

國立臺灣師範大學電機工程學系

碩士論文

指導教授：林政宏博士

植基於卷積神經網路之高效能車牌辨識系統

An Efficient License Plate Recognition System Using  
Convolution Neural Network



研究生：林永鑫 撰

中華民國一〇七年八月

# 植基於卷積神經網路之高效能車牌辨識系統

學生：林永鑫

指導教授：林政宏博士

國立臺灣師範大學電機工程學系碩士班

## 摘 要

近年來，車牌辨識系統已成為智能城市車輛管理、被盜車輛調查、交通監控等發展中的關鍵角色，車牌辨識系統有三個階段，包括車牌偵測、字元分割，與字元辨識。儘管車牌辨識系統已成功的應用於環境單純的智能停車場，但使用於監控系統中仍會面臨許多問題，例如多車道辨識，大量的交通號誌與廣告招牌，惡劣天氣與夜間拍攝的模糊傾斜圖像。本論文提出了一種高效的車牌辨識系統，首先偵測車輛，再從車輛中偵測車牌，以減少車牌偵測的誤報。再使用卷積神經網路來改善模糊圖像與近似字元的辨識效果，實驗結果顯示，與傳統的車牌辨識系統相比，該系統擁有較高的精確度。

關鍵字：車牌辨識系統、卷積神經網路、智慧都市

# An Efficient License Plate Recognition System Using Convolution Neural Network

Student : Yong-Sin Lin

Advisor : Dr. Cheng-Hung Lin

Department of Electrical Engineering  
National Taiwan Normal University

## ABSTRACT

In recent years, license plate recognition system has become a crucial role in the development of smart cities for vehicle management, investigation of stolen vehicles, and traffic monitoring and control. License plate recognition system has three stages, including license plate localization, character segmentation, and character recognition. Up to now the license plate recognition system has been successfully applied to the environment-controlled smart parking system, however it still raises many challenges in the surveillance system such as congested traffic with multiple plates, ambiguous signs and advertisements, tilting plates, as well as obscure images that are captured during bad weather and poor light conditions. In this thesis, we propose an efficient license plate recognition system that first detects vehicles and then retrieves license plates from the detected vehicles to reduce false positives on plate detection. Thereafter, the technique of convolution neural networks is applied to improve the character recognition accuracy from the blurred and obscure images. The experimental results show the superiority of the performance in the proposed method as compared to the traditional license plate recognition systems.

Keywords: License plate recognition system, convolution neural network, smart city.



## 誌 謝

回顧這兩年的碩士學習中，得到了許多老師、同學和朋友們的熱心幫助與支持。在此，我要向他們表示我誠摯的謝意！

首先，謝謝我的指導教授林政宏老師這些年來的悉心指導，從大三的專題到碩士的論文，老師不僅提供了良好的實驗環境，同時在研究上給了許多建議與指導，顧及到學生的興趣專長，幫忙尋找適合的研究題目，也謝謝老師這些年來幫我們爭取了許多參加研討會與產學合作的機會，讓我們可以接觸不同領域的知識，了解業界與學界的差異與生產產品的流程，擁有這些寶貴的經驗。

其次，非常謝謝實驗室成員們的幫忙，不論是廖重洧學長在程式撰寫、各種神奇技術上的教學，或是劉暉辰學長在上班之餘，抽空分享許多新的知識、技術，並且在我的論文中給予許多不同的建議與看法，還有實驗室最強的謝政宏大師從大三專題、產學合作到學業課程的 carry，以及實驗室學弟妹亦信、易穎、振豪、李穎、冠毅幫忙分擔實驗室的大小事務。

最後，感謝家人多年來的照顧與精神上的支持，讓我有機會讀書甚至碩士畢業。謝謝大家！

林永鑫 2018.08.08



# 目 錄

|                     |      |
|---------------------|------|
| 中文摘要 .....          | i    |
| 英文摘要 .....          | ii   |
| 誌 謝 .....           | iv   |
| 目 錄 .....           | v    |
| 圖 目 錄 .....         | viii |
| 表 目 錄 .....         | x    |
| 第一章 緒論 .....        | 1    |
| 1.1 研究背景與動機 .....   | 1    |
| 1.2 研究目的 .....      | 3    |
| 1.3 研究方法概述 .....    | 4    |
| 1.4 研究貢獻 .....      | 6    |
| 1.5 論文架構 .....      | 6    |
| 第二章 文獻探討 .....      | 9    |
| 2.1 車牌偵測 .....      | 9    |
| 2.1.1 依區域邊緣特徵 ..... | 9    |
| 2.1.2 依影像顏色特徵 ..... | 10   |
| 2.1.3 依字元排列特徵 ..... | 10   |
| 2.2 字元分割 .....      | 12   |
| 2.2.1 基於投影 .....    | 12   |
| 2.2.2 基於像素連接 .....  | 12   |
| 2.3 字元辨識 .....      | 13   |
| 2.3.1 字元像素值辨識 ..... | 13   |
| 2.3.2 字元特徵辨識 .....  | 14   |
| 2.4 物件偵測 .....      | 14   |

|   |    |
|---|----|
| 2.4.1 Region-based CNN (R-CNN) .....              | 14 |
| 2.4.2 Fast Region-based CNN (Fast R-CNN) .....    | 15 |
| 2.4.3 Faster Region-based CNN (Faster R-NN) ..... | 16 |
| 2.4.4 You Only Look Once (YOLO) .....             | 17 |
| 2.4.5 YOLOv2 .....                                | 18 |
| 第三章 研究方法 .....                                    | 21 |
| 3.1 系統流程 .....                                    | 21 |
| 3.2 車輛偵測模組 .....                                  | 22 |
| 3.3 車牌偵測模組 .....                                  | 22 |
| 3.3.1 訓練階段.....                                   | 22 |
| 3.3.2 偵測階段.....                                   | 23 |
| 3.4 字元分割 .....                                    | 24 |
| 3.4.1 二值化.....                                    | 25 |
| 3.4.2 投影裁切.....                                   | 26 |
| 3.5 字元辨識 .....                                    | 27 |
| 第四章 實驗結果 .....                                    | 29 |
| 4.1 車輛偵測實驗 .....                                  | 29 |
| 4.1.1 訓練 YOLOv2 .....                             | 29 |
| 4.1.2 動態影像偵測 .....                                | 30 |
| 4.2 車牌偵測實驗 .....                                  | 31 |
| 4.2.1 訓練 SVM 分類器 .....                            | 32 |
| 4.2.2 基於 SVM 之方法偵測車牌.....                         | 32 |
| 4.2.3 使用 YOLOv2 偵測車輛再結合 SVM 偵測車牌.....             | 33 |
| 4.3 字元分割實驗 .....                                  | 34 |
| 4.3.1 二值化.....                                    | 34 |
| 4.3.2 字元分割.....                                   | 35 |

|                   |    |
|-------------------|----|
| 4.4 字元辨識實驗 .....  | 35 |
| 4.4.1 訓練.....     | 36 |
| 4.4.2 測試.....     | 37 |
| 4.5 實驗結論 .....    | 38 |
| 第五章 結論與未來展望 ..... | 39 |
| 5.1 結論 .....      | 39 |
| 5.2 未來展望 .....    | 39 |
| 參考文獻 .....        | 40 |
| 自    傳 .....      | 45 |



## 圖目錄

|        |  |    |
|--------|--|----|
| 圖 1-1  | 車身廣告與大量字元資訊.....   | 2  |
| 圖 1-2  | 不同光線下之車牌.....  | 2  |
| 圖 1-3  | 車牌上的檢驗貼紙與車行廣告.....   | 2  |
| 圖 1-4  | 移動造成之車牌模糊.....   | 3  |
| 圖 1-5  | 車牌辨識系統流程圖.....   | 4  |
| 圖 1-6  | 從輸入影像偵測車輛位置.....   | 5  |
| 圖 1-7  | 從車輛區域偵測車牌位置.....   | 5  |
| 圖 2-1  | R-CNN 架構圖.....   | 15 |
| 圖 2-2  | FAST R-CNN 架構圖.....  | 16 |
| 圖 2-3  | FASTER R-CNN 架構圖.....  | 17 |
| 圖 2-4  | YOLO 的概述圖。輸入影像被分為 $S*S$ 個網格，每個網格會預測 $B$ 個邊界框、邊界框的信心值以及 $C$ 個物件類別的機率..... | 18 |
| 圖 3-1  | 車牌辨識系統架構圖.....   | 21 |
| 圖 3-2  | 正樣本範例.....   | 22 |
| 圖 3-3  | 負樣本範例.....   | 23 |
| 圖 3-4  | 車牌偵測訓練階段流程圖.....   | 23 |
| 圖 3-5  | 偵測階段流程圖.....   | 24 |
| 圖 3-6  | 字元分割模組流程圖.....   | 24 |
| 圖 3-7  | (左)原圖與(右)二值化後.....   | 25 |
| 圖 3-8  | 將(左)二值化後影像，進行水平投影結果(右).....  | 26 |
| 圖 3-9  | 依水平投影所裁出字元排列區域.....  | 26 |
| 圖 3-10 | 將排列字元進行垂直投影.....   | 27 |
| 圖 4-1  | COCO 2017 資料集之影像.....  | 29 |
| 圖 4-2  | 左圖為道路測拍，右圖為行車紀錄器之影像.....   | 31 |

|        |  |    |
|--------|--|----|
| 圖 4-3  | 車輛偵測後的畫面 .....                           | 31 |
| 圖 4-4  | 基於 SVM 之方法偵測車牌，產生大量誤報 .....              | 33 |
| 圖 4-5  | 使用 YOLOv2 偵測車輛再結合 SVM 偵測車牌，減少大量誤報.....   | 33 |
| 圖 4-6  | 車牌二值化 .....                              | 35 |
| 圖 4-7  | 水平投影切除上下多餘部分 .....                       | 35 |
| 圖 4-8  | 垂直投影分割獨立字元.....                          | 35 |
| 圖 4-9  | 分割後之字元.....                              | 35 |
| 圖 4-10 | 卷積神經網路架構圖.....                           | 36 |
| 圖 4-11 | 樣本有清晰(上面三張)、有模糊的影像(下面三張).....            | 36 |
| 圖 4-12 | 辨識成功之更小更模糊字元.....                        | 37 |
| 圖 4-13 | 辨識失敗之模糊近似字元，分別為 B 辨識成 8，U 辨識成 V 與 0..... | 37 |



## 表目錄

|   |    |
|---|----|
| 表 2-1 各類車牌偵測研究方法比較.....   | 12 |
| 表 2-2 各類字元分割研究方法比較.....   | 13 |
| 表 2-3 兩類字元辨識方法比較.....   | 14 |
| 表 4-1 動態影像攝影器材配置.....   | 30 |
| 表 4-2 YOLOv2 車輛偵測率 .....  | 31 |
| 表 4-3 混淆矩陣.....   | 32 |
| 表 4-4 基於 SVM 之方法偵測車牌的 RECALL RATE .....                           | 33 |
| 表 4-5 基於 SVM 之方法與 YOLOv2 結合 SVM 車牌偵測 PRECISION RATE 比較<br>表 ..... | 34 |
| 表 4-6 與其他研究方法比較.....  | 38 |
| 表 4-7 實驗總結.....   | 38 |

# 第一章 緒論

## 1.1 研究背景與動機

隨著智慧城市的發展，對於車輛密集度非常高的臺灣來說，車輛管理、被盜車輛調查和交通監控的工作量會日漸提升。過去警察為了追蹤車輛，必須調閱範圍內所有路口監視器，以肉眼過目可疑時段監視畫面、找尋特定車牌，相當的耗時與費力。所以隨科技進步，上述煩瑣辛苦之事，可透過車牌辨識系統來減輕工作量。

雖然車牌辨識系統已廣泛的應用於許多交通設施中，例如智慧停車場、交通收費系統等等。但將上述之系統架構應用於一般道路之路口監視器時，會因為複雜的環境而遇到許多挑戰。這些變因都會在偵測車牌時產生大量的誤報，使車牌辨識的準確率變低。以下介紹過去研究遭遇較多、較大的影響因子：

### 1. 複雜的道路環境：

臺灣的道路較為擁擠，並且充滿需多物件，有商店招牌、交通號誌、路名指標、車身廣告、攤販或行人、行道植栽，以及各種未知樣式與顏色的車輛。其中又以商店招牌與車身廣告最具影響，這些物件常會出現字元排列的標示(電話或編號、標語)，其特徵與車牌非常相似。



圖 1-1 車身廣告與大量字元資訊

光線與陰影的變換：

天色轉換、天氣變化皆會對在街道上所拍攝的影像產生差異。亦或是車輛大燈之類不穩定光源影響，對車牌影像的處理增加許多困難。

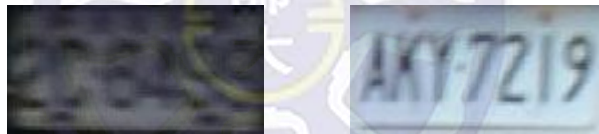


圖 1-2 不同光線下之車牌

2. 不同的車牌顏色、編碼與字體：

台灣車牌大多為白底，還有部分紅底、綠底、黃底；字則分為黑字、紅字、綠字等不同顏色；車牌的字元排列則有四至七碼，英文、數字不規則混雜；字體樣式因多次改版而有所不同，整體車牌樣式非常多樣。除此之外，更多狀況在於車牌上的車行廣告、髒汙、檢驗貼紙，以及車牌字體掉漆等等。



圖 1-3 車牌上的檢驗貼紙與車行廣告

### 3. 車輛移動造成車牌字元模糊：

車牌與字元容易因攝影設備好壞、車速不同、拍攝角度，造成不同程度模糊與傾斜情況。

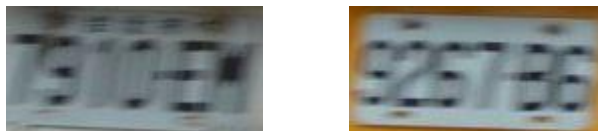


圖 1-4 移動造成之車牌模糊

除了上述車牌偵測所遭遇的困難，根據過去研究文獻，字元辨識也會因下述的問題而遭受不同影響：

1. 車牌髒汙與字元掉漆：髒汙與掉漆會使字元外型改變，增加字元的辨識困難度。
2. 車牌上多餘元件：例如貼紙、吊飾品、車牌裝飾燈，以及其他裝飾物等。
3. 字型差異：以臺灣為例，現行車牌因不同發行年份，版本不同，相同字元含有相異字型。
4. 近似字元：部分字元外型接近(如：8 與 B，2 與 Z，0 與 D)容易誤判。
5. 影像解析度：攝影設備不同，擁有不同的解析度，當解析度低時，字元呈現模糊外觀並可能夾雜更多的雜訊。

## 1.2 研究目的

目前世界上的車牌辨識系統中，大部分系統是建立在環境較單純的地方，並且一張影像中只有單一車輛，例如：停車場管理系統為單一車道出入口，攝影機是直接對著車牌的位置，並且車輛是在低速甚至靜止的狀況下進行辨識。若將這套系統運行於一般道路環境時，系統準確度與穩定性將受到影響，產生大量誤報。本研究目的即是提出一個能穩定、準確應用於道路環境的車牌辨識系統，克服上述的困境、擴增車道數目，達到多車道車牌辨識系統的目標。

### 1.3 研究方法概述

本論文提出的系統分為四個部分，包含車輛偵測、車牌偵測、字元分割，與字元辨識。

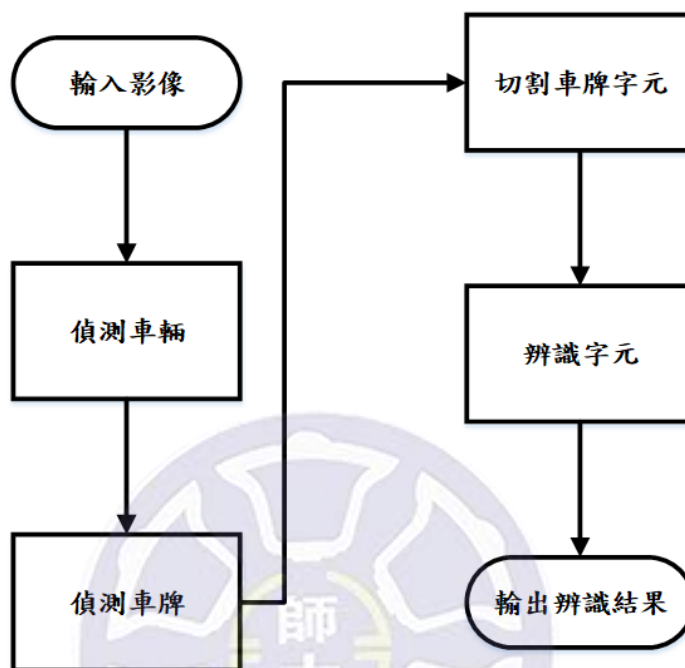


圖 1-5 車牌辨識系統流程圖

第一個車輛偵測的部分，使用 COCO 2017[1]的資料集訓練物件偵測架構 YOLOv2[2]來抓取車輛。車輛的種類分為汽車、機車、公車與卡車。這個階段會將偵測到的車輛座標輸出至下個階段。

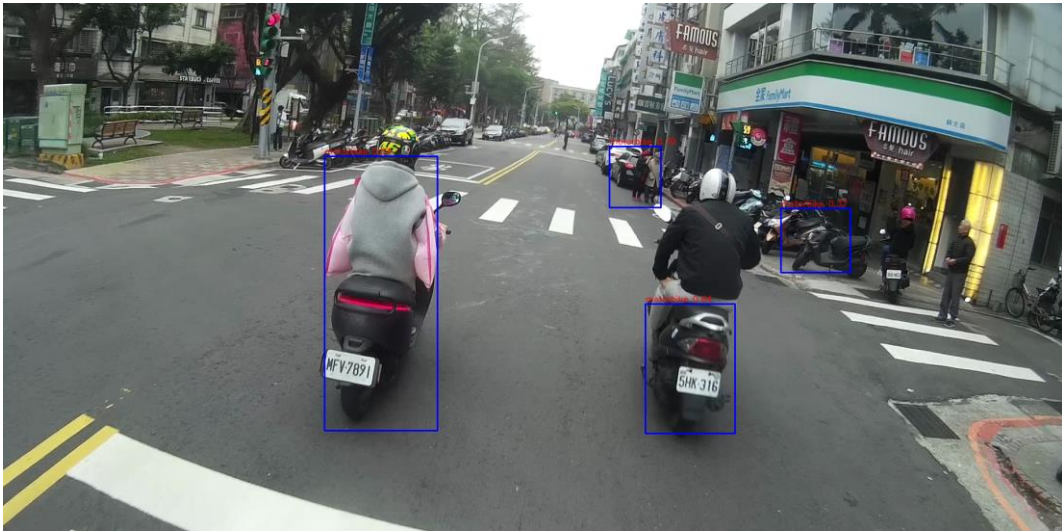


圖 1-6 從輸入影像偵測車輛位置

第二個部分車牌偵測，使用 SVM 分類器針對上階段偵測到的車輛進行偵測車牌，透過此架構可濾除許多誤報。SVM 分類器為監督式機器學習方法，訓練階段準備大量正樣本(車牌)與負樣本(非車牌)，以 HOG 方向梯度直方圖為特徵來訓練 SVM 分類器。並且車牌偵測的過程會透過縮放偵測窗(detect window)於輸入影像上滑動，以確保不同大小的車牌都會被偵測到。

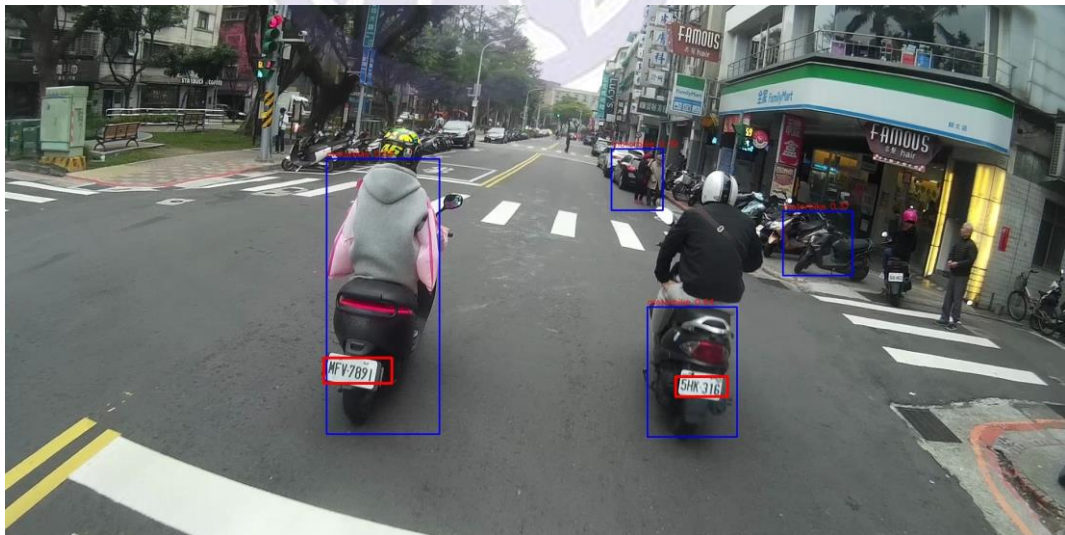


圖 1-7 從車輛區域偵測車牌位置

第三個部分，將 SVM 偵測到的車牌影像經過灰階轉換、二值化、水平與垂直投影的程序，完成字元分割。為適應道路環境多變的光影分布，以及不同髒汙

程度的車牌影像，使用了可自適應的二值化方法。

第四個部分字元辨識，本論文設計一個卷積神經網路之字元辨識架構。透過此卷積神經網路可準確的辨識車牌字元「A-Z」(不含 I、O)、「0-9」共 34 類，其中包括模糊影像與近似字元之辨識。

## 1.4 研究貢獻

本論文提出了一個高效能的車牌辨識架構，首先使用深度學習技術 YOLOv2 檢測車輛，整體車輛偵測率可達到 96.12%；再以 HOG 為特徵之 SVM 分類器從檢測到的車輛中檢索車牌，以減少車牌偵測的誤報，車牌偵測有 94.24% 的準確率。最後本論文設計之卷積神經網路(CNN)來改善模糊影像與近似字元的辨識效果，整體字元辨識率達 99.2%。透過四個階段的辨識，可以在保持召回率(Recall Rate)不變的同時，改善準確率(Precision Rate)，大幅降低誤報的狀況。與傳統的車牌辨識系統相比，此方法擁有較高的準確度與性能。

## 1.5 論文架構

本論文分為五個章節：緒論、文獻探討、研究方法、實驗結果以及結論與未來展望，以下條列各章節內容簡介。

**第一章 緒論：**概述目前車牌辨識系統的背景、研究目標以及研究方法簡介，闡述研究貢獻與論文架構。

**第二章 文獻探討：**針對與本系統、研究方法相關之文獻討論介紹。

**第三章 研究方法：**說明本系統車輛偵測、車牌偵測、字元分割、字元辨識所使用之研究方法。

**第四章 實驗結果：**介紹實驗配置與實驗過程，以及實驗結果分析。

**第五章 結論與未來展望：**最後對此研究提出的方法與實驗結果總結，並對未來

研究提出改善方向。





## 第二章 文獻探討

車牌辨識的研究發展有十年以上，過去前輩提出許多研究方法。本章節於 2.1、2.2、2.3，分別依據[3]整列過去主流架構，分為三個階段的車牌辨識系統，車牌偵測、字元分割和字元辨識的相關研究文獻； 2.4 介紹用來偵測車輛之物件偵測方法的相關文獻。

### 2.1 車牌偵測

車牌偵測，意即從輸入影像中判斷是否有車牌影像存在，找尋影像中所有車牌的位置，並輸出僅包含車牌圖像的部分。車牌偵測是系統關鍵的一環，穩定準確的偵測系統，才能進一步辨識車牌字元，也能確保系統正確率。

車牌可能以任意大小、任意位置分布於輸入影像中，所以偵測車牌的第一步，必須挑選特徵作為判斷車牌影像的依據，排除影像中多數非車牌之處。世界上各個國家發行的車牌有幾個共同特點：矩形、特定顏色牌面、數個由英文與數字混雜組成的排列字元及相對意義的字元顏色，而字元與車牌底色是以顏色差異較大的做為組合。藉著以上數個車牌的特色，可以選出作為判斷車牌區域的特徵。根據過去研究文獻與經驗，將車牌偵測方法大致分為三個主軸：(一)依區域邊緣特徵、(二)依影像顏色特徵、(三)依字元排列特徵。表 2-1 總結了各種方法的優缺點。

#### 2.1.1 依區域邊緣特徵

全球普遍都以矩形做為車牌形狀，因此這個特徵被應用於不少文獻。以邊緣偵測在圖片中尋找矩形區域的方法，被使用於[4]-[7]研究作為車牌候選區域。索貝爾運算子 (Sobel operator) 於[8]-[13]用以尋找圖片中因顏色轉換而形成的邊界 (edge)，並透過車牌邊緣兩兩垂直的特徵，找出符合矩形的區域。

霍夫轉換(Hough transform, HT)於[11][14]中被用以尋找直線、確立車牌位置，霍夫轉換的優點是可以找到略微傾斜的車牌，但缺點是耗費時間與記憶體。

整體來說，以區域邊緣作為特徵的方式是簡單方便的，但缺點是會因為影像的模糊、複雜度與交界處不明顯之狀況，導致產生過多直線交織的區域。

### 2.1.2 依影像顏色特徵

世界各國的車牌有各自政策制定的顏色，部分研究即以顏色條件作為篩選條件。[15]將影像內像素轉入 HLS 色彩空間，以預先訓練的模型進行分類。[16]則以 HLS 空間之像素值，透過神經網路作為分類程序，判斷車牌區域。[17]根據韓國的車牌顏色(白、黑、紅、綠)設計系統，只尋找黑白、紅白、綠白之顏色交界處之邊緣，透過限定特定顏色邊緣，排除大量區域，提升準確率。[18][19]透過基因演算法(Genetic algorithm, GA)，預先對車牌樣本的顏色作訓練，訓練過程中調整光影，找出車牌顏色在演算法裡的門檻值。運用平均亮度與門檻值的關係式進行是否為車牌的判斷，最後再以形狀、外觀比例作為進一步的確認。

整體來說，以顏色作為偵測車牌的基礎特徵可避免因拍攝角度導致車牌傾斜、扭曲、模糊變形的狀況，但是會因為光影與雜訊的變化而有所影響，多數使用顏色特徵的方法都需要克服光影與雜訊所造成的顏色變化。

### 2.1.3 依字元排列特徵

以臺灣為例，國內現行車牌上有四至七碼英數字元夾雜排列，產生反覆、密集的颜色轉換，同時規則排列的字元也形成一個可被偵測的「邊緣」。[20]找出類似字元排列的區域，利用霍夫轉換(Hough transform, HT)找出字元排列的上緣，若能順利找出與上緣平行之下緣，則再檢測兩平行邊緣間區域是否為排列字元，即完成車牌判斷。[21]找出可能是字元排列的區域，以神經網路進一步驗證確認。

[22]則是找出反覆對比交替的區域，認定至少是三到四個字元以上的交替次數時，判定為車牌。[23]由背景影像與字元區域的差異，找出字元排列區域後，判斷字元的寬度來進行確認。

[24]使用經訓練之 Adaboost 分類器找出字元區域，再透過以尺度不變特徵轉換(Scale-invariant feature transform, SIFT)訓練後的 SVM 分類器進行汰選。

Adaboost 為「Adaptive Boosting」的縮寫，意即自適應性強的機器學習方法。Adaboost 乃迭代演算法，Adaboost 分類器由多層分類器串接而成，而其自適應性強的關鍵在於分類器訓練時，前一層分類器錯誤分類的樣本，於下一層訓練時進行修正，同時也增加錯誤分類樣本之權重、減少正確分類樣本的權重，直到訓練完成 n 層分類器。透過權重的調整，使錯誤分類在訓練過程裡易於校正。Adaboost 分類器的分類判斷建立於訓練而得之門檻值，輸入影像依序進入各層分類器，若未達其中任一層分類器之門檻即淘汰；反之通過全部層分類器，則視為欲偵測物。

縱觀而言，以字元排列作為偵測基礎，影像源需經過灰階處理與二值化之程序，產生額外的時間消耗。優點為不易受光影影響，但當字體過度傾斜、或者是車體上有其他排列字元時(例如：廣告或者車輛編號)容易造成誤判。本文使用的 HOG 特徵亦屬此類特徵。

表 2-1 各類車牌偵測研究方法比較

| 研究方法   | 主要使用特徵            | 優點             | 缺點                           | 文獻來源      |
|--------|-------------------|----------------|------------------------------|-----------|
| 區域邊緣特徵 | 透過直線，或是矩形作為車牌判斷   | 簡單，快速          | 受限於複雜背景、具較多線條、矩形環境，或者邊緣模糊等狀況 | [4]-[14]  |
| 影像顏色特徵 | 以車牌的顏色作為特徵        | 抗車牌變形、模糊扭曲的能力高 | 受限於光影及雜訊對顏色的影響               | [15]-[19] |
| 字元排列特徵 | 透過字元排列或字元與車牌相間等特徵 | 不易受光影影響        | 計算時間較長，容易受車體上其他排列字元干擾        | [20]-[24] |

## 2.2 字元分割

在偵測到車牌後，需要做字元的分割，將車牌上的貼紙、廣告以及多餘的部分切除，用以得到獨立、單一的字元影像。整理文獻所用字元分割方法，可概分為兩類，各別介紹如下：(一)基於投影，(二)基於像素連接。表 2-2 總結了各種方法的優缺點。

### 2.2.1 基於投影

基於投影的方法是利用字元與車牌為不同顏色，在二值化後可以兩者分離。在[25]中，二值化的影像先做垂直投影，以找到各個字元的起始與結束的位置，然後以水平投影將各個字元獨立出來。此方法簡單且實用，即使車牌稍微旋轉，也能有要得將字元分割出來。但此方法需要事先了解車牌字元的排列規則，並且雜訊過大也會影響分割的效果。

### 2.2.2 基於像素連接

基於像素連接的方法是通過標記圖像中所有連接的像素來進行分割。具有相

同標記的像素被視為同一字元，此方法簡單且對於旋轉的影像非常有效，但當字元出現斷裂或是連接在一起時，就無法正確地分割字元。

表 2-2 各類字元分割研究方法比較

| 研究方法   | 優點       | 缺點          |
|--------|----------|-------------|
| 基於投影   | 簡單，快速    | 雜訊會影響投影值    |
| 基於像素連接 | 可處理旋轉的字元 | 無法處理斷裂或重疊字元 |

## 2.3 字元辨識

過去字元辨識研究也提出許多研究方法，試圖克服種種困難處。整理文獻所用字元辨識方法，可概分為兩類，各別介紹如下：字元像素值辨識，字元特徵辨識。表 2-3 總結了這兩種方法的優缺點。

### 2.3.1 字元像素值辨識

此類辨識方法以樣板比對法(Template match)為大宗。樣本比對法透過比較待辨識之字元與預先準備之各字元樣本(Template)，計算兩者之間的差值，具有最近距離之樣板即為該待測字元的辨識結果，不少研究文獻以樣板比對法進行字元辨識。

樣本比對法非常簡單、直觀，沒有過多影像處理或特徵擷取，意即過程中必須降低外部的干擾因素，以確保比對的穩定性。大部分研究於樣板比對之前，需將待辨識字元灰階轉換與二值化，減少字元影像上雜訊的影響，又如 [8][10][16][26][27][28] 將待辨識字元尺寸重設至與樣板相符，以利比對。樣板比對法仍有許多限制，例如樣板與待辨識字元必須為相同字體，無法辨識旋轉、傾斜破碎的字元，受限影像雜訊、髒汙或者是字元不完整等等狀況，都有會影響樣本比對法的準確性。

### 2.3.2 字元特徵辨識

此方法使用機器學習技術基於一個或多個特徵來辨識字元。[29]使用圖像密度為特徵訓練神經網路來辨識字元。也有一部分人使用人工神經網路(ANN)、支援向量機(SVM)、隱馬可夫模型(HMM)、卷積神經網路(CNN)等機器學習來辨識。

表 2-3 兩類字元辨識方法比較

| 研究方法    | 主要方法                      | 優點                      | 缺點                            | 文獻來源                     |
|---------|---------------------------|-------------------------|-------------------------------|--------------------------|
| 字元像素值辨識 | 樣板比對法<br>(Template match) | 簡單，沒有過多<br>處理程序         | 受限於字型差異、<br>傾斜、雜訊等            | [8][10][16]<br>[26]-[28] |
| 字元特徵辨識  | 各種提取字元特<br>徵的方式           | 對外來干擾較穩<br>定，減少多餘資<br>訊 | 特徵提取需時間；<br>特徵種類可能影響<br>辨識準確性 | [29]                     |

## 2.4 物件偵測

世界上有許多物件偵測的架構被提出，從傳統的樣板比對法(Template Matching)、可變形組件模型(DPM)[30]、尺度不變特徵轉換(SIFT) [31]，到基於深度學習的 RCNN[32]、Fast RCNN[33]、Faster RCNN[34]、Mask RCNN[35]、SSD[36]、YOLO[37]、YOLOv2[2]。此節將分析基於深度學習之物件偵測架構的相關文獻以及其優缺點。將於下文中，依序介紹與本研究使用方法相關之文獻整理。

### 2.4.1 Region-based CNN (R-CNN)

R-CNN[32]由 Ross Girshick 等人於 2014 年發表的物件偵測架構，相較於之前 VOC 2007 資料集中的最佳辨識結果，準確率提高了將近 50%，擁有 66.0 mAP 的準確率。但其缺點為處理速度相當慢，每張影像的處理時間為 20 秒。圖 2-1 為 R-CNN 架構圖。

整體而言，此系統架構可視為四個模組所組成。第一個模組為選擇性搜索

(Selective Search)，用途為提出影像中可能有待測物件之候選區域。以影像中物件的顏色相似度、紋理相似度、空間交疊相似度等特徵來計算。一張影像大約生成一千至兩千個候選區域。第二個模組為透過卷積神經網路(CNN)來提取候選區域之特徵，此卷積神經網路是由五個卷積層(convolutional layer)與兩個全連階層(fully connected layer)所組成。第三個模組將提取出來的特徵透過支援向量機(SVM)來分類，分類成事先定義的類別。最後一個模組則是以回歸的方式來校正候選框的座標。

R-CNN 最大的缺點為訓練與辨識的速度都很慢，最主要的原因為整個架構分為太多階段與選擇性搜索提出 2000 ~ 3000 個候選區域導致計算量太大。

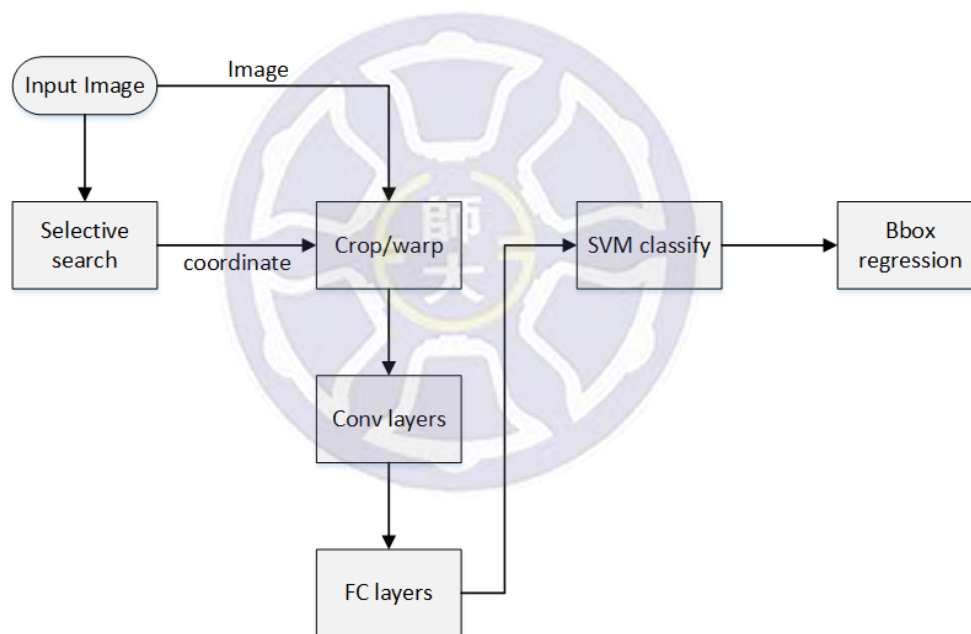


圖 2-1 R-CNN 架構圖

## 2.4.2 Fast Region-based CNN (Fast R-CNN)

Fast R-CNN[33]由 Ross Girshick 等人於 2015 年所發表，是針對現有之 R-CNN 模型改進和簡化，並提高的辨識的速度與準確性。Fast R-CNN 在 VOC 2007 資料集中有 70.0 mAP 的準確率，每張影像的處理時間為 2 秒，比 R-CNN 快了 10 倍。圖 2-2 為 Fast R-CNN 架構圖。

R-CNN 有個較大的缺點為每個候選區域都須分別通過 CNN 來提取特徵，這導致多次執行相同的計算。在 Fast R-CNN 中提出以感興趣區域(ROI Pooling)的共享計算來大幅減少特徵提取的計算量。R-CNN 的另一個缺點為計算分成太多模組，導致整體訓練過程複雜度過高，於是 Fast R-CNN 將所有的模組整合為單一模組來改善複雜度，並將物件分類與邊框回歸的計算結合，以提高訓練的速度。最後以 Softmax 分類器取代 SVM 分類器來提高辨識的準確性。

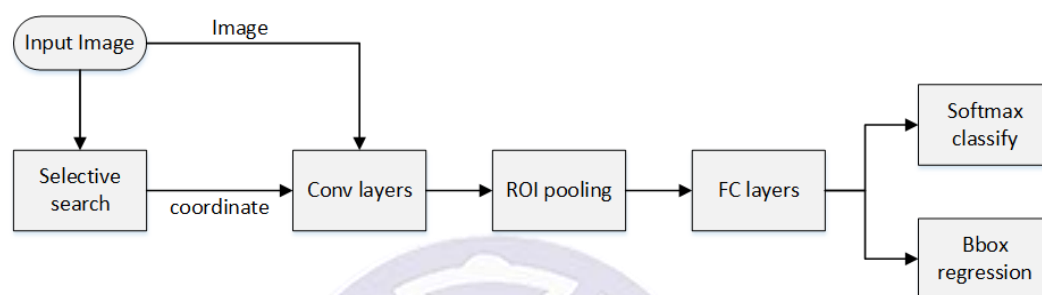


圖 2-2 Fast R-CNN 架構圖

### 2.4.3 Faster Region-based CNN (Faster R-CNN)

Faster R-CNN[34]由 Ross Girshick 等人於 2016 年提出，基於 Fast R-CNN 進一步改善，改變了傳統區域建議的方法，在準確度與速度都有所提高，辨識率有 73.2 mAP，每張影像處理時間為 140 毫秒，比 Fast R-CNN 快了 10 倍。圖 2-3 為 Faster R-CNN 架構圖。

Faster R-CNN 可視為區域提議網路(Region Proposal Network)結合 Fast R-CNN，以 RPN 取代選擇性搜索(Selective Search)，大幅提升區域提議的速度，並透過 K 個 anchor boxes 在特徵圖上滑動窗口來偵測物件與計算候選框的信心分數。其中為了提高速度，RPN 與 Fast R-CNN 共享了卷積神經網路的參數，使區域提議的階段不須額外的計算。

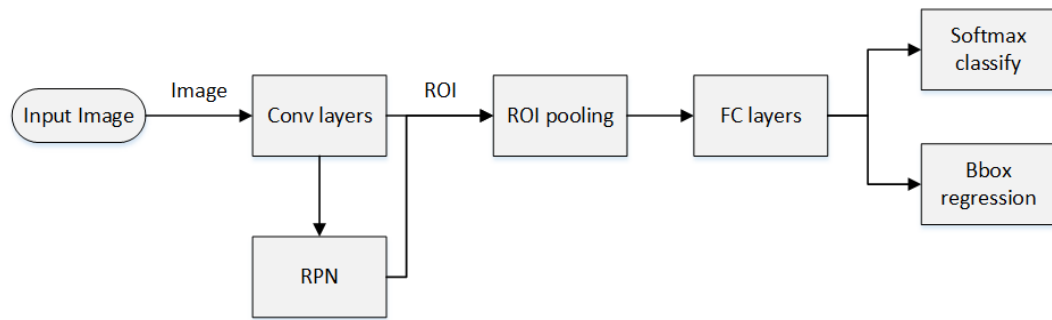


圖 2-3 Faster R-CNN 架構圖

#### 2.4.4 You Only Look Once (YOLO)

YOLO[37]由 Joseph Redmon 等人於 2015 年提出，為第一個 One Stage 的物件偵測架構，與 Faster R-CNN 等 Two Stage 的架構非常不同。One Stage 架構的辨識過程比較簡單，因為它只使用一個 CNN 架構來處理影像，並且不需要候選區域的提議，這讓 YOLO 可以更快的偵測物件，其處理一張影像的時間僅需 22 毫秒，也就是 45 FPS，比 Faster R-CNN 快了 6 倍，此速度已達到即時辨識的標準，但其辨識率只達到 63.4 mAP，明顯比 Faster R-CNN 差。

YOLO 將邊界框的預測與物件類別的辨識視為回歸問題，並只使用單一個 CNN 網路來預測結果。其計算如圖 2-4 是將輸入的影像分成  $S \times S$  個網格，每個網格會預測  $B$  個邊界框座標、邊界框的信心值以及  $C$  個物件類別的機率，最後透過非極大值抑制(NMS)輸出最佳的邊界框與物件類別。

雖然 YOLO 的辨識速度達到了即時辨識的程度，但有兩個較大的缺點為邊界框的定位較不精準，與對於影像中較小或重疊度較高的物件偵測效果較差。

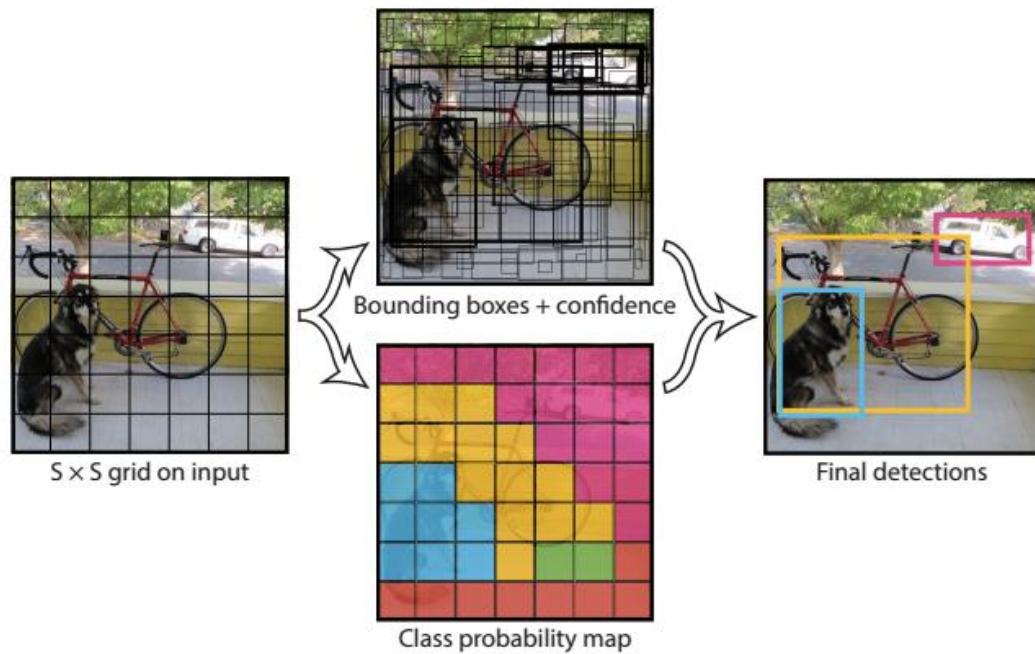


圖 2-4 YOLO 的概述圖。輸入影像被分為  $S \times S$  個網格，每個網格會預測  $B$  個邊界框、邊界框的信心值以及  $C$  個物件類別的機率。[37]

### 2.4.5 YOLOv2

YOLOv2[2] 又稱 YOLO9000 由 Joseph Redmon 等人於 2016 年提出，是基於 YOLO 來改善的版本，在 VOC 2007 的資料集中有 76.8 mAP 的準確率，並且擁有 67 FPS 的辨識速度，而其較慢 40 FPS 的版本更是擁有 78.6 mAP 的辨識率。

YOLOv2 相較於 YOLO 做了許多改進，第一，在所有卷積層後都添加批量歸一化 (Batch Normalization)，提高模型的收斂速度並避免過擬合 (Overfitting)，這增加約 2% 的 mAP 分數。第二，將原本 YOLO  $224 \times 224$  的輸入影像增加到  $448 \times 448$  來進行訓練與測試，這增加約 4% 的 mAP 分數。第三透過聚類演算法 K-means 來自動找出邊界框的預設值，並且找出在模型複雜度與準確率之間的平衡，選擇 5 個 anchor boxes 作為邊界框，這增加約 5% 的 mAP 分數。第四，添加直通層將不同通道、不同特徵圖的特徵連接起來，用以提高網路對較小物件的偵測能力，增加約 1% 的 mAP 分數。第五，為了使 YOLOv2 在不同解析度的影像上都能有效地偵測物件，因此引入了多尺度的訓練，在訓練的過程中，網路會從 {320,

352, ..., 608} 中隨機選擇一個新的解析度作為輸入圖片的大小，並每隔 10 個批次及變換一次，這使其對圖象大小的變化能有穩定辨識。

YOLOv2 透過上述的技巧在提升辨識率的同時維持即時辨識的速度，相較於 YOLO 有極大的改善。





### 第三章 研究方法

本章節將介紹本論文提出的車牌辨識系統之設計與研究方法。為了適應一般道路之複雜環境，本論文提出的車牌辨識系統包含四大模組，分別為車輛偵測、車牌偵測、字元分割、字元辨識，透過此架構可避免將交通號誌或廣告招牌是為車牌，細節將依序於本章詳述。本章各節將提及：3.1 介紹完整系架構程；3.2 介紹車輛偵測模組，說明車輛偵測研究方法；3.3 介紹車牌偵測模組，說明車牌偵測研究方法；3.4 介紹字元分割模組，敘述完整之分割流程與方法；3.5 針介紹車牌字元辨識模組說明。

#### 3.1 系統流程



圖 3-1 車牌辨識系統架構圖

本論文提出之車牌辨識系統如圖 3-1，待測影像將經過車輛偵測、車牌偵測、字元分割與字元辨識，共四個模組完成辨識。影像輸入後，由車輛偵測模組偵測影像內所有車輛所在區域；若影像內含有車輛區域，則擷取該區域(車輛本身)進入車牌偵測模組，在由車牌偵測模組將車輛上的車牌擷取下來，並將車牌影像輸入至後續字元分割、辨識等模組；反之即進入下一待偵測影像。

## 3.2 車輛偵測模組

傳統 Two Stage 的物件偵測方法，例如：Fast R-CNN、Faster R-CNN 等等，基於滑動窗口(Sliding Windows)或選擇性搜索(Selective Search)來偵測可能的目標，再以 CNN 或其他方法來確定目標是否正確。由於一般道路上拍攝照片的尺寸和環境的複雜度，使用此類物件偵測架構會非常耗時。本論文使用 One Stage 的物件偵測架構 YOLOv2[2]來偵測車輛，並以 COCO 2017[1]的資料集來訓練。車輛的種類分為汽車、機車、公車與卡車。這個階段會將偵測到的車輛座標輸出至下個階段。偵測結果如圖 1-6。

## 3.3 車牌偵測模組

偵測到車輛後，本論文以 HOG 為特徵之 SVM 分類器對車輛的影像進行車牌偵測。SVM 分類器之運用可分為「訓練階段」與「偵測階段」。

### 3.3.1 訓練階段

於訓練階段時需準備「正樣本」與「負樣本」，「正樣本」意指車牌影像，「負樣本」則是不含有車牌區域之樣本影像。本研究所用樣本來源皆為一般道路，由道路旁固定式側拍影像再進行裁切。正樣本包含汽車與機車的車牌影像。負樣本則有如：各種車輛車體、行道樹、道路路面，道路指標、商家招牌、行人、房屋等眾多非車牌區域之影像。



圖 3-2 正樣本範例

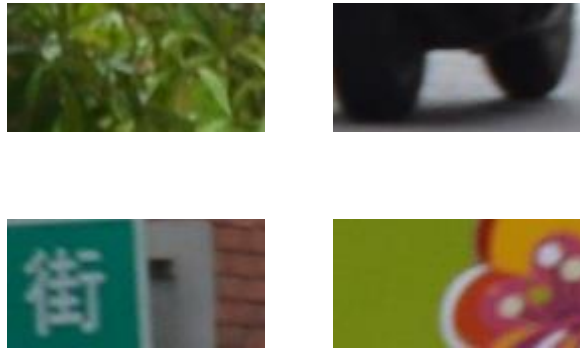


圖 3-3 負樣本範例

完成正樣本與負樣本裁切後，統一樣本尺寸進行後續 HOG 特徵提取與 SVM 訓練。訓練階段流程圖如下：

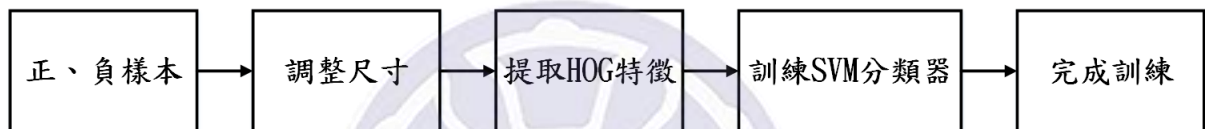


圖 3-4 車牌偵測訓練階段流程圖

### 3.3.2 偵測階段

經過訓練階段，SVM 分類器已能分類影像「是否為車牌」。由於車牌可能位於輸入影像中之任意位置，因此使用滑動偵測窗的方式逐一偵測部分影像，計算 HOG 給予 SVM 作分類判斷。同時偵測過程中也調整偵測窗尺寸，使任何位置、任意尺寸之車牌皆能被偵測。偵測結果如圖 1-7。偵測階段流程如下：

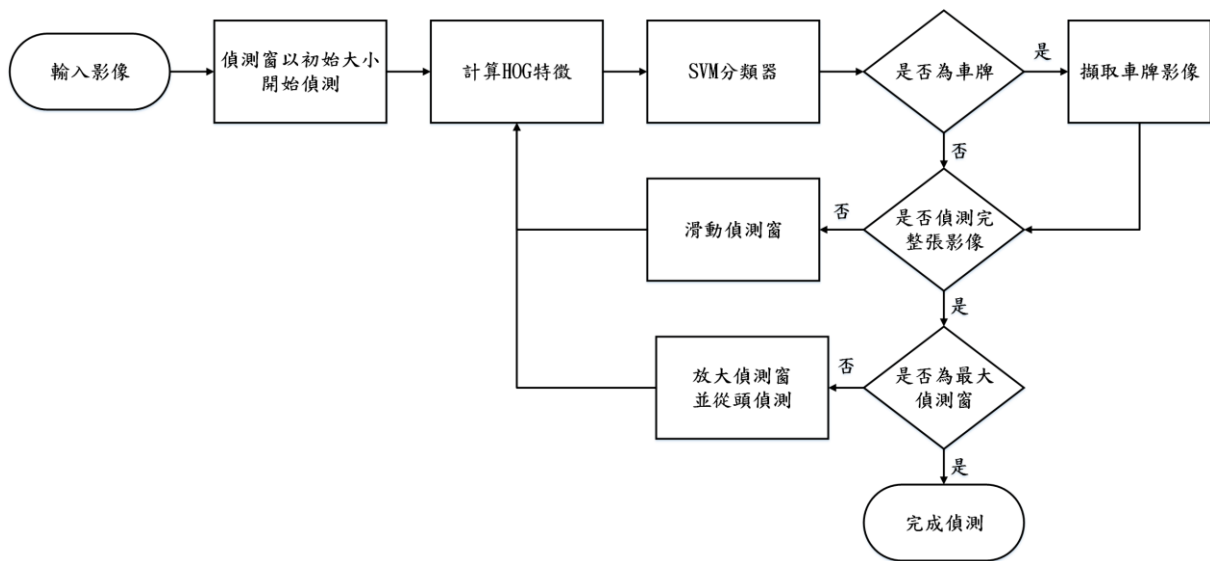


圖 3-5 偵測階段流程圖

### 3.4 字元分割

車牌偵測模組所抓取完整車牌，必須經裁切字元以外的多餘區域，將各個字元分割成單一字元以便後續辨識。字元分割模組包含幾個執行步驟：灰階轉換、二值化、水平投影裁切、垂直投影裁切，將於後文 3.4.1 二值化、3.4.2 投影裁切，依序詳述。字元分割模組流程圖如下：

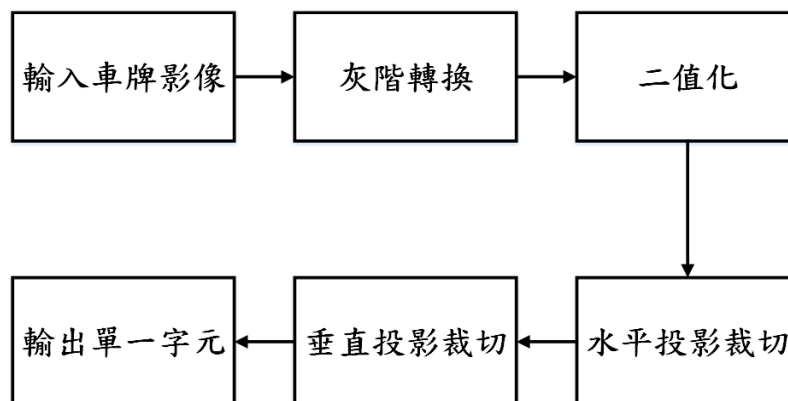


圖 3-6 字元分割模組流程圖

### 3.4.1 二值化

二值化的目的為降低影像中雜訊、加強影像對比、留下影像中感興趣的資訊；根據灰階影像定義：像素值 0 為黑色，255 為白色。透過閾值與影像中各像素進行比較，將像素值大於閾值之像素值設為 255，反之小於閾值則設為 0，使原始影像只剩黑白兩色，稱為二值化，以數學式表示為： $(R(x,y)$ 為點 $(x,y)$ 像素值， $T$ 為二值化閾值)

$$R(x,y) = \begin{cases} 255, & R(x,y) > T \\ 0, & otherwise \end{cases}$$

二值化關鍵為閾值，閾值的設置方式可分為兩種：「固定閾值」與「自適應閾值」。固定閾值，顧名思義將閾值設為一定值，此種設置方式適合用於穩定光影的環境，可透過實驗找到最佳閾值後，設為固定閾值，節省閾值計算時間。而一般道路光影變化不穩定，故固定閾值無法適用於車牌偵測。

本研究採用「自適應閾值」的方式，找出每張車牌影像各自的閾值，而這些閾值會隨影像當時環境的變化而有所不同。自適應閾值的方法有很多，舉凡像是雙峰法、P 參數法、平均法、Otsu 法等等。本論文使用之二值化閾值取決於影像中灰階像素值之最大與最小值找出影像中灰階值最大值與最小值後，以最大與最小值之平均作為二值化閾值。以數學式表示於下：

$$T = \frac{\max(R(x,y)) + \min(R(x,y))}{2}$$



圖 3-7 (左)原圖與(右)二值化後

經二值化的車牌影像，可以消除原影像中部分非感興趣之物件，將車牌字元

突顯出來，如上圖所見，車牌上的貼紙與車行廣告大多能以二值化的方式過濾。

### 3.4.2 投影裁切

二值化後的影像，由黑白二色構成，觀察二值化後的車牌影像，可發現其中車牌底色為白色、字元為黑色，並且多數黑色像素僅存於字元的位置，意即透過車牌上黑色像素分佈，可作為找尋字元排列區域之特徵。為找出黑色像素之分佈，使用影像投影作為研究方法，對二值化後的車牌影像進行水平投影，判斷車牌中排列字元的位置。水平投影，意即將同列黑色像素數目累加統計，並重新依照黑色像素數目，以黑色畫在影像上同列一側，達到將每一列黑色像素投影至同列一側的效果。



圖 3-8 將(左)二值化後影像，進行水平投影結果(右)

由上圖所示，將左圖進行水平投影至右邊垂直軸，可得到如右圖的結果。而從右圖中，可以看出中間區塊之黑色像素最密集，該區域即是原車牌影像中字元排列區域；故水平投影法能裁切掉車牌上方及下方多餘區域，保留中間字元之區域，裁切後結果如下圖：

**|K9C-039|**

圖 3-9 依水平投影所裁出字元排列區域

水平裁切後字元排列影像有幾項特徵：字元間有些許間格，以此分佈特徵為依據，進行垂直投影，可將排列字元分割成單一字元。

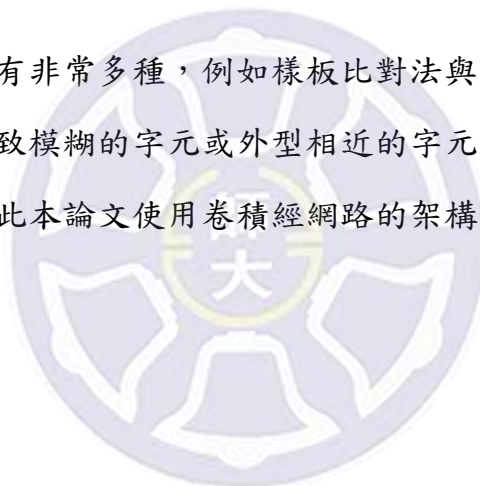


圖 3-10 將排列字元進行垂直投影

如上圖所示，將水平裁切後之車牌圖(左)，垂直投影於下方水平軸而得垂直投影圖(右)後，根據字元間的投影值低點，可判斷為字元間格，以此依序將字元分割，使各字元獨立分隔為單一字元，完成車牌上字元之分割程序。

### 3.5 字元辨識

字元辨識的方法有非常多種，例如樣板比對法與 SVM 分類器，但這些方法對於因車輛的移動導致模糊的字元或外型相近的字元如 2、Z 與 8、B，0、D 等的辨識效果不佳，因此本論文使用卷積經網路的架構來辨識字元。





## 第四章 實驗結果

此章節將說明本論文嘗試之所有實驗。對於車牌辨識系統中四大模組依序介紹，分別述於：4.1 車輛偵測實驗，4.2 車牌偵測實驗，4.3 字元分割實驗，4.4 字元辨識實驗，並於 4.5 實驗結論，進行實驗結果討論。

### 4.1 車輛偵測實驗

車輛偵測模組中，實驗包含 4.1.1 訓練 YOLOv2、4.1.2 動態影像偵測。

#### 4.1.1 訓練 YOLOv2

訓練 YOLOv2 偵測車輛的部分是使用 COCO 2017 的資料集，COCO 2017 資料集訓練資料有 18G，驗證資料 1G，測試資料 6G，資料影像如圖 4-1。

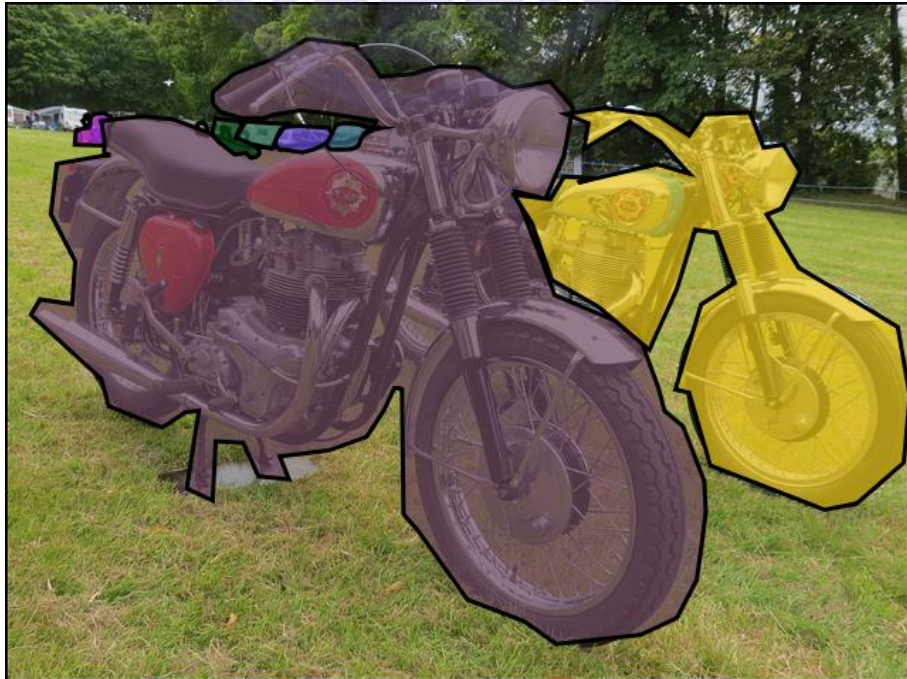


圖 4-1 COCO 2017 資料集之影像

## 4.1.2 動態影像偵測

YOLOv2 練完成後，實作以 YOLOv2 偵測道路動態影像之實驗。動態影像與靜態影像的差異，在於靜態影像較容易對焦於移動物體，而動態影像則無法。因為靜態影像拍攝時在時間上的限制較少，有充分的時間處理影像資料，而動態影像必須每秒鐘連續紀錄 30 幀(30 FPS)或 60 幀(60 FPS)畫面，所以速度要相當快，因此拍攝之影像畫質會受到限制，也會造成影像畫面較為模糊不清。這是動態影像偵測的一大難處，更會造成後續車牌偵測、字元分割、字元辨識的問題。

本實驗拍攝動態影像採用的是 1920 \* 1080 畫素、每秒 30 與 60 幀影像之規格，動態影像攝影器材配置如表 4-1，而影像來源皆為道路測拍、天橋上拍攝以及行車紀錄器等行進中車輛影像，如圖 4-2。實驗顯示，整體車輛偵測率達 96.12%，表 4-2 為分別統計汽車、卡車、公車與機車之偵測率，圖 4-3 為偵測的畫面。此階段最後會輸出畫面中車輛的座標至下一階段進行車牌偵測。

表 4-1 動態影像攝影器材配置

|        | 實驗配置           |               |
|--------|----------------|---------------|
| 使用器材   | Sony HDR-PJ675 | SJCAM SJ4000  |
| 動態影像來源 | 道路邊固定式腳架側拍     | 安裝於安全帽上之行車紀錄器 |
| 畫素     | 1920 * 1080    | 1920 * 1080   |
| 幀數     | 60 FPS         | 30 FPS        |
| 定焦距離   | 20 公尺          | X             |

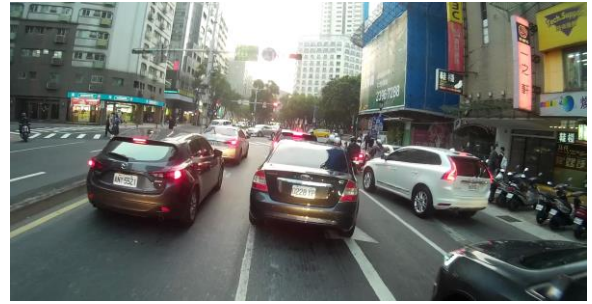


圖 4-2 左圖為道路測拍，右圖為行車紀錄器之影像

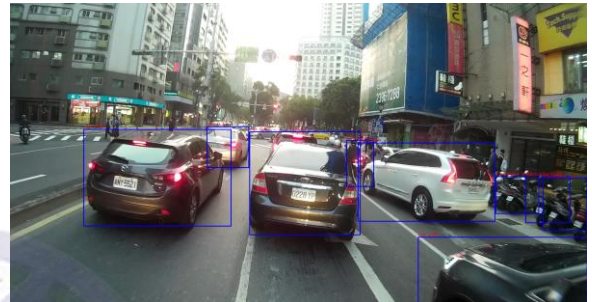
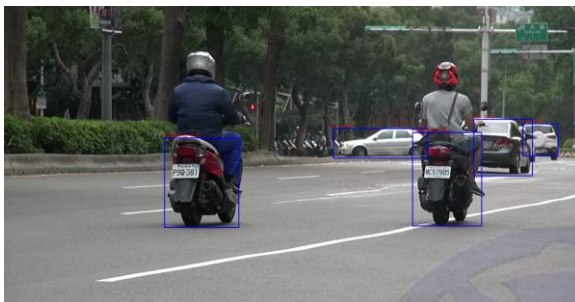


圖 4-3 車輛偵測後的畫面

表 4-2 YOLOv2 車輛偵測率

|                | 汽車                      | 卡車、公車       | 機車                |
|----------------|-------------------------|-------------|-------------------|
| <b>Video01</b> | 114/117 (97.43%)        | 3/3 (100%)  | 95/98 (96.94%)    |
| <b>Video02</b> | 124/124 (100%)          | 9/9 (100%)  | 81/94 (86.17%)    |
| <b>Video03</b> | 107/107 (100%)          | 8/8 (100%)  | 55/56 (98.21%)    |
| <b>Video04</b> | 87/89 (97.75%)          | 4/5 (80%)   | 57/64 (89.06%)    |
| <b>總量</b>      | 432/437 (98.86%)        | 24/25 (96%) | 288/312 (925.31%) |
| <b>平均</b>      | <b>744/774 (96.12%)</b> |             |                   |

## 4.2 車牌偵測實驗

車牌偵測模組的實驗中，包含 4.2.1 訓練 SVM 分類器，與比較 4.2.2 基於 SVM 之方法偵測車牌與 4.2.3 使用 YOLOv2 偵測車輛再結合 SVM 偵測車牌的差異。

#### 4.2.1 訓練 SVM 分類器

訓練 SVM 分類器必須準備大量正樣本與負樣本。本論文之實驗訓練樣本皆為道路測拍行進中車輛影像如圖 4-2 之左圖，在經過裁切而得如圖 3-2、圖 3-3。其中道路測拍影像有 1594 張，再從中裁切出 1778 張車牌影像作為正樣本，以及 5400 張負樣本進行分類器訓練。

#### 4.2.2 基於 SVM 之方法偵測車牌

訓練完 SVM 分類器後，實作以 SVM 分類器直接對動態影像偵測車牌，發現當分類器靈敏度低時，車牌召回率(Recall Rate)會因偵測不到較小的車牌而降低。將靈敏度調高時，雖然召回率因車牌都有偵測到而提升為 94.24% (表 4-4)，但卻產生大量的誤報(False Positive)導致精確率(Precision Rate)非常低如圖 4-4，這也會影響後續的字元辨識。表 4-3 為計算召回率與精確率之混淆矩陣。

$$\text{Recall Rate} = \text{TP}/(\text{TP} + \text{FN})$$

$$\text{Precision Rate} = \text{TP}/(\text{TP} + \text{FP})$$

表 4-3 混淆矩陣

|       | 相關                  | 不相關                 |
|-------|---------------------|---------------------|
| 被檢索到  | True Positive (TP)  | False Positive (FP) |
| 未被檢索到 | False Negative (FN) | True Negative (TN)  |

表 4-4 基於 SVM 之方法偵測車牌的 Recall Rate

|                    | 車牌偵測率                   |
|--------------------|-------------------------|
| <b>Video01</b>     | 114/119 (95.80%)        |
| <b>Video02</b>     | 87/93 (93.55%)          |
| <b>Video03</b>     | 133/137 (97.08%)        |
| <b>Video04</b>     | 124/137 (90.51%)        |
| <b>Recall Rate</b> | <b>458/486 (94.24%)</b> |



圖 4-4 基於 SVM 之方法偵測車牌，產生大量誤報

### 4.2.3 使用 YOLOv2 偵測車輛再結合 SVM 偵測車牌

由於上述實驗發現直接偵測車牌會產生大量誤報，因此本論文改變傳統架構，先透過 YOLOv2 偵測車輛在影像中之位置，將車輛座標傳至車牌偵測階段，再以 SVM 分類器針對車輛所在區域偵測車牌，可以減少大量誤報如圖 4-5。

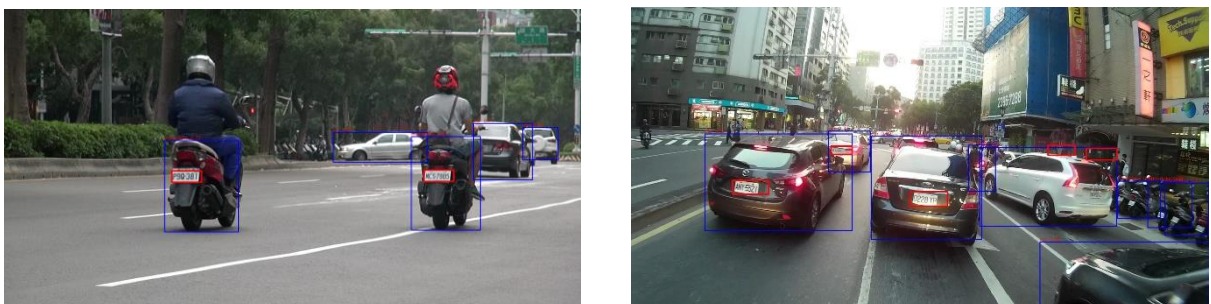


圖 4-5 使用 YOLOv2 偵測車輛再結合 SVM 偵測車牌，減少大量誤報

#### 4.2.2 基於 SVM 之方法偵測車牌與 4.2.3 使用 YOLOv2 偵測車輛再結合 SVM

偵測車牌之比較呈現於下表：

表 4-5 基於 SVM 之方法與 YOLOv2 結合 SVM 車牌偵測 Precision Rate 比較表

|        | SVM |    |           | YOLOv2 + SVM |    |           | Improve |
|--------|-----|----|-----------|--------------|----|-----------|---------|
|        | TP  | FP | Precision | TP           | FP | Precision |         |
| IMG_01 | 2   | 13 | 13.33%    | 2            | 0  | 100%      | 86.67%  |
| IMG_02 | 3   | 27 | 10%       | 3            | 4  | 42.86%    | 32.86%  |
| IMG_03 | 2   | 35 | 5.4%      | 2            | 2  | 50%       | 44.6%   |
| IMG_04 | 2   | 23 | 8%        | 2            | 1  | 66.67%    | 48.67%  |
| IMG_05 | 2   | 9  | 18.18%    | 2            | 1  | 66.67%    | 48.49%  |
| IMG_06 | 4   | 42 | 8.7%      | 4            | 2  | 66.67%    | 57.97%  |
| IMG_07 | 2   | 12 | 14.29%    | 2            | 5  | 28.57%    | 12.28%  |
| IMG_08 | 3   | 5  | 37.5%     | 3            | 0  | 100%      | 62.5%   |

### 4.3 字元分割實驗

字元分割實驗分為兩個階段，分別為 4.3.1 二值化，4.3.2 字元分割。

#### 4.3.1 二值化

二值化計算中閾值的設定非常重要，直接影響輸出的影像，閾值若設定的差，輕則車牌上的貼紙、髒污無法濾除，重則字元黏著或是消失，並且需要克服不同光影、不同色彩之變化。本實驗透過自適應閾值的方法，可成功將車牌二值化，並盡量減少字元黏著的情況，以利後續作字元分割。相較於固定閾值的做法，自適應閾值能擁有較佳的穩定性。



圖 4-6 車牌二值化

### 4.3.2 字元分割

實驗中，先將二值化後的車牌影像進行水平投影圖 4-7，把車牌上下多餘的地方切除，再透過垂直投影的方式將字元分割圖 4-8，最後得到單獨的字元圖 4-9。



圖 4-7 水平投影切除上下多餘部分



圖 4-8 垂直投影分割獨立字元



圖 4-9 分割後之字元

## 4.4 字元辨識實驗

字元辨識實驗分為兩個階段，分別為 4.3.1 訓練，4.3.2 測試。這兩個階段都

是使用自行設計之卷積神經網路(CNN)來訓練與辨識字元，圖 4-10 為網路架構圖，輸入的字元影像會先調整成 28\*28 像素的大小，分別經過兩個核心為 5\*5 大小的卷積層(Convolution Layers)與兩個 2\*2 的最大池化層(Maxpooling Layers)來提取影樣特徵，再將影像特徵透過兩個全連接層(Fully Connected Layers)的計算，最後輸出 34 種類別(英文 A~Z，不含 I、O，數字 0~9)。

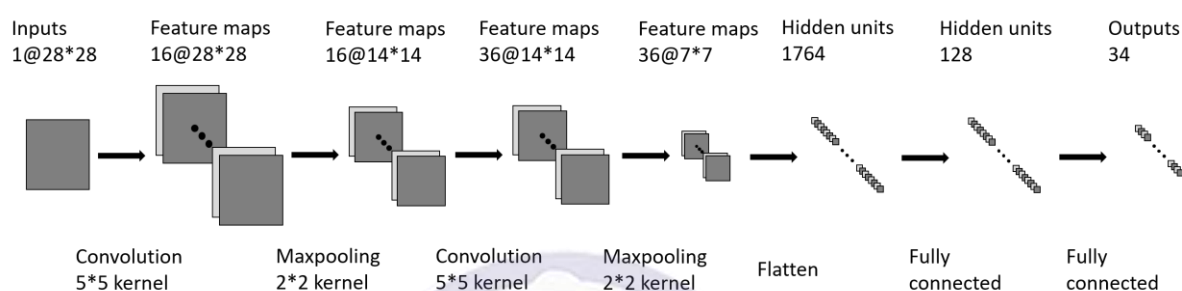


圖 4-10 卷積神經網路架構圖

#### 4.4.1 訓練

針對英文 A~Z (不含 I、O) 與數字 0~9 共 34 種字元之卷積神經網路訓練，樣本來源為由動態影像之車牌經過字元分割後所得字元影像，字元樣本數為 14,627 個，如圖 4-11 有清晰也有模糊的影像，並將所有樣本打散後隨機選取其中 60% 作為訓練樣本，40% 做為測試資料。



圖 4-11 樣本有清晰(上面三張)、有模糊的影像(下面三張)

## 4.4.2 測試

訓練完字元辨識之卷積神經網路後，以樣本剩餘之 40% 資料來測試，最後辨識率為 99.2%。除此之外，另外準備一些更為模糊之字元來測試，這些待測字元影像原始大小約為 9\*18~25\*50 像素之間，發現雖然此卷積神經網路對更小更模糊之字元影像大多都能辨識如圖 4-12，但少部分近似字元(B 與 8，U 與 V、0)仍會辨識錯誤如圖 4-13。

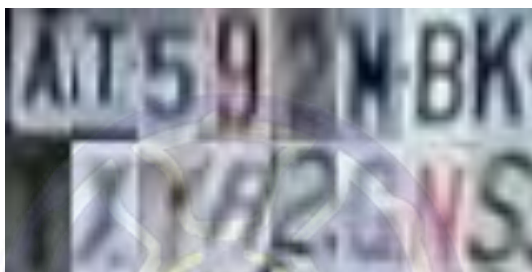


圖 4-12 辨識成功之更小更模糊字元



圖 4-13 辨識失敗之模糊近似字元，分別為 B 辨識成 8，U 辨識成 V 與 0

整體而言，以卷積神經網路為架構之字元辨識方法與其他研究方法比較皆有較佳的辨識結果如下表 4-6。

表 4-6 與其他研究方法比較

| Method                       | Number of samples | Pattern of Samples                 | Accuracy |
|------------------------------|-------------------|------------------------------------|----------|
| template matching [5]        | 1,176             | various scene and condition        | 93.1%    |
| template matching [6]        | 180               | N/A                                | 95.7%    |
| template matching [9]        | 2,340             | different weather and illumination | 98.6%    |
| multi-template matching [11] | 400               | N/A                                | 97.2%    |
| Kirsch edge detection [15]   | 2,000             | outdoor, Stopped vehicle           | 92.7%    |
| SVM-binary tree [21]         | 300               | outdoor, Stopped vehicle           | 95.2%    |
| SVM-OAR + binary tree [22]   | 260               | N/A                                | 91.9%    |
| SVM [23]                     | 1,000             | N/A                                | 98.2%    |
| SVM-binary tree [24]         | 700               | N/A                                | 96.0%    |
| K-means + SVM [39]           | 1,530             | roadside, moving vehicle           | 98.9%    |
| Our proposed CNN model       | 5,851             | roadside, moving vehicle           | 99.2%    |

#### 4.5 實驗結論

本研究提出之車牌辨識系統架構，將實驗總結呈現於下表：

表 4-7 實驗總結

|     | 車輛偵測   | 車牌偵測   | 字元辨識  |
|-----|--------|--------|-------|
| 樣本數 | 774    | 486    | 5,851 |
| 偵測數 | 744    | 458    | 5,803 |
| 偵測率 | 96.12% | 94.24% | 99.2% |

## 第五章 結論與未來展望

### 5.1 結論

不同於以往車牌辨識系統大多受限於多變光影與道路複雜環境等因素，本研究提出一個可應用於多車道偵測之車牌辨識系統。本研究具有下述特點：

1. 透過先以 YOLOv2 偵測車輛再用 SVM 偵測車牌的方式，可在保持召回率 (Recall Rate) 不變的同時，濾除大量誤報 (False Positives)，提升精確率 (Precision Rate)。
2. 設計一卷積神經網路 (CNN Model) 提高模糊與近似字元辨識的準確率。

### 5.2 未來展望

經過所有實驗與討論，整理出以下數點作未來改進方向：

1. 將 SVM 車牌偵測的部分換成 YOLOv2，以減少系統架構的複雜度。
2. 提升系統的辨識速度，若要實際應用於路口監視器必須達到即時辨識，目前系統架構較複雜導致花費過多時間。
3. 減少車上貼紙廣告造成的錯誤抓取。

## 參 考 文 獻

- [1] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. “Microsoft COCO: Common objects in context,” in *ECCV*. 2014.
- [2] J.Redmon and A.Farhadi, “Yolo9000: Better,faster,stronger,” arXiv preprint arXiv:1612.08242, 2016.
- [3] Shan Du, Member, IEEE, Mahmoud Ibrahim, Mohamed Shehata, Senior Member, IEEE, and Wael Badawy, Senior Member, IEEE, “Automatic License Plate Recognition (ALPR): A State-of-the-Art Review,” in *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, vol. 23, NO. 2, FEBRUARY 2013, pp.311-325.
- [4] M. Sarfraz, M. J. Ahmed, and S. A. Ghazi, “Saudi Arabian license plate recognition system,” in *Proc. Int. Conf. Geom. Model. Graph.*,2003, pp. 36–41.
- [5] I. Paliy, V. Turchenko, V. Koval, A. Sachenko, and G. Markowsky,“Approach to recognition of license plate numbers using neural networks,” in *Proc. IEEE Int. Joint Conf. Neur. Netw.*, vol. 4. Jul. 2004,pp. 2965–2970.
- [6] C. Nelson Kennedy Babu and K. Nallaperumal, “An efficient geometric feature based license plate localization and recognition,” in *Int. J. Imaging Sci. Eng.*, vol. 2, no. 2, pp. 189–194, 2008.
- [7] H. Bai and C. Liu, “A hybrid license plate extraction method based on edge statistics and morphology,” in *Proc. Int. Conf. Pattern Recognit.*,vol. 2. 2004, pp. 831–834.
- [8] D. Zheng, Y. Zhao, and J. Wang, “An efficient method of license plate location,” in *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2431–2438,2005.
- [9] K. Kanayama, Y. Fujikawa, K. Fujimoto, and M. Horino, “Development of vehicle-license number recognition system using real-time image processing and its application to travel-time measurement,” in *Proc. IEEE Veh. Tech. Conf.*, May 1991, pp.

798–804.

- [10] V. Kamat and S. Ganesan, “An efficient implementation of the Hough transform for detecting vehicle license plates using DSPs,” in *Proc. Real-Time Tech. Applicat. Symp.*, 1995, pp. 58–59.
- [11] C. Busch, R. Domer, C. Freytag, and H. Ziegler, “Feature based recognition of traffic video streams for online route tracing,” in *Proc. IEEE Veh. Tech. Conf.*, vol. 3. May 1998, pp. 1790–1794.
- [12] S. Zhang, M. Zhang, and X. Ye, “Car plate character extraction under complicated environment,” in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, vol. 5. Oct. 2004, pp. 4722–4726.
- [13] A. M. Al-Ghaili, S. Mashohor, A. Ismail, and A. R. Ramli, “A new vertical edge detection algorithm and its application,” in *Proc. Int. Conf. Comput. Eng. Syst.*, 2008, pp. 204–209.
- [14] T. D. Duan, D. A. Duc, and T. L. H. Du, “Combining Hough transform and contour algorithm for detecting vehicles’ license-plates,” in *Proc. Int. Symp. Intell. Multimedia Video Speech Process.*, 2004, pp. 747–750.
- [15] X. Shi, W. Zhao, and Y. Shen, “Automatic license plate recognition system based on color image processing,” in *Lecture Notes Comput. Sci.*, vol. 3483, pp. 1159–1168, 2005.
- [16] E. R. Lee, P. K. Kim, and H. J. Kim, “Automatic recognition of a car license plate using color image processing,” in *Proc. IEEE Int. Conf. Image Process.*, vol. 2. Nov. 1994, pp. 301–305.
- [17] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, “Automatic license plate recognition,” in *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 1, pp. 42–53, Mar. 2004.
- [18] S. K. Kim, D. W. Kim, and H. J. Kim, “A recognition of vehicle license plate using a genetic algorithm based segmentation,” in *Proc. Int. Conf. Image Process.*, vol. 2. 1996, pp. 661–664.

- [19] S. Yohimori, Y. Mitsukura, M. Fukumi, N. Akamatsu, and N. Pedrycz, "License plate detection system by using threshold function and improved template matching method," in *Proc. IEEE Annu. Meeting Fuzzy Inform.*, vol. 1. Jun. 2004, pp. 357–362.
- [20] J. Matas and K. Zimmermann, "Unconstrained license plate and text localization and recognition," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Sep. 2005, pp. 572–577.
- [21] S. Draghici, "A neural network based artificial vision system for license plate recognition," in *Int. J. Neural Syst.*, vol. 8, no. 1, pp. 113–126, 1997.
- [22] F. Alegria and P. S. Girao, "Vehicle plate recognition for wireless traffic control and law enforcement system," in *Proc. IEEE Int. Conf. Ind. Tech.*, Dec. 2006, pp. 1800–1804.
- [23] B. K. Cho, S. H. Ryu, D. R. Shin, and J. I. Jung, "License plate extraction method for identification of vehicle violations at a railway level crossing," in *Int. J. Automot. Tech.*, vol. 12, no. 2, pp. 281–289, 2011.
- [24] W. T. Ho, H. W. Lim, Y. H. Tay, and Q. Binh, "Two-stage license plate detection using gentle Adaboost and SIFT-SVM," in *Proc. 1st Asian Conf. Intell. Inform. Database Syst.*, 2009, pp. 109–114.
- [25] K. K. Kim, K. Kim, J. Kim, and H. J. Kim, "Learning-based approach for license plate recognition," *IEEE Signal Processing Society Workshop*, vol. 2, pp. 614–623, 2000.
- [26] K. Miyamoto, K. Nagano, M. Tamagawa, I. Fujita, and M. Yamamoto, "Vehicle license-plate recognition by image analysis," in *Proc. Int. Conf. Ind. Electron. Control Instrum.*, vol. 3. 1991, pp. 1734–1738.
- [27] T. Nukano, M. Fukumi, and M. Khalid, "Vehicle license plate character recognition by neural networks," in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst.*, 2004, pp. 771–775.
- [28] V. Shapiro and G. Gluhchev, "Multinational license plate recognition system: Segmentation and classification," in *Proc. Int. Conf. Pattern Recognit.*, vol. 4. 2004, pp. 352–355.

- [29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009, pp. 248–255.
- [30] R. Girshick, F. Iandola, T. Darrell, J. Malik, "Deformable Part Models are Convolutional Neural Networks," arXiv:1409.5403, 2014. in *CVPR*, 2015.
- [31] D. Lowe, "Distinctive image features from scale-invariant keypoints." in *IJCV*, 60 (2), pp. 91-110, 2004.
- [32] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14)*, pp.580-587, 2014
- [33] R. Girshick, "Fast R-CNN," in *ICCV*, 2015.
- [34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.
- [35] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask, "R-CNN," arXiv:1703.06870, 2017.
- [36] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. "Ssd: Single shot multibox detector," in *European Conference on Computer Vision*, pages 21–37. Springer, 2016.
- [37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," arXiv preprint arXiv:1506.02640, 2015.
- [38] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005.
- [39] W.-C. Liu, C.-H. Lin, A Hierarchical License Plate Recognition System Using Supervised K-means and Support Vector Machine, in *Proc. IEEE ICASI 2017*, Sapporo, Japan, May 13-17, 2017.
- [40] J. Redmon. Darknet: Open source neural networks in c. <http://pjreddie.com/darknet/>,

2013-2016.



## 自傳

林永鑫，1994 年出生於臺北市。

- 新北市秀朗國小
- 臺北市私立東山高級中學 國中部
- 臺北市私立東山高級中學
- 國立臺灣師範大學電機工程學系
- 國立臺灣師範大學電機工程學系研究所

## 學術成就

- Cheng-Hung Lin, Yong-Sin Lin and Wei-Chen Liu, " An efficient license plate recognition system using convolution neural networks", in Proc. of IEEE International Conference on Applied System Innovation (IEEE ICASI 2018), Chiba, Japan, 13-17 April, 2018.