

基於Hough轉換與Haar特徵串聯分類器的車牌偵測方法

License Plate Detection using Hough Transform and Haar-like Cascade Classifiers

謝禎問、張治強
大同大學資訊工程學系
中山北路3段40號
台北市104中山區
cchsieh@ttu.edu.tw

Chen-Chiung Hsieh and Chih-Chiang Chang
Dept. of Computer Science and Engineering, Tatung University
40, Sec. 3, Zhong-Shan N. Rd.,
104 Taipei, Taiwan
cchsieh@ttu.edu.jp

摘要

爲了得到監視器畫面中車牌的資訊，我們提出一個系統來對車輛的車牌部份做偵測與定位，道路監控的畫面中，車子不會像停車場這種場合可以讓我們取得較好的畫面，所以面臨旋轉車牌影像的偵測是必須考量的。我們先利用黑白相間的Haar-like特徵訓練cascaded分類器(Classifier)，據以找出畫面中的車牌，並且框出車牌的位置。首先將運動物件以背景相減(Background Subtraction)與連通區塊(Connected Component Labeling)抽取出來，並對其做霍氏轉換(Hough Transform)，找出旋轉角度，並將圖片校正。然後再以訓練好的分類器對校正過後的圖片作車牌偵測。我們提出的方法，以314/394張車牌正/負樣本做訓練，以upright和skewed Haar-like特徵訓練出upright和skewed車牌分類器，再以860張影像做測試，實驗結果的偵測率達87.4%，驗證本系統之可行性。對於水平旋轉所造成一般車牌分類器偵測不到車牌的情況，會有不錯的改善。

關鍵詞: 道路監控系統、車牌偵測、黑白特徵、串聯分類器、霍氏轉換。

Abstract

In this paper, we developed a system to automatically detect and locate license plates of vehicles. The vehicle in the real world traffic surveillance would have rotated license plate in the captured image. Due to the black-white patterns in the license plate image, cascaded classifiers using Haar-like features could be trained to detect license plate. If there is any vehicle extracted by background subtraction and connected component labeling, we can do Hough transform on the extracted vehicles for rotation correction. License plate of the corrected image could be then detected by the trained cascaded classifier. Our proposed system

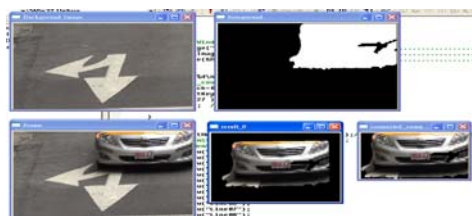
was trained with 314/394 positive/negative samples and tested with 860 vehicle images. The detection rate was 87.4% which demonstrated the feasibility of the proposed system.

Keywords: Traffic Surveillance, License Plate Detection, Haar Features, Cascaded Classifier, Hough Transform

一、引言

隨著經濟的發展，人口的成長，在道路上架設的攝影機越來越普遍。時代在進步，影像處理的技術也越來越成熟，於是以計算機程式即時監控路況來取代人工的調閱監控便成為可行，如Lai *et al.*[1]提出監控路況是否有逆向行駛的狀況發生。如果偵測到違規情況，我們希望能夠一樣由電腦程式自動取得車牌資訊，若能搭配PTZ(Pan/Tilt/Zoom)攝影機，將違規的車輛影像擷取出來放大，將有助於取得車牌資訊。

一般車牌辨識系統(License Plate Recognition System, LPRS)[2] 包含三大部分，車牌偵測(License Plate Detection, LPD)與定位(License Plate Location, LPL)，車牌字元的切割(Character Segmentation, CS)以及車牌字元的辨識(Character Recognition, CR)。其中車牌的偵測與定位，在LPRS中扮演著重要的角色，當LPL做的夠準確時，可以利用水平垂直的投影將字元自車牌區域中一個一個分割出來，分割出的字元再透過分類的方法分到0~9與A~Z，共36種的分類當中。本論文先抽取運動物體如車子等(如圖一)，再以Hough transform做旋轉校正，之後再以車牌分類器偵測，藉以降低旋轉問題對車牌偵測的影響。



圖一：偵測畫面中擷取運動物體。

二、相關研究

LPRS是個已經研究很久的主題，在車牌偵測與定位的部分，有學者[3] 以型態學(Morphology-based)的方式，定義車牌的特性為一個高對比度的區域，其灰階度會產生劇烈變化且會在小範圍內頻繁發生，透過侵蝕與膨脹等運算可以用來找出車牌的所在。也有學者以AdaBoost(Adaptive Boosting)演算法，透過學習的過程來訓練車牌分類器，再以搜尋視窗(Search Window)來掃描圖片，分辨搜尋視窗內的判斷結果為車牌或者非車牌。

1. 非基於AdaBoost之車牌偵測

Guo *et al.* [4]將RGB的色彩畫面轉換成HSI的色彩空間，接著利用Intensity-map得到具有邊緣資訊Edge-map。然後以High-Gradient-Averaging(HGA)的方法過濾出可能的車牌區域。HGA就是將Edge-map裡面梯度(Gradient)值比平均梯度值低的區域去除。這個方法可以重複做下去，Guo *et al.*是做兩次HGA。取出大致的可能車牌區域之後，再以車牌區域長寬比以及通過該區域的黑白相間的變換次數來判斷是否為車牌。以台灣的车牌

為例，正常黑白相間的次數為6到14(如圖二)，以此可以設定為過濾的門檻值。這個方法準確率可達97.1%，辨識率很高，然而這個方法的處理速度不夠快，不適合用在即時偵測。



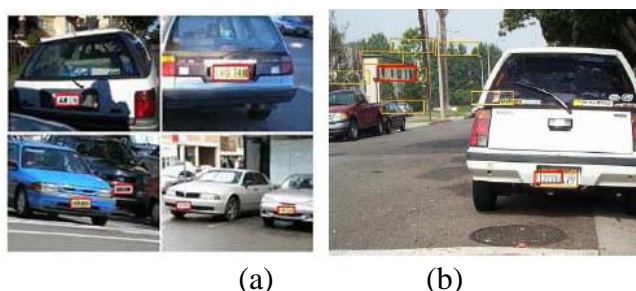
圖二: 黑白間隔次數[3]。(a)6次。(b)14次。

Chen *et al*[5] 提出一個利用Fractal dimension為基礎的車牌偵測系統，不過需要先將車牌影像正規化(Normalization)，再選擇適當的影像增強方法處理後，才可以去計算基於Fractal Dimension和其區間數值於可能車牌區域，所統計的值可據以判斷是否為車牌區域。

2. 基於AdaBoost之車牌偵測

Viola和Jones[6]提出一個AdaBoost訓練人臉分類器以偵測人臉的方法，He *et al.*[7]套用在車牌上，訓練出車牌分類器，以作為車牌的偵測，他們先做邊緣偵測，再將每一個點的垂直邊個數加起來，計算一個區域的邊密度如公式(1)，EV為通過點 (i, j) 的垂直邊數目，然後將邊密度當作全域特徵。再將Haar-like特徵當作是局部特徵，結合兩種特徵訓練出Cascade分類器，偵測結果如圖三，有些結果並不好，推測是由於在訓練時只訓練到6個階層的緣故。

$$D_E = \frac{1}{N} \sum_i \sum_j E_v(i, j) \quad (1)$$

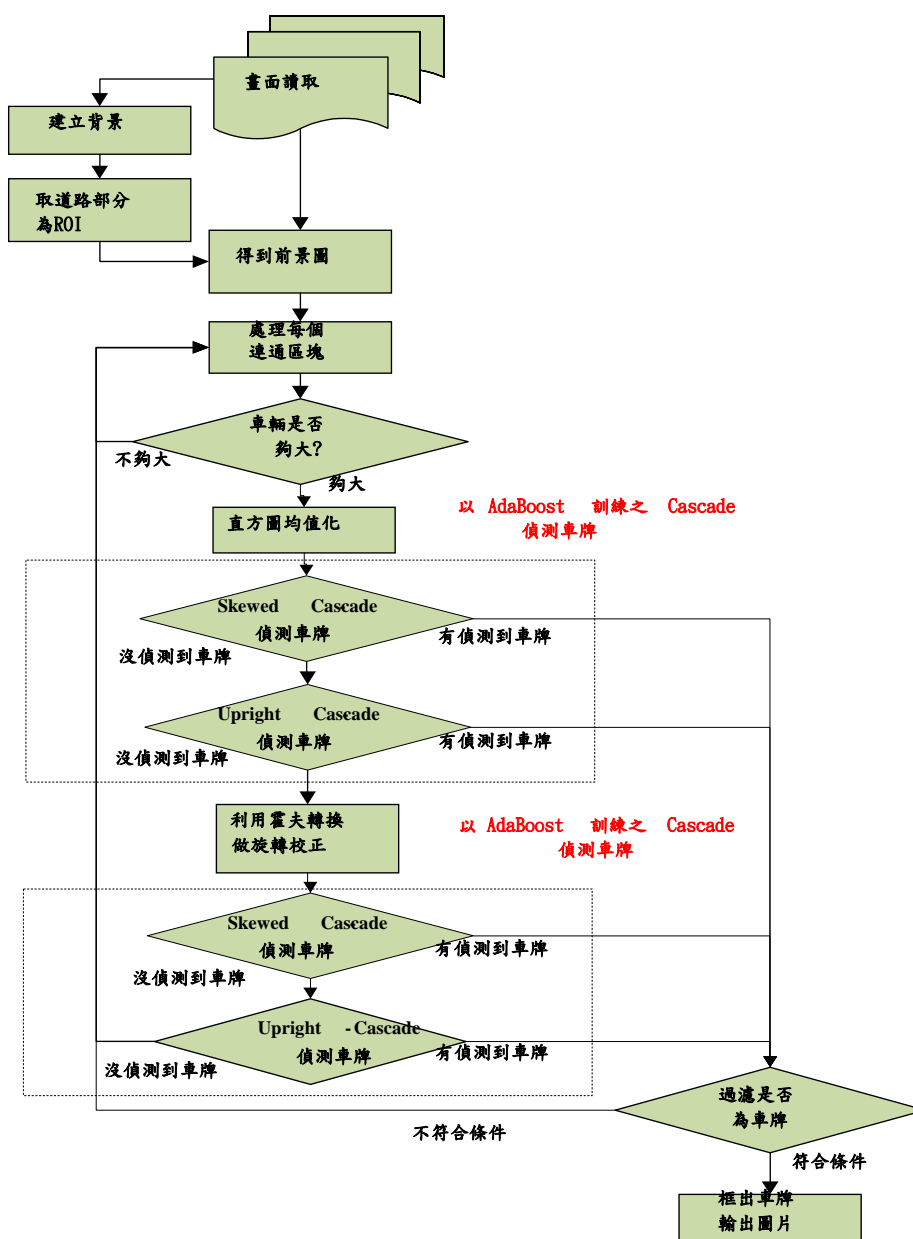


圖三: 偵測結果[7]。(a)簡單背景偵測車牌效果不錯。(b)背景複雜情況下會比較容易發生誤框車牌。



圖四: 左邊為正的Haar-like特徵，右邊為斜的Haar-like特徵[8]。

Wang和Lee[8]除了正的(Upright) Haar-like特徵，還加上斜的(Skewed) Haar-like特徵(如圖四)，以加入的Skewed Haar-like特徵一起訓練Cascade分類器，最後的Cascade分類器其辨識率可達到99.857%，這個方法的辨識率已經非常的高，然而對於旋轉角度大的車牌還是無法偵測出來，例如超過45度。雖然文獻[8]並未說明極限，但是依常理判斷超過45度應該是無法辨識的。本論文也是設定以45度為車牌旋轉最大角度，超過則判斷屬於垂直方向。



圖五: 系統流程圖。

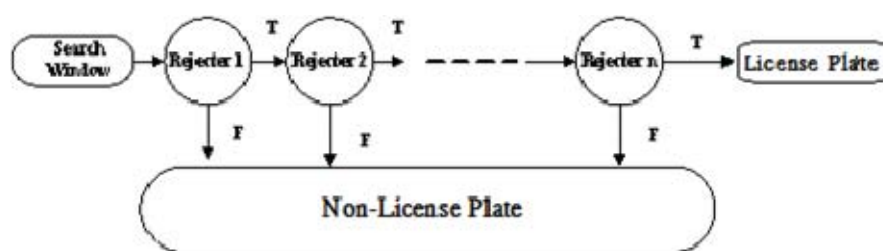
三、系統架構與實作方法

系統流程圖如圖五，我們對畫面建立背景模組，接著取得畫面中有移動的影像為前景，在前景中我們對每個連通區塊(Connected Component)作處理，包括侵蝕膨脹，以及根據面積大小來過濾，面積大小視所擷取到的車子大小而定，而車子的面積大小也與攝

影機與車子的距離有關，在本論文中，我們設定為1000 pixels。接著取出一個一個連通區塊，然後對應回原圖得到運動物體的影像，對其做Hough Transform，如果車輛有旋轉，我們就將圖片校正。然後以我們的分類器再次對校正過後的圖片做偵測車牌的動作。先以Skewed-cascade偵測，若沒有偵測到才接著以Upright-cascade偵測。

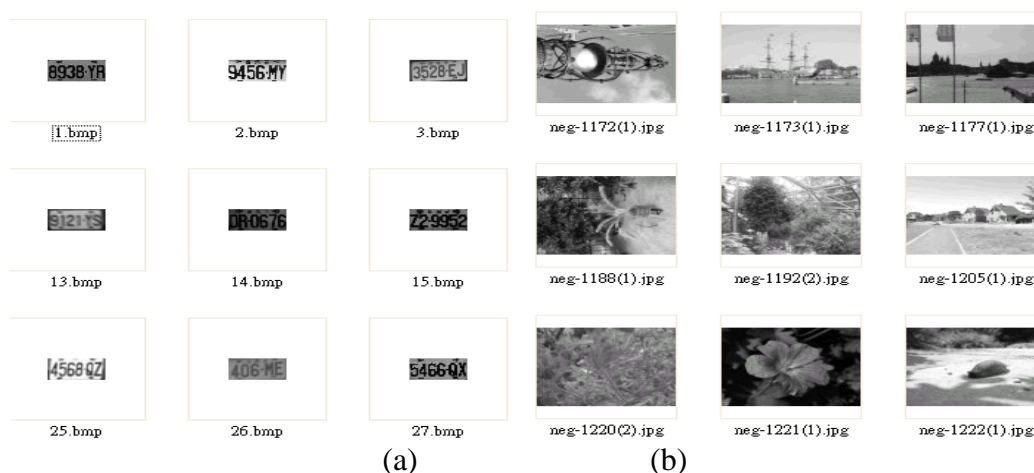
1. 訓練車牌分類器

如同人臉Cascade分類器，我們希望做出一個由多個分類器集合而成的Cascade分類器，透過這個分類器我們可以分辨待測區域是不是車牌。車牌Cascade分類器示意如圖六。



圖六: 車牌Cascade分類器示意圖。

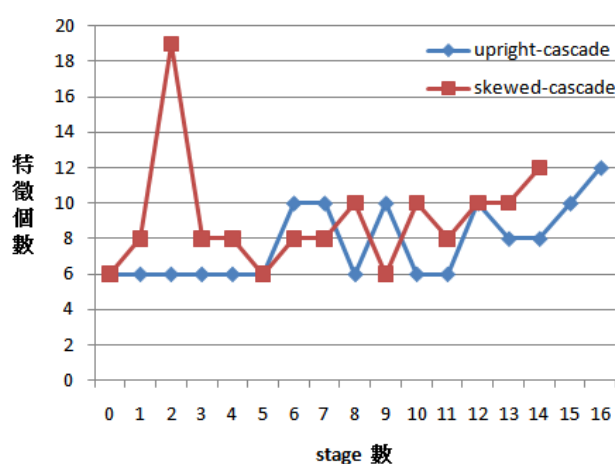
我們的目標物件不是人臉，而是車牌，以314張40×20之無旋轉車牌影像為正樣本，394張不含車牌影像之任意圖片為負樣本，負樣本的選擇可以是任意大小，但是樣本之間的差異性要越大越好，圖七為部份的訓練樣本。我們使用OpenCV(Open Source Computer Vision)的haartraining函式 [9] 來做訓練，訓練出stage為16之無旋轉車牌Cascade檔案，每一個Stage就是圖六中的一個Rejecter，為了方便，後面稱這個Cascade為Upright-Cascade。在Training過程中，目標物體越小，耗費時間較短，在往後的偵測時，無法偵測到比訓練時所用圖片小的物體，即使未完成設定的訓練過程，將已訓練好的cascade classifier取出也可直接使用。



圖七: 部份訓練樣本。(a)正樣本(b)負樣本。

2. Upright-Cascade與Skewed-Cascade

相關研究[7]提到，除了正的(Upright) Haar-like特徵，可以再加上其它斜的(Skewed) Haar-like特徵，所以我們對同樣一份的樣本，包括正樣本314張，負樣本394張，在訓練的時候考慮了Skewed的Haar-like特徵，重新訓練出一個Cascade分類器。爲了方便，訓練出的Cascade分類器，我們稱之爲Skewed Cascade。將訓練所得到的兩個XML檔案分析後，可以取出此兩Cascade分類器的分類階層(Stage)與每一階層所用到的特徵個數，如圖八所示。觀察兩者的訓練結果，我們發現，Upright Cascade和Skewed Cascade之間有些差異(如圖八)。差異的原因是因爲Skewed分類器使用較多的訓練樣本，相對特徵個數應該較多。可以看到Skewed Cascade在訓練到第二個Stage的時候用到了相對起來個數多很多的特徵數目(19個)，我們預計這會造成Skewed Cascade偵測較爲準確，但是花費時間會比較多。



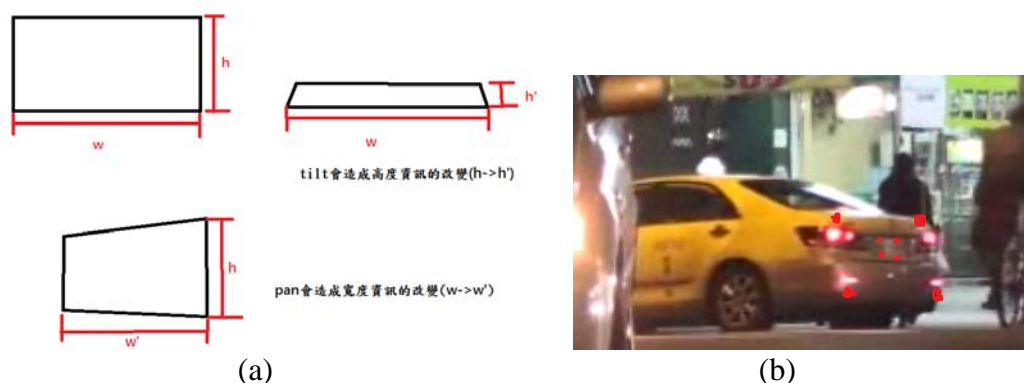
圖八: Upright-cascade與Skewed-cascade的stage數以及特徵個數。

我們使用兩種Cascade來偵測車牌，實驗的結果會在第4節詳細說明，簡單的說明觀察結果，就是與我們根據圖8所推測的結果一樣，Upright Cascade偵測車牌的TP(True Positive)偵測率比Skewed Cascade高，花費時間也比Skewed Cascade短，但是誤判(False Positive, FP)的比率比Skewed Cascade來的高。由於誤判的部分須要透過過濾程式排除，所以爲了時間的花費，我們在實作的時候，對每張影像先以Skewed Cascade判斷一次是否有車牌，如果沒有，才以Upright Cascade再次的做偵測。

3. 偵測旋轉車牌-以Hough Transform進行旋轉校正

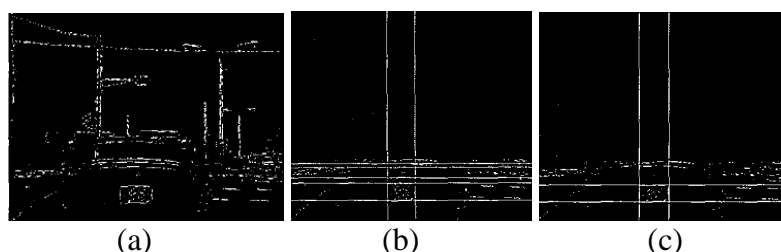
如果車牌不正就會很直接的影響我們的偵測與定位結果，所以我們希望能在LPL的動作之前就能先做好旋轉校正，能直接幫助我們接下來的偵測率。但是旋轉概分成旋轉方向與車牌面平行的水平旋轉(Rotation)，還有旋轉方向與車牌面垂直的旋轉，垂直旋轉是由上下傾斜(Tilt)和左右側轉(Pan)而合成。基於需要使用Cascade偵測車牌，車牌的形狀很重要，如果是垂直方向的旋轉，由於在三度空間中成像，會造成視覺上看到的車牌長寬比資訊的破壞(如圖九(a))，這種旋轉如果要校正，可能要去用別的方法找出車牌邊框，或者要找出車子本身精確的四個點資訊(如圖九(b))，接著才能校正回來，然後再使

用cascade去偵測車牌，但是如果能找到這些資訊，似乎也沒必要再用Cascade去偵測車牌了。所以我們只針對水平方向的旋轉作校正，我們希望用簡單的動作，得到最大的效益。



圖九: 車牌側轉。(a)側轉及傾斜會破壞長寬比例示意。(b)實例。

道路交通領域用到Hough Transform的研究相當多，因為交通號誌多是圓形，道路線是直線，這些都能利用Hough Transform，偵測需要的目標。Hough Transform由Paul Hough在1962年提出，1972年的時候Duda *et al.* [10] 提出利用Hough transform的方法來偵測圓弧或直線。Yanamura *et al.* [11] 在論文中直接使用Hough Transform對邊緣圖找出直線，再由這些直線過濾出最可能是車牌邊緣的直線來定位車牌(如圖十)。Wen *et al.* [12] 則提出以3D Transform 先對車牌影像作扭曲校正，再以Hough Transform來估算車牌的位置與旋轉的角度。

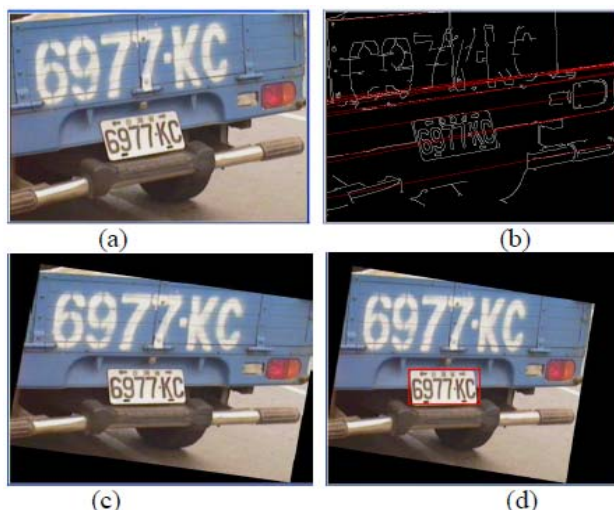


圖十: 利用Hough transform找車牌[11]。(a)對原圖做邊緣偵測(b)去掉上方區域部份，然後找出可能直線(c)過濾出車牌區域。

建立在車牌都是無旋轉的掛在車輛上的前提下，我們想出了一個很直觀的方法，雖然簡單但卻很有效的車牌校正法。相較一些事後的校正，例如找到車牌之後利用文字的水平軸或水平邊界來找直線校正角度，或者像人臉找到了五官之後再以兩眼角連線與水平軸來做校正，我們提出的想法是在偵測到我們的目標物之前就去做旋轉校正。最主要的用意就是我們旋轉的目的，是爲了幫助我們使用車牌分類器來偵測車牌，而不是爲了之後的字元分割或者字元辨識的工作。

在偵測到車輛的影像中，我們預計最長的直線都會是剛好與車牌的水平軸呈平行的直線段，因為對一般的轎車來講，垂直方向比較不易形成最長直線，大部分會有一個樓梯形的彎曲。經過Hough Transform測試，確實水平方向比較容易得到最長直線。我們將

Hough Transform的Vote參數設定100，並且在斜率過濾上設定與垂直軸夾角不超過經度0.52的直線就不考慮，然後對剩下的線條做42個區間來分類他們的經度，取眾數做整張圖片的旋轉校正。校正之後的圖片我們再透過cascade分類器對畫面偵測車牌，偵測到車牌之後將車牌定位，以紅框標記出來，整個流程以一個實際的例子來說明，如圖十一。



圖十一: 以Hough Transform對影像旋轉校正再偵測車牌實例。(a)原圖。(b)Hough Transform偵測直線。(c)旋轉校正。(d)校正後的偵測車牌。



圖十二: 車牌過濾實例。(a)黑白交替次數小於5，框選範圍將不會顯示為偵測到車牌。(b)黑白交替次數大於5，框選範圍將會顯示為偵測到車牌。

4. 車牌過濾

如我們先以Skewed-cascade來偵測車牌，如果找不到車牌再接著以Upright-cascade偵測一遍，需要作車牌過濾動作的時機，只有在偵測到車牌的時候，加上車牌過濾的機制，可以降低我們框錯的機率，也就是非車牌而誤框(False Positive, FP)的情況。由於我們訓練的Cascade，正樣本的大小是40x20，所以我們在偵測車牌可能區域的時候，搜尋框都是以40x20的比例去放大作搜尋，故一般常用的長寬比就沒必要再拿來過濾我們偵測出來的車牌區域。我們用來過濾的條件，第一個是最簡單，也很有效的，也就是設定面積的大小，以640x480的畫面，我們設定面積大於6000的都過濾掉，就能過濾掉本來一些FP的情況。

類似[3]所提出的做法，接著我們再以框出來的車牌區域為ROI(Region of interest)來做過濾的處理。首先我們算出區域內的直方圖分布，然後找出最高的峰值，計算峰值左邊或者右邊其餘值之總和哪邊較高，往該方向移動10個灰階度，並以該點為門檻值作二

值化。然後對圖片作中值濾波過濾雜訊，而後得到的影像，我們接著在其高度的中點作水平線，計算黑白交錯次數，我們設定至少要5次才是我們認為可能的車牌區域(如圖十二)。過濾的效果也是在第4節會詳細說明，大致上來說過濾的效果不錯，但是花費的時間不算低。

四、實驗結果與分析

本章節將介紹我們的一些實驗結果以及數據。包含我們實驗所使用的硬體設備，Upright-cascade(UC) 以及Skewed-cascade (SC) 的偵測效率與時間，還有以實驗數據說明正常車牌影像的整體偵測率，最後討論Hough transform處理對車牌偵測的影響以及處理時間。

1. 車牌過濾實驗環境

本系統所使用的開發平台為Windows Vista Home Premium，Intel®Core™i7 CPU 860 @2.8GHz的處理器，DDR3 8GB的記憶體。開發環境為Visual C++ 6.0，影像處理函式使用OpenCV 1.0版，影像大小都縮放為640×480 pixels的影像來處理。

2. Upright Cascade及Skewed Cascade的偵測效率與時間

我們總共測試了860張圖片，每張圖片都分別單純以Upright Cascade與Skewed Cascade作偵測。其中100張大小為320×240 pixels，660張大小為640×480 pixels，100張大小為2048×1536 pixels，我們都縮放為640×480的大小來處理。測試結果如表1。我們首先觀察到由320×240放大成640×480的圖片，偵測效果比較差，而由2048×1536縮小成640×480的偵測結果，跟原本就是640×480的圖片偵測結果比較接近，差異沒有太大。這應該是由於圖片放大會產生模糊化的現象，所以造成一些資訊的失真。

表一: Upright Cascade(UC)及Skewed Cascade(SC)偵測效率

圖片大小	320×240	640×480	2048×1536	平均
測試張數	100	660	100	
UC TP個數與百分比	45 45%	560 84.8%	68 68%	673 78.3%
UC FP個數與百分比	3 3%	62 9.4%	7 7%	72 8.4%
UC平均偵測時間(ms)	11.2	11.3	11.3	11.3
SC TP個數與百分比	31 31%	467 70.8%	58 58%	556 64.7%
SC FP個數與百分比	0 0%	11 1.7%	2 2%	13 1.5%
SC平均偵測時間(ms)	13.4	13.5	13.5	13.5

第二個觀察到的就是Upright Cascade偵測到確實為車牌(True Positive, TP)和偵測到不是真正車牌(FP)的次數都比Skewed Cascade的TP以及FP高，所以我們最後用來偵測車

牌的方法選擇先以Skewed Cascade偵測，因為其FP機率較低，如果沒有偵測到車牌，接著才使用Upright Cascade再偵測一次。

第三個部份是過濾車牌的部份。整體來講，有車牌過濾的流程，能夠提升我們偵測車牌的準確率。表1內的FP數據是沒有使用車牌過濾流程去處理的數據。若加上3.4的兩個過濾條件，則FP幾乎都排除掉了，結果如表二。沒有過濾掉的情況都是車身上噴漆的文字，或者是FP本身有框到部份車牌。

第四個部份是觀察偵測時間，如同我們在3.2所預測，實驗的結果Upright Cascade偵測的速度大約為11ms(毫秒)，比Skewed Cascade的速度13ms快。我們實際車牌偵測的做法是先以Skewed Cascade來做車牌偵測，當Skewed Cascade偵測不到車牌時，接著才以Upright Cascade來做車牌偵測，當兩個Cascade分類器都偵測不到車牌時，就視為偵測失敗。我們將前面用來偵測車牌的860張測試圖片，以這種雙層Cascade的方法來偵測車牌，結果如表三，效果比單獨使用Upright Cascade或Skewed Cascade都好。

表二: 加上車牌過濾後FP的的情況以及所花時間

圖片大小	320×240	640×480	2048×1536	平均
測試張數	100	660	100	
UC FP個數與百分比	0 0%	12 1.9%	1 1%	13 1.5%
SC FP個數與百分比	0 0%	3 0.5%	0 0%	3 0.3%
SC平均偵測時間(ms)	23.3	23.3	23.2	23.3

表三: 雙層Cascade之偵測效率

	TP數	(TP/總數)×100%
UC直接偵測到車牌	673	78.3%
SC直接偵測到車牌	556	64.7%
以雙層Cascade 偵測到車牌	714	83.0%

3. Hough Tnsform對車牌偵測的影響及處理時間

我們將前面860張測試圖片再次做實驗，如果以雙層Cascade偵測失敗，那麼就做Hough transform旋轉校正，之後再次以雙層cascade偵測，結果如表四，提升的效果並不多，大約提高4%。並非Hough transform旋轉校正的效果不好，而是Hough transform主要是針對水平的旋轉校正，對於側轉或者上下傾斜的車牌幫助不大。

表四: 加上Hough transform對偵測率的提升

	TP數	(TP/總數)×100%
以雙層Cascade偵測一次	714	83%
偵測失敗的影像以Hough transform 處理後再以雙層Cascade偵測一次	752	87.4%
提升偵測車牌數量與百分比	38	4.4%

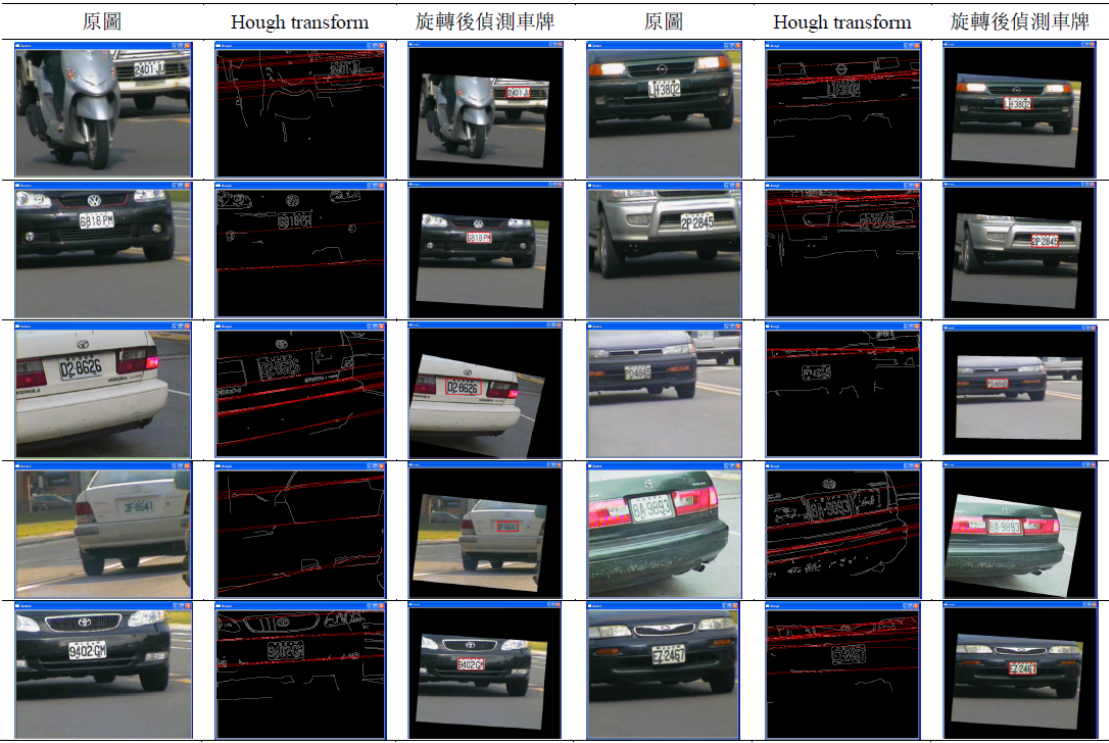
表五則為Hough Transform以及經過Hough Transform之前後偵測車牌所花費的時間。其中的Hough Transform是包含了Hough Transform找直線的時間加上旋轉圖片的時

間，而Hough Transform旋轉校正後的圖片多補上的黑色背景會造成再次偵測的時間稍微增加。表六為我們實際透過Hough Transform之後能夠幫助到偵測車牌的例子。

表五: Hough Transform及偵測平均花費時間

	平均花費時間(ms)
以UC偵測車牌	13.5
以SC偵測車牌	11.3
Hough transform花費時間 (包含找直線與影像旋轉)	60.5
經過Hough transform校正 以UC偵測車牌	19.3
經過Hough transform校正 以SC偵測車牌	17.2
過濾程式花費時間	23.3

表六: Hough Transform後順利偵測到車牌之實驗結果



五、結論

本篇論文爲了在真正的交通監控畫面中取得車牌資訊，除了研究如何以分類器偵測車牌和作車牌的定位，更重要的是爲了配合真實上的情況，提出以Hough transform對影像作旋轉校正再以cascade分類器做車牌偵測。然而Hough transform運算的時間比較長，在處理上我們的想法是降低不必要的背景，減少處理的影像部份，我們實際將影像中車輛的部份單獨取出來處理，影像大小至少降爲三分之一，Hough transform的時間也降到20ms以下，但是車流量過大，車速過快，反光或者陰影等，都會影響到切割的品質與準確度。我們未來研究的方向是希望以PTZ攝影機的功能，將車輛的部份單獨放大到足以

處理車牌影像即可，兼顧影像資訊以及減少影像處理的部份，藉以提高整個偵測的效率與速度，此方面也有部分文獻[13]可供參考。

[謝啓]

本研究課題得到國科會基金（NSC 99-2622-E-036 -001 -CC2）資助，特此致謝。

參考文獻

- [1] Lai, C.H., Wu, B.S. and Hsieh, C.C., “Adaptive road model for traffic safety applications,” *Information Technologies, Applications, and Management Conference (2010 ITAMC)*, 高雄, 2010, June 25.
- [2] Huang, Y.P., Chen, C.H., Chang, Y.T. and Sandnes F.E., “An intelligent strategy for checking the annual inspection status of motorcycles based on license plate recognition,” *Expert Systems with Applications*. 2009, 36(5): pp. 9260-9267.
- [3] Hsieh, J.W., Yu, S.H., and Chen, Y.S., “Morphology-based license plate detection from complex scenes,” *IEEE International Conference on Pattern Recognition*, 2002: pp. 176 – 179.
- [4] Guo, J.M. and Liu, Y.F., “License plate localization and character segmentation with feedback self-learning and hybrid binarization techniques,” *IEEE Transactions on Vehicular Technology*, 2008, 57(3): pp. 1417 – 1424.
- [5] Chen, B., Cao, W.L., and Zhang, H.C., “An efficient algorithm on vehicle license plate location,” *Proceedings of the IEEE International Conference on Automation and Logistics*, 2008: pp. 1386-1389.
- [6] Viola, P. and Jones, M. “Robust real-time object detection,” *Int. J. Computer Vision*, 2004, 57(2): pp. 137–154.
- [7] He, X., Zhang, H., Jia, W., Wu, Q., and Hintz, T., “Combining global and local features for detection of license plates in a video,” *Proceedings of Image and Vision Computing*, New Zealand, 2007: pp. 288–293.
- [8] Wang, S.Z. and Lee, H.J., “A cascade framework for a real-time statistical plate recognition system,” *IEEE Transactions on Information Forensics and Security*, 2007, 2(2): pp. 267–282.
- [9] Seo, N., “Tutorial: OpenCV haartraining,” <http://note.sonots.com/SciSoftware/haartraining.html>.
- [10] Duda, R.O. and Hart, R.E., “Use of the Hough transform to detect lines and curves in pictures,” *CACM*, 1972, 15(1): pp. 11-15.
- [11] Yanamura, Y, Goto, M, Nishiyama, D., Soga, M., Nakatani, H., and Saji, H., “Extraction and tracking of the license plate using Hough transform and voted block matching,” *IEEE Intelligent Vehicles Symposium*, 2003: pp. 243 – 246.
- [12] Wen, C.Y., Yu, C.C., and Hun, Z.D., “A 3-D transformation to improve the legibility of license plate numbers,” *Journal of Forensic Sciences*, 2002, 47(3): pp. 578-585.

- [13] Chiu, S.H., Lu, C.P., and Wen, C.Y., “A motion detection-based framework for improving image quality of CCTV security systems,” *Journal of Forensic Sciences*, 2006, 51(5): pp. 1115–1119. DOI: 10.1111/j.1556-4029.2006.00205.x

