

深度學習YOLOv4物件偵測技術建立即時動態車牌辨識系統

劉振隆¹，劉恩睿^{2*}

摘要

建立一個入侵偵測系統(Intrusion Detection System, IDS)上，傳統作法可能採用如圖像相似度來進行比對與分析，此種方法雖簡單快速，但主要缺點為無法明確知道所偵測到的物件為何物。鑒於目前屬於電腦視覺領域正在發展中的 YOLO (You Only Look Once)物件偵測技術係以深度學習為基礎，相較其他物件偵測技術上，YOLO 具有高效率的物件偵測能力並能準確地識別出物件之種類與位置定位。另外，車牌辨識系統的建立在現今科技執法中可進行違規取締、車流分析與交通疏導等任務，尤其在過去執法人員為了追蹤違規車輛，必須透過調閱大區域範圍內的有用監視器，利用肉眼過目可疑時段的畫面來找尋特定車牌，相當的耗時與費力，但若透過車牌辨識系統的有效執行將可大幅度減輕工作量。故本文將採用 YOLOv4 偵測技術應用於車牌輪廓(contour)與定位(position)偵測，再以光學字元辨識(Optical Character Recognition, OCR)以及採用字元分割(Character Segmentation)之卷積神經網路(Convolutional Neural Network, CNN)技術執行車牌字元辨識，以建立高效率的 YOLO-CNN 即時動態車牌辨識系統。技術建立有效的車牌辨識系統。本文使出這種以深度學習為基礎之 YOLOv4 物件偵測技術應用於車牌辨識，將進行靜態照片與動態影片之物件偵測分析以及靜態停放與動態行駛之台灣汽機車的車牌辨識分析上，本文並將以 TKinter 建立整合 GUI (Graphic User Interface)系統可顯示車輛、車牌偵測與定位影像以及車牌辨識之呈現。

關鍵字：YOLO 物件偵測技術、光學字元辨識(OCR)、卷積神經網路(CNN)、車牌偵測、TKinter、整合 GUI 系統

¹劉振隆，義守大學資訊管理學系 教授。Email: jlliu@isu.edu.tw

²劉恩睿(通訊作者)，國立臺東大學綠色與資訊科技碩士學位學程 助理教授。Email: ejliu@nttu.edu.tw

Data Visualization of Unstructured Text Mining Analysis

Jenn-Long Liu¹, En-Jui Liu^{2*}

Abstract

In the establishment of an Intrusion Detection System (IDS), traditional methods may use techniques such as image similarity to compare and analyze, which although simple and fast, have the main drawback of not being able to clearly identify the detected objects. Considering the development of the computer vision field, the YOLO (You Only Look Once) object detection technology, based on deep learning, has higher efficiency in object detection compared to other techniques and can accurately identify the type and location of objects. Additionally, the establishment of a license plate recognition system in current technology enforcement can perform tasks such as violation detection, traffic analysis, and traffic guidance. In the past, law enforcement officers had to search for specific license plates by manually reviewing footage from useful monitors over a large area, which was time-consuming and labor-intensive. However, through the effective implementation of a license plate recognition system, the workload can be significantly reduced. Therefore, this article will use YOLOv4 detection technology to detect license plate contours and positions, and use Optical Character Recognition (OCR) and Convolutional Neural Network (CNN) techniques for character recognition and segmentation, respectively, to establish a high-efficiency YOLO-CNN real-time dynamic license plate recognition system. The article will analyze static photos and dynamic videos using YOLOv4 object detection technology for license plate recognition in Taiwan's automobiles and motorcycles, and create a GUI system with TKinter to display the vehicle and license plate detection and location images, as well as the license plate recognition results.

Keywords: YOLO Object Detection Technology, Optical Character Recognition (OCR), Convolutional Neural Network (CNN), License Plate Detection, TKinter, Integrated GUI System

¹ Jenn-Long Liu, Professor, Department of Information Management, I-Shou University, Email: jlliu@isu.edu.tw

² En-Jui Liu (Corresponding Author), Assistant Professor, Master Program of Interdisciplinary Program of Green and Information Technology, National Taitung University, Email: ejliu@nttu.edu.tw

壹、簡介

一般在我們的日常生活中常用到許多科技方法工具去執行偵測(Detection)動作，這將使得我們在工作或執行特殊任務上更加有效率與便利性。例如在日常生活中普遍利用雷達偵測即時的氣候情況，有利於我們在出門時選擇攜帶雨具與否或穿衣的決定；利用超音波測量距離，便利於我們在做距離的衡量，若裝在汽車上便能幫助駕駛在倒車時有用的輔助設備。而在日常生活中使用最多的就是影像的偵測，譬如在醫療方面，醫生可以透過顯微鏡去偵測病人體內的大腸息肉(黃威銘, 2020)，或是利用血液抹片檢查出的圖像來做生理異常的分析(Sigrid, 2021)，進一步為病患作醫療診治。在生物物種研究應用方面上，我們可以透過鏡頭內的影像對昆蟲的路徑進行偵測及分析，有助於學者的觀測，還能利用定點攝影機及鏡頭，針對不同種類的鳥類在不同時間點的遷徙，可以分析鳥類的習性或是察覺瀕臨絕種的問題，進一步做出適當的保護措施。在交通方面，我們可以透過影像去偵測車牌，不僅可以讓執法人員去偵測贓車，或是透過路口的監視影像及行車紀錄器，執行科技執法，將有違規的事項及情節進行取締(蘇浩平, 2017)。對於在停車場出入口的管制上，影像偵測將有利於管理人員做好車流的控管(洪文斌, 2021)，也能利用在高速公路的過路繳費系統上，針對車輛進行自動繳費扣款的動作(林永鑫、林政宏, 2018)，方便用路人免去繳費繁瑣的步驟，能提升作業效率。

由上述可知偵測技術對於人們在日常生活與執行特殊任務的重要性，Redmon 和 Farhadi 於 2015 年提出了基於單一神經網路的目標檢測系統：YOLO (You Only Look Once)，YOLO 可以一次性預測多個 Box 位置和類別的卷積神經網路，它能實現端到端(end-to-end)的目標檢測和識別，而 YOLO 最大的優勢就是執行速度快與偵測物件具有高準確性(林永鑫、林政宏, 2018)。由於 YOLO 並未使用滑動視窗 (sliding window) 及提取 proposal 的網路訓練方式，而是直接使用整張圖片影像的訓練模型，如此可以讓目標和背景區域有更好的區分及辨別，也更能抓取到我們所需要偵測的物件。在 YOLO 偵測技術的發展上，第一代的 YOLO 是為了解決之前 two-stage 方法把檢測問題分成兩部分，多階段的訓練導致優化困難，所以用一個網路輸出和類別實現了 one-stage 的方法(楊松儒, 2019)。而第二代 YOLO (YOLOv2)的產生是為了解決第一代檢測性能低及在檢驗時受到數據集標籤的一些限制，故提出了聯合訓練方法和數據集的結合方法像是 ImageNet 和 COCO，經由訓練得知的模型為 YOLOv2。第三代 YOLO (YOLOv3)的提出是在 YOLOv2 的基礎上做改良，像是邊界框預測、類別預測、多尺度預測還使用新網路 Darknet-53 作為提取特徵之用。第四代的 YOLO (YOLOv4)是對第三代的一個改進，提出了一個總和所有檢測技巧的方法，經過篩選及排列，找出最佳的組合(catOneTwo, 2020; Bochkovskiy et al., 2020)。由於 YOLOv4 具有相當複雜之深度學習架構且能很有效的準確偵測出各種物件與定位，目前已廣被應用於各種物件偵測領域上。

對於深度學習領域的發展上拜 1980 年代機器學習的興起之賜，讓電腦能夠透過從大量的資料中找出規律來學習、並自行更正錯誤及進行人工優化。1986 年時 Hinton 提出「反向傳播算法」用來訓練人工神經網路，不過當時技術還尚未成熟，所以提出不久後便逐漸沒落。直到 2006 年，Hinton 提出了受限玻爾茲曼機和深度信念網路兩種概念(芮嘉瑋, 2021)。深度學習是機器學習的方法之一，描述著透過模仿生物的神經網路運作方式的模型，以達成類似生物的決定與判斷能力(維基百科, 2023a)。深度

學習的特性就是透過「學習」的方式，經由大量的訓練樣本，最後得出訓練模型可以去做後續預測或求最佳解的研究。最基本深度學習模型共有 3 個層級，包含輸入層、隱藏層及輸出層。人類的神經元會接受其他神經元輸入的刺激，透過總刺激大小，最後決定輸出多少。而人工神經網路若要實現擬人腦的這種思維效果，則可在網路架構中加上權重(weights)、偏值(bias)及非線性激活函數(activation function)來求解出最後的輸出(馮玄明等, 2017)。在深度學習中重要的神經網路架構有三種，全連結神經網路、遞迴神經網路及卷積神經網路(陳信希等, 2020)。全連結神經網路(fully connected NN)是最常見的神經網路，其通常搭配其他網路的模式做使用。遞迴神經網路(RNN)，用來描述動態時間行為(維基百科, 2022)，所以擁有預測能力，會把每一次的輸出儲存在暫存記憶空間裡，而當次的輸入會與前一次的輸出整合起來後作為當次的輸出結果，通常可應用於在行為預測、語音辨識等工作上。而卷積神經網路(Convolutional Neural Network, CNN)是一種前饋神經網路，可以利用反向傳播演算法，在結構上擁有卷積層、線性整流層、池化層及完全連接層(維基百科, 2023b)，通常應用於圖像識別與影像識別上。卷積神經網路係通過模組不斷的疊加或是加上多層的類神經網路後形成深度學習的模型。整個卷積神經網路係透過稀疏交互(sparse interactions)、參數共享(parameter sharing)及等變表示(equi-variant representations)方式來改進訓練的效率及效能。卷積層與池化層可以重複使用，主要是依照系統架構的複雜程度而決定次數多寡。卷積層是利用卷積核來檢查不同的特徵，而把不同特徵的強度分別記錄下來。圖像經由不同卷積核的滑動視窗與權重計算乘積和後(示意圖，如圖 1(a))，最後會輸出特徵圖。池化層主要特色是對輸入的特徵圖進行壓縮、以減少計算成本、防止過度擬合、並提取主要特徵，且具有特徵不變性。池化層常用的有 2 種方式，包含平均池化層及最大池化層。平均池化層，目的是對鄰近區域內的特徵求平均，能很好的保留背景，計算方法則是範圍區域內的值相加後平均，即為新的輸出值。最大池化層，目的是對鄰近區域的特徵點內取最大值(示意圖，如圖 1(b))，能更好的保留紋理特徵，更佳突顯邊緣，且擁有更好的減少雜訊功能，計算方法則是在範圍區域內的值取得最大值，即為新的輸出值。在 CNN 中，完全連接層之目的是做最後的整合與分類，在完全連接層中的每個神經元都會與上一層的所有神經元有連結，再進行輸出(維基百科, 2023b)，如圖 2。計算方法則是每一個的輸出都看成為前一層的每一個結點乘以一個設定的權重 w ，再加上一個偏值(bias)做調整來得到，最後帶入激活函數中，得到真正的輸出。大致上而言，卷積神經網路之深度學習模型可以應用於物體追蹤、語音辨識、影像辨識、行為辨識及語音助理等領域。

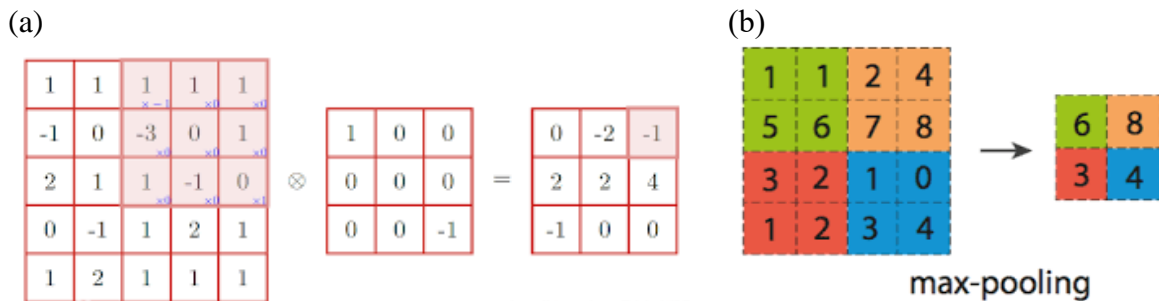


圖 1. (a)卷積層滑動與運算及(b)池化層(最大池化層)特徵取值示意圖

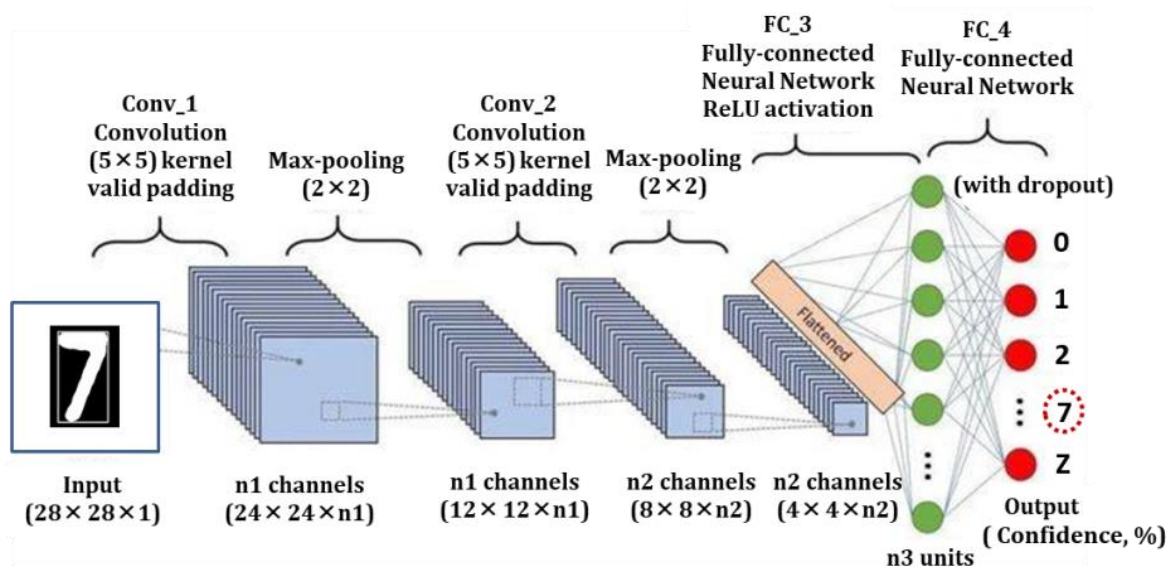


圖 2.卷積神經網路(CNN)示意圖

鑒於車牌辨識系統已成為違規取締、科技執法、交通疏導等發展中的關鍵角色，過去執法人員為了追蹤違規車輛，必須透過調閱大量範圍內的所有監視器，利用肉眼過目可疑時段的畫面、找尋特定車牌，相當的耗時與費力。所以隨著科技進步，上述繁瑣辛苦之事，可透過車牌辨識系統來減輕工作量。傳統車牌辨識是找尋特徵，在經過基本影像處理過後，藉由找尋特徵再做分群之流程(周弘裕, 2019)，如常用之 k-means 分群方法(郭弘裕, 2017)或 SVM (Support Vector Machine) (Cortes & Vapnik, 1995)。然而傳統車牌偵測與辨識方法較適用於正面視角的情況下，若遇到車輛車牌傾斜及模糊影像容易造成車牌定位錯誤或準確率大幅下降，故本研究將採用以深度學習為基礎之高效率 YOLOv4，透過大量的車牌影像訓練出有效的車牌訓練模型進行學習，可提升車牌輪廓與定位之偵測準確性，再將此車輛擷取出來的車牌影像經過灰階(gray scale)、透視變換(perspective transformation)影像轉正、圖像之二值化(binanzation)、形態學(morphology)之侵蝕(erosion)/膨脹(dilation)凸顯影像的形狀特徵、高斯模糊(Gaussian blur)或中值(median blur)做濾波(filtering)影像平滑等 Open CV 影像處理程序後，提供後續車牌辨識流程(林宏軒, 2019)。本研究建立之 YOLO-CNN 車牌辨識整合系統將可以有效地使用於智能停車場、執法人員追蹤車輛及 AI 智慧城市車流管制之服務應用上。

貳、研究方法

一、YOLO 偵測技術

一般而言，物件辨識有 one-stage 和 two-stage 兩類的方法，RCNN (Region-based Convolutional Neural Networks) (Girshick, 2015) 系列為屬於 two-stage 的方法，先得到候選區域(region proposals)對影像物件定位後再進行識別。由於多了一層需要處理 proposals 的結構，故辨識速度較慢；而 YOLO 物件偵測技術為人工智慧與電腦視覺領域，屬於 one-stage 的單個卷積神經網路(single-stage CNN)架構來處理影像，2015 年 6 月 Redmon 等人提出第一代 YOLOv1 (Redmon et al., 2016)，其整個偵測過程分為 3 個階段：(1)將影像縮放到 448*448，(2)通過神經網格進行檢測和分類，以及(3)NMS 抑

制(Non-max Suppression)流程，輸出最終結果時對物體的定位和分類可以一次接續完成。首先，當輸入影像通過神經網路後，YOLO 就可以直接輸出對應的邊界框(bounding box, bbox)的位置以及其最可能類別的可信度(confidence)，不僅運算流程非常直覺，同時物件識別速度比 two-stage 之 CNN 架構快很多。YOLO 處理物件偵測時，將邊框的預測與物件的類別視為迴歸問題，使用一個 CNN 架構來處理影像，經由多層的影像分析，不僅判斷會越來越精確，也可以讓 YOLO 快速地對於整張圖片進行物件的偵測。在 YOLOv1 偵測物件流程中，首先是將圖片分割，在計算上將輸入的影像分成 $S \times S$ 個網格，每個網格細胞(grid cell)會預測 B 個 bbox，每個 bbox 會有 5 個預測值： (x,y,w,h,c) ，分別為邊界框的中心位置 (x,y) 、寬度 (w) 、高度 (h) 及邊界框的可信度 $(c = \text{Prob}(\text{Object}) \times \text{IoU}_{\text{pred}}^{\text{truth}})$ ，編碼後可得到 $S \times S \times (B \times 5 + C)$ 維張量(tensors)的輸出，如圖 3 所示。根據可信度值的不同，偵測到的物件會有不同粗細的邊框，最後將利用非極大值抑制(NMS)流程，透過利用 IoU (Intersection over Union，深度學習辨識後產生的 predicted bonding box 與真實的 ground-truth bounding box，計算兩者 bounding boxes 的交集面積與兩者聯集區面積的比值)比較預設定之一個閾值 (threshold)，來定義重複區域，其次找出比較的基準，在此重複區域中找出預測機率最高的 bounding box，接著透過捨棄其他 $\text{IoU} \geq 0.5$ 的 bounding boxes，如此可將多餘的候選邊界框濾除，完成最佳的邊界框與物件類別。在最後輸出結果時，將某一個邊框預測的可信度值 $(C = \text{Pr}(\text{Object}) \times \text{IoU}_{\text{pred}}^{\text{truth}})$ 和邊框所屬的網格預測的第 i 個類別機率 $(\text{Pr}(\text{Class}_i | \text{Object}))$ 相乘，即可得到每個 bounding box 所屬第 i 個類別可信度值(如方程式(1)所示)，最後進行影像物件偵測輸出(Neubeck & Van Gool, 2006)。YOLOv1 的網路架構示意圖，如圖 4 所示，在此網路架構中有 24 個卷積層(convolutional layers)以及 2 個全連接層(fully connected layers)。

$$c = \text{Prob}(\text{Class}_i | \text{Object}) \times \text{Prob}(\text{Object}) \times \text{IoU}_{\text{pred}}^{\text{truth}} = \text{Prob}(\text{Class}_i) \times \text{IoU}_{\text{pred}}^{\text{truth}} \quad (1)$$

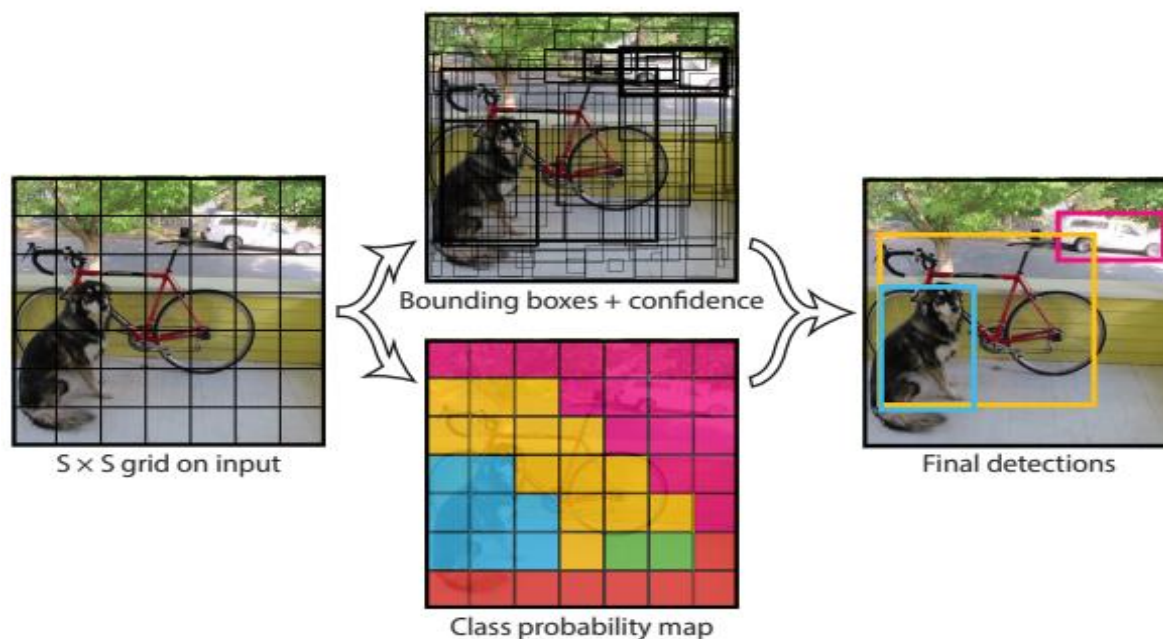


圖 3. YOLO 物件偵測運算方式示意圖(Redmon et al., 2016)

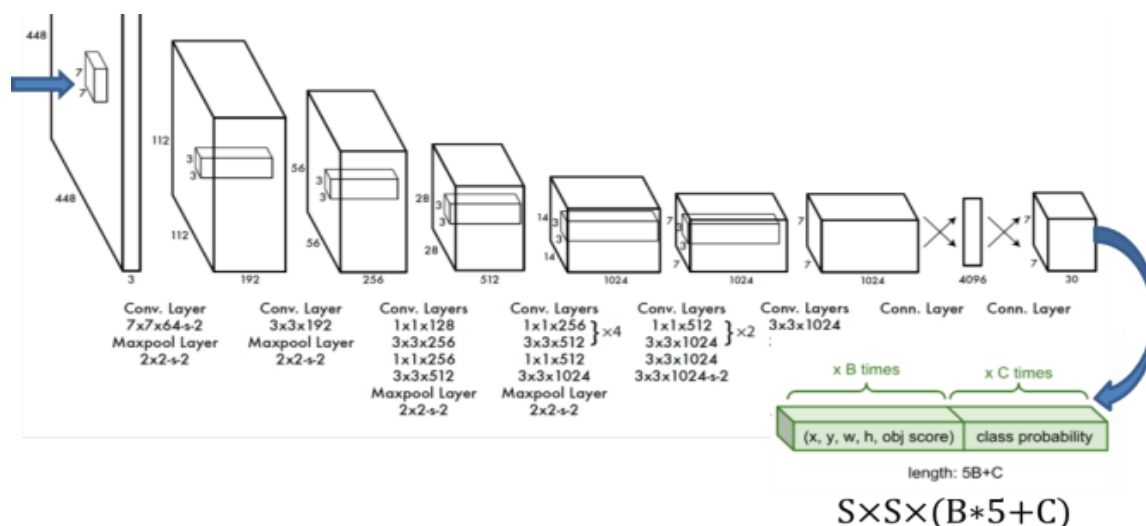


圖 4. YOLOv1 網路架構示意圖(修改自 Neubeck & Van Gool, 2006)

在前述影像物件偵測時先選出物件候選框並判斷選出之物件候選框是否為物件，當一個物件可能產生多個候選框情況下，進行 NMS 流程去消除多餘物件框找到最佳的框，如圖 5 所示。演算流程步驟為：(a)在偵測到的物件中候選邊框中找出最高可信度值的 bbox，並將此 bbox 歸入「確定是物件集合」內，(b)然後將此選擇的 bbox 與其他 bbox 各別計算 IoU 後，將計算出來 IoU 值大於設定閾值($\text{IoU} \geq 0.5$)之 bbox 的可信度度值設定為 0 (用以刪除多餘物件框產生)。依序重複步驟(a)與(b)直到所有的 bbox 都被比對完成，此時，在「確定是物件集合」的物件框就是最後結果，詳細之 NMS 的演算流程如下：

The processes of Non-Maximum Suppression (NMS):

input $B = \{b_1, b_2, \dots, b_n\}$, $C = \{c_1, c_2, \dots, c_n\}$, and NMS_t

B : the list of initial detection boxes; C : the list of corresponding detection confidences;

NMS_t : the specified threshold of Non-Maximum Suppression

begin

$D \leftarrow \{ \}$

while $B \neq \text{empty}$ **do**

$m \leftarrow \text{argmax } C$

$M \leftarrow b_m$

$D \leftarrow D \cup M$; $B \leftarrow B - M$

for b_i in B

if $\text{IoU}(M, b_i) \geq NMS_t$ **then**

$B \leftarrow B - b_i$; $C \leftarrow C - c_i$

end

end

end

return D, C

end

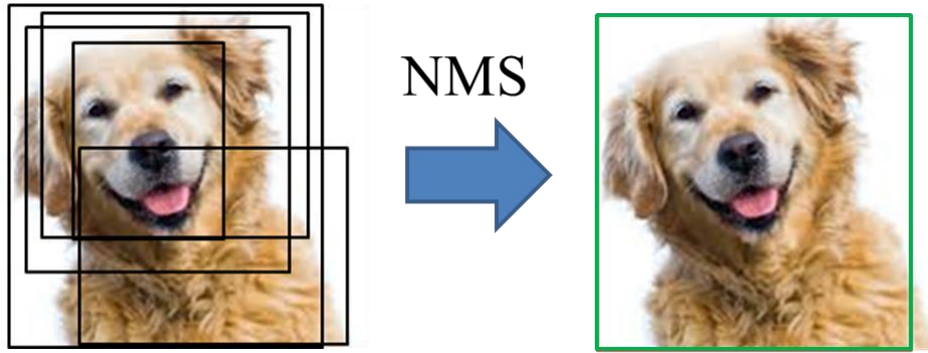


圖 5. NMS 流程演算後之最佳物件輸出結果

由於 YOLOv1 一個網格只預測兩個框，一個框只給出一個分類，如果一個格子出現多個物件，特別是小物件很多時，卻只能檢測出其中一個物件，這是 YOLOv1 方法的一個缺陷。另外對於小物件，小的 IoU 誤差也會對神經網路在優化過程中造成很大的誤差函數 (loss function) 影響，從而降低了物體檢測的定位準確性從而影響到檢測效果。進一步地，2017 年 Redmon 和 Farhadi 在 CVPR 上提出之 YOLOv2 建立在 v1 版本之高處理速度的基礎上，進行預測更準確(better)、速度更快(faster)與識別物件更多(stronger)三個面向進行了改進(Redmon & Farhadi, 2017)。其中擴充套件已可檢測達 9000 種不同物件，故命名為 YOLO9000。在 2018 年 4 月改良版的 YOLOv3 引進了更好的基礎分類網路：darknet-53 (仿自 ResNet，準確率與 ResNet-101 或 ResNet-152，但執行速度更快) (Redmon & Farhadi, 2018)。而分類器不再使用 Softmax，分類損失改採用二分類交叉損失熵(binary cross-entropy loss)，然而 YOLOv3 隨著 IoU 的增大時效能會下降，表示 YOLOv3 不能很好地契合 ground truth。直到第四代的 YOLOv4 (Bochkovskiy et al., 2020) 可謂是集大成者的偵測技術，三位作者中第一位為參與 YOLO github 來自俄羅斯的 Alexey Bochkovskiy，另外兩位則來自台灣中央研究院資訊所的廖弘源所長與王建堯博士。YOLOv4 在 AI Rewind 2020: A Year of Amazing Papers 榮獲 2020 年度最驚艷的論文之一。三位作者提出的 YOLOv4 框架有(1) Backbone：CSPDarknet53、(2) Neck：SPP，PANet 與(3) Head：YOLOv3，經由實證結果顯示為一種高效而強大的物件偵測模型，它使每個人都可以使用 1080 Ti 或 2080 Ti GPU 訓練超快速和準確的目標檢測器，YOLOv4 開源碼公開於 <https://github.com/AlexeyAB/darknet>。複雜的網路架構，如圖 6 所示 (Li et al., 2020)。

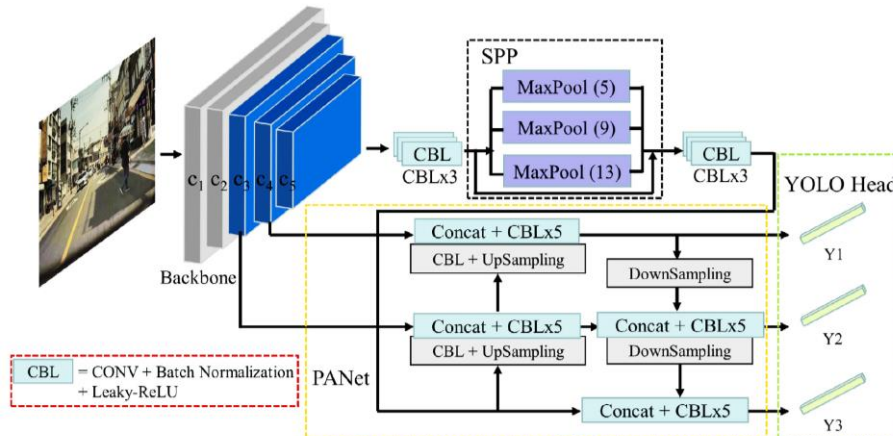


圖 6. YOLOv4 網路架構圖(Redmon et al., 2016)

由於在偵測的過程中會遇到梯度消失的問題，為了解決此問題，有人在之前曾經提出利用 ResNet、DenseNet 等卷積神經網路來解決，但是利用 ResNet 就會使得前面的網路層學不到東西，然而 DenseNet 會有太多重複的複製，讓不同層的參數學習多餘的資訊浪費計算量，所以後來改良出局部殘差網路 (partial residual networks, PRNet)，將資訊分流減少計算以提升運算速度，接著再延伸出跨階段局部網路 (cross stage partial network, CSPNet)，利用分割、分流、合併路徑，大幅減少計算，並增加學習多元的目標(郭雅欣、黃曉君，2020)，CSPNet 簡化結構部分圖，如圖 7 所示。YOLOv4 進一步將 ResNeXt50、Darknet53 分別與 Cross Stage Partial Network (CSPNet) 結合，組成了 CSPResNeXt50 和 CSPDarknet53，而 CSPDarknet53 即為 YOLOv4 的主幹 (Backbone)。

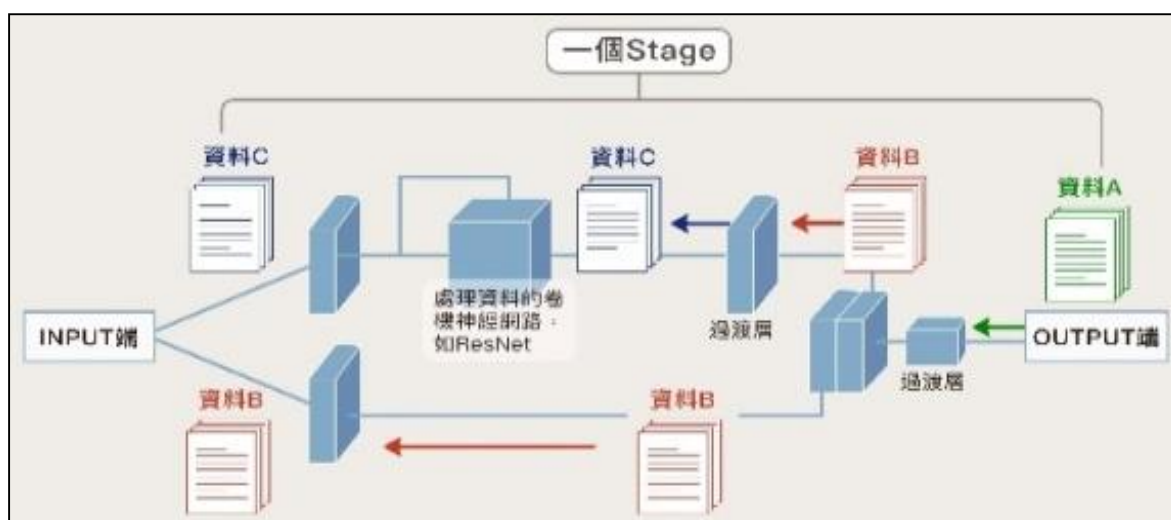


圖 7. CSPNet 簡化結構的部分圖(楊松儒，2019)

二、車牌辨識(以 OpenCV 影像處理+OCR/CNN 字元辨識)

光學字元辨識 (Optical Character Recognition, OCR)最早的概念是在 1929 年由 Tausheck 提出，它是對於圖像檔案資料的文字進行分析辨識處理，取得版面的文字及資訊的過程。而 OCR 進行分析時有五個步驟分別為：輸入、前期處理、中期處理、後期處理及輸出。輸入是對於不同的圖像格式，會有不同的壓縮方式與不同的儲存格式。在前期處理會利用二值化簡單將圖片的內容分為前景與背景，為了更快的計算，會先將彩色的圖進行處理，讓圖片留下前景與背景各個資訊，所以二值化的過程也稱為黑白化，也會用到圖像降噪，是針對各圖像的噪點進行去噪的過程，最後會用到傾斜校正，由於在拍照過程中難免會有物件傾斜的問題，所以需要利用校正技術使圖片中的物件轉正，且符合完全的水平及垂直的對齊(徐莘杰等，2011)。在中期的處理會先運用版面分析，將圖片分段落，根據圖像的多樣性或複雜度，尋求最佳化，再來需要進行字元的切割，因為若直接進行 OCR 分析將會極大限制 OCR 的效能，還有字元的辨識，會提取圖片中物件的特徵，若文字的位移、斷筆、旋轉等影響，也會對特徵的提取增加困難度，最後要進行版面的還原，在圖像辨識之後的文字需要輸出為原始文件，進行排列、段落、位置的編排。在後期的處理，會根據特定語言的上下文進行結果的校正。最後的輸出是將辨識出的字元以特定的格式進行文字輸出(維基百科，2018)。

由於車牌畢竟是一張影像圖片，不是純書寫文字影像，OCR 辨識過程中極容易產

生誤判結果，故本研究進一步亦使用深度學習網路 CNN，透過訓練多種書寫字體的 36 個字元(1234567890 及 A-Z)進而產生訓練模型作為車牌字元分割後的辨識技術(CNN 字元辨識)。另外，由於使用 OCR/CNN 字元辨識時，若是車牌傾斜、反轉等車牌位置擁有不確定的因素，若不進行校正之動作，將會影響之後的字元分割處理以及字元辨識之問題。所以需要使用 OpenCV 進行車牌校正及影像處理，使得辨識準確率更高。車牌圖像的校正有縮放、翻轉、透視變換等操作。縮放主要的目的是讓圖像變成需要的大小。翻轉的主要目的，是把圖像轉正為需要的位置，分為水平翻轉、垂直翻轉、水平與垂直同時翻轉。透視變換的主要目的是把圖像投影到另一個新的視平面，做所謂的透視變換(perspective transformation)(如圖 8(a))。透視變換是將圖片投影到一個新的視平面(viewing plane)，也稱作投影映射(projective mapping) 它是二維(x,y)轉換到三維(u,v,w)、再到另一個二維(x',y')空間的映射。運算方式是透過原始影像四邊形的四個頂點座標(x,y)和變換後影像四邊形的四個頂點座標(x',y')(自行設定影像長與寬尺寸)，如圖 8(b)，即可求得變換矩陣。座標轉換方程式如下：

$$[u, v, w] = [x, y, 1] \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (2)$$

影像轉換後的座標(x',y')=(u/w,v/w)。其中轉換矩陣 $\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ 中 $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ 表示線性轉換，如縮放(scaling)、剪切(shearing)和旋轉(rotation)， $[a_{31} \ a_{32}]$ 表示平移， $[a_{13} \ a_{23}]^T$ 產生透視變換。一般在經過透視變換之後的圖片通常不是平行四邊形。

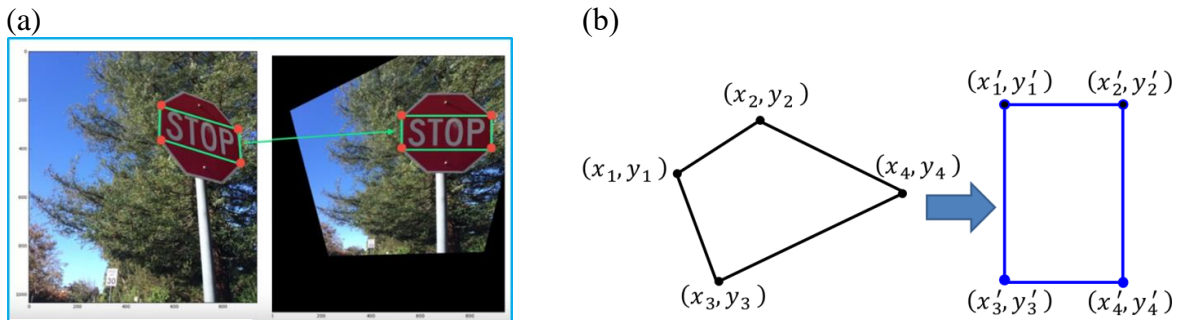


圖 8. (a)影像透視轉換示意圖及(b)圖像位置轉換前後座標

本研究以影像處理 OpenCV 執行方式如下：

```
# get transform matrix
```

```
M = cv2.getPerspectiveTransform(X, X') # X:影像轉換前之座標; X':影像轉換後之座標
```

```
# apply transformation for an image (img) to img with size of (trans_width, trans_height)
```

```
Img = cv2.warpPerspective(img, M, (trans_width, trans_height)) #輸出轉換後的影像。
```

另外，當車牌位置轉正後就要進行車牌辨識，但由於背景環境問題等，容易造成車牌模糊等因素難以辨識，所以需使用前期的二值化與圖像降噪，中期的字元切割、字元辨識與版面還原，直到最後的輸出。原始圖像，須經過二值化的處理，是利用設

定一個閾值，分為固定閾值及自適應閾值(維基百科，2019)，固定閾值是將閾值設為一定值，適用於環境擁有穩定的光源，而自適應閾值可以隨著當下光源變化而有所改變(林永鑫、林政宏、2018)。主要是將圖像變成只有黑與白，如此一來可以將不容易發現的資訊有效的顯示出來。而在圖像中的雜質也會顯示出來，所以需經過圖像降噪方式把雜質濾出，主要方式是利用濾波器達成保留紋理特徵與突顯邊緣。再來需要進行字元的切割，將車牌分割成獨立的字元，因為車牌的字體可能因為其他因素造成字元沾黏、斷筆等，通常是使用水平或垂直投影來達成字元分割。分割後，進行 OCR/CNN 字元辨識。在一系列的辨識途中可能會造成與原車牌字體排列的不同，所以需要進行版面的恢復，達到與原車牌相同的排列，最後進行輸出(有解無憂，2021)。

參、研究流程

本計畫擬先建立以 Python Tkinter 介面設計的 GUI 整合系統，在此 GUI 整合系統中選擇我們所輸入的車輛影像，利用 YOLO 進行車牌的邊緣偵測，擷取出車牌輪廓與定位後，再透過 OpenCV 的影像處理判斷車牌是否有傾斜的狀況，若(a)有傾斜的情形時則須進行車牌校正，利用縮放、翻轉及透視變換使圖片的車牌方正整齊，之後進入車牌灰階、二值化、濾波降噪、線條膨脹/侵蝕、自適應直方圖均衡化及車牌牌面淨化等車牌影像處理程序，最後輸出辨識完成的字元；若(b)車牌沒有傾斜的狀況時，則無須執行車牌轉正程序，直進入車牌影像處理及 OCR 字元辨識程序。本研究為將擷取之車框去除車牌字元以外之不必要車牌鑽孔鎖螺絲孔汙點、縣市車籍文字影像以及將車框影像能縮減尺寸到車牌字元邊緣，以增加車牌號碼之正確辨識率，故額外進行車牌牌面淨化的工作，如圖 9 所示。經由此程序可以得到一個較為“乾淨的”車牌牌面影像後，再進入車牌影像 OCR/CNN 字元辨識程序。車牌辨識系統之 YOLO 車牌偵測、透視轉換車牌轉正及 OCR 車牌字元辨識三大階段，示意圖如圖 10(a)。詳細之 OCR 車牌辨識運算流程，如圖 10(b)所示。



圖 9.車牌牌面淨化圖

(a)

(b)

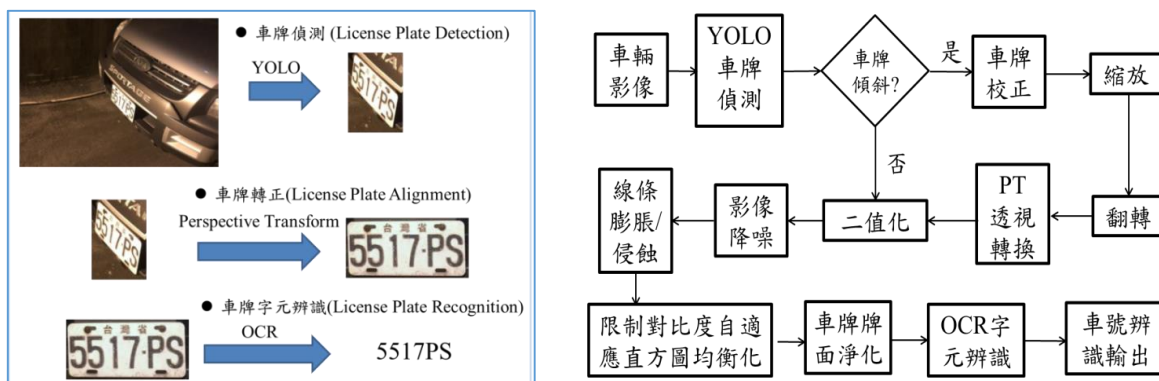


圖 10.車牌辨識系統之(a)車牌偵測、轉正與字元辨識階段及(b)OCR 運算流程圖

本研究採用之 CNN 之 API 串接模組如下：

```
def get_model(num_classes):
    K.clear_session()
    model = Sequential()
    model.add(Conv2D(32, (3, 3), input_shape=(28, 28, 3), activation="relu"))
    model.add(BatchNormalization())
    model.add(Conv2D(32, (3, 3), activation="relu"))
    model.add(BatchNormalization())
    model.add(Conv2D(32, (5, 5), activation="relu", strides=2, padding="same"))
    model.add(BatchNormalization())
    model.add(Dropout(0.4))
    model.add(Conv2D(64, (3, 3), activation="relu"))
    model.add(BatchNormalization())
    model.add(Conv2D(64, (3, 3), activation="relu"))
    model.add(BatchNormalization())
    model.add(Conv2D(64, (5, 5), strides=2, padding="same", activation="relu"))
    model.add(BatchNormalization())
    model.add(Dropout(0.4))
    model.add(Flatten())
    model.add(Dense(128, activation="relu"))
    model.add(BatchNormalization())
    model.add(Dropout(0.4))
    model.add(Dense(num_classes, activation="softmax"))
    model.compile(
        loss="sparse_categorical_crossentropy",
        optimizer=optimizers.Adam(lr=0.0001),
        metrics=["accuracy"],
    )
    return model
```

當建立完成上述 CNN 模組後，即可進行 YOLO 結合 CNN 之車牌辨識系統之

YOLO 車牌偵測、透視轉換車牌轉正及字元分割後結合 CNN 車牌字元辨識三大階段，示意圖如圖 11(a)。詳細之 CNN 車牌辨識運算流程，如圖 11(b)所示。

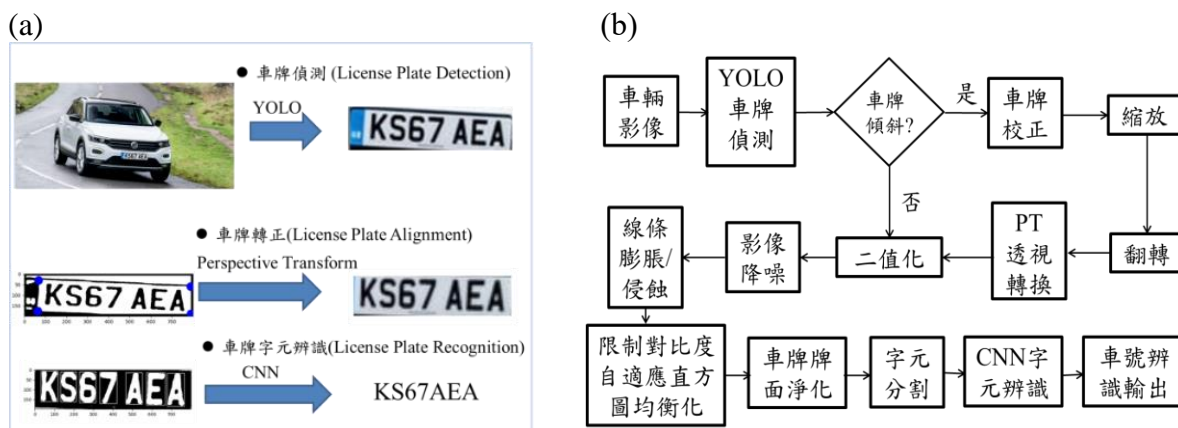


圖 11.車牌辨識系統之(a)車牌偵測、轉正與字元辨識及(b)CNN 車牌辨識運算流程圖

本文除了使用光學文字辨識 (OCR)去辨識文字車牌之外，亦使用卷積神經網路 (CNN)作為車牌辨識的工作，CNN 訓練模組如下：

```

model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(height, width, channel)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(36, activation='softmax')) #model.add(layers.Dense(35, activation='softmax'))
model.summary()
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
    
```

肆、研究結果與討論

一、YOLO 偵測靜態照片中的物件

電腦視覺領域的 YOLO 設計有一個深度學習框架 darknet 的環境，透過訓練過程可以建立各類物件的偵測模型。如圖 12(a)為透過 YOLO 物件偵測模型可以正確偵測到照片中有兩個人(people)與風箏(kite)的物件，在被偵測到的物件上方可標示出此物件歸類的百分比，由於影像清晰，故偵測到的三個物件可信度分別為 94%、99%與 100%。當應用偵測都會城市街景時即可偵測到相當多的物件存在，如圖 12(b)中可偵測出三個人(people)騎腳踏車(bicycle)，一位行人(people)在走路，綠色框的一台公共汽車(bus)及影像模糊的 8 台汽車(car)等。顯示 YOLOv4 可正確偵測到物件及正確的物件名稱。

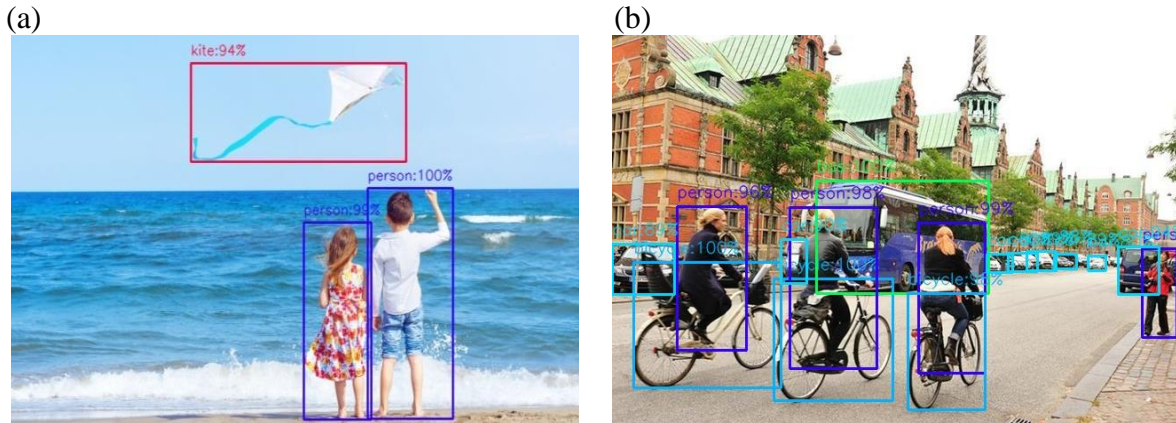


圖 12.靜態照片中被偵測到的物件結果

二、YOLO 偵測動態影片中的物件

為能應用到實際現實生活中動態物件的偵測，YOLO 物件偵測可以將動態影片 (video) 中的每幀 (frame) 影像進行偵測後，再以 OpenCV 影像處理儲存成動態 avi 影片檔。由 avi 影片檔中擷取出來之圖 13(a)-(b) 畫面中顯示影片內可偵測出一位女生肩背著手提袋 (handbag)、一位女生肩上有黃色框手提袋 (handbag)、一位女生背部綠色框背包 (backpack)、其他多位藍色框行人 (people) 在走路、一台汽車 (car) 與三輛腳踏車 (bicycle)。由上述 YOLOv4 物件偵測結果顯示，YOLO 偵測技術當可更進一步應用於動態的物件偵測上，故本研究將進一步將此優異的 YOLOv4 物件偵測技術應用於尺寸大小不同之汽機車車牌偵測上，並將採用光學文字辨識 OCR 進行所偵測出來之車牌做車牌字元辨識的工作。



圖 13.動態影片中被偵測到的物件結果

三、YOLO 偵測靜態汽機車車牌

一般而言，汽機車車牌在一張照片影像中所佔的比率很低，車牌定位難度較高，雖然傳統影像處理方式可訂出車牌邊界達成車牌定位，但透過 YOLO 物件偵測技術所建立的汽機車車牌訓練模型，可達成精準且高效率的車牌定位目標。圖 14(a)-(b) 顯示 YOLO 可正確偵測出歪斜之靜態汽車與機車的車牌輪廓 (contour) 與位置 (position)。

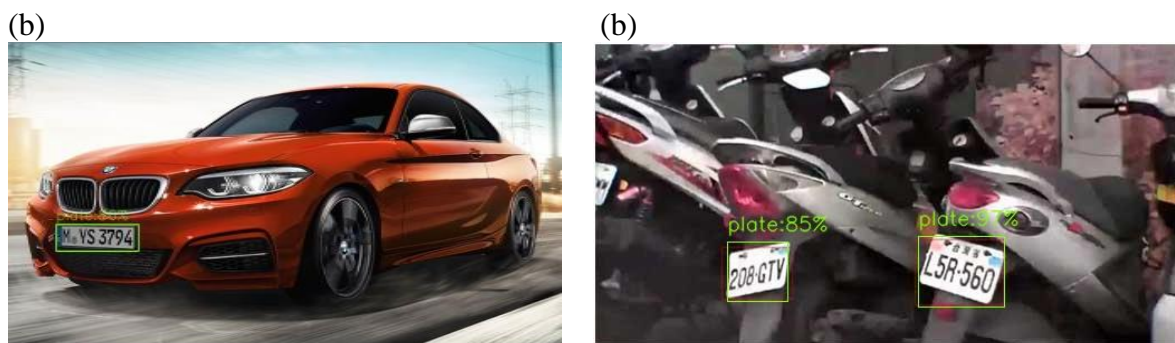


圖 14.靜態停車之汽機車被偵測到的車牌顯示

四、YOLO 偵測動態汽機車車牌

進一步地，本研究應用 YOLO 偵測技術於動態行車之車牌偵測上。使用之參考影片來源，可參考：<https://www.youtube.com/watch?v=a79yoYezgFs>。分析結果呈現於圖 15(a)-(b)，顯示 YOLO 可同時偵測到大部分汽機車之近方大車牌與遠方小車牌、清晰車牌與動態偵測到之模糊車牌、同向與對向車道之車輛車牌，同時也能偵測出正向與歪斜之汽車與機車的車牌輪廓與位置。

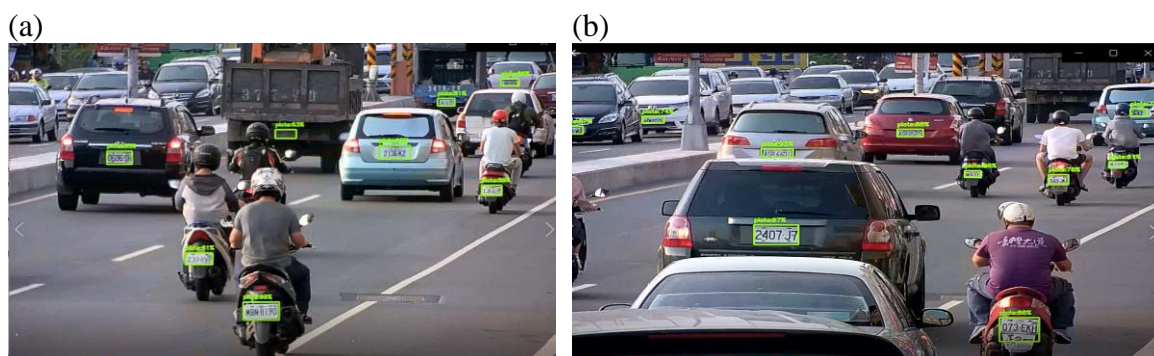


圖 15.動態行車之汽機車被偵測到的車牌顯示

五、YOLO 偵測結合 OCR 之靜態車輛車牌辨識

經由上述結果顯示 YOLO 之偵測能力也相當適合於動態車輛移動時的車牌偵測上，因此，以 YOLO 結合 OCR/CNN 字元辨識技術來做為靜態與動態車輛之車牌辨識工作將是非常適合的。YOLO 物件偵測技術可以快速且有效地偵測出靜態停放與動態行駛之汽機車車牌輪廓與定位。進一步地，本文將利用 YOLO 偵測出之車牌透過車牌影像灰階化、透視轉換校正車牌傾斜、二值化、高斯模糊(Gaussian blur)與中位模糊(median blur)做邊緣模糊處理等影像處理後，進行 OCR 車牌文字辨識，在本研究中所進行 OCR 之車牌文字辨識：`text = pytesseract.image_to_string(roi, config='-c tessedit_char_whitelist=0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ --psm 8 --oem 3')`。圖 16(a)-(b)分別為各一台汽車與機車於靜態停放時(影像清晰)之車牌辨識結果。兩張影像之車牌均能正確辨識無誤。若當車牌相當模糊、車牌輪廓過小或車牌具有高度傾斜時，OCR 的文字辨識準確度就顯得不足，適合的方式為使用深度學習網路(deep learning network)架構以車牌訓練模型來進行車牌文字辨識預期會比較有高的辨識準確度，這將是下一節中本文使用之 CNN 字元辨識的重要成果。



圖 16.靜態停放之汽機車被偵測到的車牌辨識結果

六、YOLO 偵測結合 CNN 之動態車輛車牌辨識

在動態汽機車行駛之車牌辨識結果，如圖 17(a)-(b)所示，雖然這兩張車牌顯示因車輛於動態行駛時所拍攝影像較為模糊，為能有效地辨識正確車牌字元，本研究亦進一步使用了較為複雜之 CNN 字元辨識技術於動態車牌之車牌辨識工作上。由圖 17(a)-(b)兩張動態汽機車影像圖顯示，本研究使用 YOLO 物件偵測技術結合 CNN 字元辨識可將汽機車上的車牌文字正確地完成辨識出來。圖 17(a)中 CNN 字元辨識結果中可以正確辨識出汽車車牌(2136KZ)。進一步地，在動態行駛之多輛汽、機車及車牌影像模糊的辨識工作上，由圖 17(b)顯示 CNN 字元辨識結果中不僅可以辨識出較為清處的廂型車車牌(4437LE)外，亦可以同時正確辨識出較小框的機車車牌(NRJ7167)以及遠方模糊的汽車車牌(9372GA)。由上述之結果顯示，本研究發展出來之字元分割後的影像應用 CNN 字元辨識模組具有相當準確的動態車牌字元辨識能力。

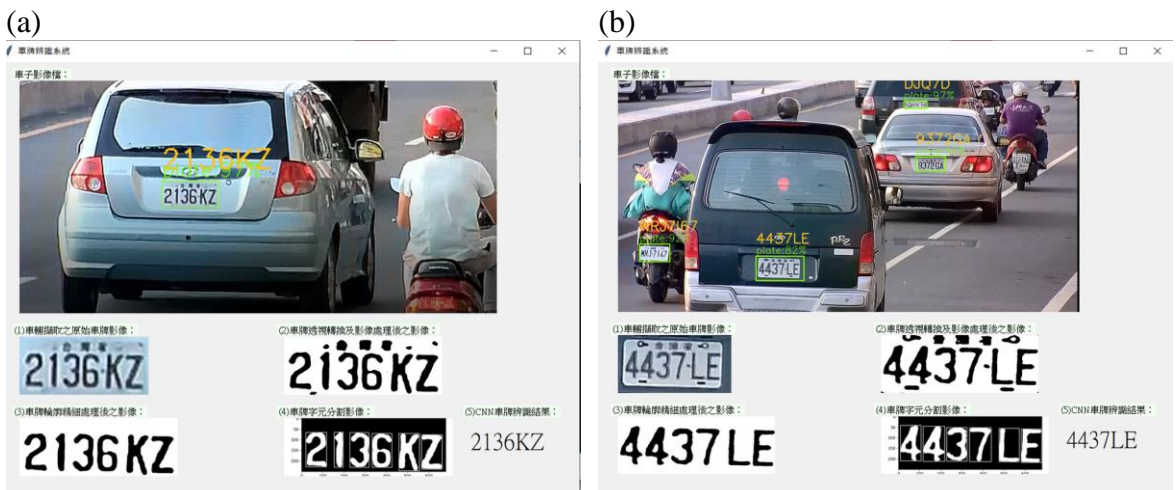


圖 17.動態行車之汽機車被偵測到的車牌辨識結果

伍、結論

物件偵測在實務應用上相當具有價值，好的偵測技術有助於提升偵測出物件類別的正確性，可以廣泛應用於農產品分類、醫療影像疾病分類、入侵系統或車牌辨識等系統。本文主要著重於將此物件偵測技術應用於車牌辨識系統上，車牌辨識系統的應用有助於執法人員在巡邏時隨時對於路邊四周的車輛進行偵測贓車的動作並進行立即的違規取締；利用科技執法，藉由路口監視器、路燈、及行車紀錄器，取締闖紅燈、逆向行駛、違停等交通違規或作為重大事故的判決；也可以藉由在出入口的設置鏡頭，進行停車場出入的管控，針對進出的車輛做流量管制，節省去管理人員多餘的重複動作，增加工作效率；還有也能針對停車場或是高速公路上的車輛進行自動扣款繳費的收費系統。本研究使用優異的 YOLOv4 偵測技術應用於靜態停放與動態行駛之台灣汽機車牌定位。首先以 YOLOv4 進行動態照片與動態影片的物件偵測，由結果顯示 YOLOv4 具有優異的動、靜態影像內物件的偵測效能。進一步地，本研究將此 YOLOv4 應用於台灣汽機車之車牌輪廓與定位以提升偵測車牌的可靠度。當車牌定位後，本研究使用了 OpenCV 進行判別車牌是否有傾斜的情況，進行縮放、翻轉及透視變換使車牌圖像的水平與垂直對齊方正，以利於後續 OCR/CNN 提高判斷車牌號碼字元的準確率。由靜態停放與動態行駛之台灣汽機車牌辨識結果顯示，本文結合 YOLOv4 偵測車牌與 OCR 車牌字元辨識應用於靜態車輛之車牌辨識具有良好的效果。進一步地，在本研究上亦發展了 YOLOv4 車牌偵測結合 CNN 深度學習模型進行車牌字元分割及字元辨識，由本研究的分析結果顯示，本研究發展的技術大幅度地提高了這種“YOLOv4 車牌偵測結合 CNN 車牌字元辨識”應用於動態行駛之多輛汽機車牌辨識的準確性。

參考文獻

- catOneTwo (2020)，一文讀懂 YOLOv1,v2,v3,v4 發展史，知乎，取自
<https://zhuanlan.zhihu.com/p/139128562>
- Sigrid, C. (2021)，五分鐘認識用 YOLO 做血液抹片物件偵測，MDBros，取自
<https://medium.com/mdbros/五分鐘認識用-yolo-做血液抹片物件偵測-object-detection-be3c77e13228>
- 有解無憂(2021)，OpenCV+Python 影像處理入門教程（四）幾何變換，取自
<https://www.uj5u.com/qita/270791.html>
- 周弘裕(2019)，AI 車牌辨識與傳統車牌辨識的差異，取自
<https://www.youtube.com/watch?v=nLxAg452n0g&t=22s>
- 林永鑫、林政宏 (2018)，植基於卷積神經網路之高效能車牌辨識系統，國立臺灣師範大學電機工程學系碩士論文。
- 林宏軒(2019)，DNN 車牌辨識於智慧城市的應用，工業技術研究院資訊與通訊研究所。
- 芮嘉璋(2021)，深度學習神經網路之運作，*北美智權報*，285。
- 洪文斌(2021)，人工智慧應用於停車場管理系統，*中台山月刊*，257，52-55。
- 徐莘杰、郭仲源、邱俊傑、楊明學 (2011)，車牌辨識系統，中華大學資訊工程學系專題期末報告。
- 郭弘裕(2017)，基於 RGB-D 之影像分割方法，義守大學資訊工程研究所碩士論文。

- 郭雅欣、黃曉君(2020)，一眼揪出你有沒有超速！世界第一物件偵測技術：YOLOv4，研之有物，中央研究院，科技新報報導，取自 <https://technews.tw/2020/11/09/yolov4-hong-yuan-mark-liao/>
- 陳信希、郭大維、李傑、高虹安、吳信輝、王建堯、王家慶、李宏毅、張智星、曾新穆、蔡炎龍、鄭文皇、蘇上育(2020)，*人工智慧導論理*（初版，31-32 頁），全華圖書股份有限公司。
- 馮玄明、陳華慶、翁慶昌、賴麗雲(2017，11 月)，深度學習神經網路人體影像辨識自動模型設計，中華民國第二十五屆模糊理論及其應用研討會，屏東縣：國立虎尾科技大學。
- 黃威銘(2020)，基於資料擴增與深度學習的改善磁控膠囊內視鏡辨識腸道之研究，國立台灣大學電機資訊學院電機工程學系碩士論文。
- 楊松儒(2019)。以深度學習為基礎之路面破損與閘栓檢測系統，國立台灣師範大學電機工程學系碩士論文。
- 維基百科(2018)，光學字元辨識。取自：<https://zh.wikipedia.org/wiki/光學字元辨識>。
- 維基百科(2019)，二值化，取自 <https://zh.wikipedia.org/wiki/二值化>。
- 維基百科(2022)，循環神經網路，取自 <https://zh.wikipedia.org/wiki/循環神經網路>。
- 維基百科(2023a)，人工神經網路，取自 <https://zh.wikipedia.org/wiki/人工神經網路>。
- 維基百科(2023b)，卷積神經網路，取自 <https://zh.wikipedia.org/wiki/卷積神經網路>。
- 蘇浩平(2017)，開放環境下之車牌偵測，國立台灣科技大學電機工程系碩士論文。
- Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. (2020), "YOLOv4: Optimal Speed and Accuracy of Object Detection," retrieved from <https://arxiv.org/pdf/2004.10934.pdf>
- Cortes, C., & Vapnik, V. (1995), Support-Vector Networks. *Machine Learning*, 20, 273-297.
- Girshick, R. (2015, December), "Fast R-CNN," *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, 1440-1448.
- Li, Y., Wang, H., Dang, L. M., Nguyen, T. N., Han, D., Lee, A., Jang, I., & Moon, H. (2020), "A Deep Learning-Based Hybrid Framework for Object Detection and Recognition in Autonomous Driving," *IEEE Access*, 194228 - 194239. doi: 10.1109/ACCESS.2020.3033289
- Neubeck A., & Van Gool, L. (2006), Efficient Non-Maximum Suppression, 18th International Conference on Pattern Recognition (ICPR'06), IEEE。
- Redmon, J, Divvala, S., Girshick, R., & Farhadi A. (2016), "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA.
- Redmon, J., & Farhadi, A. (2017), "YOLO9000: Better, Faster, Stronger," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, 6517-6525.
- Redmon, J., & Farhadi A. (2018), "Yolov3: An Incremental Improvement," arXiv 2018, arXiv:1804.02767, 2018.