

干擾狀況下之交通標誌偵測及辨識

Traffic Sign Detection and Recognition in Noisy Outdoor Scenes

楊修銘 劉昆灝 劉昭麟

國立政治大學資訊科學系

{ g9013, s8838, chaolin }@cs.nccu.edu.tw

摘要[†]

交通標誌的偵測和辨識，是在智慧型運輸系統中幫助駕駛人，使他們能夠更安全的駕駛汽車。這篇論文分為兩個部分，一是交通標誌的偵測，我們利用了交通標誌特定的顏色、特殊的外形、以及內部的特徵，配合一些判斷的規則來認定交通標誌的存在，另一個是交通標誌的辨識，將偵測到的交通標誌做一些正規化的處理，如調整圖片解析度，去除雜點等，可得到一張 16x10 的圖片，將其配合離散餘弦轉換所得到的特徵，輸入至類神經網路，即可判斷出正確的答案。我們的系統目前可以偵測圓形及三角型交通標誌，成功率約為八成五，而在辨識方面則以 45 個三角型標誌為樣本，辨識成功率約為七成。

1. 簡介

智慧型運輸系統(Intelligent Transportation Systems)乃指應用電腦、通訊等科技，提供一些即時資訊，使得行車之安全及效率，能夠獲得改善，像是導航系統、交通管理最佳化和支援公共運輸等，都是屬於其範疇。而我們的研究乃是利用電腦視覺(Computer Vision)的技術，來做交通標誌的偵測，另外利用類神經網路的技術來做辨識的部分。在張耀升的研究中，他提到了，先對圖片做 saturation enhancement，再利用顏色及邊的偵測，來找出交通標誌的所在[1]。在 Escalera [3]裡面，他們也提到利用顏色的特徵，再加上 Optimal Corner Detection [9]的方法來偵測出交通標誌的所在。在 Luo et al.的研究中提到，利用了 fractal texture segmentation 的方法來找出交通標誌[8]。三者都有不錯的結果，不過他們的論文並沒有提到是否會有偵測錯誤的情形或是將偵測

錯誤的標誌留待後面的類神經網路再做處理。我們希望能夠在偵測時就減低錯誤率使得送進類神經網路中的圖形都是正確的標誌，減低辨識複雜度，因此在利用顏色、邊的偵測、物件偵測和過濾後，還會在後面加上一些特徵值的判斷來直接將不屬於我們需要的圖形刪除。效果不錯，有約九成的成功率。

另外有許多人做過利用類神經網路來做交通標誌的辨識，例如 Luo et al.，他們採用了 receptive field neural network，32x32 的輸入、10 個輸出另外加上 4 層隱藏層，用來辨識九個標誌，雖然效果不錯，不過辨識需 4 秒的時間[6]，另外 Luo et al.，採用 reconfigurable neural network，45x45 個輸入，十個輸出，也約需 1 秒的時間來做辨識[7]，因此我們在辨識的部分也有特別注意辨識的時間，盡量增加速度，另外我們除了把剪下的交通標誌當作 ANN 的輸入外，還加進了一些特徵值(如 DCT)當作輸入，希望能加強辨識效果。除此之外，我們增加了可辨識標誌的數量，目前我們已能夠做到交通部制定的 45 個三角型標誌的辨識。

大部分的 paper 都只是稍微的帶過了前端交通標誌偵測的部分，且對於偵測交通標誌時會有的特殊遮蔽現象，譬如遭車體或樹木遮蔽，或是交通標誌扭曲..等等現象並沒有多加描述，因此我們希望能就這方面特殊交通標誌的偵測與辨識來做深入的探討，

以下是我們的研究成果，第二節為交通標誌的偵測部分，第三節為交通標誌的辨識，第四節討論系統的設計與分析，第五節是實驗結果，第六節則為討論。

2. 交通標誌偵測

我國的交通標誌分為四種：(1)警告標誌 (2)禁制標誌 (3)指示標誌 (4)輔助標誌，本系統目前著重在偵測與辨識正三角形的警告標誌與圓形的禁制標誌。一般常見的交通號誌，通常都有醒目的顏色或是

[†]發表於台灣人工智慧學會第七屆人工智慧與應用研討會論文集第 555 頁到第 560 頁，台灣台中，Nov. 2002.

特定的形狀來突顯它的所在，例如警告標誌，就是由紅色的外框、白色的底以及三角形的外形所組成，因此在做交通號誌的偵測時，通常都是以這兩種特徵配合其他的特徵來定位交通標誌[3]。以下我們以圖一的處理為例，說明整個系統的辨識過程。



圖一

2.1 顏色 (Color Cue)

警告及禁制標誌，都有一個鮮明的紅色外框，因此，利用 RGB 的值定義一個“紅色”的臨界值區域，利用這個區域去檢查輸入圖片(圖一)的像素，將可以過濾出一些可能是交通標誌的候選人，在我們的系統裡面，這個臨界值區域定義成：

$$R_a = \begin{cases} R_b, & \text{if } (R_b - G_b) > 20 \ \& \ (R_b - B_b) > 20 \ \& \ R_b > 20 \\ 0, & \text{otherwise} \end{cases}$$

$$G_a = 0$$

$$B_a = 0$$

也就是 R 值至少比 G 和 B 值都大 20 以及 R 值本身也要大於 20 以上，這樣設定的原因是根據實驗的結果以及藉由設定值為 20 使得此系統不致於過度靈敏，導致過濾出不必要的像素。當然，直接定義某個 RGB 值域為紅色值域也是可以的，不過由於 RGB 的值容易受到光的影響，而產生某部分的數值偏高而某部分的數值偏低，因此採用上述值差的定義法，比較不會受到光害的影響。另外，採用 RGB 值互除的方法也可以避免光害的影響，甚至可能比我們系統定義的方法更加靈敏；不過靈敏的系統，反而會在圖片上選出過多跟交通標誌無關的雜點，而影響了後面的偵測，因此採用算是折衷的相減方法。另外，我們也暫不考慮 HSI 系統，因為光的改變雖然不會對它造成什麼影響，但是要去計算 HSI 系統會花費較久的時間。

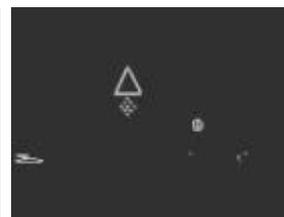
2.2 邊的偵測 (Edge Detection)

利用顏色偵測後，結果會如圖二所示，只留下我們認為紅色範圍內的像素，此時對這張圖片做“Laplacian of Gaussian”的 Edge Detection 處理[4]，得

到一個臨界值，將比臨界值細的線段給忽略掉，結果如圖三所示，依照紅色的像素產生許多的邊。



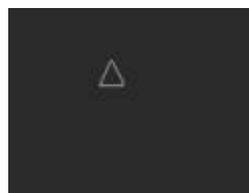
圖二



圖三

2.3 物件的偵測 (Object Detection)

做完 Edge Detection 後，還必須確認哪些“邊”是屬於同一個物件所有，而那些“邊”只是些小雜訊。我們系統所採取的做法，是由上到下，由左到右去掃描圖片，發現非零像素後，則尋找其八個鄰近格子(8-connected neighborhood)，查看是否有其餘非零像素，直到找不到後，就把找到的這幾個像素定義為屬於某個物件的像素，做完後就可以定義出屬於這張圖片的所有物件，再將這些物件中，擁有像素數目在六十點以下的物件刪除，如圖四所示。我們將這些物件定義為是圖片上的雜訊，即使這些物件是交通標誌，在六十點像素以下已經是過小的標誌，對於系統後段的辨識將是一大考驗，我們暫時不做這方面的辨識，因此在此處把它給刪除，在我們的實驗裡面，只有一個交通標誌因而刪除。剩下的物件就是系統選定最終的候選人，我們利用此物件的最小和最大的行列值，由原本的輸入圖上剪下此物件，如圖五所示，然後放入其他的特徵值做篩選。



圖四



圖五

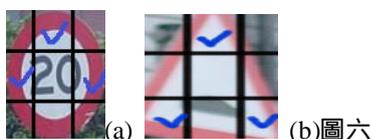
2.4 特徵篩選

我們選定了幾個特徵來判斷選出的物件是否是交通標誌：(1)物件的長寬比例：一般的交通標誌只要不是太過分的扭轉，或是視線角度的問題，通常其長寬比會在一定的範圍內。(2)物件是否在某個列的上方：對於迎面而來然後從左右視角消失的交通標誌，

大部分都會在視野的某個列值上方，而不會在過分下方，除非是特殊視角。(3)物件最中央像素的 R-G 值差 (4)物件最中央像素的 R-B 值差 (5)物件最中央像素的 G-B 值差：選定(3)(4)(5)的原因是一般的交通標誌不管其角度為何，通常在物件最中間的一點都會落在交通標誌的資訊區內，即當中只會有白或黑兩種像素值(除了有紅色斜線的禁制標誌和紅綠燈標誌)，因此可以設定其最中間像素的值域來判斷。根據實驗的結果，我們選定了物件的長寬比例要在 0.7 和 1.3 之間，而在 640*480 的圖片中，座標系統原點定於左上角，交通標誌的列定義其必須位於第 350 列的上方，而物件最中間的像素值差，R-G 值小於 33，R-B 值小於 46，G-B 值小於 33。在判別的方法中，我們採用投票制，也就是一種特徵投一票的做法，得到五票的物件，直接判定為是我們目前可處理的交通標誌，而得分在 3 分或是 4 分的物件，則要進入下一階段的形狀判定，得分在 3 分以下的物件，則直接淘汰。

2.5 形狀判定

由於某部分的交通標誌在做上面五種特徵值的判別時，會因為某些細部的原因，如視角的關係導致看到的交通標誌不是呈現較正常的長寬比而是一個細長的長方形，因此沒有辦法得到 5 分而被認定為不是交通標誌，因此我們將得分較高，也就是仍舊有機會是交通標誌的物件，再拿來做形狀的判定，看看此物件是否是三角形或是圓形，如果判定是，由於之前的五個特徵值的得分也頗高，因此也可以直接判斷屬於交通標誌。判斷三角形或是圓形的方法，則如圖六(a)(b)所示，將物件長寬都等分成三等份，也就是將物件劃分成九個等份，若是三角形的標誌，則其三個頂點，會分別落在右下、左下以及上方中間等份，若是圓形標誌，則其 0 度、90 度和 180 度的頂點，會分別落在右方中間、上方中間以及左方中間等份，利用此特性，即可判斷出哪些物件屬於三角形和圓形的交通標誌，而哪些不是。



3. 交通標誌辨識

當我們把交通標誌從一張照片中偵測出來以後，接下來就是辨識的工作，在本文中，我們將辨識三角形的交通標誌，也就是我國交通部所規定的各個標誌，三角形的交通標誌共 45 個標誌[2]，為了解決此項問題，我們建構一個三層的 feed-forward back-propagation neural network 來辨識三角形標誌，在三角形標誌的網路中，其具有一層的隱藏層與 46 個 output(即對應於 45 個目標標誌，另一個 output 代表輸入的影像不屬於其他 45 個目標影像)，hidden layer 的大小為 8Q 以下將分別從三個部分來探討實驗的內容，即前置處理(影像正規化)，網路的訓練，以及實驗的數據及結果。

3.1 輸入影像的正規化(Input Image Normalization)

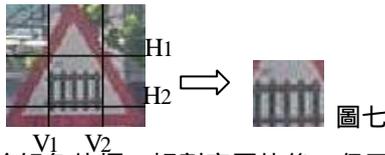
為了節省空間以及加速運算，我們必須在輸入的影像上作有效的處理以期能用最少的成本來保留每個影像上的特徵值。詳細的方法可以分成幾個步驟，在每個步驟中，我們都以三張圖片來示範處理結果：

1.調整圖片大小：在本實驗中，我們將以 80x70 的影像來進行操作，故需要將來源的輸入影像調整成我們所需要的大小。

2.剪裁圖片：經過觀察，發現三角形的交通標誌最具特徵的地方是在標誌的內部區域，也就是扣除紅色外框後，內部的一部份區域，在三角形的辨識網路中，我們剪裁下 32x30 的影像作為最具特徵的部分，這個 32x30 的影像幾乎可以完整地將標誌中的內部影像完全剪裁下來，剪裁後的影像會是原始影像中如下圖七所示四條直線所圍成的區域：

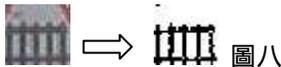
根據 training data 得到的資料顯示，大部分切割出來的是交通標誌的圖片，其最具有特徵的地方都會落在中央偏下的地方，在觀察過後可以發現兩條垂直線 V1 及 V2 將影像中最不具特徵的大部分外圍區域排除在外，兩條水平線 H1 及 H2 亦有如此的作用，處理過後就可以捨去多於不必要的影像儲存空間，也不會喪失原本影像中具有的特徵，即使是被樹葉或陰影遮蔽的圖片也是如此，而某些存有缺角或是有小幅歪斜的交通標誌也會在可容忍的範圍內，因此我們劃定了此四條線，其對於所有的交通標誌候選人幾乎都能截取影像中最具特徵的地方，除了極少部分缺角相當

嚴重的標示外。本實驗就以這四條線的所在位置為基準，過濾掉大部分的影像。



3.扣除紅色外框：切割完圖片後，仍可能會有一些在影像角落的像素值為紅色，我們利用前述的 threshold 來偵測明顯的紅色的像素，並將偵測到的紅色像素改成白色值。

4.轉換圖片格式：將輸入的影像由原本的三維圖片(RGB)，轉換成灰階影像，調整亮度後保留亮度較低的像素，在我們的實驗中，我們把每張來源影像的平均灰階值調整為 100(影像中每個像素的灰階值範圍是 0~255,0 表示黑色，255 表示白色)，並保留灰階值小於 70 的像素，最後再將影像轉為 binary image，使影像中的差異更明顯，也縮小了影像的儲存空間及增加處理的方便性，圖八描述出此項步驟的效果。



5.去除雜點：保留影像中佔有較大範圍的連接區域，把雜訊出現的機會降到最低，增加辨識率，最後的影像是一個 32x30 的 binary image，即影像中的像素值為 0 或 1。

6.降低影像解析度：將 32x30 降低解析度，降為 16x10，處理過後影像為灰階影像，新影像的每一個像素值是由來源影像的 2x3 格像素值的平均所決定，此過程可用下列的式子表示：

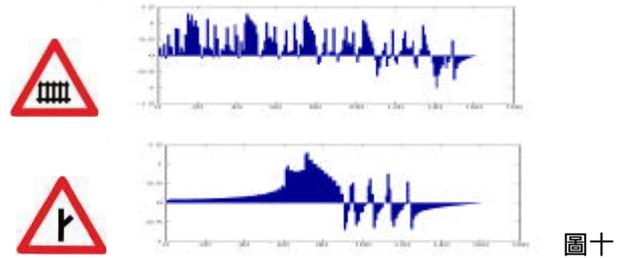
$$I_2(i, j) = \frac{1}{6} \sum_{m=2i-1}^{2i} \sum_{n=3j-2}^{3j} I_1(m, n)$$

其中 I_2 表示目標影像， I_1 表示來源影像，上述的式子是先算出六格的像素值之和，再算出平均值，圖九說明降低解析度造成的效果。



7.離散餘弦轉換：在圖形的處理技術中，離散餘弦轉換 (Discrete Cosine Transform) 是分析影像特徵的方法之一，本實驗中將上個步驟所得到的圖形 (如圖九) 經由離散餘弦轉換，得到圖形的部分特徵，並

將此特徵一併放入類神經網路中，以圖十說明標誌影像與其經由轉換過後得到的向量圖表。



3.2 訓練數據集(Training Patterns)的取得

一組具有足夠特徵的 Training Set 是訓練有效的類神經網路的必要條件，本實驗利用完美的交通標誌影像來訓練網路，之後再將完美的影像經過處理來製造出其他帶有雜訊的訓練數據集，再交替用這些訓練數據集來訓練我們所建構的交通標誌辨識網路，以下列出各種訓練數據集的產生方法與實際例子：

- (1)完美的影像：總共有 45 張完美的三角形標誌圖片。
- (2)加入雜訊的影像：在每張完美的圖片上，利用 MATLAB 加上 variance level 為 0 到 0.8 的 Gaussian noise，以 0.005 為間距來製造具有雜訊的圖片，也就是每張完美的標誌圖片都另外產生出 160 張不同等級的雜訊影像，這組數據集在三角形的交通標誌中共有 160x45 張圖片，下圖十一所示由左至右為加入 variance 0 至 0.8 的 Gaussian noise 之後的影像。



- (3) Shifted images：為了能讓網路充分適應標誌裡些微的 displacement，我們將完美的標誌圖片往八個方向做 circularly shift，shift 的範圍是上下左右各 3 個像素，每個標誌共產生 48 個位移過的影像(7x7-1)，這組數據集在三角形的交通標誌中共有 48x45 張圖片。
- (4) Rotated images：將完美的影像與(3)中所述的位移影像做旋轉，順時鐘與逆時鐘各旋轉 1~10 度，每個標誌共產生 20 張旋轉過的影像，下圖十二為往順時鐘方向旋轉 0 至 10 度的結果。



- (5) 真實的影像：除了上述的四種訓練數據來源以外，我們也收集了一部分的真實交通標誌影像作為類

神經網路的訓練數據。

4. 設計與分析

整個系統的詳細設計流程如下所述。

4.1 兩階段判別法

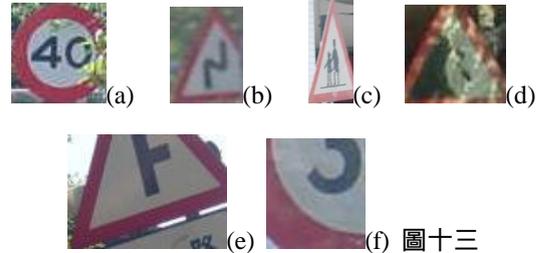
我們在判別的方法中採用了兩階段判別法，而不把形狀判定的方法一起加入特徵篩選的地方投票，目的是區分某些特徵值對於某些交通標誌有比較好的效果，因此如果放在一起投票，很有可能比較偏好形狀判定方法的交通標誌會被其他特徵值給判定出局，而偏好其他方法的交通標誌則會因為加入判定形狀的特徵而被判定出局，因此才會設計成，將得五分的直接判斷成交通標誌，另外在用比較鬆的規則，將得三分以上的物件在送入作第二層的判斷，如此則在兩階段完畢後可以留下最多判斷正確的交通標誌。

4.2 被遮蔽及有雜訊的交通標誌偵測

在路上設置的交通標誌往往會因為一些因素，如路樹、樹蔭、視野角度、和車子等等問題而造成一些遮蔽或是雜訊(Noise)如圖十三所示[5]，在我們的系統內也有考慮到這些情況而且對於某些情況仍然可以做到偵測的能力。在形狀判定的部分，由於判定三角形和圓形是採用九等份的方法來判斷，因此即使如圖十三(c)的圖形已經扭轉的非常厲害，仍然會落在我們所設定的範圍內，在三角形的判斷中，除非交通標誌是順時針或是逆時針旋轉到一定程度以上，否則都會被判定為是三角形標誌，而圓形的判定則是不管圖形如何旋轉，都會落在我們所設定的圓形範圍內。另外一種常見的是因為遮蔽而產生缺角的情形，如圖十三(a)(b)(e)(f)。我們設定的五種特徵篩選以及在形狀判定所採用的等分判定方法，對缺角的交通標誌有一定的容忍性，因此系統仍然會將此類交通標誌偵測出來以供辨識。另外有雜訊的交通標誌如圖十三(d)，也可以經由上述的方法偵測出來，因此整個系統對於一定程度內的被遮蔽及有雜訊的交通標誌都能順利偵測。

目前較有問題的偵測，則是如果正好有紅色物體環繞在交通標誌的旁邊，或是交通標誌的紅色外圈因為過度骯髒或老舊而退色，則在第一階段的顏色過濾時會出問題(Centroid Noise)。有時候在夜晚拍照時，

交通標誌會因為角度問題而沒辦法反光，導致完全呈現黑色，此時會造成系統無法辨識，比較有效的解決辦法可能是直接使用夜間適用的攝影機，或是在做攝影時，對連續的輸入做比對，因為拍攝角度的問題，應該會有幾張呈現出交通標誌的影像，此時就有機會可以成功的偵測。



5. 實驗結果

我們在實驗中所使用的資料是由數位相機在街頭拍攝交通標誌得來，共計 150 張，所採用的解析度是 640x480，裡面有包含如陰天、晴天、夜晚各種氣候拍攝的照片以及正常、缺角及被樹蔭擋住的各式交通標誌。我們挑選了其中的 50 張當作訓練資料，剩下的 100 張照片則是當作測試資料，在挑選訓練資料時有特別注意到挑選各種樣式及遮蔽的交通標誌以求的訓練之客觀性。在實驗的過程中，我們測量了由輸入的圖片上所剪下來的物件，包含在特徵篩選內所提到的物件的長寬比、最大的列、中間像素的 R-G、R-B 和 G-B 值差，再利用 precision 和 recall 兩樣指標，定義出各樣特徵所要採用的臨界值，此臨界值會讓兩樣指標能夠同時達到一定的水準。剩下的 100 張圖片當作測試資料來作實驗。實驗的結果如 Table 1，precision 是系統剪下物件中被辨認是交通標誌而實際上是交通標誌的比例，recall 是系統剪下的物件中實際是交通標誌與被辨認為是交通標誌而實際上的確是交通標誌的比例，若再加上沒被系統剪下的物件而其實確是交通標誌的即是整體系統 recall。

	以五種特徵做偵測		加入形狀判定	
	Training	Testing	Training	Testing
Precision	0.84	0.85	0.92	0.91
Recall	0.84	0.83	0.89	0.87
System Recall	0.84	0.75	0.85	0.82

Table 1

$$\text{Precision} = \text{SR}_{\text{yes}} / \text{S}_{\text{yes}}$$

$$\text{Recall} = \text{SR}_{\text{yes}} / \text{SCR}_{\text{yes}}$$

$$\text{System Recall} = \text{SR}_{\text{yes}} / (\text{SCR}_{\text{yes}} + \text{SUCR}_{\text{yes}})$$

S_{yes} ：系統剪下的物件中被認為是交通標誌的數量。

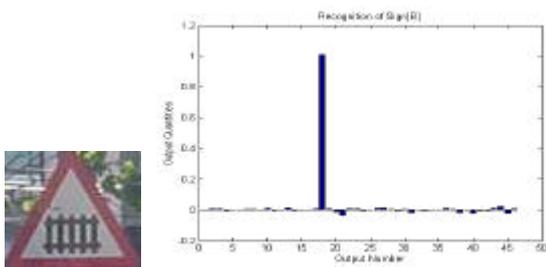
SR_{yes} ：系統剪下的物件中被認為是交通標誌而實際上也是交通標誌的數量

SCR_{yes} ：系統剪下的物件中實際上是交通標誌的數量。

SUCR_{yes} ：系統未剪下而實際上是交通標誌的數量。

另外在辨識方面，根據多次實驗，在 Intel Celeron 900MHZ 的 PC 上測試，判斷一張已剪下的交通標誌候選人圖片為何種交通標誌約須 280ms，圖十四這張圖，為實驗的測試數據，左圖為輸入的待辨識影像，右圖為網路輸出結果的長條圖，橫軸是類神經網路中 output 層的各個 neuron，縱軸代表個 neuron 的輸出量，output 的輸出量愈大，代表輸入的影像愈可能是該 output 所代表的標誌影像，在輸出的數據中，實驗的設定是將最高分及次高分者都當作辨識的結果。在辨識方面，利用上述 3.2 節的訓練資料，能夠得到 99% 的正確率，而在測試資料中，可以得到 68% 的正確率。

另外系統整體在測試資料方面的辨識率約在 58%，其中辨識錯誤的物件有很大一部分都是旋轉過後或是有缺角的交通標誌。



圖十四 (sign[B]及其辨識結果，正確辨識出 sign18)

6. 討論

雖然我們的方法可以有效的偵測出大部分有雜訊或是被遮蔽的交通標誌，但還是有一些例外的情形，尤其是當處於黑夜或是下雨天，這時照到的交通標誌最難以被偵測出來，如果能夠利用 HSI 系統並且降低其計算複雜度或是將輸入圖片做一些 saturation 的改變，也許能克服惡劣天氣對我們的影響，這也是

未來我們所努力的目標，另外，利用類神經網路來辨識交通標誌雖然有一定的成果，不過在有遮蔽的情況下，還是會有辨識不良的情形發生，也是改進的重點。

感謝

感謝台中市政府交通局謝長明先生及政大資料所黃上銘同學提供交通標誌資料及照片。

參考文獻

- [1] 張耀升，*戶外交通標誌辨識之研究*，國立交通大學電機與控制工程研究所碩士論文，民國 83 年。
- [2] 教育資料館，*道路交通標誌標線號誌設置規則*，民國 87 年。
- [3] A. de la Escalera, L. E. Moreno, M. A. Salichs and J. M. Armingol, Road traffic sign detection and classification, *IEEE Trans. on Industrial Electronics*, 44(5), 848-859, 1997.
- [4] R. Haralick and L. Shapiro, *Computer and Robot Vision*, Vol. 1, pp. 346-351, 1992.
- [5] N. Kehtarnavaz and A. Ahmad, Traffic sign decognition in noisy outdoor scenes, in *Proc. IEEE Int. Conf. on Intelligent Vehicles*, pp. 460-465, 1995.
- [6] R. Luo, H. Olapalli, and D. Hislop, Autocorrelation, in *Proc. Int. Conf. Industrial Electronics, Control, Instrumentation and Automation*, pp. 700-705, 1992.
- [7] R. Luo, H. Potlapalli, and D. Hislop, Translation and scale invariant landmark recognition using receptive field neural network, in *Proc. Int. Conf. Intelligent Robots and Systems*, pp. 527-533, 1992.
- [8] R. Luo and H. Potlapalli, Landmark recognition using protection learning for mobile robot navigation, in *Proc. Int. Conf. Neural Networks*, pp. 2703-2708, 1994.
- [9] K. Rangarajan, M. Shah, and D. Van Brackle, "Optimal corner detector," *Computer Vision, Graph., Image Processing*, vol. 48, no. 2, pp. 230-245, Nov. 1989.