

車牌辨識系統之實作

學生：林郁佐

指導老師：林峻鋒

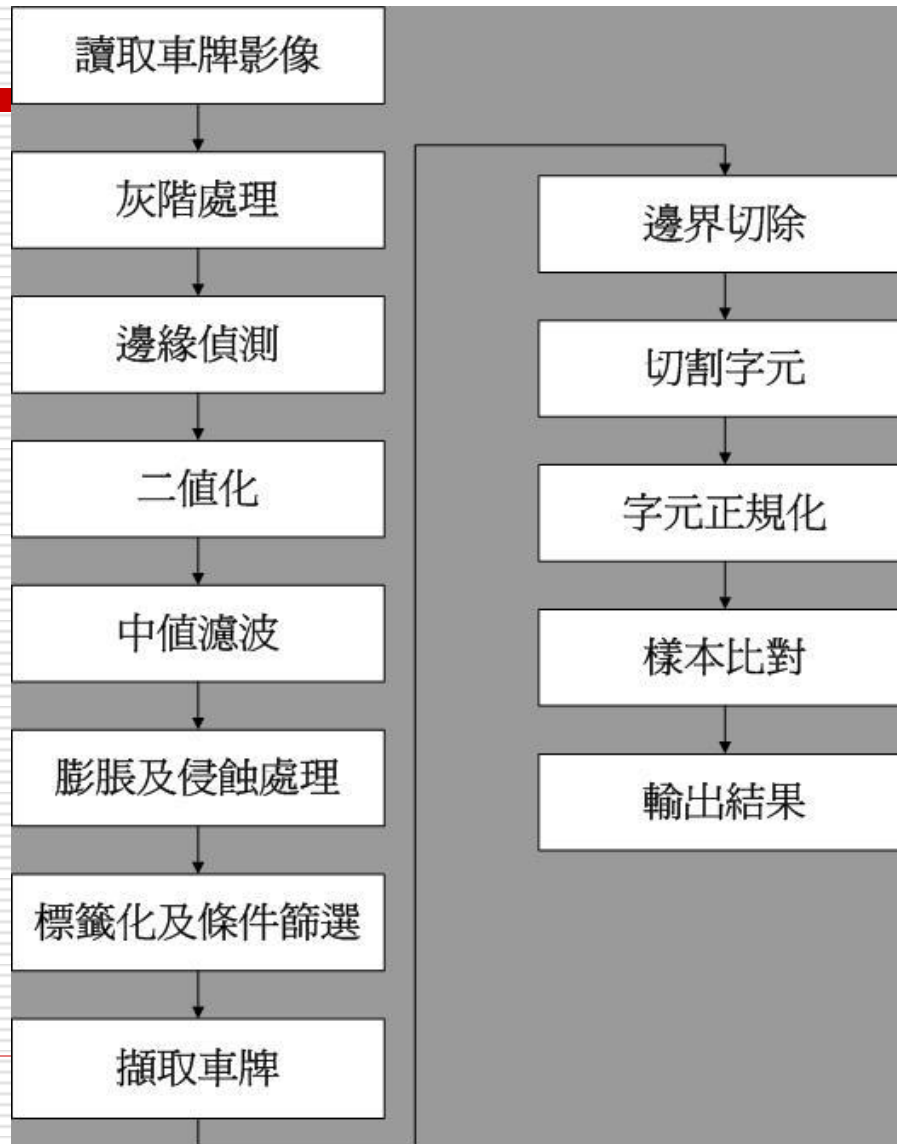
大綱

- 研究動機
 - 系統流程
 - 實作步驟
 - 實驗結果
 - 未來研究方向
-

研究動機

由於汽車已經變成人們主要的交通工具，車輛的管理顯得更為重要，傳統以人工進行管理的方法不符合經濟成本，於是想藉由對車牌的辨識，應用於對車輛的管理。如此一來可以大大地減少人力，在系統技術成熟的情況下，也能避免人為錯誤的發生

系統流程



灰階處理

$$\text{Gray} = 0.3R + 0.59G + 0.11B$$



原始影像



灰階影像

邊緣偵測

邊緣指的是灰階值落差極大的地方，而車牌區塊內的字元部份，會形成許多密集的垂直邊緣，這個特性讓我們能夠對車牌區域進行尋找。

本專題使用Prewitt遮罩取得影像內明顯的垂直邊緣。

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

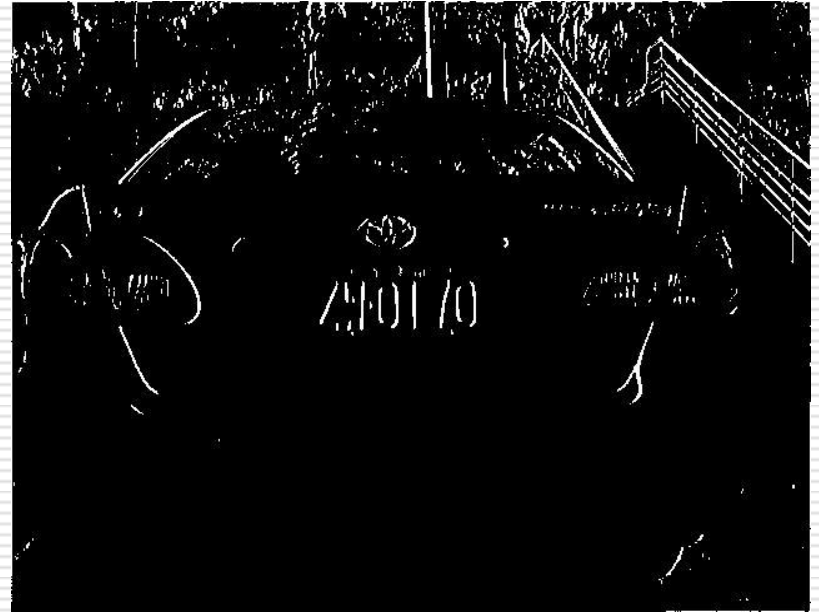
Prewitt遮罩係數



經過Prewitt遮罩後

二值化

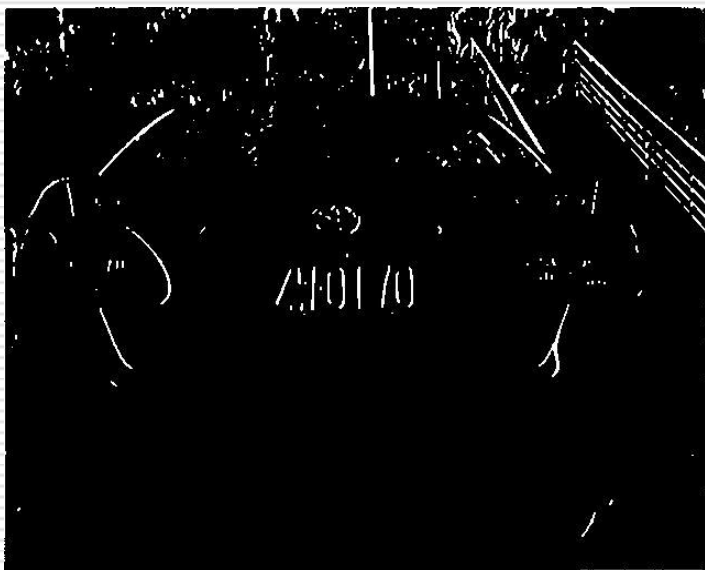
先得到影像中最大跟最小的像素值，
其總合的一半即為初始的門檻值 T ，
接著將影像裡大於此門檻值的像素值
加總起來取其平均值 O_{sum} ，
小於此門檻值的像素也同樣加總起來
取得其平均值 B_{sum} ，再把 O_{sum} 跟 B_{sum}
加起來除以二得到 T_n ，若 T_n 不等於 T ，
則將門檻值 T_n 設為新的門檻值 T ，
直到 T_n 等於 T 為止，
這樣取得的 T 即是當作二值化依據的門檻值。



經過二值化以後的影像

中值濾波

中值濾波的目的是消除影像內的雜訊，其原理是找出相鄰八個像素的灰階值，並將目前的點設成八個像素的中間值。



經過一次中值濾波



經過二次中值濾波

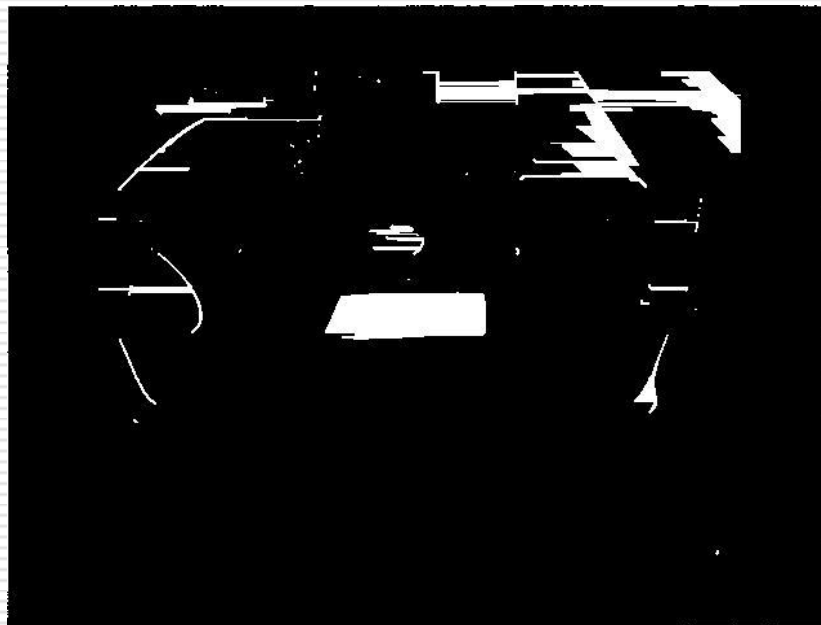
膨脹及侵蝕處理

膨脹處理:若一定範圍內有任一點為白點，則目前這點亦為白點

侵蝕處理:若一定範圍內的所有點皆為白點，則目前這點才為白點

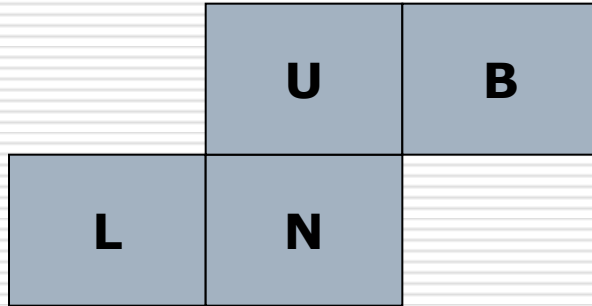


經過膨脹處理



經過侵蝕處理

標籤化



L:目前像素的左邊

N:目前的像素

U:目前像素的上面

B:目前像素的右上方

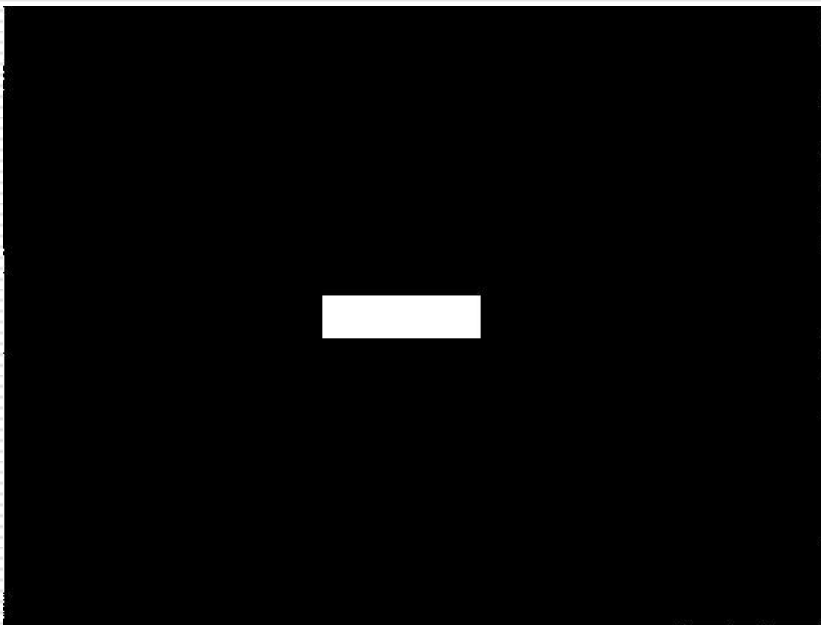
M:標籤

```
for(影像裡的每一列){  
  for(影像裡的每一行){  
    N=目前的像素且為白點  
    if(如果L、U、B皆未有標籤)  
      N=新的標籤  
    else if(B已標籤且L、U皆未標籤)  
      N=B的標籤;  
    else if(U已標籤且L未標籤)  
      N=U的標籤;  
    else if(L已標籤且U未標籤)  
      N=L的標籤;  
    else if(L、U皆有標籤且標籤相同)  
      N=U的標籤;  
    else if(L、U皆有標籤且標籤不同)  
      N=U的標籤;  
    將此條水平線目前像素之前  
    跟L一樣的標籤設成U的標籤  
  }  
}
```

條件篩選

- 一. 區塊的像素總合必須大於 **1600 pixels** 。
 - 二. 區塊的長寬比例要在 **1.5~5.0** 之間。
 - 三. 從經過上面兩個條件篩選以後的區塊中，選擇最下方的區塊。
-

條件篩選(續)



經過篩選後的區塊



擷取出的車牌影像

邊界切除



原始車牌影像

上下邊界的切除線找尋方法為從中點往左右延伸
30個pixels，往上下水平尋找，直到找到整條
水平線為白色或白點個數大於90%



切除上下邊界

從左邊往右邊搜尋每條垂直線，一但發現整條
垂直線白點個素少於95%，則停止搜尋並將
此條垂直線標記為切除線



切除左右邊界

切割字元



字元切割線

切割字元是每次找出字元切割的起點跟終點再進行切割，每次切割完的終點即為下一個字元切割的起點，而切割的終點為起點加上一個字元的寬度，在起點跟終點附近找到黑白像素差距最大的垂直線，即為字元的切割線，重複此步驟直到切出六個字元。另外，針對字元「-」及「1」有進一步的處理。



切割出的字元

字元正規化

本專題採用線性正規化，將字元正規化成**15*45**的大小，
正規化的方法是以**15*45**的座標推斷欲正規化字元
在此大小裡的座標是黑點或白點。

The image displays six characters, 'Z', 'W', '0', '1', '7', and '0', rendered in a highly stylized, pixelated font. Each character is composed of a grid of black and white pixels, representing a binary or binary-coded decimal (BCD) format. The characters are arranged horizontally and are all the same size, reflecting the normalization process described in the text.

正規化以後的字元

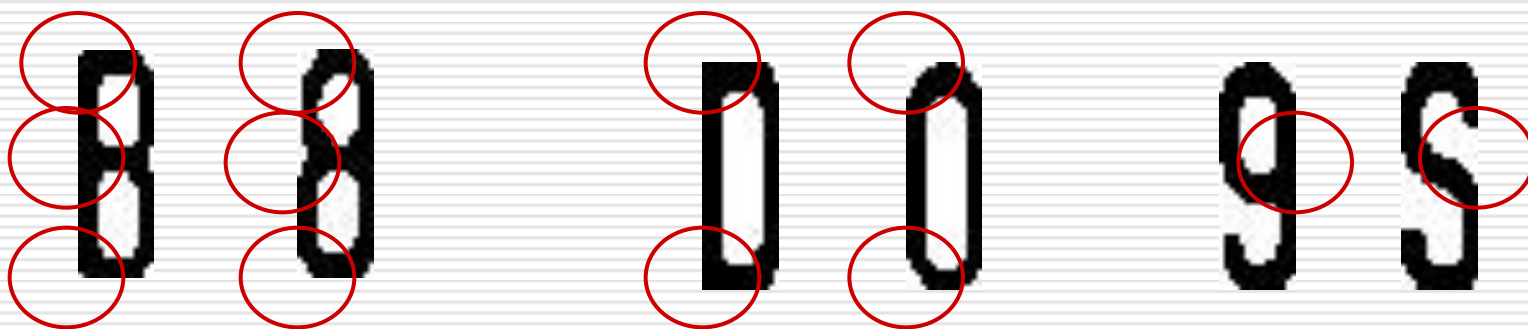
樣本比對

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

樣本字元

樣本比對(續)

針對容易辨識錯誤的幾組字元，進行第二次重點區域的比對。



實驗結果

實驗車牌影像:131張。

辨識正確張數:81張。

辨識率:61.83%。

總共辨識字元:672個。

辨識正確字元:604個。

字元辨識率:89.88%。



系統介面

未來研究方向

本專題的辨識率不盡理想，其原因可能是因為影像的品質包括陰影、角度、亮度的影響，或是影像處理的演算法，專題所使用的一些演算法未必能在每張影像裡得到預期的結果，未來將克服影像品質所帶來的影響，如；傾斜角度的矯正、影像對比度的增強。另外，將會尋找或者自行設計更能達到預期結果的影像處理演算法，待車牌的辨識率達到一定目標以後，將會再搭配相關硬體，如攝影機，以能夠構成一套完整的車牌辨識系統。
