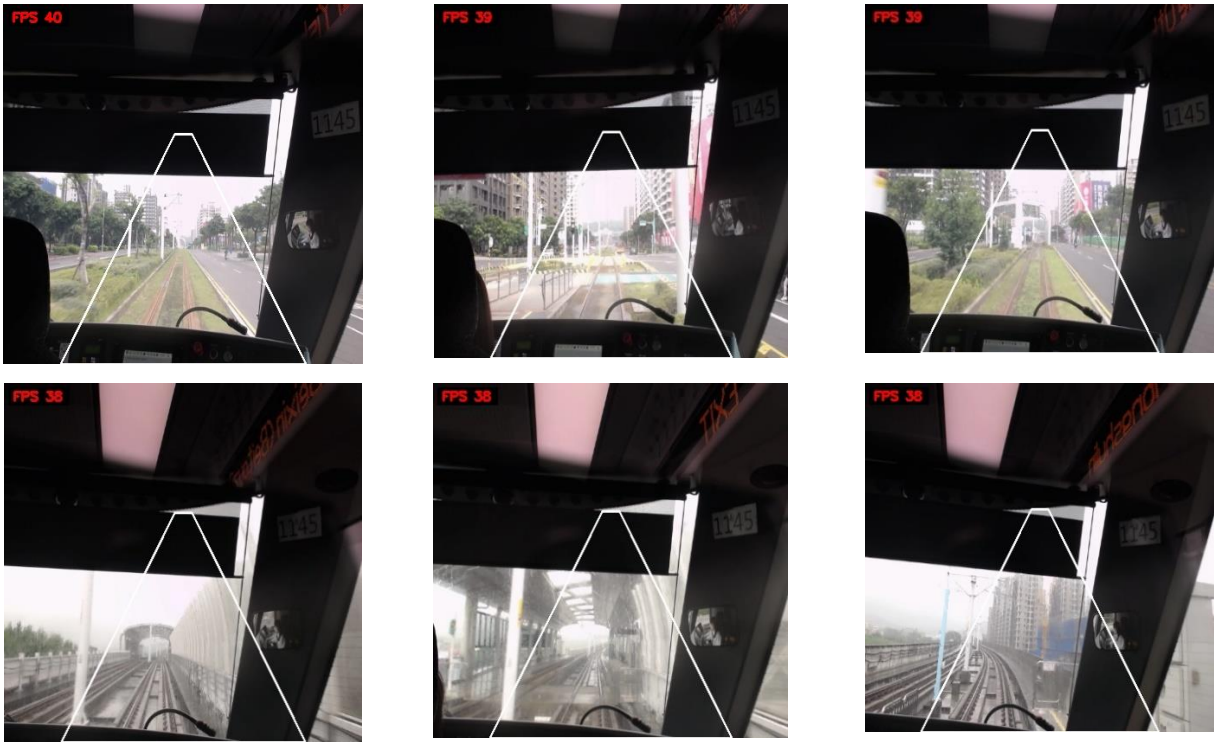


圖 106 YOLOv4-Tiny 在 A(下圖)/B(上圖)型路權情形 (圖片來源：研究者自行拍攝)

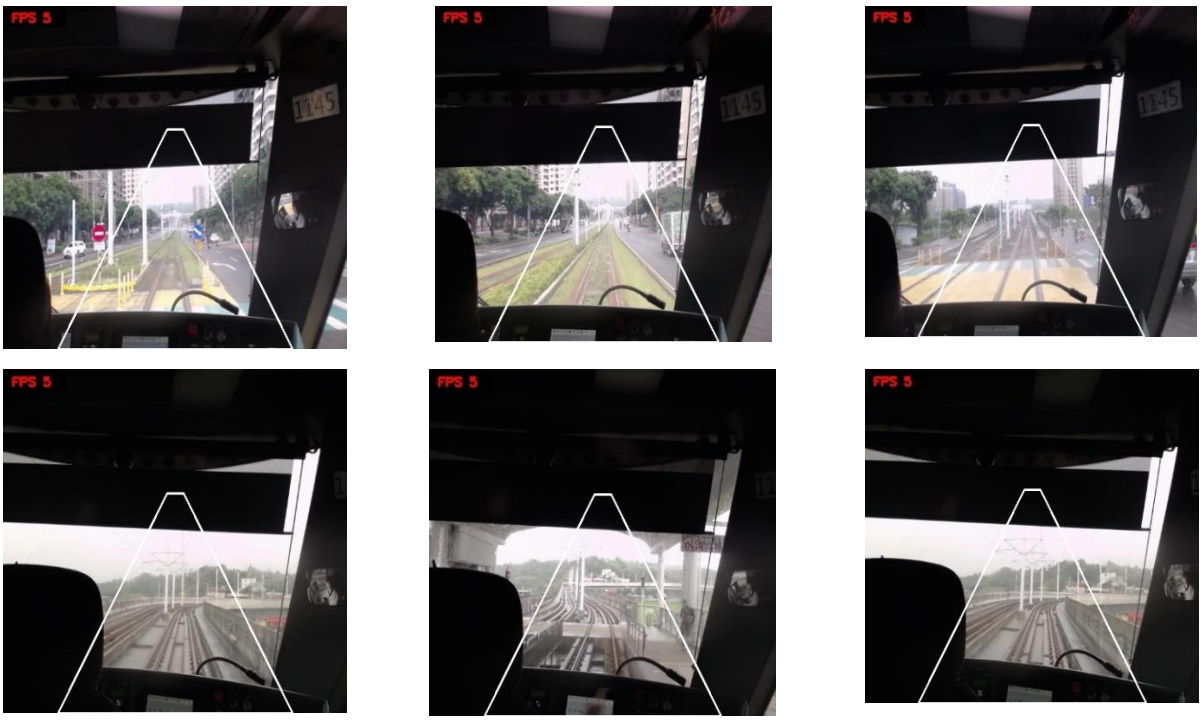


圖 107 YOLOv4 在 A(下圖)/B(上圖)型路權情形 (圖片來源：研究者自行拍攝)

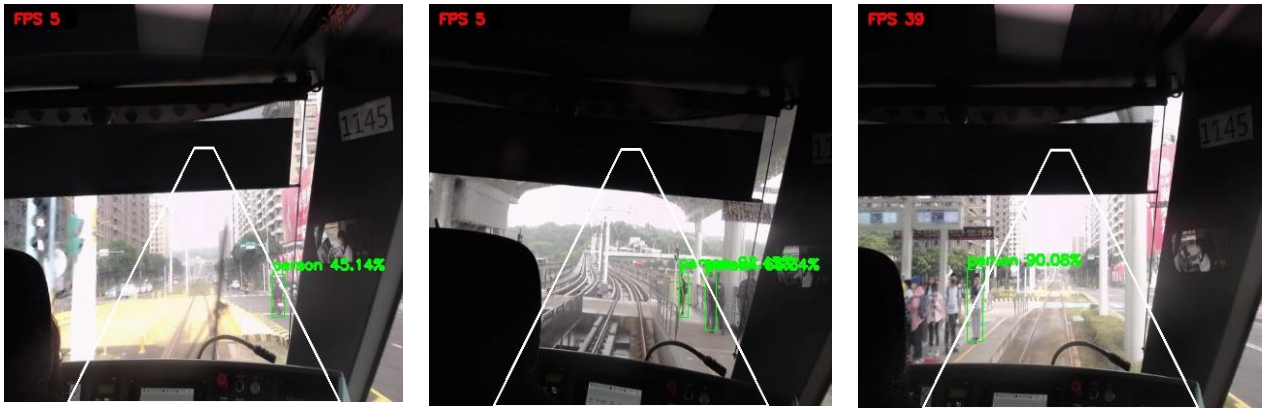
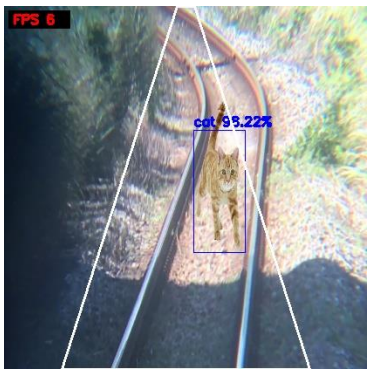


圖 108 YOLOv4、YOLOv4-Tiny 未限制辨識區時，可辨識到道旁的人
(圖片來源：研究者自行拍攝)

圖 109 YOLOv4 會車情形
(圖片來源：研究者自行拍攝)

結果與討論：YOLOv4、YOLOv4-Tiny 實測上可以運行在不同場域，在特殊路段如橋下光影變化時，沒有誤判情況，也能運行在雨天和陰天。北捷在會車時，也沒有誤判發生。實測時在沒有設辨識區時，如月台上的人，是可以辨識到，顯示本系統可以有效運作的。將異物 P 入下圖中，顯示不同場域也能偵測，如下圖。



YOLOv4 在臺鐵內灣支線辨識情況

YOLOv4-Tiny 在北捷辨識情況

圖 110 異物偵測情形
(圖片來源：研究者自行拍攝)

十. 研究加裝望遠鏡頭的辨識效果，讓列車有足夠煞車距離。

(一) 探討使用望遠鏡頭錄製影像與正常鏡頭錄製影像的品質。

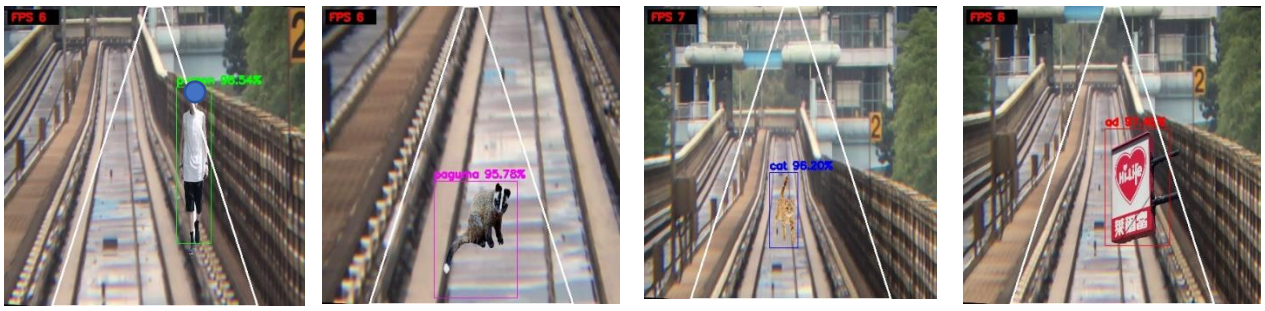
1.實驗目的：使用十倍望遠鏡頭和 iphone 6s 錄製影像，提高辨識距離，讓列車有足夠煞車距離，如下圖 111、圖 112。北捷站間距離平均約 1.5km，從圖 110 可看到 1.5 km 對站的情形。

2. 實驗變因：使用十倍望遠鏡頭在 YOLO 系列辨識效果。

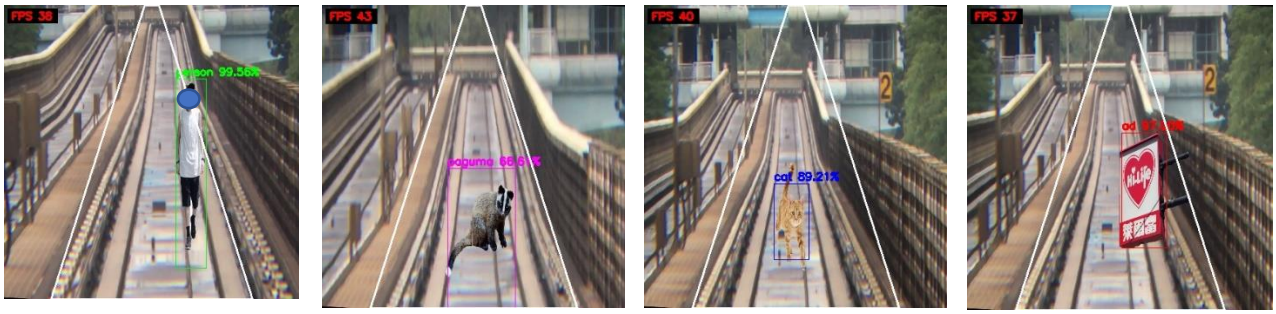
3. 實驗結果：加裝望遠鏡頭可以增加煞車預警距離，無論 YOLOv4、YOLOv4-Tiny、YOLOv6 均能辨識異物，效果如下：

圖 111 望遠鏡頭+手機
(圖片來源：研究者自行拍攝)

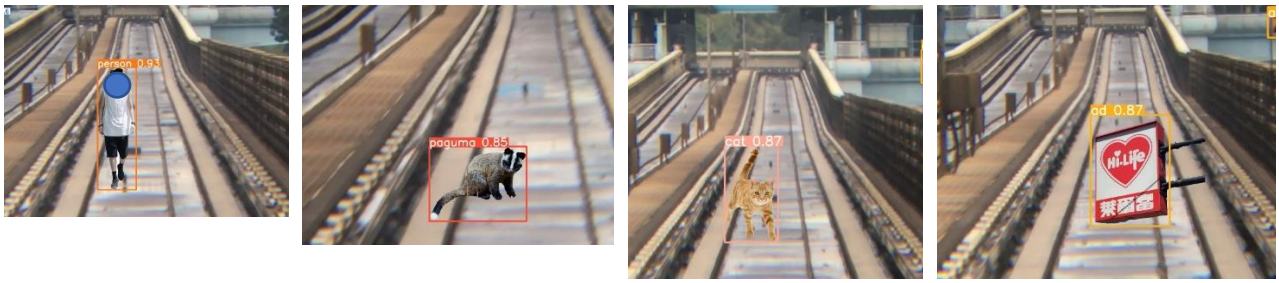
圖 112 使用望遠鏡頭畫面比較
(圖片來源：研究者自行拍攝)



YOLOv4 加望遠鏡頭辨識效果



YOLOv4-Tiny 加望遠鏡頭辨識效果



YOLOv6 S 版加望遠鏡頭辨識效果

YOLOv6 N 版加望遠鏡頭辨識效果

圖 113 比較加望遠鏡頭辨識效果 (圖片來源：研究者自行拍攝)

結果與討論：如同上方圖所示，使用望遠鏡頭達到了 65-70 公尺外的預警，在錄製軌道的影像時，發現使用望遠鏡頭，因列車行駛晃動，影像晃動劇烈，但整體清晰度是可使用的。建議加裝三軸穩定器和電子防震來穩定影像。

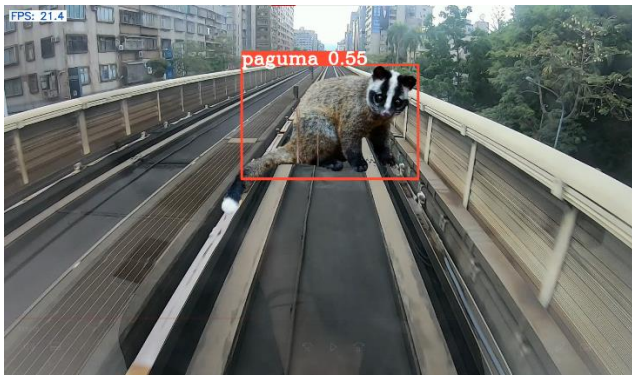
十一、 比較內嵌式系統 Nvidia Xavier NX 在異物偵測的效能。

(一)、實驗 Yolov4、Yolo4-Tiny、Yolov6 在 Nvidia Xavier NX 幀率和辨識的效果。

1. 實驗目的：探討不同 YOLO 架構在 Nvidia Xavier NX 幀率和辨識的效果。
2. 控制變因：200 張數的資料集，YOLOv4、YOLOv4-Tiny：訓練次數為：50000 iterlations。YOLOv6：S 版、N 版訓練次數為 400 epochs。
3. 實驗結果：YOLOv4 約在 10 FPS、YOLOv4-Tiny 約在 50 FPS、YOLOv6 S 版約在 20 FPS、YOLOv6 N 版則在 30 FPS。辨識效果如下圖所示：

YOLOv4 約為 10 FPS

YOLOv4-Tiny 約為 50 FPS



YOLOv6 S 版約為 20 FPS

YOLOv6 N 版約為 30 FPS

圖 114 比較 Nvidia Xavier NX 辨識的 FPS
(圖片來源：研究者自行拍攝)

結果與討論：(1). 將 YOLOv4、YOLOv4-Tiny、YOLOv6 S 版、YOLOv6 N 版的 FPS 整理如表 3，發現 YOLOv4、YOLOv4-Tiny 在 Xavier 的辨識速率都提高，YOLOv4-Tiny 甚至可高達約 50FPS；而 Yolov6 在內嵌式系統的速率均下降許多，因是其演算法的特性。唯 N 版模型仍在目標幀率 20FPS 之上，分別為在 TX2 為 26FPS，在 Xavier 約為 30FPS。

(2). 系統若考慮高準確率和幀率 YOLOv6 N 版為最適當，若考慮較高的幀率，則可選擇 YOLOv4-Tiny。

(3). 本研究提出 YOLOv4-Tiny 和 YOLOv6 N 版的效能優於作者 Wang, Y.和 Yu, P.在其論文中提的 SSD 的方法，其 mAP 為 89%，在 TX2 的幀率為 25.9FPS。

表 3. FPS Comparison

Platfrom / Method	mAP(%)	GPU-MX150	Jetson TX2	Xavier
YOLOv4	93	~6	~5	~10
YOLOv4-Tiny	88	~40	~35	~50
YOLOv6-S	95	~25	~15	~20
YOLOv6-N	95	~43	~26	~30

陸、研究過程中問題

本研究遇到的問題如下：

一、樹和招牌物件難定義

原先定義類別(Class)為：「貓、果子狸、人、樹、廣告牌」，但是發現樹這個類別比較難定義，因一個樹後面還有很多樹，所以要如何認定呢？廣告牌也有同樣問題，要在軌道上定義「倒下的廣告牌」，並不容易定義。如圖 115。解決方法將樹這個類別先去除，並將廣告牌分為「正常」(Good ad.)與「倒下的」(fall ad.)類別，所以規劃目前的類別為：貓、果子狸、人、正常廣告牌(Good ad.)、倒下廣告牌(fall ad.)。

在第二次訓練時發現準確度也沒有很高，主要原因是招牌的背景太亂，圖片模糊，又有倒下與正常倒置難以訓練，因次類別改為改：貓、果子狸、人、正常招牌(ad.)。並自行拍攝招牌的圖片進行訓練，準確度也有相對的提升。如圖 67。

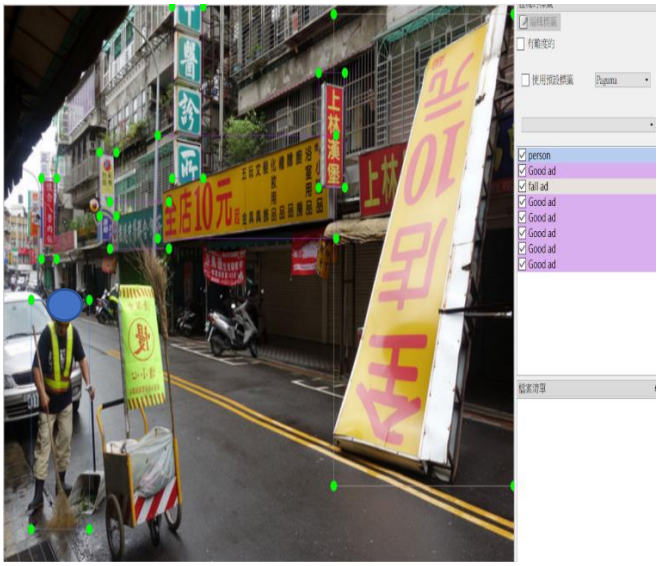


圖 115 廣告牌分為正常(Good ad.)與倒下(fall ad.)。
(圖片來源：Google 網站)

圖 116 較清晰，也背景不亂
(圖片來源：研究者自行拍攝)

二、實測時誤判情形：

實測時發現系統有時會將對向軌道上停放的列車誤判為「招牌」，或將號誌燈辨識成「人」，見下圖，這是因為模型類別中沒有這些類別，可以在訓練模型時，加入這些軌道旁易出現的類別，或特殊景致，使系統可以區分出，或本實測因在列車上，並未調整調整「辨識區」符合現地軌道的區域，若調整「辨識區」符合軌道區域，就能排除道旁的列車或號誌燈，避免誤判。

內灣線誤判對向車　　內灣線誤判號誌燈　　北捷誤判號誌燈

圖 117 誤判的情況
(圖片來源：研究者自行拍攝)

三、果子狸容易誤判成貓：

果子狸為野生動物，其多來自網站圖片，只有特寫鏡頭，對於遠近和不同角度照片甚少。果子狸和貓的特徵相像，甚難區分。本系統測試時，發現僅 YOLOv6 系列可以辨識出但信心值仍偏低，如圖 94、圖 95。顯示其演算法特徵擷取和影像資料增補(Data Augmentation)有其特點。未來可以再適當調整，取得更多果子狸樣本數，來解決果子狸辨識度差的問題。

四、隧道和夜晚問題：

對於北捷的隧道為封閉路段，相對安全，也不允許以遠光燈照射。故本研究目前著重於白天和傍晚為實驗條件。夜晚和隧道環境建議可以使用高功率的紅外線攝影機來補強。

柒、結論與應用

本系統將攝影機架設在列車頭，並加入望遠鏡頭，達到 70 公尺外的預警。以北捷文湖線為實驗對象，時速 70 公里車速，以幀率 20 FPS 為目標。採用可見光攝影機與 AI 物件偵測(Object Detection)的技術，提前確認是何異物，並採取不同煞車措施，讓列車提高安全性和降低誤點率。YOLOv4-Tiny、YOLOv6 的運行的幀率均在本研究的目標幀率值之上，在 GPU-MX150 上運行 YOLOv4-Tiny 和 YOLOv6 N 版可達約 40 FPS，為目標值的 2 倍左右，如圖 77，本系統達到高辨識速率和準確率，YOLOv6 的平均準確率(mAP)達 95%。也使用內嵌式系統 Jetson TX2 完成到內灣鐵路、北捷、淡海輕軌車廂實測。最後在高階內嵌系統 Nvidia Xavier 上運行，YOLOv4-Tiny 高達 50FPS，YOLOv6 N 版也有 30 FPS 的速率。本系統也對不同天候條件進行實測，陰雨天均能辨識。對於夜晚則建議採用高功率的紅外線攝影機做異物偵測。「高效能 AI 軌道異物入侵系統」，可以有效減少列車事故的發生，並保障人車的安全。

捌、未來展望

一、使用 TensorRT 對 AI 模型優化，達到高的幀率，滿足高速度運行偵測的需求。

二、使用更高階嵌入式系統進行實測，達到高速度偵測和 Edge 運算的需求。

三、加入更多類別和增加異物資料集數量，如汽車、工程車、交通號誌、和列車等，使模型可以更準確和穩健。

四、增加果子狸資料集的數量和照片的多樣性，使其易於辨識。

玖、參考文獻

1. 嚴文廷、曹馥年、林雨佑 (2021)。*台鐵近 60 年最嚴重意外：太魯閣號事故 49 死，213 人輕重傷*。取自報導者網址 https://www.twreporter.org/a/taiwan-railway-408-taroko-accident

2. 鄭瑋奇 (2022)。*又見異物入侵軌道 台鐵局長：將嚴懲段長等 3 人*。 取自自由時報網址 https://news.ltn.com.tw/news/life/breakingnews/3813080

3. 曹悅華 (2021)。*台鐵完成 11 處邊坡落石告警！路段曝光 明年還要做 15 處*。取自聯合新聞網址 https://udn.com/news/story/7266/5997870

4. 李宜秦 (2022)。*異物入侵威脅安全！台鐵建 26 處「落石告警系統」 靠 AI 提高辨識度*。取自 Etoday 新聞雲網址 https://www.ettoday.net/news/20220206/2174684.htm

5. 盧逸峰、游明煌、邱瑞杰、邱瓊玉(2022)。*台鐵平溪線列車 1 天 2 度撞落石*。取自聯合報網址 https://udn.com/news/story/7266/6610308

6. 郭逸 (2015)。*文湖線列車行進間遭掉落招牌砸中 幸無人傷亡*。取自自由時報網址 https://news.ltn.com.tw/news/society/breakingnews/1347923

7 Alexey Bochkovskiy and Chien-Yao Wang and Hong-Yuan Mark Liao (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *Computer Vision and Pattern Recognition*. arXiv:2004.10934.

8. AlexeyAB/darknet (2021). *YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet*. Retrieved from Github, Web site: https://github.com/AlexeyAB/darknet

9. 張家銘 (2021)。*YOLOv4 產業應用心得整理 -張家銘*。取自 AI 台灣人工智慧學校網址 https://aiacademy.tw/yolo-v4-intro/

10. 李謦伊 (2020)。 *YOLOv4 訓練教學*。取自謦伊的閱讀筆記網址 https://medium.com/ching-i/yolo-c49f70241aa7

11. 王智偉 (2019)。*FPS 幀數是什麼？24fps 、30fps 、60fps 有什麼區別？*取自 Rene.E Laboratory 網址 https://www.reneelab.net/frames-per-second.html

12. 李謦伊 (2020)。*YOLO 演進 — 3 — YOLOv4 詳細介紹*。取自謦伊的閱讀筆記網址 https://medium.com/ching-i/yolo%E6%BC%94%E9%80%B2-3-yolov4%E8%A9%B3%E7%B4%B0%E4%BB%8B%E7%B4%B9-5ab2490754ef

13. Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, Xiaolin Wei (2022). YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *Computer Vision and Pattern Recognition,* arXiv:2209.02976.

14. ANAI: An 'All-in-One' No Code AI platform. *YOLOv6 : Explanation, Features and Implementation.* Retrieved from Medium, Web site: https://medium.com/@ANAI-DemocratizingAI/yolov6-explained-in-simple-terms-c46a0248bddc

15. meituan/YOLOv6 (2022). *YOLOv6: a single-stage object detection framework dedicated to industrial applications.* Retrieved from Github, Web site: https://github.com/meituan/YOLOv6

16. Bernat Puig Camps. *Error analysis for object detection models.* Retrieved from Data Science at Microsoft, Web site:https://medium.com/data-science-at-microsoft/error-analysis-for-object-detection-models-338cb6534051

17. YOLO detection (2021)。*Learning YOLO detection algorithm*。取自拿了橘子跑網址 https://wuhongyui.github.io/posts/yolo/

18. Tommy Huang (2018)。*深度學習-什麼是 one stage，什麼是 two stage 物件偵測件*。取自 Medium 網址 https://chih-sheng-huang821.medium.com/%E6%B7%B1%E5%BA%A6%E5%AD%B8%E7%BF%92-%E4%BB%80%E9%BA%BC%E6%98%AFone-stage-%E4%BB%80%E9%BA%BC%E6%98%AFtwo-stage-%E7%89%A9%E4%BB%B6%E5%81%B5%E6%B8%AC-fc3ce505390f

19. 張順祥 (2021)。*提升安全 台鐵平交道增設自動偵測障礙物*。取自中央社網址 https://www.rti.org.tw/news/view/id/2088124

20. 王嘉慶(2021)。*驚險一瞬間 高鐵「預警神器」發威 救了全車乘客*。取自中時新聞網址 https://www.chinatimes.com/realtimenews/20210808001948-260405?chdtv

21. 臺北市政府捷運工程局(2021)。*捷運工程叢書 精進版*。取自臺北市政府捷運工程局網址 https://ebook.dorts.gov.taipei/#n3

22. COCO. *COCO Dataset*. 取自 COCO 官網址 https://cocodataset.org/#download

23. Jason Chen (2020)。 *【教學】從 COCO Dataset 中提取所需的類別資料*。 取自 Jason Chen Blog 網址 https://jason-chen-1992.weebly.com/home/coco-dataset

24. 林雅雯、謝禎冏、黃維信、謝尚琳、洪瑋宏、李明德 (2020)。*扣件缺失辨識系統建置研究*。港灣季刊，(116)， 20-32。

25. Yohannes, Ervin and Lin, Chih-Yang and Shih, Timothy K. and Hong, Chen-Ya and Enkhbat, Avirmed and Utaminingrum, Fitri (2021). Domain Adaptation Deep Attention Network for Automatic Logo Detection and Recognition in Google Street View. *IEEE Access, 9,* 102623-102635. doi:10.1109/ACCESS.2021.3098713.

26. 國家運輸安全調查委員會 (2021)。 *110 年鐵道列車紀錄裝置普查報告*。國家運輸安全調查委員會。

27. Leoliao (2016)。*存储进程输出 Queue*。取自莫頓 PYTHON 網址 https://mofanpy.com/tutorials/python-basic/multiprocessing/queue

28. 張嘉鈞 (2021)。*NVIDAI Jetson Nano 深度學習應用-使用 OpenCV 處理 YOLOv4 即時影像辨識*。取自 DesignSpark 網址 https://www.rs-online.com/designspark/nvidai-jetson-nano-opencvyolov4-cn

29. Jetson TX2 開發人員套件與模組。*Jetson TX2*。取自 Nvidia 網址 https://www.nvidia.com/zh-tw/autonomous-machines/embedded-systems/jetson-tx2/

30. 維基百科。*臺灣鐵路管理局*。取自維基百科網址 https://zh.m.wikipedia.org/zh-tw/%E8%87%BA%E7%81%A3%E9%90%B5%E8%B7%AF%E7%AE%A1%E7%90%86%E5%B1%80

31. 維基百科。*內灣線*。取自維基百科網址 https://zh.m.wikipedia.org/zh-tw/%E5%85%A7%E7%81%A3%E7%B7%9A

32. 羅心樂 (2022)。*鐵道守護者_高準確率 AI 鐵道辨識異物入侵系統設計之研究*。2022 台灣國際科學展覽會。

33. Wang Y, Yu P. A Fast Intrusion Detection Method for High-Speed Railway Clearance Based on Low-Cost Embedded GPUs. Sensors (Basel). 2021 Nov 1;21(21):7279. doi: 10.3390/s21217279. PMID: 34770584; PMCID: PMC8587953.

【評語】190005

本作品具有實用價值。本作品在北捷文湖線的現場拍攝軌道影片和相片，然後把四種不同物件的照片貼上軌道現場相片後再使用預先訓練好的 Yolo 模型進行物件種類的辨識。此外還量測不同 Yolo 版本執行在不同硬體平台上可以支援的最高 frame rate。目前系統能辨識的物件種類只有四種，不符合真實世界可能會發生的情境（例如：台鐵的開放式鐵軌上可能會出現很多不同種類的物件如汽車、摩托車和腳踏車等），建議未來可以擴充所要辨識物件種類的數目來增強此作品的實用性。另外，目前此作品是把不同物件的照片貼上軌道現場相片來當成測試的場景，因為這些物件的相片是從一些非軌道現場取來貼上使用而不是拍攝於真正掉落在軌道上的場景，因此測試所得的 accuracy 可能與真實情場景所得的 accuracy 不同，建議未來能改善這個問題。