

國立勤益科技大

DT  
448.6  
6081  
100  
225084

電子工程系研究所碩士班

碩士論文

基於樣板比對之車牌自動辨識系統

Automatic Vehicle License Plate  
Recognition System Based on Template  
Matching

研究生：羅智群

Student: Zhi-Chun Luo

指導教授：劉正忠

Advisor: Chen-Chung Liu

國立勤益科技大學圖書館



225084

中華民國 一 百 年 六 月

# 國家圖書館

## 博碩士論文電子檔案上網授權書

本授權書所授權之論文為授權人在國立勤益科技大學電子工程系  
100 學年度第 2 學期取得碩士學位之論文。

論文題目：基於樣板比對之車牌自動辨識系統  
指導教授：劉正忠

茲同意將授權人擁有著作權之上列論文全文（含摘要），非專屬、  
無償授權國家圖書館，不限地域、時間與次數，以微縮、光碟或其  
他各種數位化方式將上列論文重製，並得將數位化之上列論文及論  
文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢  
索、閱覽、下載或列印。

上列論文為授權人向經濟部智慧財產局申請專利之附件或相關文件之一（專利  
申請案號：\_\_\_\_\_），請於 \_\_\_\_\_ 年 \_\_\_\_\_ 月 \_\_\_\_\_ 日後再將上列論  
文公開或上載網路。

因上列論文尚未正式對外發表，請於 103 年 7 月 6 日後再將上列論文公開  
或上載網路。

授權人：羅智群

親筆簽名及蓋章：羅智群

民國 100 年 7 月 6 日

電話：\_\_\_\_\_ 傳真：\_\_\_\_\_  
聯絡地址：嘉義縣番路鄉下坑村菜公店108之24之7號  
E-Mail：monkeymk2001@yahoo.com.tw

國立勤益科技大學  
研究所碩士班  
論文口試委員會審定書

本校電子工程系碩士班 羅智群 君

所提論文 基於樣板比對之自動車牌辨識系統

合於碩士資格水準，業經本委員會評審認可。

論文口試委員會：

召集人：

委員：

喻石生

喻石生

喻石生

劉正峯

劉正峯

指導教授：

所長：

電子工程系主任 林熊徵

中華民國一百 年 六月

# 基於樣板比對之車牌自動辨識系統

研究生:羅智群 指導教授:劉正忠

國立勤益科技大學電子工程系研究所碩士班

## 中文摘要

為實現智慧交通系統之理想，自動車牌辨識系統絕對是關鍵的一環，然而自動車牌辨識系統包涵了車牌定位、車牌字元切割和車牌字元辨識三大部份。此三項功能必須依序進行，完成定位後方能進行字元的切割，將車牌上的各個字元，切割成數張的獨立字元影像，方能進行之後的辨識。車牌定位之目的為在影像中，尋找可能的車牌位置，本論文提出之車牌定位演算法主要的步驟有色彩空間轉換、適應性背景更新、去除雜訊、框取車輛、截取輪廓、邊緣細化、刪除連續邊、框取車牌以及調整車牌框架大小等步驟；而字元切割則是對車牌定位後所得的影像進行切割，將車牌中的每個字元獨立出來，本篇論文利用了霍夫轉換法，來找出車牌邊緣的直線，利用此條直線的斜率來進行變形車牌的矯正，並且針對車牌影像計算邊緣梯度圖，利用此圖來進行車牌範圍的收縮，使之範圍收縮至僅剩字元的部份，取得僅剩的字元區塊後，即可利用投影法來進行字元的切割。取得個別字元

的影像後，就是進行最後的字元辨識，本篇論文採用樣版比對的方法來進行辨識，並加以改進，字元資料庫共有 175 張影像，分別為 10 個阿拉伯數字及 25 個英文字母 (字母 “O” 除外) 各 5 張影像，且皆已正規化大小為  $20*40 \text{ pixel}^2$ 。樣版比對時為計算兩張圖的差異點，然後再根據差異點的多寡，給予對應的分數，每五張相同數字或字母的分數會進行加總，因此總共會有 35 個分數。此分數最高者即為判定之字元。分數計算完成後，則找出評分最高的一組，作為此張輸入字元影像的辨識結果，本論文僅使用了極低的樣本數，但最後的辨識率亦可達九成五以上。降低了樣本數在程式設計的觀點上，不僅僅只是降低了記憶體耗用，亦直接的減少了執行的迴圈數，增加了執行的效能，表示著我們可以使用更少的資源，更快的速率，達成極高的辨識效果。

# **Automatic Vehicle License Plate Recognition System Based on Template Matching**

**Student: Zhi-Chun Luo, Advisor: Dr. Chen-Chung Liu**

Department of Electronic Engineering  
National Chin-Yi University of Technology, Taiwan

## **Abstract**

Nowadays, vehicle license plate (VLP) recognition system has become a key to lots of traffic related applications, such as road traffic monitoring, traffic analysis, and parking lots access control, etc. Accurately detecting the VLP from a vehicle image, extracting the VLP number from the detected VLPs, and recognizing the extracted VLP number are the main stages of vehicle license plate recognition (VLPR) system. They greatly control the overall recognition accuracy and processing speed of the whole system. The main aim of this research is to find a high effective VLP recognition system. The functions of this system consists of locating the multiple VLPs of moving vehicles from a video traffic image sequence, extracting VLP number from detected VLP, and recognizing the extracted VLP number. Main steps in the VLP locating stage are color space transform, adaptive background updating, noise removal, vehicle frame extraction, edge detecting and thinning,

long edge deleting, VLP segmentation, and VLP normalization steps. In the VLP number segmentation stage for an extracted VLP, the proposed algorithm adopts the Sobel operator to obtain the edge map of the VLP, uses Hough transform to obtain the longest straight line segment as the horizontal reference line to calibrate the pose of the extracted VLP, utilizes horizontal boundaries and vertical boundaries detections to trim the VLP region, and employs projection scheme to extract the characters in trimmed VLP region. In the character recognition stage, the VLP character image database is first constructed with 175 normalized character images (10 Arabic numbers and 25 English characters (A ~ Z, except character "O"), and 5 images for each character). The extracted character is normalized into a standard size and is then compared to each element of the character image database. Each comparison will obtain its score according the difference between the extracted character and a database element so that a character has 5 scores. The 5 scores of each character are then summed. The extracted VLP character is identified as the character whose score sum is the maximum of all sore sums and is larger than the threshold defined empirically. The experiment results show that the presented algorithm can (i) correctly localize the VLPs even in overlapped vehicles situation, (ii) effectively extract the VLP number even from a distorted VLP caused by the shifting of relative position between the vehicle and the camera, and (iii) accurately identify the extracted VLP numbers.

## 致謝

回想著當初推甄勤益電子所的一段離奇的過程，彷彿就只是昨天才發生的事，轉瞬間已經兩年，在目前的人生規劃中，全心全意學習的階段，差不多也是到此結束，說到此應該先感謝我的家人，我的爸爸、媽媽還有弟弟，全力的支持我，讓我能毫無後顧之憂的準備學業。在往後開始的日子，雖說仍要不斷的充實自己，但屆時已經不如現在，種種的壓力必須要開始去面對。但至少很慶幸的是，研究所這兩年的期間我學習到的真的很多，不僅僅是專業上的知識，更重的是研究的方法甚至態度，這當然是要感謝指導教授 劉正忠 副教授無私、毫無保留的教誨。在就讀研究所這兩年的期間，實驗室的幾位學長，張智欽、邱奕禮、胡冠男、詹殷宗，皆對學生的研究上有著不小的建議與指教，更於日常的某些細節上，悉心的提醒。還有兩年期間秦璿祐及鐘培中同學在這兩年的期間，不斷的相互討論、提醒及勉勵。最後再回想從小到大，學生這樣一路走來，都還算平順，雖然在大學期間曾經的迷惘過，但至今卻也還算幸運的，在畢業後找到一份自己理想中的工作；到如今學生能有如此的境遇，不禁要感謝學生國中時間的導師 陳彩蔭 老師，學生的性格養成，絕對都要完全歸功於陳老師。說到這就像“陳之藩-謝天”一文中所說的：『無論什麼事情，得之於人者太多，出之於己者太少，因為需要感謝的人太多了，就感謝天



吧!』感謝那些曾經幫我禱告過的所有弟兄姐妹，更感謝主保守我，  
從小至今的一家平安，更願在未知的將來一樣喜樂。



# 目錄

中文摘要 .....	III
ABSTRACT .....	V
致謝 .....	VII
目錄 .....	IX
圖目錄 .....	XI
表目錄 .....	XIV
<b>第一章 緒論 .....</b>	<b>1</b>
1.1. 研究背景與動機 .....	1
1.2. 研究目的 .....	2
1.3. 論文系統架構 .....	3
<b>第二章 文獻探討 .....</b>	<b>4</b>
2.1. 車牌定位 .....	4
2.2. 車牌字元切割 .....	6
2.3. 車牌字元辨識 .....	7
<b>第三章 相關原理介紹 .....</b>	<b>9</b>
3.1. 常用色彩空間 - RGB、HSI、YIQ .....	9
3.1.1. RGB .....	10
3.1.2. HSI .....	11
3.1.3. YIQ .....	14
3.2. 邊緣偵測 .....	14
3.3. 霍夫轉換 (HOUGH TRANSFORM) .....	15
<b>第四章 車牌辨識系統 .....</b>	<b>16</b>
4.1. 車牌定位 .....	16
4.1.1. RGB 轉 HSI 色彩空間 .....	17
4.1.2. 適應性背景更新 .....	18
4.1.3. 框取車輛 .....	21
4.1.4. 輪廓擷取 .....	23
4.1.5. 輪廓細線化 .....	25
4.1.6. 刪除連續邊 .....	29
4.1.7. 框取車牌 .....	30

4.1.8. 調整車牌框架大小 .....	31
4.2. 字元切割 .....	32
4.2.1. 色彩空間轉換及像素值調整 .....	33
4.2.2. 車牌傾斜校正 .....	34
4.2.3. 範圍收縮 .....	38
4.2.4. 字元切割 .....	41
4.3. 車牌字元辨識 .....	48
<b>第五章 實驗結果 .....</b>	<b>53</b>
<b>第六章 結論與未來展望 .....</b>	<b>63</b>
<b>參考文獻 .....</b>	<b>65</b>
<b>作者簡介 .....</b>	<b>73</b>



# 圖目錄

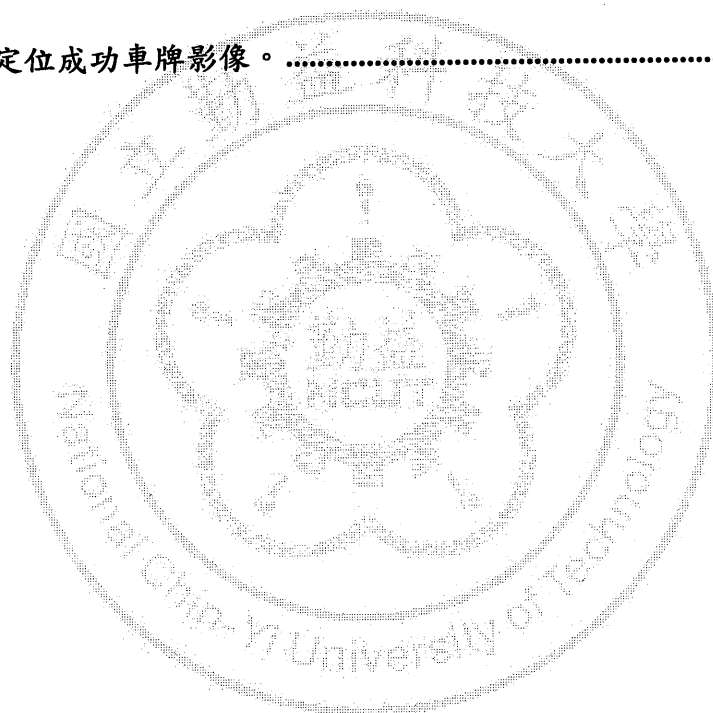
圖 3-1：色彩相加原理示意圖 .....	11
圖 3-2：HSI 色彩模型 .....	13
圖 3-3：色調表示方法示意圖 .....	13
圖 4-1：車牌定位演算法的步驟圖 .....	17
圖 4-2：HSI 色彩空間模型 .....	18
圖 4-3：當前影像 .....	19
圖 4-4：背景影像 .....	19
圖 4-5：物件影像 .....	19
圖 4-6：8x8 等分區塊的物件影像 .....	21
圖 4-7：3x2 遮罩 .....	22
圖 4-8：物件樹的建立 .....	22
圖 4-9：框選車輛框架 .....	23
圖 4-10：邊緣輪廓影像 .....	25
圖 4-11：輪廓像素(P)與其鄰接點 .....	26
圖 4-12：Hilditch 細線化範例 .....	27
圖 4-13：ZS 細線化前後比較圖 .....	28
圖 4-14：刪除連續編過後影像 .....	29

圖 4-15：車牌框架在車輛框架內走訪情況.....	30
圖 4-16：框取重疊車輛之車牌.....	31
圖 4-17：台灣實際車牌影像.....	32
圖 4-18：字元切割工作流程圖。.....	33
圖 4-19：色彩空間轉換結果圖.....	34
圖 4-20：傾斜車牌範例。.....	35
圖 4-21：傾斜車牌未經傾斜校正之字元切割結果.....	35
圖 4-22：傾斜車牌經傾斜校正後之字元切割結果.....	36
圖 4-23：灰階影像之 Canny 邊緣梯度圖.....	37
圖 4-24：經傾斜校正之結果圖.....	37
圖 4-25：另一張車牌經傾斜校正後之結果圖.....	39
圖 4-26：範圍收縮方法示意圖。.....	39
圖 4-27：假設之最極端狀況之車牌。.....	39
圖 4-28：範圍收縮之結果範例.....	40
圖 4-29：部份車體雜訊無法去除範例.....	41
圖 4-30：垂直投影範例.....	42
圖 4-31：垂直投影量放大圖.....	42
圖 4-32：較不精確之車牌定位範例.....	46
圖 4-33：圖 4-32 之車牌影像完成範圍收縮的結果.....	46

圖 4-34：因車體雜訊造成字元切割失敗範例.....	46
圖 4-35：字元切割工作流程圖.....	47
圖 4-36：車牌字元辨識流程圖.....	49
圖 4-37：車牌字元之假設影像.....	50
圖 4-38：圖 4-37 之(a)、(b)差異點示意圖。.....	50
圖 4-39：字元 ”1” 與 ”I” 特徵比較方法；(a)數字”1”；(b)字母”I”。.....	51
圖 5-1：車牌定位圖形使用者界面.....	54
圖 5-2：車牌字元切割及辨識圖形使用者界面.....	54
圖 5-3：成功車牌定位範例 1.....	55
圖 5-4：成功車牌定位範例 2.....	56
圖 5-5：成功定位多部車輛範例.....	57
圖 5-6：部份車牌字元辨識範例.....	61
圖 5-7：部份車牌字元辨識範例.....	62

# 表目錄

表 4-1：各步驟結果表格.....	42
表 4-2：車牌號碼元素一覽表.....	43
表 4-3：輸入車牌影像遠近比較表.....	44
表 5-1：實驗環境列表.....	53
表 5-2：部份定位成功車牌影像。.....	58



# 第一章 緒論

## 1.1. 研究背景與動機

自福特首先利用了管線 ( Pipeline ) 作業生產汽車,使得相同的人力資源狀況下,得以大幅的提升了汽車產量,再伴隨著工商業急遽發展經濟起飛,造就了現今汽車如此普及的局面;車輛的數量與流量迅速增加,社會秩序也隨之惡化,因此車輛管理問題亦日趨重要,此時若依舊由人工作業來管理交通工具,勢必管理上會漏洞百出,因此智慧交通系統 ( Intelligence Transportation System ) 的發展絕對是必然的,而且亦是邁向智慧生活環境 ( Intelligence Life Environment ) 之目標,極為重要的一環。然而智慧交通系統中最為關鍵的,絕對不可或缺的功能為自動車牌辨識系統 ( Automatic License Plate Recognition System )。舉凡停車管理系統、開口出入管制系統等等,皆必須先辨識出目標的車輛號碼,進而才能繼續後續的管理動作。基於此需求新的車輛自動車牌辨識技術不斷的被提出來。而在車輛自動車牌辨識技術中,針對監視設備所記錄下來的動態影像來進行辨識已成為主流。因為此一技術對於警方人員在管理監控車輛上,有著莫大的幫助,可以藉由調閱犯罪現場之監視影片來尋找可能的嫌疑車輛,假使使用人力檢查監視影像不但耗時費神,也很容易拖垮調閱者之精



神而大大降低車輛追蹤的效率，因此自動車牌辨識系統對於治安的改善亦有著莫大的幫助。智慧交通系統在不需人力的操作下能自動地監控車輛，所以近年來，車牌辨識系統已成為智慧交通系統的主要發展方向。

## 1.2. 研究目的

自動車牌辨識系統已成為在智慧交通系統中最重要的一個部份。在眾多現代交通監視方法中，自動車牌辨識系統扮演一個核心的腳色，它可用在區域安全的控管上來節省時間以及減少交通的繁忙，用在停車場系統上也能節省車輛管理的人力和金錢，達到監視區域的安全控制、協助交通執法。為達全自動智慧交通系統之終極目標，在完全無需人力操作，僅於必要時輸入目標之車輛號碼，即可尋找出車輛；為實現此一目標，首先必須開發出一個適應性極廣，同時辨識率又極高的自動車牌辨識系統。本論文之自動車牌辨識系統，針對動態的監視影像，即時的定位影像中的車牌位置，並且截取車牌區塊的影像，緊接著將車牌影像中的車牌字元，逐個分割成個別的獨立影像，取得個別的車牌字元影像後，即可針對個個字元進行辨識。

### 1.3. 論文系統架構

本論文分成六個章節，第二章文獻探討，介紹截至目前為止車牌辨識的相關研究；第三章相關原理介紹，某些在專業上被視為常識 (Common Sense) 的方法、原理、技術，在第三章會先行簡述說明之，以利於非從事相關研究的人士，能更順利的閱讀本文章；第四章為本論文所提出的自動車牌辨識系統；第五章實驗結果；最後第六章為結論。



## 第二章 文獻探討

為實現智慧交通系統之理想，自動車牌辨識系統絕對是關鍵的一環，至今已有許多相關的研究被發表，每篇相關文獻著重的部份皆不盡相同，但都不外乎三大項目，車牌定位、車牌字元切割和車牌字元辨識。此三項功能必須依序進行，完成定位後方能進行字元的切割，將車牌上的各個字元，切割成數張的獨立字元影像，方能進行之後的辨識。本章將依序針對此三大功能來進行探討。

### 2.1. 車牌定位

車牌定位之目的，為框選影像中可能的車牌區域，針對車牌定位已有許多的方法被提出[01~16]，最直覺的方法是利用車牌的特徵條件來定位，所謂的車牌特徵包括了車牌的結構、顏色、紋理、邊界點、外觀比例等等，但此些特徵皆會受限於某種特定的條件下，比如說外觀的比例，會因拍攝器材和車輛之間夾角的改變，而使得成像中的車牌外觀變形；另外顏色亦會因為迎光或背光的情況下，而有所差異。G. H. Ming 等人[14]提出利用邊界特徵 (edge feature)，配合車牌區域擁有高密度之特性之方法，並且自動忽略了影像中密度最高及最低的區域，如此方可縮小搜索的範圍，減少搜尋的時間，但同時也可能發

生忽略的區域中包含了車牌區塊，而造成定位的失敗；另外 D. Zheng 等人[12]不僅使用了邊界特徵和密度特性，他們還提出了影像強化的方法，改善影像中的因光線造成的問題，雖然稍微提升了準確度，卻亦增加了額外的運算負擔。

再者由於各個國家的車牌編號規則不盡相同，使得固定的特徵值只能適用於固定的區域範圍，超出範圍外，車牌的特徵改變，會使得整體的成功率大幅的下降。因此使用要使得車牌辨識能適應於各種不同車牌特徵，勢必要讓辨識系統本身具有學習的能力，因此有許多學者提出了利用類神經網路，再配合模糊理論來完成一個具自我學習能力的車牌辨識系統。S. H. Park 等人[13]即是利用類神經網路來實現車牌辨識系統，他們將影像由 RGB 轉換至 HIS 色彩空間，接著將影像分成許多小區塊，最後再交由類神經網路判斷每一小方塊是否為車牌的一小部份。另外 N. Zimic 等人[15]、Y. A. Syed 等人[05]及 S. L. Chang 等人[17]亦有使用了類神經網路，配合模糊理論，來完成車牌的定位，N. Zimic 等人作法為定義明亮與黑暗的變化程度，用來定位出屬於車牌的像素；而 Y. A. Syed 等人[05]與 S. L. Chang 等人[17]則定義彩色邊界的模糊集合。然而類神經網路的適應性高，但卻需要極大量的運算，與事先的訓練。還有一做法為透過直線的掃瞄，並且配合車牌邊緣密度高的特性，最後藉由組合各條掃瞄線，邊緣變換最高

的地方來進行定位。

邊界特徵對於車牌定位絕對是一相當重要的特徵，本文是利用像素密度計算來定位出車牌的位置，用來框取車牌的車牌框架，再依照比例算出車牌框架寬度，並且事先透過一些前處理的方法，以增加定位時的速率與準確率的提升。

## 2.2. 車牌字元切割

再完成定位取得車牌影像後，並無法針對此影像直接進行辨識，以台灣國內車牌為例，由總長六碼的英文字母與阿拉伯數字所組成，而且還有分別英文字母在阿拉伯數字之前或之後，如此龐大的資料庫，實在無法直接進行辨識。因此必須進行字元的切割，所定位所得的車牌影像，逐一切割出每個車牌號碼字元，如此辨識的樣本資料庫可得到極大的簡化。一般最常見的方法是使用投影法[18~23]，首先對車牌影像進行二值化運算[24~26]，再將二值化後的影像正交投影至橫軸和縱軸，然後再利用車牌號碼與號碼之間，並不會相連會留有間隙的特性，即可切割出每個車牌字元。此述所使用的投影法，皆為正交投影，因此倘使車牌定位的成像，因當初拍攝時的角度問題，造成車牌影像的傾斜，會使得利用投影法切割時，遭遇到極大的困難，因此本篇論文使用 Hough Transform [27~30] 來找出影像中最長的直

線，以此直線當成校準線，將傾斜的車牌矯正，以適應之後投影法的切割。

### 2.3. 車牌字元辨識

相較於手寫辨識，車牌字元是從影像中所截取的部份區塊，缺少了字元筆劃步驟的資訊，但由於車牌的字元皆為統一印刷的字體，有固定的樣式與大小，因此辨識上會較為簡單，但是切割所得的車牌字元常會因為拍攝角度的原因，而造成車牌字元的歪斜或扭曲；亦或因為拍攝距離遠近的關係，而造成解析度不適應。上述的問題皆會造成辨識的成功率下降，許多研究提出了各自的辨識系統，有些著重於辨識速率的提升，有些則著重於辨識率的增加；這些辨識系統常用的辨識方法大概有兩類，類神經網路[17, 31~33]和樣版比對[34~36]，另外還有小部份的論文使用字元結構特徵來進行辨識[37]。在 P. Comelli[36]等人的研究中主要為辨識羅馬地區的車牌，P. Comelli 所採用的辨識方法為樣版比對，在 P. Comelli 論文中提及由於羅馬地區的車牌前兩字元代表州郡，為固定的字元組合，第三個字元開始方是車牌編號，因此 P. Comelli 建立了兩組樣版，一組為所有州郡代字元的集合，另一組則是所有可能的車牌字元組合。上述此種透過預知的編號排列方式來進行分類，此種方法被稱為先驗知識 (Priori

Knowledge) ，使用先驗知識的方法來提升整體的辨識率，效果往往會因車輛的增加導致車牌號碼的不足，而使得編號的規則破例，最後讓整個車牌辨識系統必須重新修改。雖然此種方法的適應性低，但其觀念卻被許多其它車牌辨識相關的研究所延伸應用，藉以改善辨識的效能，特別是某些外形相近的字元，如”0”與”D”、”1”與”I”、”2”與”Z”、”8”與”B”……，此些類似的字元在許多的車牌辨識中往往會遭遇到困難，因此在[33~35, 38]等研究皆使用先驗知識的方法來解決類似字元的辨識。

另外類神經網路具有非線性，抗雜訊及應適性等優點，因此近年隨著個人電腦運算能力的提升，使得許多的研究者採用類神經網路來進行辨識[17, 31~33, 39~43]，如 M. H. T. Brugge [42]等人使用離散細胞元類神經網路 (DTCNN) ，並藉由車牌編號規則的輔助，辨識率可達 98.51%，還有魏銷志 [45] 利用 SimNet 進行辨識，對於解析度僅有 640\*480 的影像亦可達 91.77% 的辨識成功率。

## 第三章 相關原理介紹

本章將會介紹在此論文中，會使用到的一些重要基本原理，如本篇論文所採用到的 RGB、HSI、YIQ 色彩空間，另外還有在字牌定位與車牌字元切割時皆會用到的邊緣偵測計算方法，還有霍夫轉換 (Hough Transform) 用來校正傾斜的車牌。這些原理方法已相當的成熟，並且廣泛的被應用，稍後的研究當中，亦會直接的使用這些方法，是故在此章中，預先概要式的說明這些方法其原理為何，以利於閱讀稍後章節時，更能把握各個步驟之關鍵與重點。

### 3.1. 常用色彩空間 - RGB、HSI、YIQ

在影像處理的領域上，對於色彩空間的了解，絕對是不可或缺的，而且可以提升影像處理的效率。色彩空間定義一種具體的描述方式，通常會是一組固定位元長度的數值，用以表示某種顏色。以 RGB 色彩空間為例，通常會將紅色 (R channel)、綠色 (G channel) 和藍色 (B channel) 各以一個位元組的大小來描述之，組合後方可顯示黃色、紫色等等不同的顏色。以下便會介紹此篇論文中使用到的 RGB、HSI 和 YIQ 三種色彩空間。



### 3.1.1. RGB

RGB 即為 Red, Green, Blue 的縮寫，顧名思義此一色彩空間利用色彩相加原理，透過混合紅綠藍三種顏色，來表示其它各種顏色，以圖 3-1 為例，假使我們定義 R 為  $100_B$ ，G 為  $010_B$ ，B 為  $001_B$ ，表示我們僅用三個位元，來描述部份的顏色，其它可以表示的顏色即如圖 3-1 所示，此時黃色(RG)即為  $110_B$ ，淺藍色(GB)即為  $011_B$ ，紫色(BR)即為  $101_B$ ，白色(RGB)即為  $111_B$ ，而黑色(RGB')即為  $000_B$ ，因此我們假設用三位元長度的 RGB 色彩空間，能表示的顏色，即為圖 3-1 所示的八種可能。然而在色彩學上，顏色不是唯一的屬性，另外還有對比度和亮度的調整，在 RGB 的色彩空間一樣能對這些屬性進行調整，對比度的調整必須讓 RGB 平衡往某一原色邁進；而亮度的調整，則是保持紅綠藍三種顏色間的相對比例，增加數值來增強亮度，反之減少數值即為調暗。

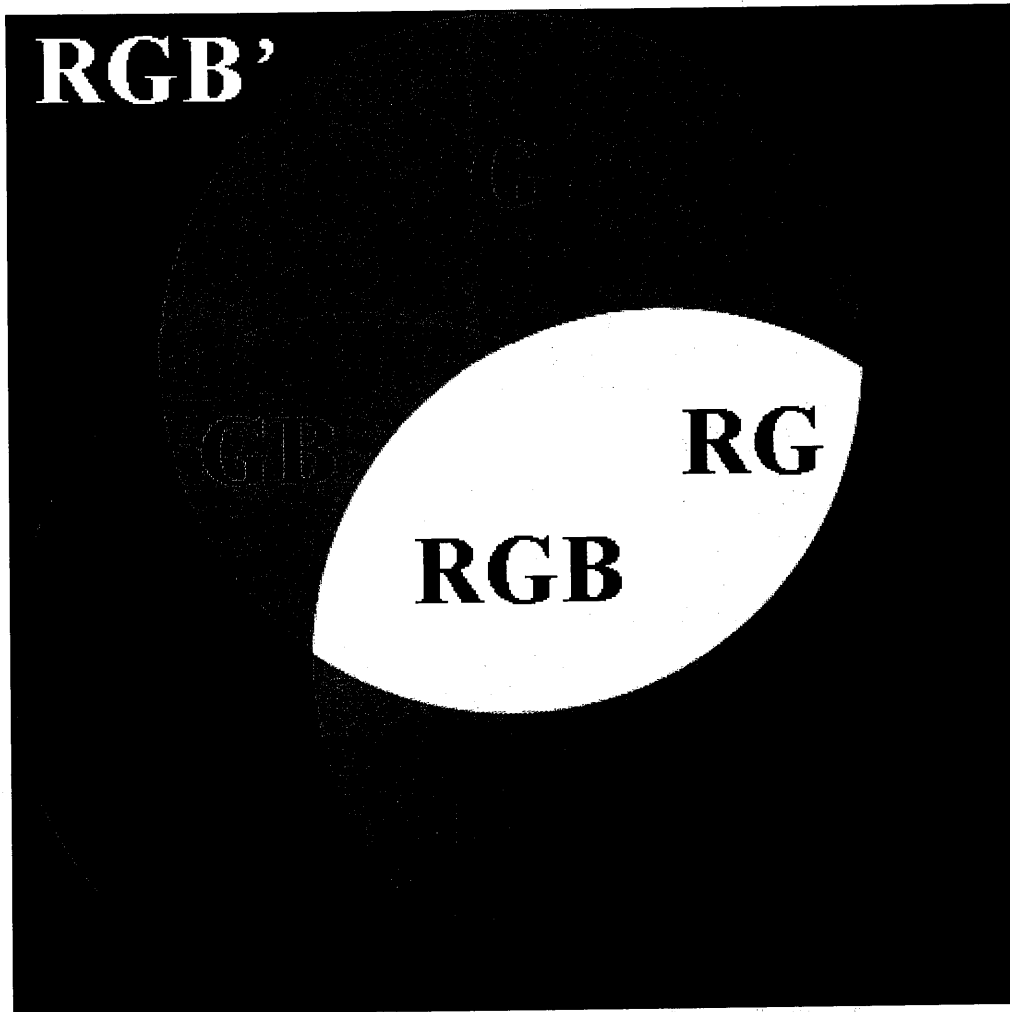


圖 3-1：色彩相加原理示意圖。

### 3.1.2. HSI

HSI 色彩空間用色調 (Hue)、飽和度 (Saturation) 和亮度 (Intensity) 來描述色彩，其模型是按著人類的視覺系統來設計，如圖 3-2 所示。我們可以看到 HIS 色彩空間是用一個圓錐空間模型來描述，這模型雖然複雜，但卻能把色調、飽和度和亮度的變化表現得十分清楚。首先色調即為圓錐上同一亮度 (Intensity) 的橫切面，其表示方法為角度值，介於 0~360 度之間，如圖 3-3 所示。接著是飽和度，

亦稱彩度，指的是色彩的純度，色彩中含某單一顏色成分的多少，其表示範圍為 0~1，飽和度的調整，即變動主要顏色與其它顏色之間的比率。最後亮度的調整，其概念和 RGB 色彩空間上的做法是一樣的，保持紅綠藍三種顏色間的相對比例，變動數值來增加減少明亮度，亦即圖 3-2 中垂直軸的數值變化，表示的範圍亦為 0~1，但模型為何會設計為圓錐型，而不是圓柱型，此是因為要配合人的視覺系統，因為人的視覺對於亮度的敏感度，遠高於對顏色濃淡的變化的敏感度，亦即眼睛感覺物體變亮的時，同時會覺得覺體的顏色變淡，因此才設計為圓錐型。在影像處理中常採用 HSI 色彩空間，因為它可以分開處理各個屬性，而且是分別獨立的，HSI 和 RGB 兩個色彩空間，其實是相同的物理量，只是表示法不同而已，兩者之間存在著轉換關係，在稍後系統設計時，將會說明兩者如何互相轉換。

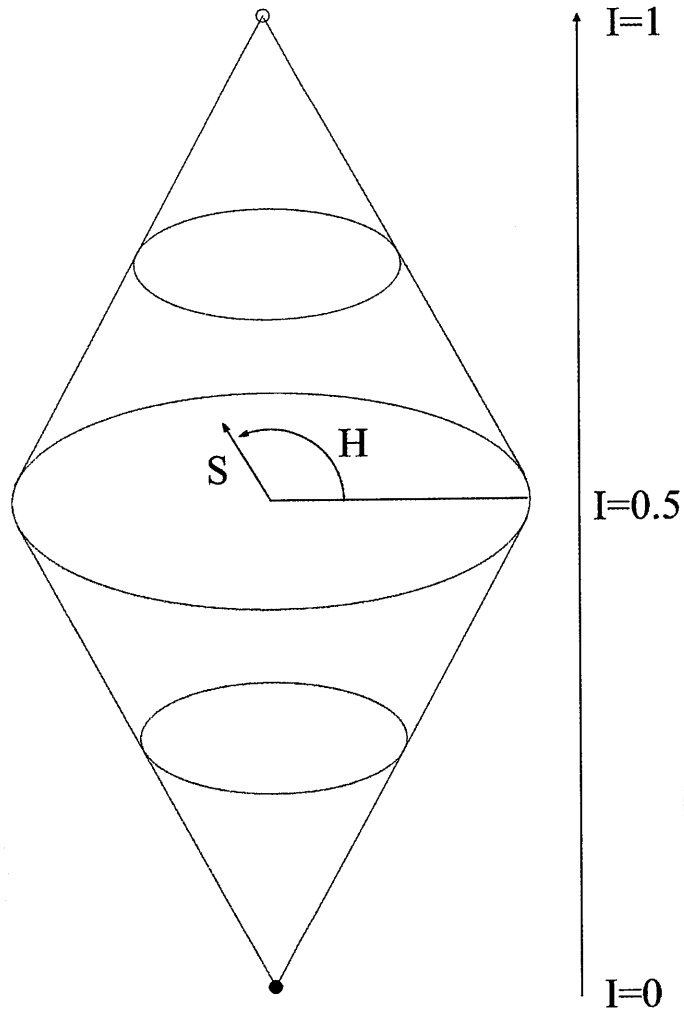


圖 3-2 : HSI 色彩模型

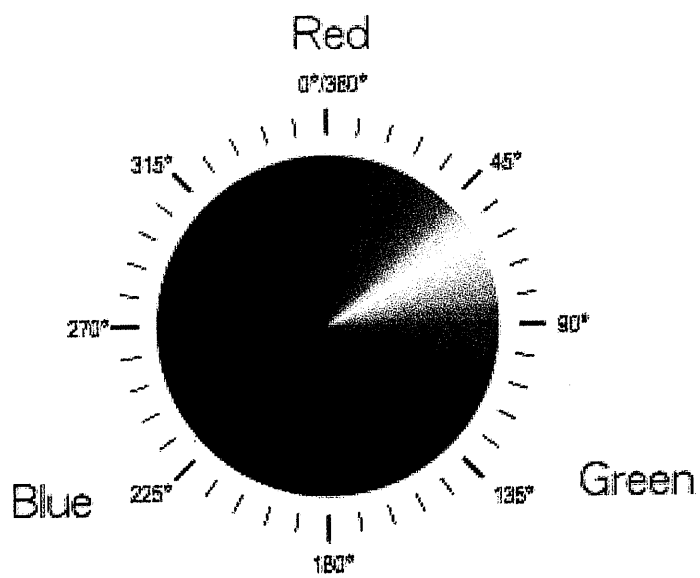


圖 3-3 : 色調表示方法示意圖。

### 3.1.3. YIQ

YIQ 色彩空間常見於北美的電視系統，屬於 NTSC (National Television Standards Committee) 系統。Y 是指顏色的明亮度 (Brightness)。其實 Y 就是圖像的灰度值 (Gray value)，而 I 和 Q 則是指色度 (Chrominance)，即描述圖像色調及飽和度的屬性。在 YIQ 系統中，Y 分量代表圖像的亮度信息，I、Q 兩分量則是攜帶顏色信息，I 分量代表從橙色到青色的顏色變化，而 Q 分量則代表從紫色到黃綠色的顏色變化。將彩色圖像從 RGB 轉換到 YIQ 色彩空間，可以把彩色圖像中的亮度信息與色度信息分開，分別獨立進行處理。同樣 RGB 和 YIQ 兩者之間亦存在著轉換關係，將一併在稍後系統設計時介紹之。

## 3.2. 邊緣偵測

影像中兩區域之間的分隔線即為邊緣，因為在同一個區塊中，鄰近週圍的點，其像素值相近，不會有明顯大幅度的變化，但當跨越到其它的區塊後，兩區塊間可能於色調、明亮度或彩度，三個屬性中任何一個或以上，起了明顯的變化，因而產生了兩個不同的區塊，區塊之間勢必會產生了邊緣。而邊緣的偵測方法研究，迄今已發展多年，且已趨近於成熟，常見的方法有 Sobel、Prewitt、Rebert、Laplace、

Canny...等。

### 3.3. 霍夫轉換 (Hough transform)

Hough transform 在 1962 被提出，為影像處理中識別幾何形狀的一種方法，霍夫轉換在識別圖形時，不會受到圖形旋轉的影響。最簡單的霍夫轉換是在影像中識別直線，一般常用斜率式來表示直線。斜率式如公式(3-1)所示，其中  $m$  為斜率，而  $k$  值為  $y$  軸的位移值。

$$k = y - mx \dots\dots\dots(3-1)$$

這表示參數平面( $m-k$ )中的一條直線。因此，圖像中的一個點對應參數平面中的一條直線，圖像中的一條直線對應參數平面中的一個點。對圖像上所有的點作霍夫變換，最終所要檢測的直線對應的一定是參數平面中直線相交最多的那個點。這樣就在圖像中檢測出了直線。基於霍夫轉換的改進方法還有許多，如圓、圓弧、橢圓、矩形等等的偵測，其中還另有一個重要的方法“廣義霍夫轉換”，可以用來檢測任意形狀。

## 第四章 車牌辨識系統

車牌辨識系統通常會被分為三大主要部份，其分別為車牌定位、字元切割和字元辨識。車牌定位之目的為在影像中，尋找可能的車牌位置；而字元切割則是負責對車牌定位後所得的影像進行切割，將車牌中的每個字元獨立出來；取得各別字元的影像後，就是進行最後的字元辨識了。本章節將會依序介紹此三大部份的詳細作法及步驟。

### 4.1. 車牌定位

首先車牌定位 (License Plate Location, LPL) 為字元辨識前最重要的部份，因為這關係到整個字元辨識速度以及之後的辨識正確率，假使車牌定位失敗，後續的車牌字元切割和車牌字元辨識將無從做起。本論文提出之車牌定位演算法的步驟圖如圖 4-1，可看到圖中共有八個步驟；步驟一“G&B”為色彩空間轉換、步驟二“BR”背景雜訊去除、步驟三“VF”為框取車輛、步驟四“AT”為截取輪廓、步驟五“ET”為邊緣細化、步驟六“SED”為刪除連續邊、步驟七“VLF”為框取車牌，最後步驟八“ALF”為調整車牌框架大小；稍後章節中會詳細介紹此八個步驟。

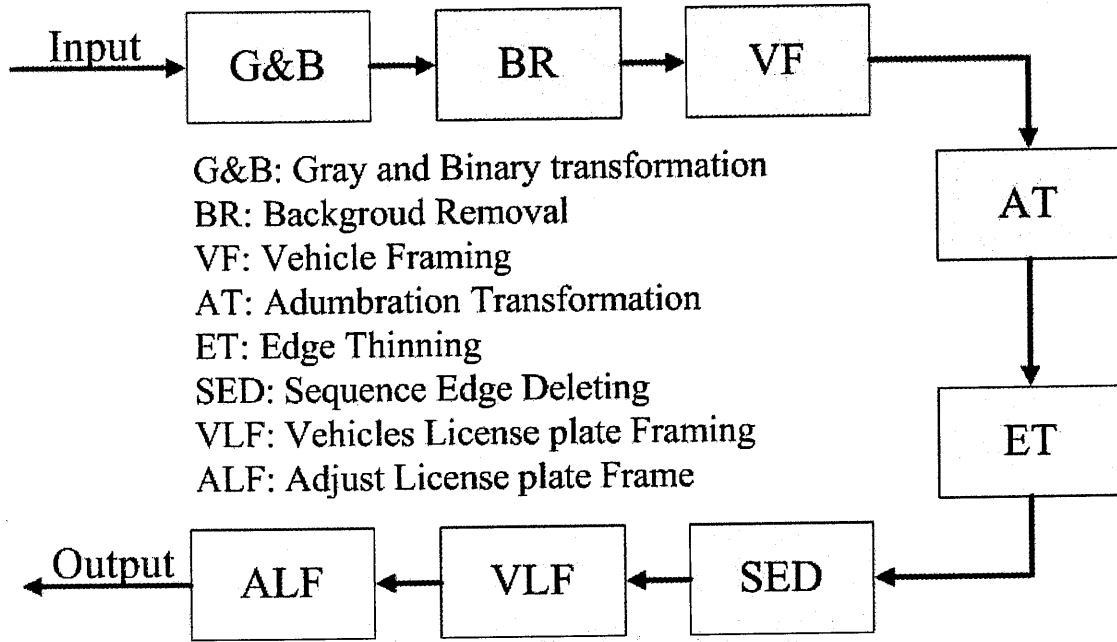


圖 4-1：車牌定位演算法的步驟圖

#### 4.1.1. RGB 轉 HSI 色彩空間

在本論文的車牌定位演算法步驟中，影像的輸入是由 CMOS Sensor 所擷取的 RGB 彩色影像轉成 HSI 色彩空間的影像，其轉換公式如(4-1)~(4-4)所示。

$$\theta = \cos^{-1} \left\{ \frac{0.5 \times [(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^{1/2}} \right\} \dots\dots\dots(4-1)$$

$$H = \begin{cases} \theta, & \text{if } B \leq G, \\ 360 - \theta, & \text{if } B > G. \end{cases} \dots\dots\dots(4-2)$$

$$S = 1 - \frac{3 \times [\min(R, G, B)]}{(R + G + B)} \dots\dots\dots(4-3)$$

$$I = (R + G + B) / 3 \dots\dots\dots(4-4)$$



其中  $H$  是彩度 (Hue) ，其值介於 0 到 360 之間，每個  $H$  值代表著不同的顏色， $H$  為 0 表示沒有色彩，是純灰色，如果  $H$  超過 360，則取 360 之餘數； $S$  代表色彩飽和度 (Saturation) ，其值介於 0 到 1 之間，數值越大表示顏色越鮮豔，數值越小則反之； $I$  表示亮度 (Intensity) ，也就是灰階度。圖 4-2 為 HSI 色彩空間模型。在本論文所提出的自動車牌定位系統中就是在  $I$  平面 (I-plane) 上做前處理，以及利用  $S$  與  $H$  的色彩資訊來定位車牌位置。

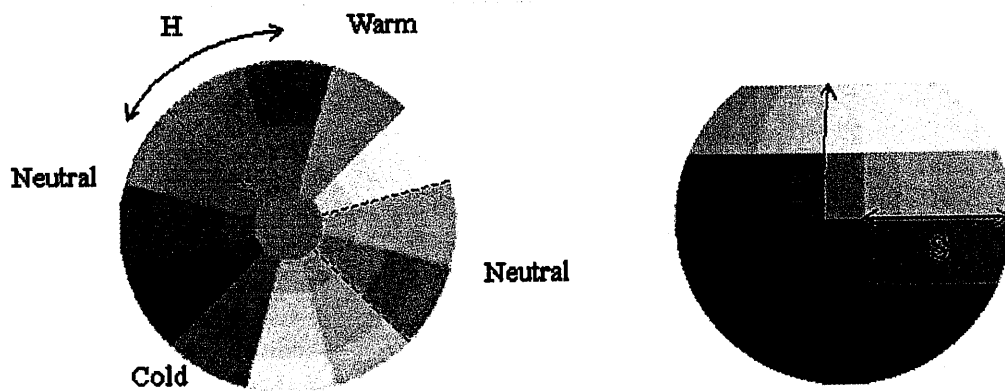


圖 4-2：HSI 色彩空間模型

#### 4.1.2. 適應性背景更新

本論文之車牌定位演算法是在  $I$  平面 (I-plane) 上用背景相減演算法 (Background Differing) 來擷取連續影像的中的移動物件，直接將當前影像  $X$  (圖 4-3) 與背景影像  $X_{bg}$  (圖 4-4) 逐個像素比較得到物件影像  $X_{obj}$  (圖 4-5)。



圖 4-3：當前影像



圖 4-4：背景影像

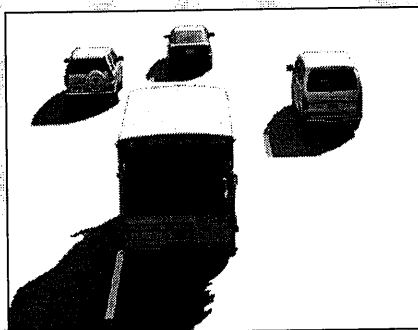


圖 4-5：物件影像

取出物件影像後將物件像素設為白色 (pixel value = 255) ，其餘背景區域則把像素設為黑色 (pixel value=0) ，故此時的物件影像  $X_{obj}$  即為黑白的二值影像。但是背景相減演算法有很大的缺點，就是對於環境的細微變化，如光線、陰影等條件特別敏感，以至於會產生非常多的雜訊。所以本文中是使用型態學的侵蝕與膨脹演算法來對物件影像  $X_{obj}$  中的雜訊濾除。

由於本篇論文所提出的車牌辨識系統是在室外環境下運作的，所

以室外環境下背景影像  $X_{bg}$  不僅會受到光線與物件陰影的影響，也會受到時間的遷移而改變，不僅如此攝影機也有可能會被外力所移動，因此在適當的時候更新背景影像  $X_{bg}$  是需要的。本論文提出一個適應性的背景影像更新演算法 (An Algorithm of Updating Background for Self-Adaptive)。

背景影像更新演算法的第一步先將物件影像  $X_{obj}$  分成 64 個區塊 ( $Block_i, j=1...64$ )，如圖 4-6 所示。若某個區塊中白色像素個數不少於區塊內全部像素個數的 33.33%，則此區塊即被標記起來，若物件影像  $X_{obj}$  中被標記的區塊超過 40 個的話，則視為環境已改變，此時就以一張當前的影像取代背景影像成為新的背景影像，如公式 (4-5)、(4-6) 所示。其中  $B_h$  與  $B_w$  為區塊的長和寬， $p_i$  為區塊內第  $i$  個像素的像素值。

$$Block_j = \begin{cases} 1, & \text{if } \sum_{i=1}^{B_h \times B_w} p_i \geq \frac{255(B_h \times B_w)}{3} \\ 0, & \text{otherwise} \end{cases} \dots\dots\dots(4-5)$$

$$X_{bg} = \begin{cases} X, & \text{if } \sum_{j=0}^{64} Block_j \geq 40 \\ 0, & \text{otherwise.} \end{cases} \dots\dots\dots(4-6)$$

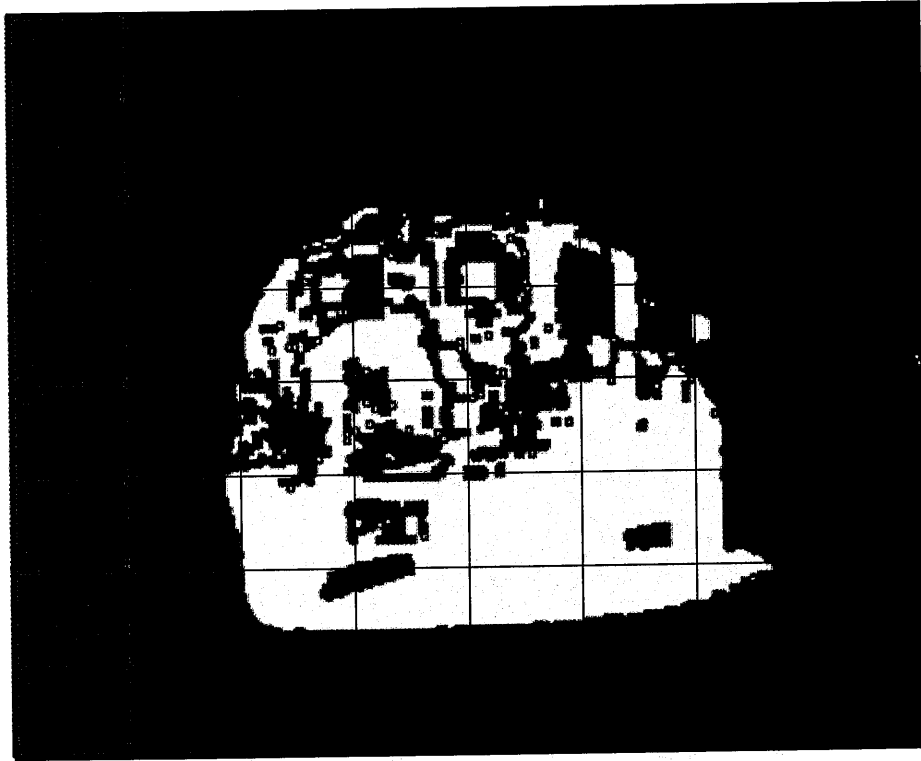


圖 4-6：8x8 等分區塊的物件影像

#### 4.1.3. 框取車輛

在物件影像  $X_{obj}$  中除了車輛以外，還有其它可能會移動的物件，例如行人、動物以及距離太遠的車輛，這些移動物件根本沒有可辨識的車牌資訊，無法辨識出車牌，若納入系統運算，不僅會提高計算量，也會影響到辨識的結果。所以本論文是利用物件的大小來判斷物件是否為車輛物件，框取車輛步驟如下：

- A. 開始利用一個 3x2 的遮罩(3x2 Mask)，如圖 4-7。此遮罩之 root 移動順序為由左而右，由上至下之走訪整張物件影像。
- B. 當 3x2 遮罩上的 root 找尋到第一個白色像素時，將此像素的座標

位置被當作是樹狀結構的「根節點」。

- C. 依照圖 4-7 上 3x2 遮罩的箭頭順序建立其子節點。
- D. 然後此次搜尋到的各個子節點，將成為下次搜尋的父節點，如此重複步驟 3，直到此移動物件的像素都被搜尋完畢為止。
- E. 當移動物件所有的像素都被搜尋過後，此移動物件就會形成一個樹狀結構，如圖 4-8 所示。而在這裡稱此樹狀圖為物件樹(Object Tree)。

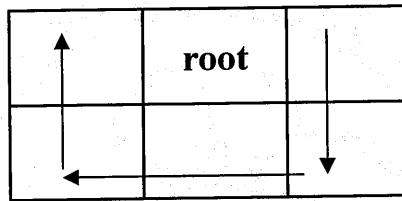
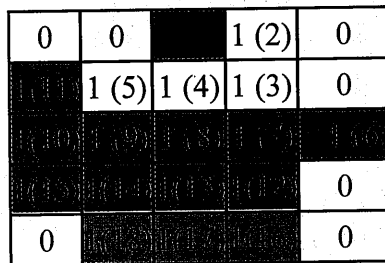
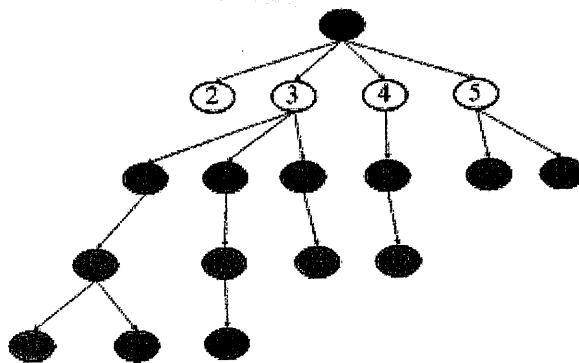


圖 4-7：3x2 遮罩



(a)



(b)

圖 4-8：物件樹的建立(a)物件影像的示意圖，1 為白色像素，0 為黑色像素，括號內的數字順序為長樹過程中的節點順序；(b)為物件影像(a)之物件樹。

F. 當物件樹建構出來之後，若節點數大於門檻值  $k$  就將此物件判定為車輛，再利用樹狀結構上每個節點所對應的位置來框取車輛，而用來框取物件的矩形框架稱為車輛框架 (Vehicle Frame)，此框架框出了整個車輛的範圍，但為縮小範圍減少運算量，在取得初步的框架後，會將此框架的高度自動縮小為原來的三分之二，將較靠近車頂的三分之一部份先行去除，其結果如圖 4-9-(c) 所示。

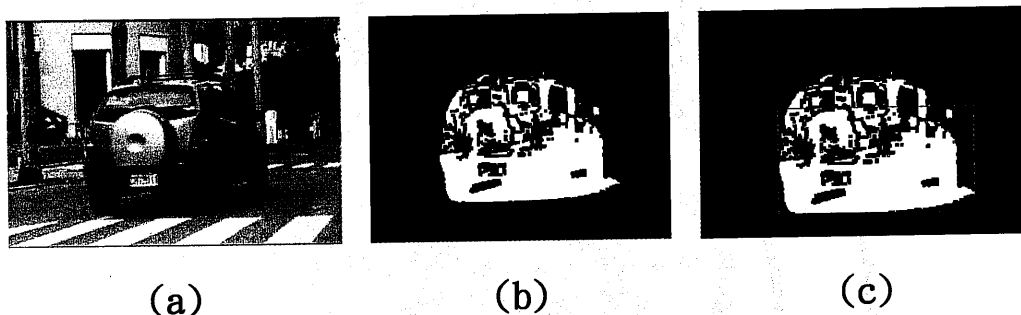


圖 4-9：框選車輛框架；(a)原始影像；(b)物件影像；(c)車輛框架。

#### 4.1.4. 輪廓擷取

在國內有九成的車牌是白底黑字所組成的，而本論文所提出的自動車牌定位演算法也是針對此類的車輛做車牌定位，所以在做輪廓擷取之前必須要在 Vehicle Frame 中找出車子的白色區域，這是為了減少擷取到車牌以外的輪廓。在 4.2 節中有提到，當輸入影像一進來就將輸入影像轉成 HSI 色彩空間的影像，而白色在 HSI 空間中 I 的值為 255，H 與 S 的值皆為 0，所以本論文就是根據這樣的特性來找出車輛白色的區域，再對這些白色區域做輪廓擷取，如圖 4-10。而擷

取輪廓的方法是利用 Sobel 運算子，Sobel 是影像處理中的運算子之一，主要用作邊緣檢測。在車體的外觀上，垂直與水平的線條是最多的，又為了避免擷取多餘的輪廓，本文是取角度為 45 度與-45 度之邊緣，其公式如下。

$$Mask_{45} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \dots\dots\dots(4-7)$$

$$Mask_{-45} = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \dots\dots\dots(4-8)$$

$$G = \sqrt{G_a^2 + G_b^2} \dots\dots\dots(4-9)$$

其中  $G_a$  與  $G_b$  分別藉由  $Mask_{45}$  與  $Mask_{-45}$  遮罩運算，所得的 45 與 -45 度的邊緣之影像。擷取車輛輪廓的目的為縮小目標範圍，進而取出車牌的輪廓，從圖 4-10 中可以觀察到，車牌區域的輪廓分佈極為密集。本文即利用此一特性來定位車牌位置。

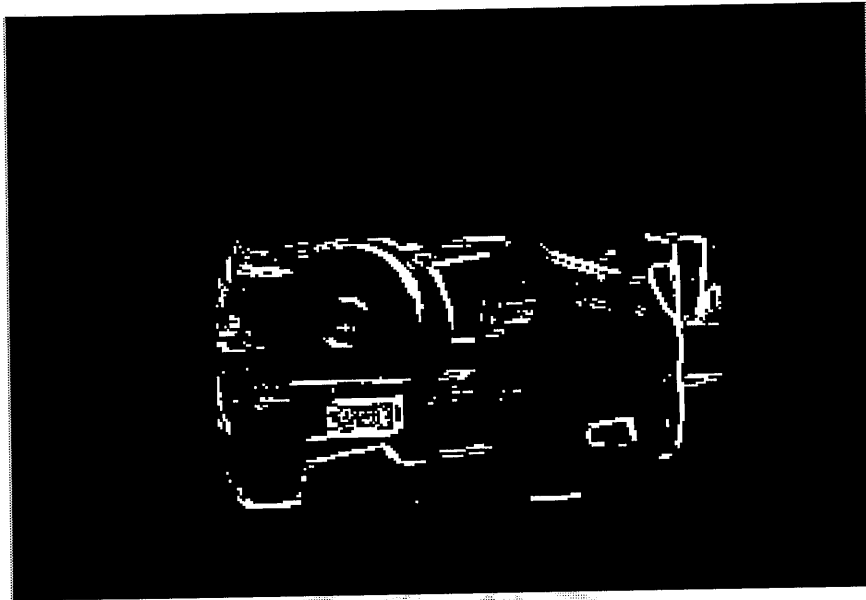


圖 4-10：邊緣輪廓影像

#### 4.1.5. 輪廓細線化

利用 Sobel 演算法縮小範圍，取得車輛的邊緣之後，為了再簡化邊緣，本文使用細線化演算法來取出車子的骨架，而細線化演算法有幾類，以下詳述其中兩大常用方法：

##### A. Hilditch 細線化[47]:

Hilditch 細線化法是從輪廓像素  $P$  與其八個鄰接點，如圖 4-11 所示，藉由判斷中心像素與其八個鄰接點的關係，來決定  $P$  點是否該消除，其步驟如下：

(1) 令  $B(P)$  為  $P$  八個鄰接點由黑色(pixel value=0)變白色(pixel value=255)的次數， $W(P)$  為  $P$  鄰接點為白色的個數。

(2) 定義連接數  $N_c$ (Connectivity Number):



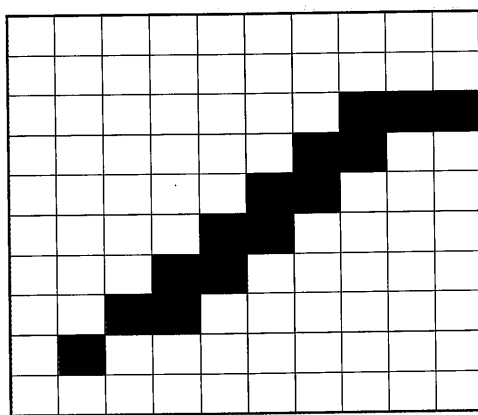
$$N_c = \sum_{k=1,3,5,7} \{(1 - P_k)[1 - (1 - P_{k+1})(1 - P_{k+2})]\} \dots\dots\dots(4-10)$$

在公式 4-10 中，當  $k=7$  時， $k+2=1$ ，其中 P 點與 8 個鄰接點的排列方式如。

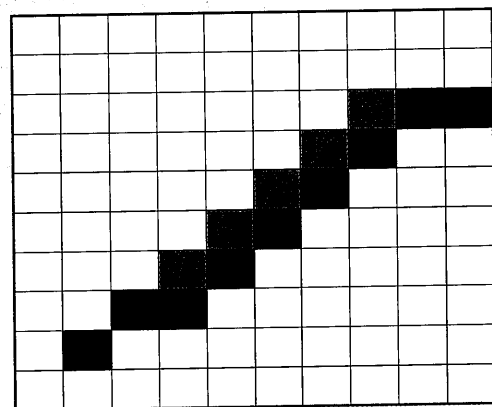
P6	P7	P8
P5	P	P1
P4	P3	P2

圖 4-11：輪廓像素(P)與其鄰接點

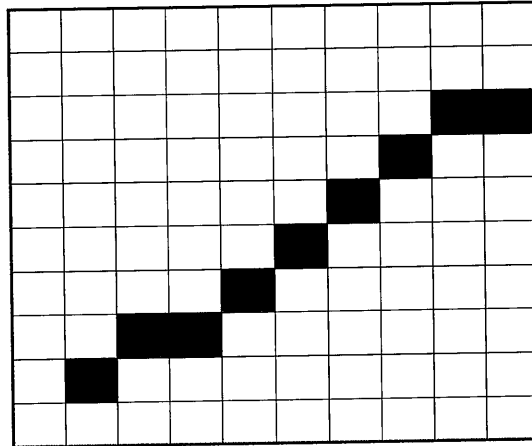
(3) 當  $N_c=1$  時，表示 P 點為邊界點，也就是待消除點，但並不是  $N_c=1$  時 P 點就該被消除，如果同時存在  $W(P)=1$  時，表示 P 點為輪廓的端點，所以需保留。圖 4-12 為 Hilditch 細線化的過程，其中綠色為待消除的像素。



(a)



(b)



(c)

圖 4-12：Hilditch 細線化範例；(a)Hilditch 細線化前影像；(b)標示待消除點；  
(c)Hilditch 細線後影像。

#### B. ZS 演算法[47]:

ZS細線化演算法，因為是由Zhung and Suen在1984年提出的，此為一平行式的細線演算法 (Parallel Thinning Algorithm)。ZS演算法同樣會針對如圖4-10所示的八個鄰近點做運算，主要有兩個步驟，第一步為針對以P點為中心的右半部鄰近點 ( $P_8$ 、 $P_1$ 、 $P_3$ ) 以及下半部鄰近點 ( $P_1$ 、 $P_3$ 、 $P_5$ ) 做運算；第二步則是對左半部 ( $P_3$ 、 $P_5$ 、 $P_7$ ) 及上半部 ( $P_5$ 、 $P_7$ 、 $P_1$ ) 做運算，以下說明其運算及消除的法則。首先定義 $A(P)$ 為P之鄰近點由0變1的次數，0表示為背景，1表示為輪廓像素，並且令 $B(P)$ 為鄰近為輪廓像素點的總和。在第一個步驟時只要符合公式4-11時即將該P點去除。第二步時假使符合公式4-12，亦將該P點去除。此方法以疊代方式，一層一層刪除物件邊界的像素，最後保留中心的骨架(skeleton)，而

疊代終止的條件為以下四點：ZS細線化同時要遵守端點不能侵蝕、相連部分不能過度消去、細線化後的骨架保持完整性、不能收縮太多。圖4-13為經過ZS細線化產生的結果。

$$\begin{aligned}
 2 \leq B(P) \leq 6, \\
 A(P) = 1, \\
 P_7 \cdot P_1 \cdot P_3 = 0, \\
 P_1 \cdot P_3 \cdot P_5 = 0.
 \end{aligned}
 \tag{4-11}$$

$$\begin{aligned}
 2 \leq B(P) \leq 6, \\
 A(P) = 1, \\
 P_3 \cdot P_5 \cdot P_7 = 0, \\
 P_5 \cdot P_7 \cdot P_1 = 0.
 \end{aligned}
 \tag{4-12}$$

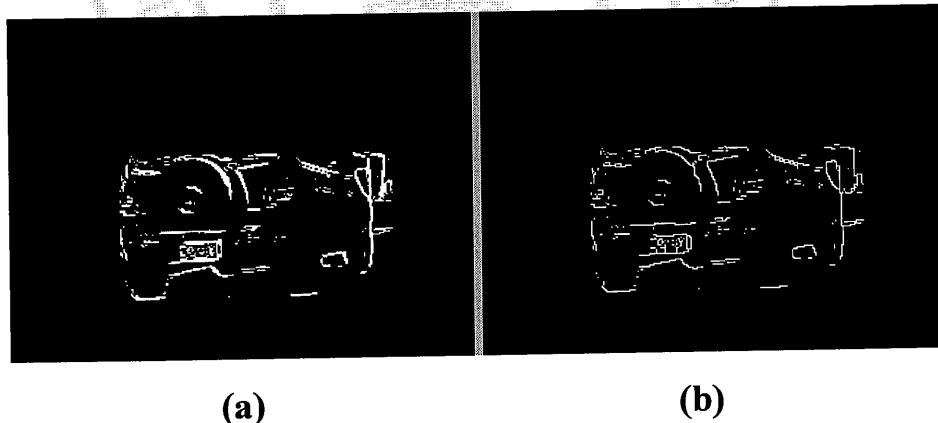


圖 4-13：ZS 細線化前後比較圖；(a)細線化前影像；(b)細線化過後影像

在一般的情況下，ZS 與 Hilditch 這兩種細線化法是搭配使用的，假使單獨僅用 ZS 演算法來取得骨架的話，部份的骨架線條的寬度可能還有兩個像素以上。假使單獨只用 Hilditch 細線化法來計算骨架的

話，所產生的骨架容易偏離輪廓中心。本論文中細線化的目的為減少不必要的輪廓，因此為了避免增加計算時間，所以僅使用 ZS 細線化法來簡化輪廓。

#### 4.1.6. 刪除連續邊

在取得車輛的細線化後的骨架，若水平和垂直方向有連續 10 個以上的白點，意即不間斷的水平和垂直線段超過 10 個像素的話，就將此水平或垂直線段刪除，因為車牌的部份，無論為水平和垂直方向皆不會有連續太長的線段，如此不僅可以再度刪除多餘無用的邊緣資訊，同時亦保證保留了車牌部位的骨架不會被刪除，經過上述步驟之後，大部分的白點像素一定都會分佈在車牌的部份，如圖 4-14 所示。

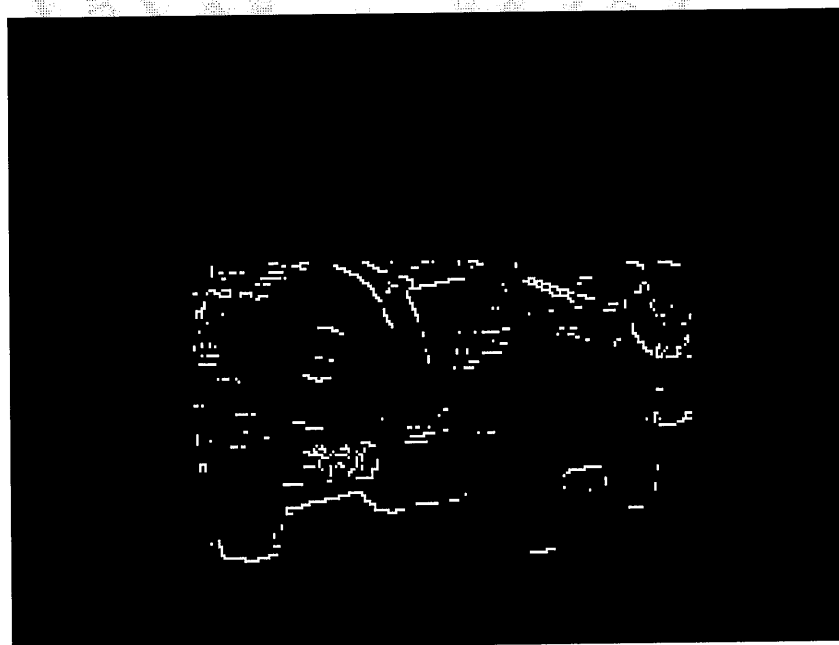


圖 4-14：刪除連續邊過後影像

#### 4.1.7. 框取車牌

此一步驟，首先計算出骨架像素密度，藉此定位出影像中車牌的位置，事先需先定義用於框取車牌的框架 (Plate Frame)，其寬度訂定為車輛框架 (Vehicle Frame) 最長邊的 1/8，而車牌框架的高度則定義為寬度的 62.5%，再依照此一定義的框架框取車牌影像。

訂定出車牌框架之長寬後，將車輛框架當看成是一張小圖，又稱為殘餘影像 (Residual image)、Plate Frame 當作是一個遮罩，此遮罩在殘餘影像中由左至右，由上到下走訪，如圖 4-15，當 Plate Frame 移動一次，便計算一次 Plate Frame 內的骨架像素個數( $N_M(e)$ )，而最大  $N_M(e)$  值之 Plate Frame 位置即為車牌的位置。

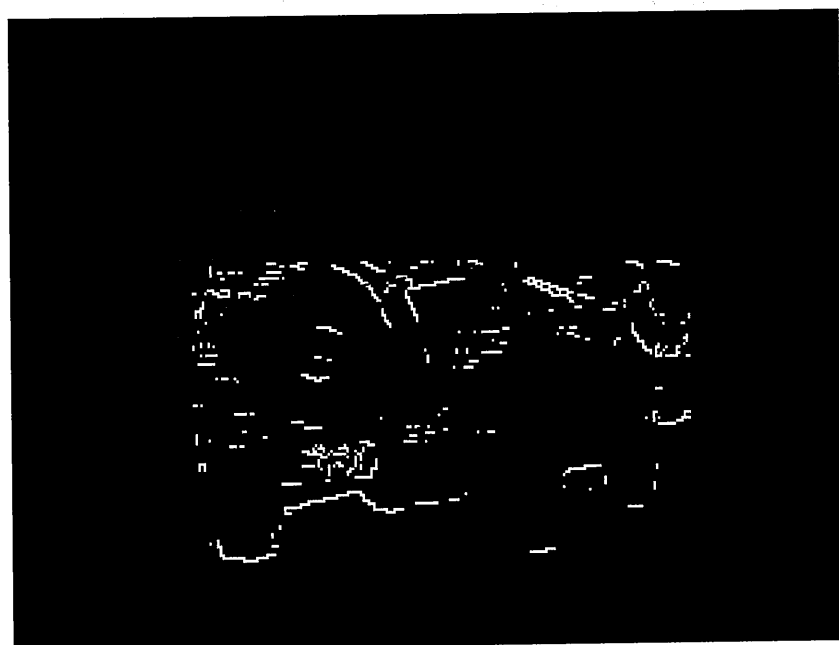


圖 4-15：車牌框架在車輛框架內走訪情況

若另外一個 Plate Frame 之像素個數總合值大於像素值總合最大值的 80%，表示 Residual image 內有兩部車子的重疊的影像，並框取出車牌，如圖 4-16 所示。

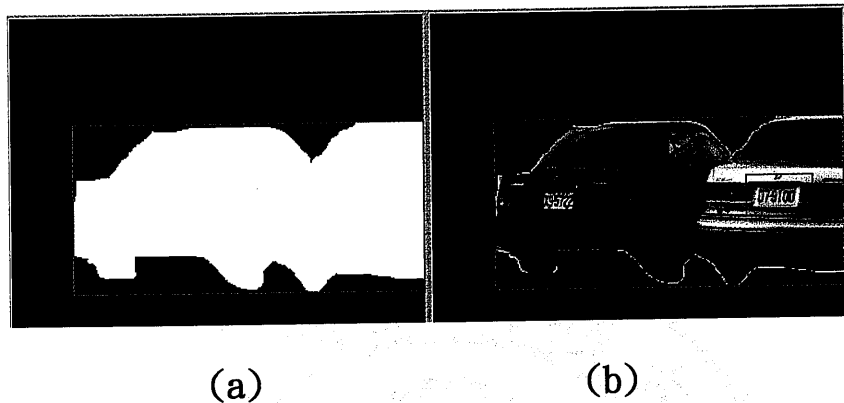


圖 4-16：框取重疊車輛之車牌

#### 4.1.8. 調整車牌框架大小

當明確的定位到車牌的位置後，可由圖 4.20 可以看出 Plate Frame 都比車牌大很多，所以調整 Plate Frame 的大小是必須的。調整車牌框架大小之作法和在 Residual image 內定位出車牌位置很類似。當 Plate Frame 定位在車牌位置上之後，我們將 Plate Frame 看成是一個 Residual image，用並將上一步驟中的框架，長跟寬皆縮小為原來的 95%，並再次如圖 4-15 所示走訪，計算新框架內的骨架像素的總和  $NM(e)$ 。找到  $NM(e)$  最大值的位置，然後計算此位置 mask 內的骨架像素密度值，若密度未大於 0.6，重複此步驟，直到密度大於 0.6，最後得到 mask 的大小和位置就是 Plate Frame 最終的大小和位置。

## 4.2. 字元切割

在取得了車牌影像之後，必須先行將車牌影像中字元，逐一按照順序的獨立出來，以台灣的車牌為例，如圖 4-17 所示，車牌號碼為一組由英文字元或字元所組成的號碼，而且長度固定為六碼。較為早期之車牌，在車牌號碼的上方會有較小的繁體中文字，標明省份或縣市別；另外車牌上亦會預留好螺絲孔。因此再進行字元切割時，必須先將這些多餘的資訊先一步去除；而且再進行字元切割時，還會碰到許多的問題，如拍攝時的光線問題；拍攝時的角度問題，造成車牌影像的傾斜。

本篇論文提出了一些方法，來解決上段所描述的問題，其方法流程如圖 4-18 所示，大致上可分成色彩空間的轉換和像素值的調整、車牌傾斜的校正、範圍的收縮以及最後的切割。以下便分成四個小章節，分別來介紹此四個步驟。



圖 4-17：台灣實際車牌影像

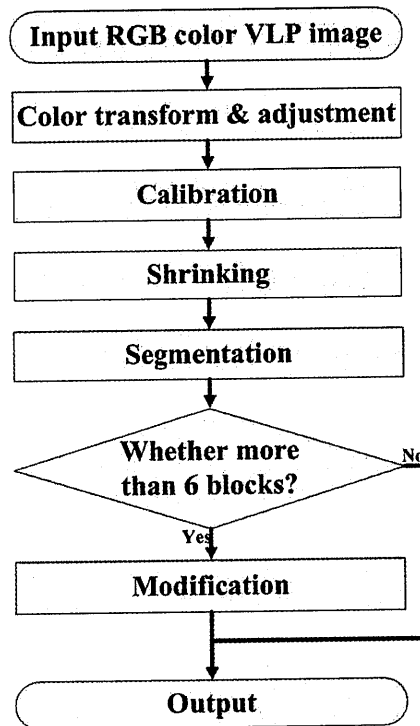


圖 4-18：字元切割工作流程圖。

#### 4.2.1. 色彩空間轉換及像素值調整

首先一開始輸入的車牌影像為彩色影像，在台灣的車牌雖有些特殊的車種，如大客車、大貨車、聯結車…等，為方便區別，會有不同底色，或不同的字體顏色，這只是為求方便肉眼區別車種，並不會出現相同一模一樣的車牌編號，然而僅以顏色來區分成不同的車輛，是故色彩的資訊對於後續的辨識，並無任何的幫助，因此為求運算上的效率，以及節省記憶體的空間，便先行將影像由 RGB 色彩空間，依公式(4-13)轉換至 YIQ 色彩空間，並取 YIQ 之 Y 值為灰階值。其結果如圖 4-19 所示。



$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & -0.274 & -0.322 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \dots\dots\dots(4-13)$$

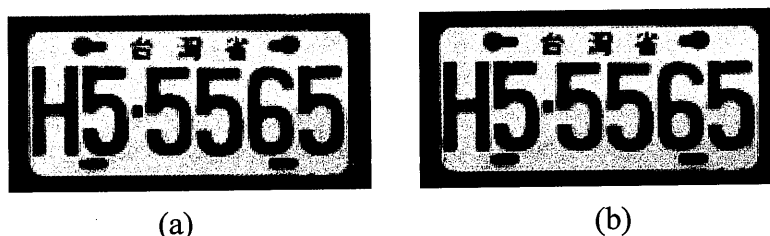


圖 4-19：色彩空間轉換結果圖；(a)定位完成後之彩色車牌影像；(b)車牌影像灰階圖。

#### 4.2.2. 車牌傾斜校正

部份車牌可能於拍攝時的角度問題，而造成車牌的傾斜，如圖 4-20 所示。此一問題對於字元切割後續的工作影響甚鉅，甚致會導致字元辨識時的失敗，以圖 4-20 為例，此張車牌影像如不經車牌傾斜的校正，直接進行後續的步驟時，其結果如圖 4-21 所示。然而假使此張車牌進行傾斜校正的話，則可透過本篇論文所提出的方法，獲得準確的校正，並且可以提升後續字元切割或辨識的準確率。圖 4-20 有經過傾斜校正之結果，如圖 4-22 所示，比較圖 4-21 與圖 4-22 之結果，可明顯得知經傾斜校正後之結果較為優異，而且適合後續的字元辨識。下段文章將會詳細解釋如何進行傾斜的校正。

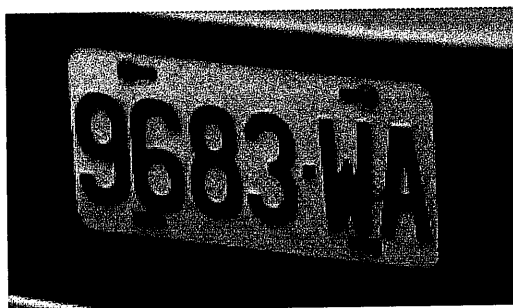
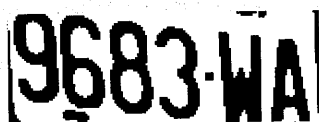


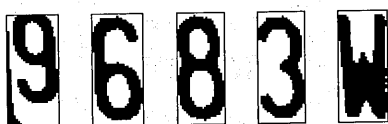
圖 4-20：傾斜車牌範例。



(a)

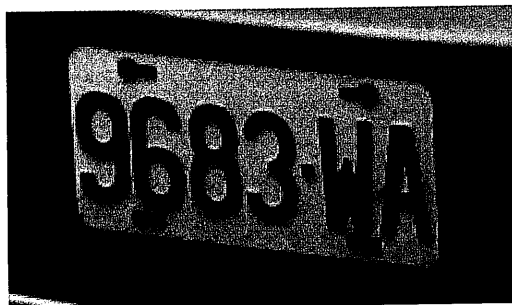


(b)

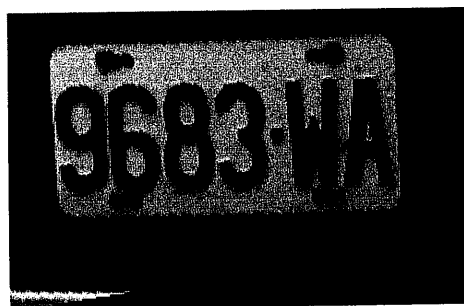


(c)

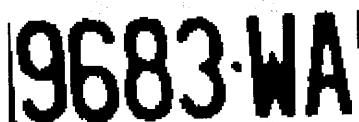
圖 4-21：傾斜車牌未經傾斜校正之字元切割結果；(a)將圖 4-20 轉為灰階結果圖；(b)未經傾斜校正直接進行範圍收縮之結果；(c)未經傾斜校正之字元切割結果。



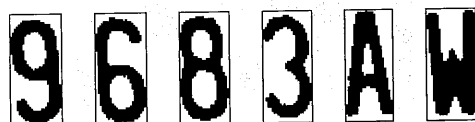
(a)



(b)



(c)



(d)

圖 4-22：傾斜車牌經校正後之字元切割結果；(a)將圖 4-20 轉為灰階結果圖；  
(b)傾斜校正後之結果；(c)經傾斜校正直接進行範圍收縮之結果；  
(d)經傾斜校正之字元切割結果。

車牌傾斜的校正首先需計算原本灰階車牌影像的 Canny 邊緣梯度圖，因為本篇論文的車牌校正方法，是使用霍式轉換 ( Hough Transform ) 來找出影像中最長的一條直線，並以紅線加以標記之，如圖 4-23 所示；利用此條標記紅線之斜率，可計算出每一直行需要向上或向下平移多少像素；其校正之結果圖 4-24 如所示；其中會將

邊緣梯度圖一併校正並儲存之，此一動作是為進行後續範收縮時使用之，在下一小節時將會說明之。

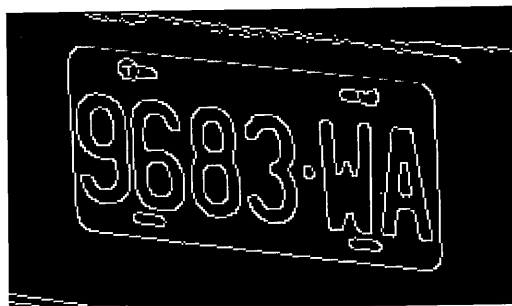
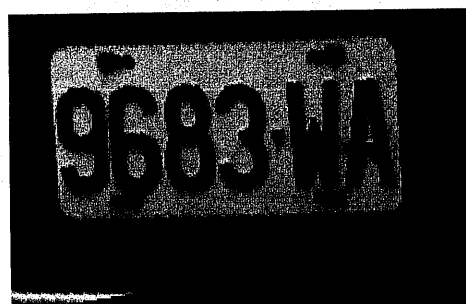
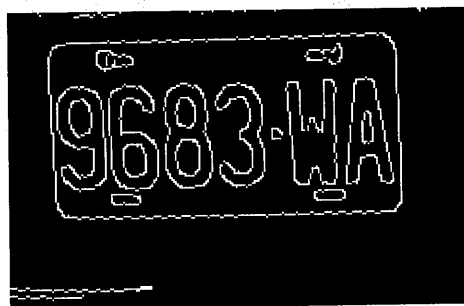


圖 4-23：灰階影像之 Canny 邊緣梯度圖，並且以紅線加以標記圖中最長之直線。



(a)



(b)

圖 4-24：經傾斜校正之結果圖；(a) 經傾斜校正之灰階結果圖；(b) 經傾斜校正之邊緣梯度圖。

經由比較圖 4-21 與圖 4-22 之結果，可以確定傾斜校正之步驟是必要而且極為重要的，圖 4-22 所顯示之結果可以得知，字元得以被精準的切割出來；反觀圖 4-21 之結果，不僅部份的字元切割不夠準

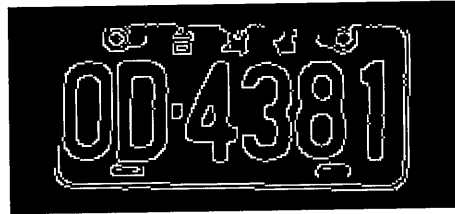
確，將螺絲孔的部份涵括在字元影像中，而且還將車體部份的雜訊一併當成字元而切割出來。因此結果顯示了此一步驟明顯改善了切割的成功率，而且相信此結果亦會大大提升了，後續進行辨識時的成功率。

#### 4.2.3. 範圍收縮

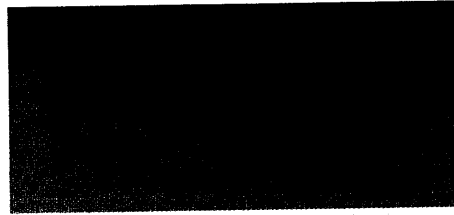
上一小節中介紹了傾斜校正的方法，並亦展示了校正完成後的結果，此一結果包涵了太多不必要的雜訊，如車牌號碼上方標示的台灣省資訊，以及上下方所預留的螺絲孔，和左右兩邊的車體部份，這些資訊都是多餘的，因為並不會出現相同的車牌號碼，而不同省份或縣市別的情況。因此為求後續進行字元切割時的方便，須先行將範圍收縮至更為貼近車牌號碼的區域，目的為將上述多餘的雜訊先行去過濾去除。

圖 4- 25 為另一張車牌經過傾斜校正後之範例，在前一小節中有提及，校正時不僅針對灰階圖，亦會對邊緣梯度圖進行校正，因為此一步驟必須再次利用到邊緣梯度圖，此一步驟的方法如圖 4- 26 所表示，一開始由高度一半的地方，開始往上及往下搜尋，假使如圖 4- 27 所假設之車牌號碼，亦至少還會有 12 次的邊緣變換，因此由影像高度一半的地方，往上搜尋邊緣變換低於 12 次的地方，是為上邊界；往下搜尋邊緣變換低於 12 次的地方，是為下邊界。接著再由水平兩旁往內收縮至有邊緣變換的地方即停止，利用在邊緣圖上所搜尋到的

結果，來對校正後的灰階圖進行收縮，其結果如圖 4-28 所示。



(a)



(b)

圖 4-25：另一張車牌經傾斜校正後之結果圖；(a) 經傾斜校正之灰階結果圖；(b) 經傾斜校正之邊緣梯度圖。

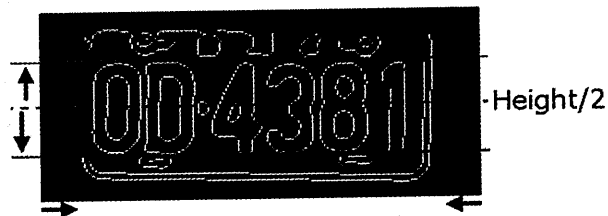


圖 4-26：範圍收縮方法示意圖。

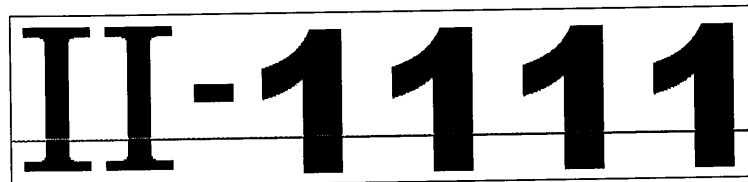


圖 4-27：假設之最極端狀況之車牌。

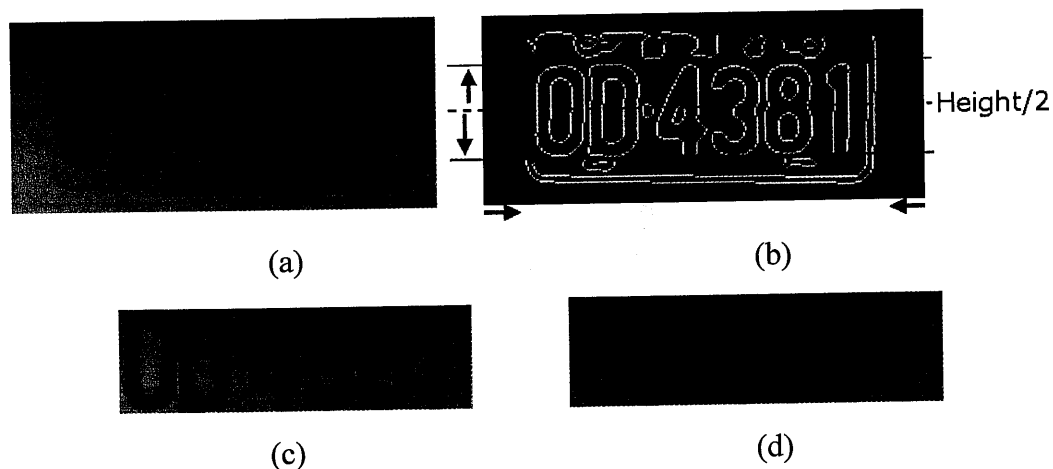
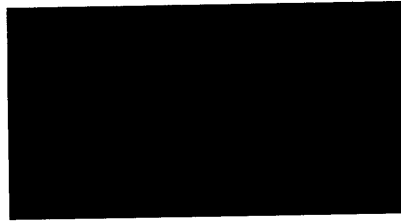


圖 4-28：範圍收縮之結果範例；  
 (a)經矯正後之車牌影像灰階圖；(b)範圍收縮方法示意圖；(c)範圍收縮結果圖；  
 (d)將(c)之結果進行高度調整，固定為 55 像素。

圖 4-28 所顯示之結果，經過範圍收縮後，省份、縣市別，還有預留的螺絲孔等雜訊，已經去除，雖然有時左右兩旁車體的部分無法去除，如圖 4-29 所示，但此一部份將會在字元切割時，獲得解決。另外於圖 4-28 之 (d) 小圖，為將收縮的結果進行放大或縮小，將影像的高度調整為固定 55 像素，此步驟目的為進行字元切割時，是用投影法來進行字元的切割判斷，原理將在下一小節中加以詳細說明之。



(a)



(b)




圖 4-29：部份車體雜訊無法去除範例；(a) 經矯正後之車牌影像灰階圖；(b) 已經高度正規化之範圍收縮結果圖。

#### 4.2.4. 字元切割

在完成範圍收縮並且將高度正規化之後，必須先將範圍收縮的結果影像二值化，目的是為了使用投影法，來進行字元的切割判斷；其結果如表 4-1 所示。針對二值化的結果圖，會進行垂直投影量的計算，如圖 4-30 所示，此方法為投影法中的垂直正交投影，而以程式設計的角度來看此投影法，其實不過是對圖片所構成的二維矩陣，進行每一行的加總，形成與圖片寬度等寬的一維陣列。仔細觀察計算後的投影量，為方便觀察，圖 4-31 為另一張完成投影量計算的結果圖，並且加以放大和標記，標記的地方其投影量皆等於零，且往上對應至車牌字元的地方，恰皆為字元與字元之間的時間隔，利用此一特性，搜尋投影量連續的區塊，即為可能的字元位置，接著只要介斷連續區塊的寬度是否足夠，只要符合訂定的寬度區間內，即判定為字元儲存之。



表 4-1：各步驟結果表格

步驟	圖片影像	圖片維度 (單位：像素)
範圍收縮		高度：50 寬度：197
高度正規化		高度：55 寬度：216
二值化		高度：55 寬度：216

2908-FW

(a)

2908-FW  


(b)

圖 4-30：垂直投影範例；(a)範圍收縮後結果圖；(b)垂直投影量計算結果。

7068-UC  


圖 4-31：垂直投影量放大圖。

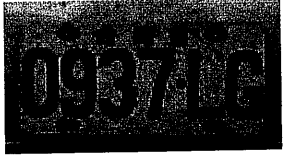


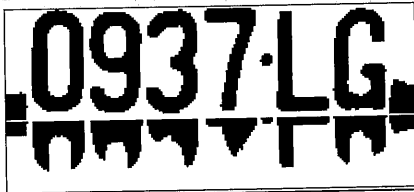




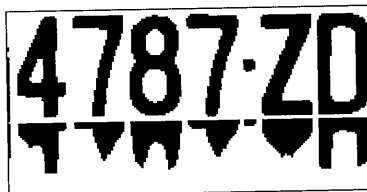

但上述的字元切割方法，大致上可行，但事實上卻存在二個小問題。第一個問題為假使車牌號碼含有阿拉伯數學 “1” 或英文字母

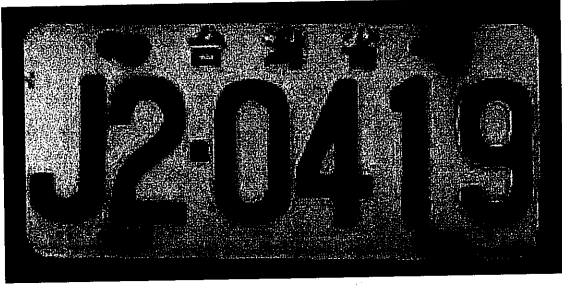


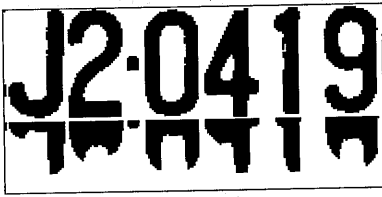

“ I ” 時，此兩個字元的寬度明顯會比其它字元窄上許多，參考表 4-2 所示，因此數字 “ 1 ” 和字母 “ I ” ，必須另外訂定寬度範圍，獨立出來當成特別情況 ( Special Case ) 另外判斷之。至於還有第二個問題，假使一開始輸入的車牌，因為距離遠近的問題，造成拍攝所得的車牌影像尺寸不一，使得固定的字元寬度門檻，無法適應太大會大小的字元，因此在範圍收縮的小節中有提及，收縮完成後會進行圖片高度的正規化，目的就是為了將字元調整致適當的大小，以適應字元切割所訂定的寬度門檻，表 4-3 展示了三張遠近距離不同的車牌，但皆可成功切割出車牌字元影像。

表 4-2：車牌號碼元素一覽表

0	1	2	3	4	5	6	7	8	9
<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
A	B	C	D	E	F	G	H	I	J
<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>	<b>J</b>
K	L	M	N	O	P	Q	R	S	T
<b>K</b>	<b>L</b>	<b>M</b>	<b>N</b>	保留	<b>P</b>	<b>Q</b>	<b>R</b>	<b>S</b>	<b>T</b>
U	V	W	X	Y	Z				
<b>U</b>	<b>V</b>	<b>W</b>	<b>X</b>	<b>Y</b>	<b>Z</b>				

表 4-3：輸入車牌影像遠近比較表

車牌號碼	處理步驟	結果影像	圖片維度 (單位：像素)
0937-LG	原始車牌影像		寬度：137 高度：76
	範圍收縮		寬度：137 高度：37
	高度正規化		寬度：203 高度：55
	垂直投影量		寬度：203 高度：91
	字元切割		寬度：25~26 高度：55
4787-ZD	原始車牌影像		寬度：200 高度：101
	範圍收縮		寬度：180 高度：54
	高度正規化		寬度：183 高度：55
	垂直投影量		寬度：183 高度：91
	字元切割		寬度：24~26 高度：55

J2-0419	原始車牌影像		寬度：278 高度：138
	範圍收縮		寬度：258 高度：75
	高度正規化		寬度：189 高度：55
	垂直投影量		寬度：189 高度：91
	字元切割		寬度： 26~28, 13 高度：55

另外部份車牌於定位時，可能框選了過多的車體部份，如圖 4- 32 所示，此張車牌定位影像在完成了範圍收縮後，如圖 4- 33 所示，再將之進行後續的字元切割，其結果如圖 4- 34 所示，在圖 4- 34-(d)所顯示的結果中，車牌兩旁的車體部份，被誤判為字元，為解決此一問題，在完成第一次字元切割時，會判斷所得的字元數是否符合，國內的車牌號碼長度，假使不符則進行修正切割，如圖 4- 35 中紅色框框標記處。

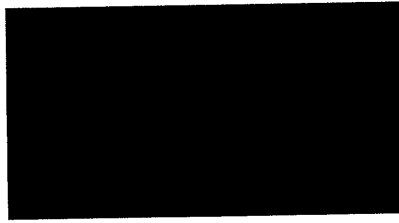


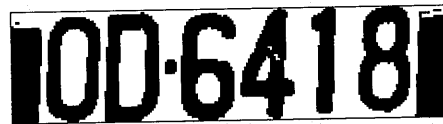
圖 4-32：較不精確之車牌定位範例。



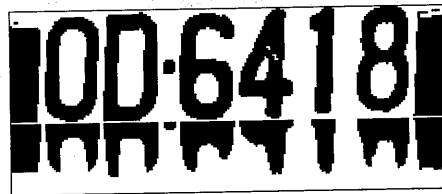
圖 4-33：圖 4-32 之車牌影像完成範圍收縮的結果。



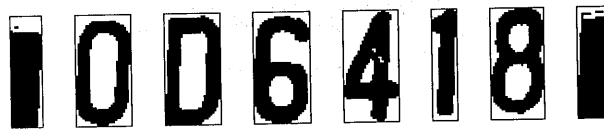
(a)



(b)



(c)



(d)

圖 4-34：因車體雜訊造成字元切割失敗範例；(a)將圖 4-33 高度正規化結果；(b)二值化結果；(c)垂直投影量計算圖；(d)字元切割結果。

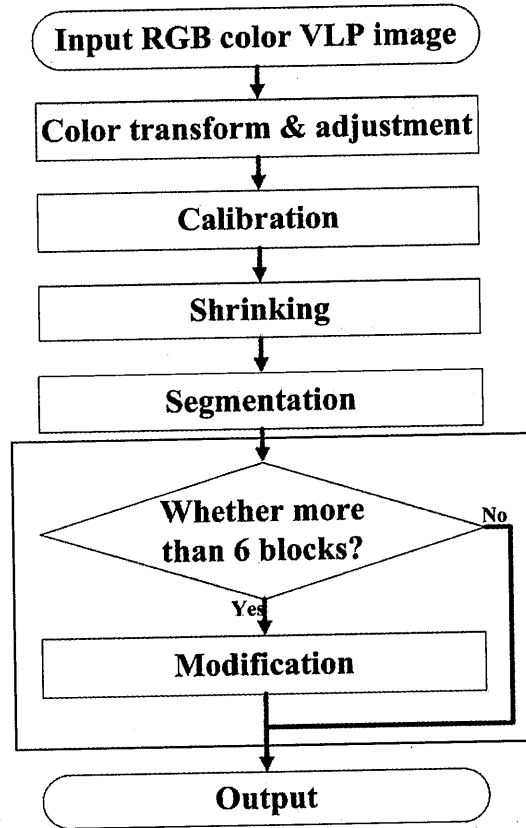


圖 4-35：字元切割工作流程圖。

修正切割的方法則是根據國內車牌編號規則來完成，藉由搜尋出車牌號中的“-”符號，並且判斷“-”符號座落於影像中的左半邊，還是右半邊。假使“-”符號座落於左半邊，則按照國內的車牌編碼規則，於“-”符號左邊會有兩碼，而在“-”右邊會有四碼；假使“-”符號座落於影像右半邊，則“-”符號左邊會有四碼，而在“-”右邊會有二碼。如此方可將水平左右兩端的車體雜訊，將之去除。

### 4.3. 車牌字元辨識

本篇論文之辨識流程如圖 4- 36 所示。首先輸入的影像須先與預先建立的資料庫中的每張圖片，進行比對並計算分數。分數的計算 (calculate score value) 為本辨識方法的精髓，故以下將詳細介紹評分的標準。本論文的辨識方法採樣版比對，並加以改進，字元資料庫共有 175 張影像，分別為十個阿拉伯數字及 25 個英文字母 (字母 “O” 除外) 各五張影像，且皆已正規化大小為  $20*40 \text{ pixel}^2$ 。因此欲辨識的影像輸入後，皆會正規化成為  $20*40 \text{ pixel}^2$  之大小，然後進行樣版比對，直接計算兩張圖的差異點，然後再根據差異點的多寡，給予一個分數，每五張相同數字或字母的分數會進行加總，因此總共會有 35 個分數。此分數最高者即為判定之字元。分數計算完成後，則找出評分最高的一組，作為此張輸入字元影像的辨識結果。

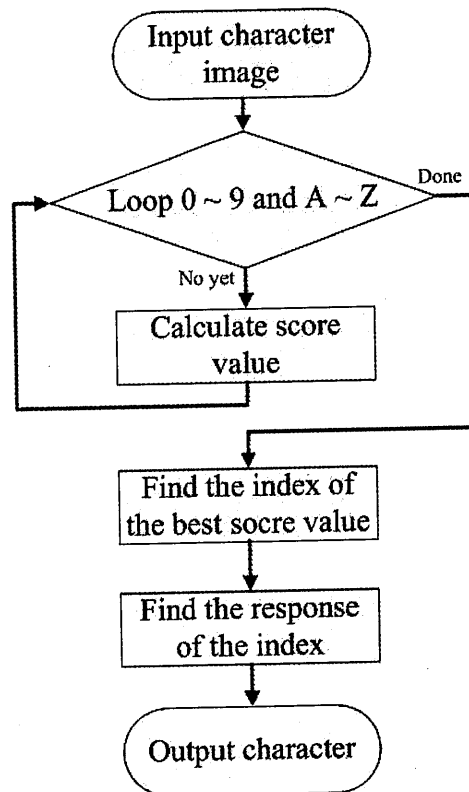


圖 4-36：車牌字元辨識流程圖。

至於評分的標準，本段文章將加以詳細介紹，首先大概解釋一下何謂樣版比對，圖 4-37 為兩張不同字元的假設影像，每一小格表示一個像素，比對兩張像可得差異點，如圖 4-38 之綠點所示，計算所有綠色點的個數，然後依照預先設定好的量尺表，如公式(4-14)所示；以圖 4-37 為例， $X$  為比對後所得的差異點個數， $N_1, N_2, N_3, \dots, N_{n-1}, N_n$  為預訂好的門檻，依序遞增；而  $a_1, a_2, a_3, \dots, a_{n-1}, a_n$  為預訂好的分數，依序遞減，也就是說差異點越多的，分數越低，差異點越少的，分數越高，而且總共只有 50 個像素，因此  $N_1, N_2, N_3, \dots, N_{n-1}, N_n$  就只介於 0~50 之間。計算完差異點找出對應分數後，此為比對一張樣本的分數，然而預設的資料庫，每個車牌字元會有五張樣本，所以五張樣本



的分數加總後，為最後的評分標準，所以最後總共會有 33 個分數 (0, 2~9, A~H, J~N 和 P~Z)，因車牌號碼中英文字母 “O” 為避免與數字 “0” 混淆，保留英文字母 “O” 不用，另外數字 “1” 與字母 “I” 亦會被視為特別情況，另外判斷之。因此僅有 33 個分數，其中最高者，其對應所得的樣本組別即為辨識所得之字元。

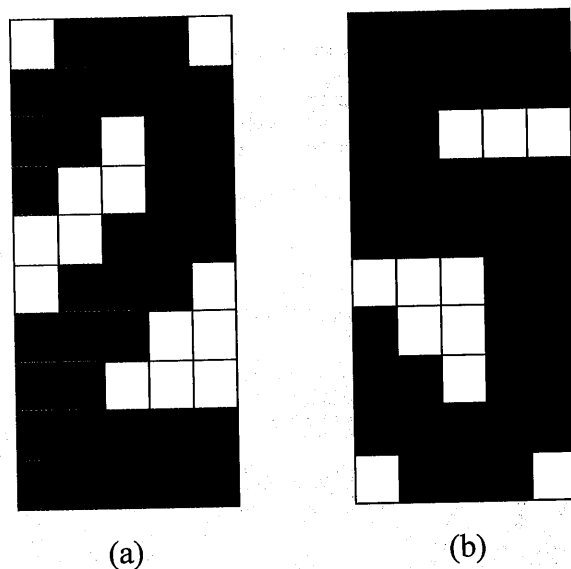


圖 4-37：車牌字元之假設影像；(a)目標字元”2”；(b)樣本字元”5”。

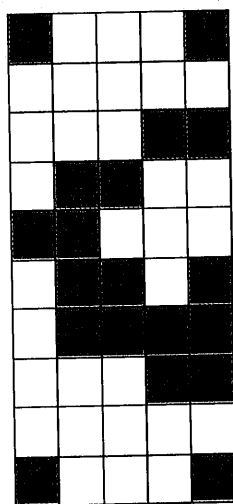


圖 4-38：圖 4-37 之(a)、(b)差異點示意圖。

$$\begin{aligned}
0 \leq X \leq N_1 & \quad \text{Score} = a_1 \\
N_1 < X \leq N_2 & \quad \text{Score} = a_2 \\
N_2 < X \leq N_3 & \quad \text{Score} = a_3 \\
& \quad \vdots \\
N_{n-1} < X \leq N_n & \quad \text{Score} = a_n \quad \dots\dots\dots(4-14)
\end{aligned}$$

where

$$0 < N_1 < N_2 < N_3 \dots < N_{n-1} < N_n < 50 \text{ pixels,}$$

$$a_1 > a_2 > a_3 \dots > a_n > 0$$

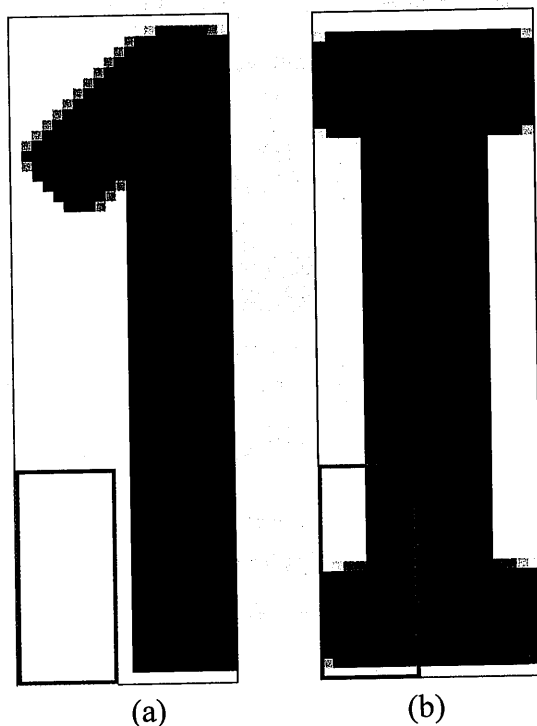


圖 4-39：字元 ”1” 與 ”I” 特徵比較方法；(a)數字”1”；(b)字母”I”。

然而上述的樣板比對方法中，並未將數字 “ 1 ” 與英文字母 “ I ” 列入比對的資料庫中，此兩個字元將會被獨立出來判斷之，觀察圖 4-39 之比較圖，假使今天欲進行辨識的字元，符合字元切割時

所訂定的窄字元寬度區間，此時藉由判斷圖中綠色框選的範圍之像素密度，即可判斷出該字元為“1”或“I”。



## 第五章 實驗結果

本論文之實驗環境如表 5- 1 所示。為方便系統開發之緣故，除 Camera 必須到戶外定點架設，錄製實驗用之動態影像，其餘實驗設備皆設置於實驗室內，Camera 錄製完影像後，再從記憶卡複製影像至電腦進行後續的實驗。進行實驗的電腦，中央運算單元(CPU)為 Intel 的四核心系列，處理速度為 2.66 GHz；記憶體因為受限於作業系統的限制，實際可用的僅有 3.5 GBytes；運行的作業系統為 Windows XP Professional Service Pack 3，而選用的編譯器為 Visual Studio 2005，撰寫的程式則是使用 C++。

另外為方便展示上一章節所述之各步驟結果，本論文設計了兩個簡單的圖形使用者界面 ( Graphic User Interface, GUI )，將車牌定位獨立展示，如圖 5- 1 所示；另外車牌字元切割和辨識結合在同一使用者界面，如圖 5- 2 所示。

表 5- 1：實驗環境列表

中央運算單元(CPU)	Intel Core 2 Quad 2.66GHz
記憶體(RAM)	3.5 GBytes
攝影機(Camera)	Sony DV
作業系統(O.S.)	Windows XP Professional SP3
編譯器(Compiler)	Visual Studio 2005
程式語言(Programming Language)	C++

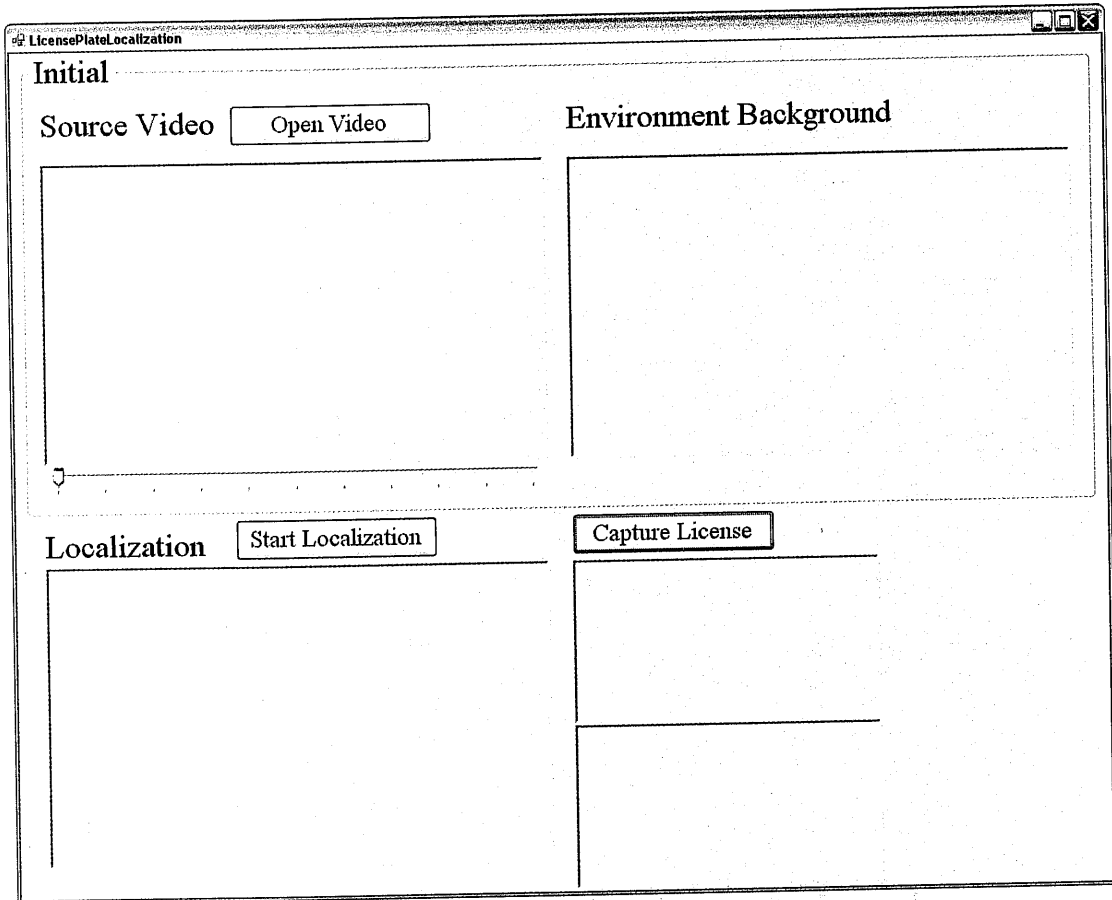


圖 5-1：車牌定位圖形使用者界面 (License Plate Localization GUI)。

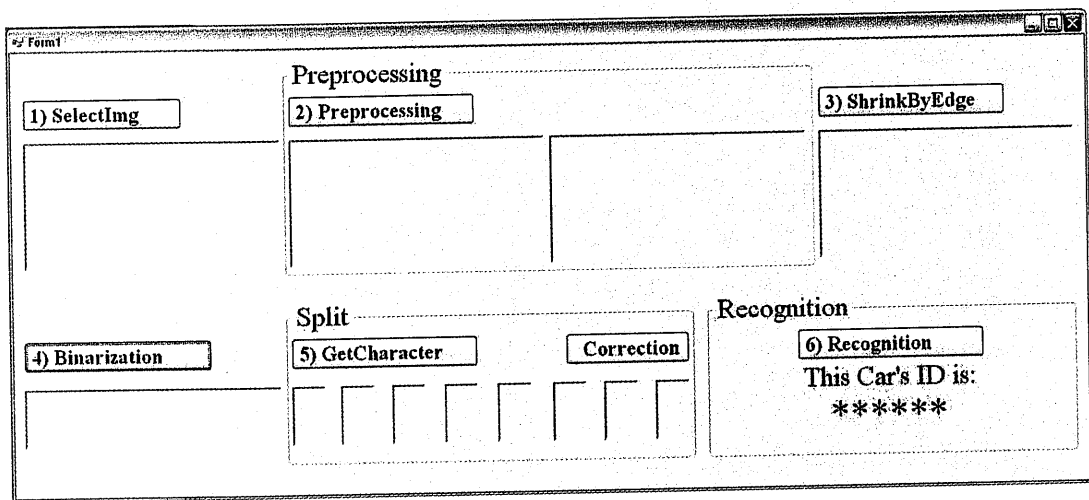


圖 5-2：車牌字元切割及辨識圖形使用者界面 (License Plate Character Segmentation & Recognition GUI)。

首先車牌定位實驗，由 Sony DV 錄製了數段影像，總共時長約 128 小時，由人工計數總共有 387 台不重覆的車輛，總共成功定位了 363 台車輛，定位成功率為 93.7%，其中部份定位成功影像如圖 5-3 及圖 5-4 所示。另外本車牌定位演算法亦能成功同時定位多部車輛，如圖 5-5 所示。

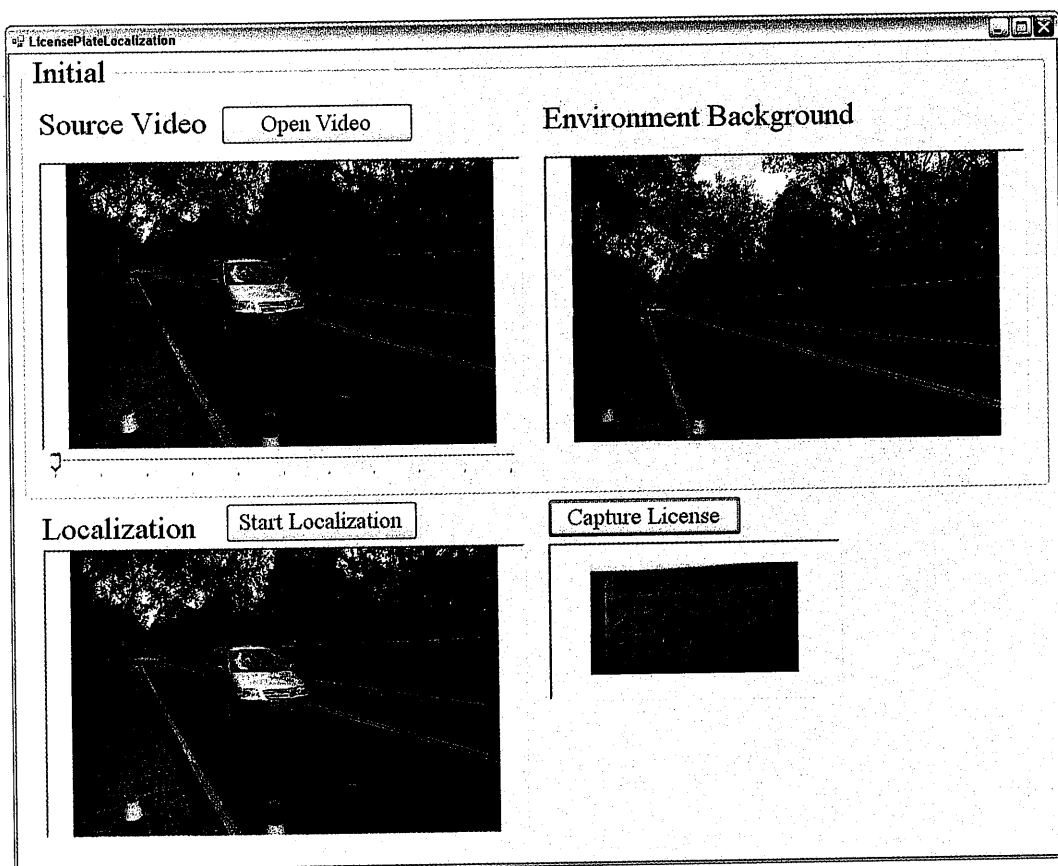


圖 5-3：成功車牌定位範例 1。

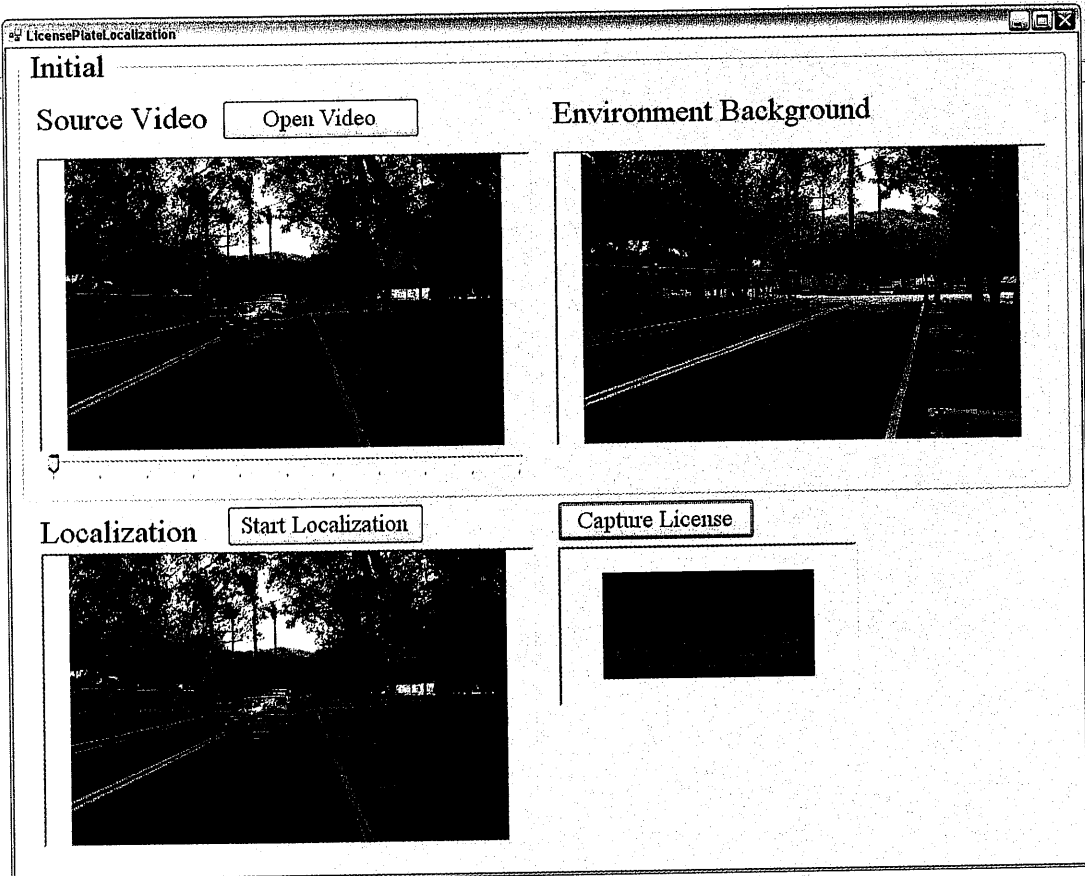


圖 5-4：成功車牌定位範例 2。

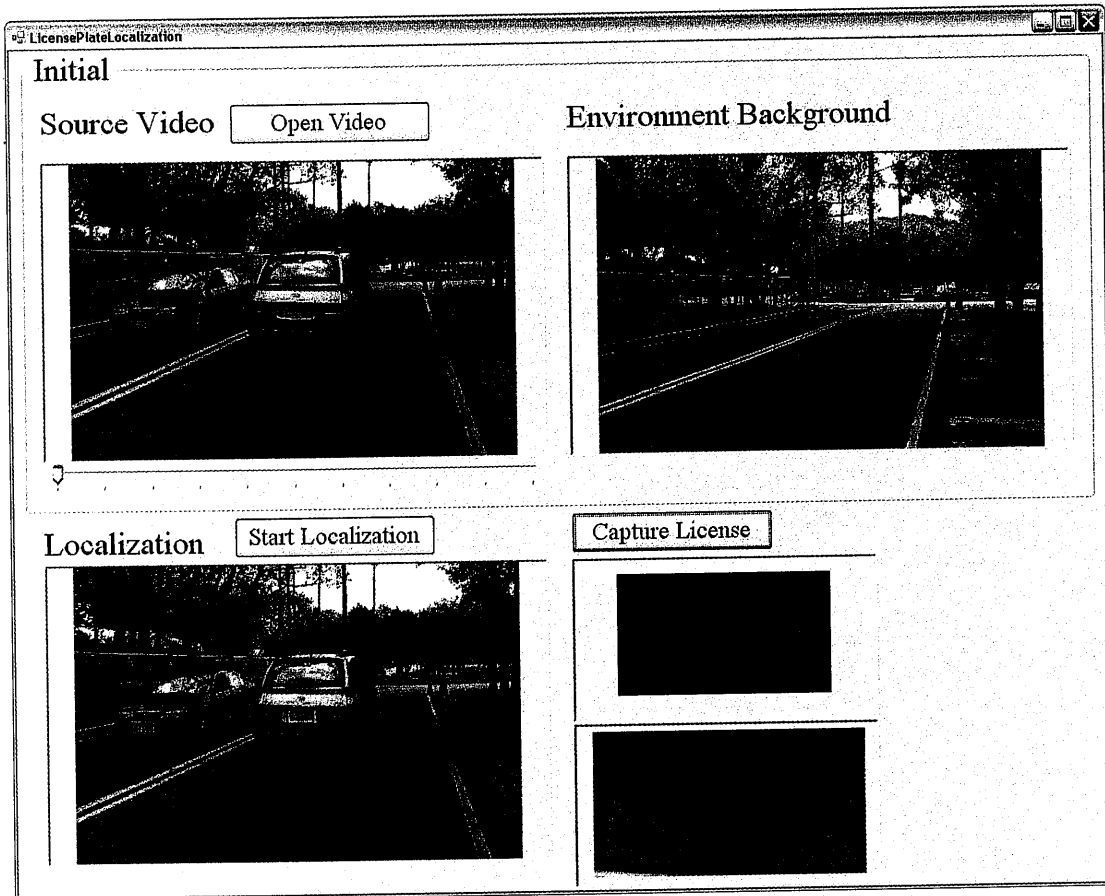


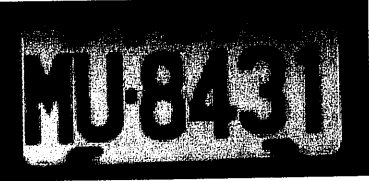

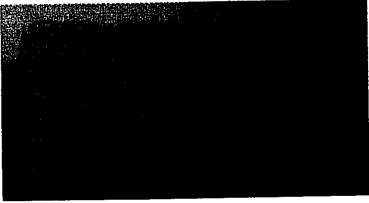





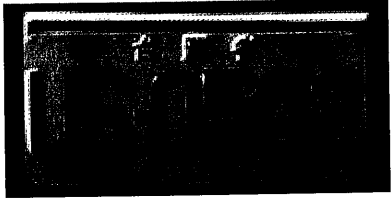
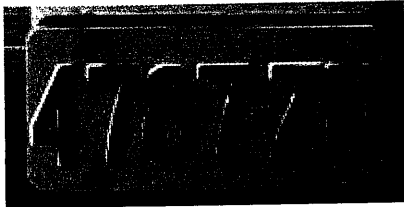
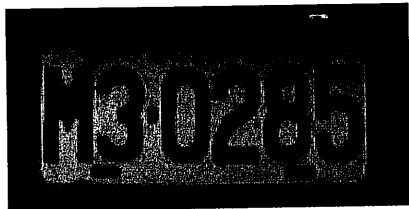

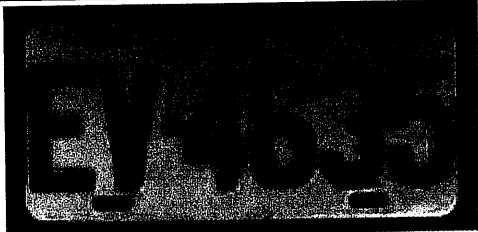
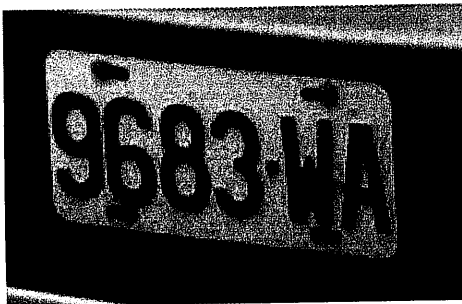
圖 5-5：成功定位多部車輛範例。

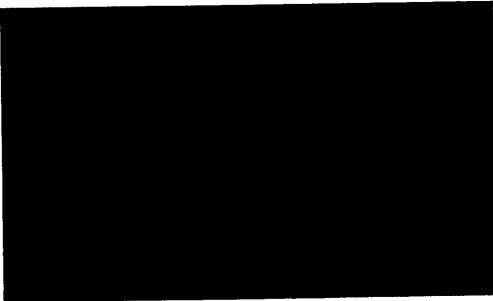

在成功定位的 363 輛車牌中，將得到 363 張不同的車牌號碼，這 363 張車牌號碼影像，即為後續的車牌字元切割及辨識的測試樣本，以下僅列部份定位成功車牌影像，如表 5-2 所示；觀察表 5-2 可以發現，完成定位後所取得的車牌影像，其圖片尺寸大小並不固定，因為本論文所使用的方法，並非使用固定的框架來框取車牌影像，僅檢查框取所得的影像的長寬比例是否符合標準，這樣的定位方法可適應性更高。



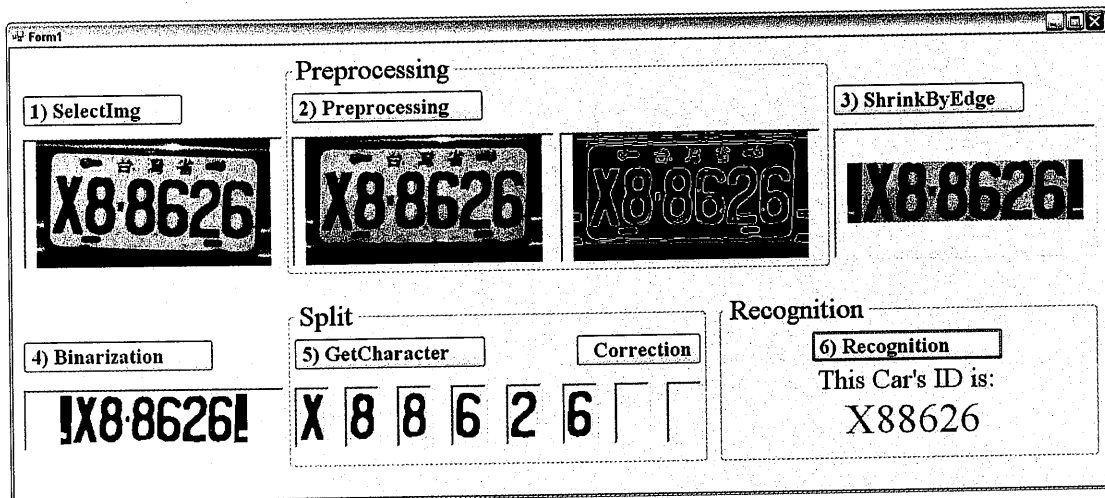
表 5-2：部份定位成功車牌影像。

成功截取車牌影像	影像尺寸資訊(單位：像素)
	寬：161 高：85
	寬：167 高：83
	寬：185 高：90
	寬：187 高：93
	寬：183 高：98
	寬：188 高：92
	寬：193 高：92
	寬：196 高：93

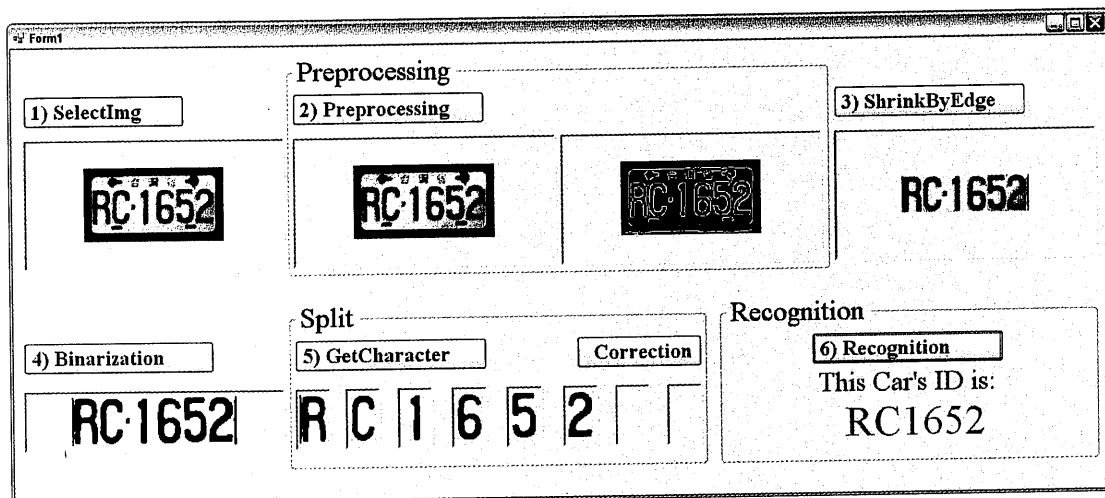
	<p>寬：192 高：97</p>
	<p>寬：200 高：101</p>
	<p>寬：201 高：101</p>
	<p>寬：229 高：114</p>
	<p>寬：236 高：115</p>
	<p>寬：229 高：148</p>

	<p>寬：245</p> <p>高：147</p>
	<p>寬：270</p> <p>高：149</p>

如果的定位方法雖然增加了定位的可適應性，但是如此一來亦增加了字元切割的難度，因為切割的方法為採用正交投影量來進行，並且判斷字元的寬度是否符合在訂定的範圍內。但比較表 5-2 車牌影像尺寸最小與最大的差異，實在過於懸殊，倘使我們訂定的寬度範圍要同時適應如此差異懸殊的車牌尺寸，勢必會造成字元切割的成功率大幅下降，原因為訂定的區間過於廣泛，使得會將雜訊判誤為字元，因此本篇論文的車牌字元切割演算法，經過第四章所述的一連串的前處理方法，亦成功的完全解決尺寸大小不一的問題，其結果如圖 5-6 所示。



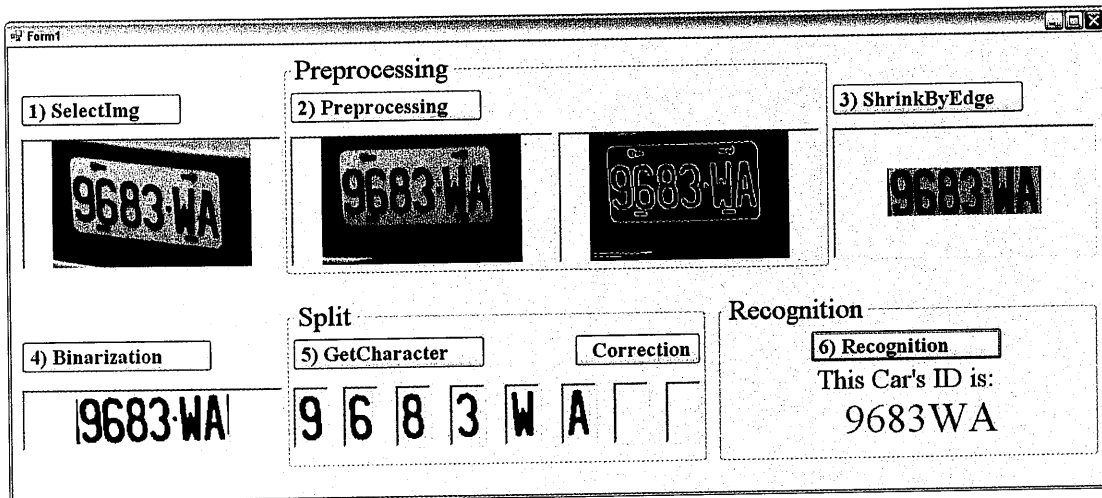
(a)



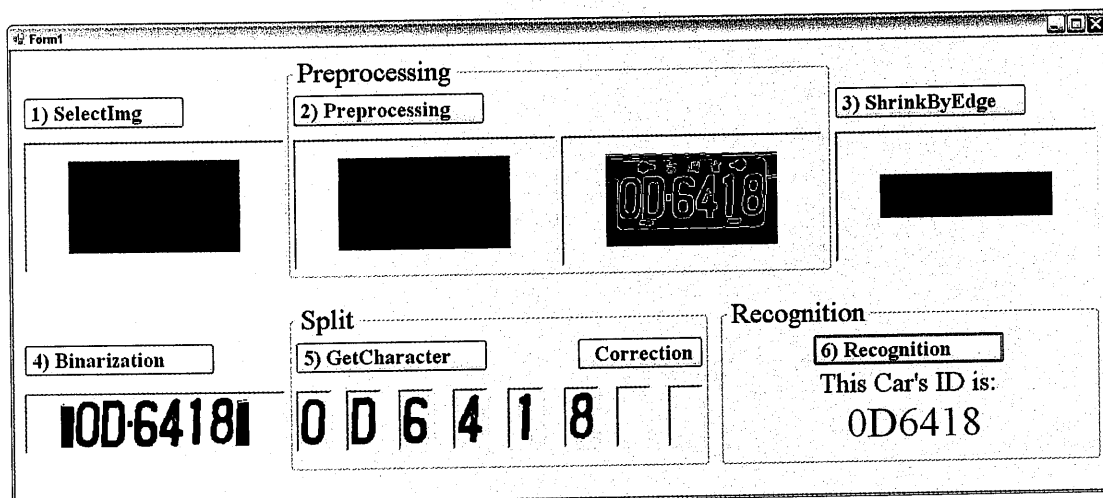
(b)

圖 5-6：部份車牌字元辨識範例。

再者於表 5-2 之中，有兩張車牌成像嚴重傾斜的範例，此類車牌在本論文的字元切割演算法中，亦能成功的矯正回長方形的影像，然後再成功切割出字元並辨識之。其結果如圖 5-7 所示。



(a)



(b)

圖 5-7：部份車牌字元辨識範例。

## 第六章 結論與未來展望

本論文實際完成了整個車牌辨識系統，其依序包涵了三項功能，首先為車牌的定位，然後車牌字元切割，最後為車牌字元辨識。車牌的定位功能，可以即時定位移動中的車輛，即使是有兩部車子重疊或是有車輛以外的移動物件，甚至是在複雜的背景中都能正確的定位出車牌的位置，平均定位出車牌位置的時間大約也在 0.3 秒鐘左右。而且本系統也有針對白底黑字車牌做最佳化，所以即使是車牌附近的輪廓很複雜也能找到車牌位置。不管如何，本系統的處理速度已被改善的夠快，以致於能即時的且精確的定位出畫面中多輛車牌的位置。接著本論文的車牌字元切割演算法，亦能適應於各種情況的車牌，如傾斜的車牌成像，各張的車牌可能尺寸大小不一的問題，還有光線變化使得每張車牌明亮度不一，這些問題在本論文的車牌字元切割演算法，皆可以順利的得到解決，並大幅提高了車牌字元切割的成功率。最後車牌字元辨識的方法，利用簡單的極低的樣本數，達到了九成以上的辨識成功率。縮小了樣本數，不僅成功的大幅降低運算的處理時間，同時亦減低的記憶體耗用，這表示著我們可以用更少的硬體資源，達到了相同的辨識成功率。這意味著或許不久的將來，可能把整個系統移植至嵌入式上運算，而且能達到與電腦上相同的成果。

完整的車牌辨識系統所應包涵的三大項功能，在本論文已經都全數完成。在過去礙於硬體上製程的技術，使得嵌入式系統和一般個人電腦間，硬體的資源差異十分懸殊，但如今製程上的技術不斷的提昇，再加上本論文的演算法利用了較低的硬體資源，亦能達到頗高的辨識率，在未來的發展，或許真的可以嚐試把個車牌辨識系統，移植至嵌入式上運行，並且可以達到相同的成效。



## 參考文獻

- [01]. 張智欽, “移動中車牌定位之嵌入式系統”, 國立勤益科技大學電子工程學系碩士論文, 民國 98 年。
- [02]. k. Yalcun and M. Gokmen, “Visual license plate tracking,” Proceedings of the 12<sup>th</sup> IEEE Conference on Signal Processing and Communications Applications, 2004, pp. 623-626.
- [03]. V. Shapiro and G. Gluhchev, “Multinational license plate recognition system: segmentation and classification,” Proceedings of the 17th International Conference on Pattern Recognition, 2004, vol. 4, pp. 352-355.
- [04]. Z. Qin, S. Shi, J. Xu, and H. Fu, “Method of license plate location based on corner feature,” Proceedings of the 6th World Congress on Intelligent Control and Automation, 2006, vol. 2, pp. 8645-8649.
- [05]. Y. A. Syed and M. Sarfraz, “Color edge enhancement based fuzzy segmentation of license plates,” Proceedings of the 9th International Conference on Information Visualisation, 2005, pp. 227-232.
- [06]. P. Dubey, “Heuristic approach for license plate detection,” Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, 2005, pp. 366-370.
- [07]. T. S. Chen, H. H. Wu, and C. H. Lai, “Multi-target car license plate



- detection from complex environment,” Proceedings of the IEEE-Eurasip Conference on Nonlinear Signal and Image Processing, 2005, pp. 34.
- [08]. S. Zhang, M. Zhang, and X. Ye, “Car plate character extraction under complicated environment,” Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2004, vol. 5, pp. 4772-4726.
- [09]. S. Yohimori, Y. Mitsukura, M. Fukumi, N. Akamatsu, and N. Pedrycz, “License plate detection system by using threshold function and improved template matching method,” Proceedings of the IEEE Annual Meeting of the Fuzzy Information, 2004, vol. 1, pp. 357-362.
- [10]. T. D. Duan, D. A. Duc, and T. L. H. Du, “Combining Hough transform and contour algorithm for detecting vehicles’ license-plates,” Proceedings of the International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004, pp. 747-750.
- [11]. Q. Wu, H. Zhang, W. Jia, X. He, J. Yang, and T. Hintz, “Car plate detection using cascaded tree-style learner based on hybrid object features,” Proceedings of the IEEE International Conference on Video and Signal Based Surveillance, 2006, pp. 15-20.
- [12]. D. Zheng, Y. Zhao, and J. Wang, “An efficient method of license plate location,” Pattern Recognition Letters, In Press, Corrected Proof, Available online, 22 June 2005.

- [13]. S. H. Park, K. I. Kim, K. Jung, and H. J. Kim, "Locating car license plate using neural networks," *Electron. Letters*, 1999, vol. 35, NO. 17, pp. 1475-1477.
- [14]. G. H. Ming, A. L. Harvey, and P. Danelutti, "Car number plate detection with edge image improvement," *Proceedings of the 4th International Symposium on Signal Processing and Its Applications*, 1996, vol. 2, pp. 597-600.
- [15]. N. Zimic, J. Ficzkowski, M. Mraz, and J. Virant, "The fuzzy logic approach to car number plate locating problem," *Proceedings of the International Conference on Intelligent Information System*, 1997, pp. 227-230.
- [16]. S. Luis, M. Jose, R. Enrique, and G. Narucuso, "Automatic car plate detection and Recognition through intelligent vision engineering," *Proceedings of the IEEE 33rd International Conference on Security Tech.*, 1999, pp. 71-76.
- [17]. S. L. Chang, L. S. Chen, Y. C. Chung, and S. W. Chen, "Automatic license plate recognition," *IEEE Transactions on Intelligent Transportation Systems*, March 2004, vol. 5, no. 1, pp. 42-53.
- [18]. Y. Cui and Q. Huang, "Character extraction of license plates from video," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 502-507.
- [19]. H. Hontani and T. Koga, "Character extraction method without prior knowledge on size and position information," *Proceedings of the IEEE International Conference on Vehicle Electronics*, 2001, pp.

67-72.

- [20]. 呂炎州，「不需要字元切割的車牌辨識法」，靜宜大學資訊管理學系碩士論文，民國 93 年。
- [21]. J. Zhang, X. Fan, and C. Huang, "Research on characters segmentation and characters recognition in intelligent license plate recognition system," Proceedings of the Conference on Chinese Control, 2006, pp. 1753-1755.
- [22]. R. Dogaru, I. Dogaru, and M. Glesner, "Emergent computation in semi-totalistic cellular automata: applications in character segmentation," Proceedings of the International Conference on Semiconductor, 2004, vol. 2, pp. 451-454.
- [23]. Y. Feng, M. Zheng, and X. Mei, "A novel approach for license plate character segmentation," Proceedings of the IEEE International Conference on Industrial Electronics and Applications, 2006.
- [24]. N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on System, Man and Cybernetics, 1979, vol. 9, NO. 1, pp. 62-66.
- [25]. Y. Feng, M. Zheng, and X. Mei, "A novel binarization approach for license plate," Proceedings of the IEEE Conference on Industrial Electronics and Applications, 2006.
- [26]. Gonzalez, Digital image processing 2/e, Addison-Wesley Publishing Company, 2002.

- [27]. S. Mao, X. Huang, and Min Wang, "An adaptive method for Chinese license plate location," The 8th World Congress Intelligent Control and Automation (WCICA), 2010, pp. 6173-6177.
- [28]. Jing-Ming Guo and Yun-Fu Li, "License Plate Localization and Character Segmentation With Feedback Self-Learning and Hybrid Binarization Technique," IEEE Transactions on Vehicular Technology, vol. 57, 2008, pp. 1417-1424.
- [29]. Wu Guo-ping, Cheng Shi, Ao Min-si, and Lei Hui, "Slant Correction of Vehicle License Plate Based on Feature Point and Principal Component Analysis," International Conference on Computer Science and Software Engineering, 2008, pp. 487-490.
- [30]. Chu-Duc Nguyen, Mohsen Ardabilian, and Liming Che, "Real-time license plate localization based on a new scale and rotation invariant texture descriptor," International IEEE Conference on Intelligent Transportation Systems (ITSC), 2008, pp. 956-961.
- [31]. 蔡銘鑫, "小波轉換和類神經網路應用於車牌辨識" 朝陽科技大學資訊工程學系碩士論文, 民國 93 年。
- [32]. J. A. G. Nijhuis, M. H. ter Brugge, K. A. Helmholt, J. P. W. Plum, L. Spaanenburg, R. S. Venema, and M. A. Westenberg, "Car license plate recognition with neural networks and fuzzy logic," IEEE International Conference on Neural Networks, 1995, vol. 5, pp. 2232-2236.

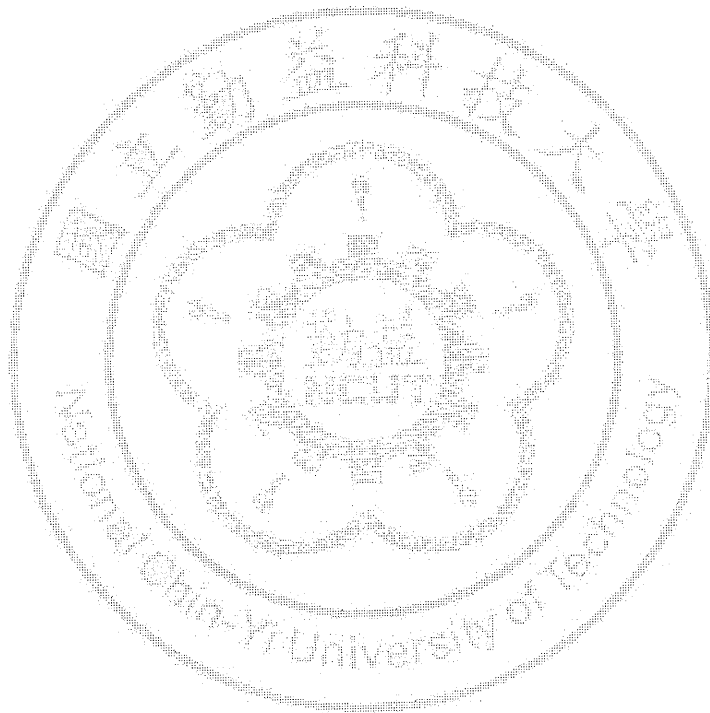
- [33]. C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, E. Kayafas, "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications," *IEEE Transactions on Intelligent Transportation Systems*, 2006, vol. 7, pp. 377-392.
- [34]. C. Wu, L. C. On, C. H. Weng, T. S. Kuan and K. C. Ng, "A macao license plate Recognition System," *Proceedings of the 4th International Conference on Machine Learning and Cybernetics, Guangzhou, 2005*, vol. 7, pp. 4506-4510.
- [35]. Y. P. Huang, S. Y. Lai, and W. P. Chuang, "A Template-based model for license plate recognition," *IEEE International Conference on Networking, Sensing and Control*, 2004, vol. 2, pp. 737-742.
- [36]. P. Comelli, P. Ferragina, M. N. Granieri, and F. Stabile, "Optical recognition of motor vehicle license plates," *IEEE Transactions on Vehicular Tech.*, 1995, vol. 44, pp. 790-799.
- [37]. D. Pruegsa and T. Arit, "Thai Vehicle License Plate Recognition Using the Hierarchical Cross-correlation ARTMAP," *IEEE International on Intelligent Systems*, 2006, pp. 652-655.
- [38]. H. A. Hegt, R. J. de la Haye, and N. A. Khan, "A high performance license plate recognition system," *IEEE International Conference on Systems, Man, and Cybernetics*, 1998, vol. 5, pp. 4357-4362.
- [39]. S. N. H. S. Abdullah, M. Khalid, R. Yusof, and K. Omar, "License plate recognition using multi-cluster and multilayer neural networks," *Proceedings of the Information and Communication*

- Technologies, 2006, vol. 1, pp. 1818-1823.
- [40]. X. Chi, J. Dong, A. Liu, and H. Zhou, "A simple method for Chinese license plate recognition based on support vector machine," Proceedings of the International Conference on Communications, Circuits and Systems, 2006, vol. 3, pp. 2141-2145.
- [41]. V. Seetharaman, A. Sathyakhala, N. L. S. Vidhya, and P. Sunder, "License plate recognition system using hybrid neural networks," Proceedings of the IEEE International Annual Meeting of the Fuzzy Information, 2004, vol. 1, pp. 363-366.
- [42]. M. H. T. Brugge, J. H. Stevens, J. A. G. Nijhuis, and L. Spaanenburg, "License plate recognition using DTCNNs," Proceedings of the 5th IEEE International Workshop on Cellular Neural Networks and Their Applications, 1998, pp. 212-217.
- [43]. Y. Cheng, J. Lu, and T. Yahagi, "Car license plate recognition based on the combination of principal components analysis and radial basis function networks," Proceedings of the 7th International Conference on Signal Processing, 2004, vol. 2, pp. 1455-1458.
- [44]. T. Nukano, M. Fukumi, and M. Khalid, "Vehicle license plate character recognition by neural networks," Proceedings of the International Symposium on Intelligent Signal Processing and Communication Systems, 2004, pp. 771-775.
- [45]. 魏銷志, "動態多標的車牌辨識系統之研究", 元智大學資訊


研究所碩士論文，民國 89 年。

[46]. 李奇霖，“車輛牌照自動辨識系統-使用類神經網路”，淡江大學資訊工程學系碩士論文，民國 88 年。

[47]. 江東毅，“由影像輸入之機械臂書法系統”，國立台灣科技大學電機工程研究所系碩士論文，民國 91 年。



## 作者簡介

中文姓名:	羅 智 群	
Name:	Zhi-Chun Luo	
生日:	1986 年 01 月 28 日	
籍貫:	臺灣 嘉義縣	
學歷:	高中 國立嘉義高級中學  大學 南華大學 資訊工程學系  研究所 國立勤益科技大學 電子工程研究所	
論文發表:	<ol style="list-style-type: none"> <li>1. <b>Chen-Chung Liu, Zhi-Chun Luo</b>, Chih-Chin Chang, "Multiple Vehicle License Plates Localization and extraction of Vehicle License Plate Number for Moving Vehicles," <i>International Journal of Advanced Information Technologies (ISSN: 1996-2568)</i>, Vol. 4, No.2, pp.136-151, December, 2010.</li> <li>2. <b>Chen-Chung Liu, Zhi-Chun Luo</b>, "Vehicle License Plate Detection and Character Segmentation System Synthesized with Embedded System PXA270," <i>2010 International Conference on High-Speed Circuits Design (ISBN: 978-957-21-7823-2)</i>, Taiwan, pp. 39-47, October 25-29, 2010.</li> <li>3. <b>Chen-Chung Liu, Zhi-Chun Luo</b>, "Extraction of Vehicle License Plate Number Using License Plate Calibration," <i>The IET International Conference on Computing Theory, Technologies and Applications (ISBN: 9781849192088)</i>, Taiwan, pp. 187-192, August 4-6, 2010. [indexed by EI Compendex and IET]</li> <li>4. <b>Chen-Chung Liu, Zhi-Chun Luo</b>, "An extraction Algorithm of Vehicle License Plate Numbers Using Pixel Value Projection and License Plate Calibration," <i>2010 International Symposium on Computer, Communication, Control and Automation (IEEE</i></li> </ol>	



*Catalog number: CFP10741-PRT, ISBN: 9781424455669*), Tainan, Taiwan, pp. 256-259, May 5-7, 2010. **[indexed by EI Compendex and ISTP]**

5. **Chen-Chung Liu, Zhi-Chun Luo** , Chih-Chin Chang, "Multiple Vehicle License Plates Localization and extraction of Vehicle License Plate Number for Moving Vehicles," *2010 International Conference on Advanced Information Technologies (ISBN: 9789867043306)*, Taichung, Taiwan, pp. 1-12, April 23-24, 2010.
6. **Chen-Chung Liu, Zhi-Chun Luo**, Chen Lin, "A Vehicle License Plates Recognition System Based on Maximum Score Decision Principal," *2010 The 4<sup>th</sup> Cross- Strait Technology, Humanity Education and Academy- Industry Cooperation Conference (ISBN: 9789860255478)*, Taichung, Taiwan, pp. EEE9-1~ EEE9-16, December 4, 2010.
7. **Chen-Chung Liu, Zhi-Cyun Luo**, "The Design and Implementation of a Web Page Replacement Server for Mobile Devices," *2009 Workshop on Consumer Electronics (ISBN: 978-957-21-7330-5)*, Taichung, Taiwan, pp. 1155-1161, December 6, 2009.