

98 年度大專生參與國科會計畫

研究成果報告

國立清華大學 資訊工程學系 雷禹恆

(一)摘要

本計畫的目的是透過電腦視覺，影像處理，與機器學習的相關技術，協助校園內的視障師生，使他們在日常活動及行走時，更能辨別道路上的邊界、來車等突發狀況。我們將針對需要偵測的物體，分析並擷取其特徵，並利用機器學習的方法進行模型建立與識別。最後，我們在電腦上實作出圖形介面的即時偵測結果，以模擬視障者所配帶在身上的智慧型裝置。

關鍵字：視覺輔助，機器學習方法，物件偵測。

(二)研究動機與研究問題

電腦視覺是一門研究如何使機器「看」的科學，利用攝影機或相機代替人眼的視力，並用電腦（處理器）代替人腦，對於影像做進一步的處理、決策，成為便於人們使用的結果。若利用電腦視覺代替人眼視覺來解決人類的問題，對於視覺障礙者，意義是相當大的。也因此，本計畫期許能幫助視障朋友解決視覺上的缺憾，並藉由專題研究，盡一份社會責任。

清大的校園內有不少視障師生，學校也成立清大盲友會提供協助。在一次與某位視障老師的訪談中，我們發現先天的全盲與後天的全盲，在適應環境上有很大的差異。先天全盲者，往往從小就受過啓明學校的訓練，對於行走時的平衡保持，路線記憶相當熟悉；但是後天全盲者，少了啓明訓練的條件，可能連直線行走都有問題（一般人閉上眼睛直線行走也是很困難的）。因此有些視覺輔助嘗試提供的資訊，例如「前方十公尺」，對於某些視障朋友可能沒什麼意義。如果計畫目的是想協助最有困難的視障者的話，掌握哪些資訊對他們才有意義，就顯得十分重要了。

本計畫的研究重點，在於對某些欲識別的樣式(pattern)，如車道標線、人行道邊界、路上車輛，分析其在影像上的特徵，再透過適當的分類器來判斷接收到的資料是不是任何一種我們想要辨認的樣式。如果是的話，再給予使用者適當的輸出提示，以利使用者對於當前路況的判斷。

(三)文獻回顧與探討

傳統的視覺輔助系統，如導盲杖、導盲犬等，往往提供的資訊有限，或是取得的時間、金錢代價昂貴，無法提供一般視障人士必要的幫助。另一方面，儘管具備 GPS 導航的手機或 PDA 已越來越普及，但由於這些導航系統在設計時並未考慮視障人士的需求，多位視障人士在實際使用測試後發現，現行的 GPS 導航

系統仍無法提供他們所需的資訊 [1]。因此，專門針對視障人士需求設計的電子式的導盲系統，或稱「電子式行進輔具(Electronic Travel Aids; ETA)」仍是各國家積極研究的方向。

文獻 [2] 提到，一般的 ETA 多採用偏物理的感測，例如超音波、紅外線、雷射等，不僅成本昂貴，所提供的資訊有時過於「被動」，只有在視障者將遭遇障礙物時，才被輔具引導閃避。如能以電腦視覺的方式，模擬人眼看到的真實畫面，再配合可行的演算法，將可化被動為主動，得到畫面中對於路況判斷有意義的資訊。又以數位化的方式建構一套 ETA，可大大降低物理儀器的昂貴成本，僅需要一台等級尚可的攝影機、處理器、輸出裝置，加上設計過的演算法即可做到。

在物件偵測（如：人、車、建築物）領域，已有很多基於機器學習的方法被提出。Viola 與 Jones [3] 在 CVPR'01 曾提出過一個物件偵測的訓練架構，把 AdaBoost 演算法的強分類器串接成一個決策樹(a cascade of decision tree)，而影像特徵則是使用基於亮度值(intensity-based)的矩形(又稱 Haar-like) 特徵。優點是偵測快速，缺點是矩形特徵本身不夠準確。

另一方面，Dalal 與 Triggs [4] 在 CVPR'05 曾經提出了基於梯度(gradient-based)的 HOG (Histogram of Oriented Gradients) 特徵，是將 Lowe [5] 在 IJCV'04 提出的 SIFT 文獻後半段獨立出來使用，而訓練架構則是直接使用現成的 SVM。優點是 HOG 準確率高，缺點是 HOG 特徵向量的維度太高，動輒高達數千維，如果要在影像裡跑滑動視窗(sliding window)找尋物件的話，速度會太慢，在影片方面的應用會更不理想。

因此，為了結合前兩者的優點，本計畫實作的主要參考方法是國內陳昱廷和陳祝嵩 [6] 在 2008 年所發表的文獻。影像特徵部分，基於亮度值和基於梯度值的特徵皆使用以加強準確率；而訓練架構部分，使用修改自 [3] 的 cascaded AdaBoost 來達到快速偵測。

(四)研究方法及步驟

(0)假設

為了讓計畫容易進行，並把問題簡化，我們限制研究的範圍在國立清華大學的室外校園，此區域是視障者比較需要幫助，及容易遇到突發狀況的地方。又考量光線問題，我們把條件限制在「白天、好天氣」的理想情況，以避免雜訊造成的問題複雜度。理想中的最終目標是利用筆記型電腦配上網路攝影機，以模擬可攜式的視覺輔助設備。

(1)車道標線與人行道邊界 - 使用 Hough 轉換

車道邊線部分欲完成的功能是，當路上有一條或多條標線時，如何引導使用者靠馬路邊走，遠離馬路中央。典型的清大車道如圖 1.(a)所示，希望偵測到的結果如圖 1.(b)所示。我們觀察到，校內車道的中間大多是雙黃線，這是一個有利的現象；而車道兩側幾乎都是單黃線，或是可停車的白線。我們可以依據「偵測到幾組線」，和「線在影像中的位置」，來引導使用者必須向哪邊靠，例如當有一組偏左的雙線時，則引導使用者靠右；當有一組單線、一組雙線時，則引導使用者靠近單線；當有三組線時，則引導使用者靠向較近的左線或右線。

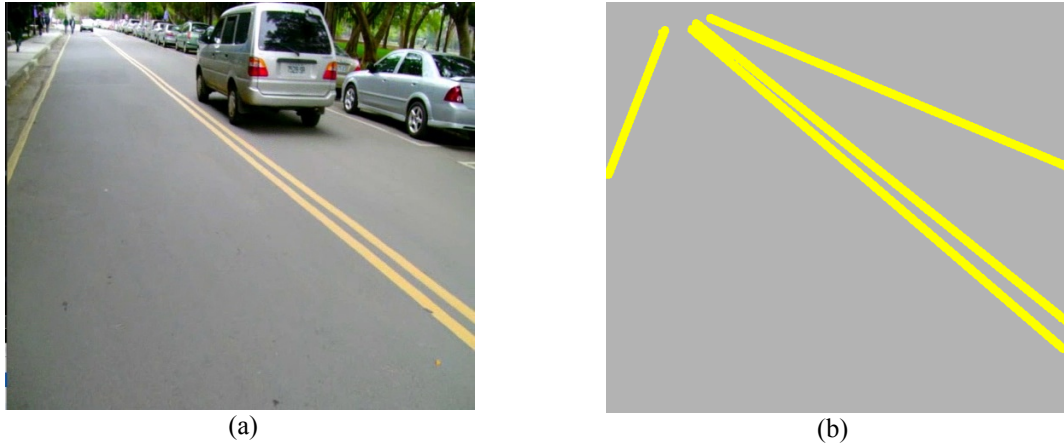


圖 1：(a)典型的清大校內車道，與(b)被偵測的結果示意圖。

人行道邊界欲完成的功能，是在使用者快超越邊界之前給予提示。通常人行道與界外區的差異明顯，邊緣可以用影像處理的方法判斷出來，在此邊界落在中間附近的時候，代表使用者已經靠近邊界，如圖 2：



圖 2：(a)邊緣明顯的清大人行道，與(b)被偵測的結果示意圖。

車道邊線與人行道邊界皆可採用 ρ - θ 版本的 Hough 轉換 [7][8] 來偵測直線。首先將灰階影像套用簡單的濾波器而得到邊緣影像(edge image) 對於邊緣影像中的某一條直線，其直線方程式可表示成：

$$\rho = x \cos \theta + y \sin \theta \quad (1)$$

其中 ρ 代表直線至原點的距離， θ 代表法線與 x 軸所交的夾角，如圖 3(a)所示。通常直線方程式是以 ρ 、 θ 作為參數，在 x - y 空間中畫直線，而 Hough 轉換的原理，是反將 x , y 作為參數，在 ρ - θ 空間中（也稱作參數空間(parameter space)）

畫曲線，如圖 3(b)。此弦波曲線代表的意義，是該點所對應所有的 (ρ, θ) 組合。

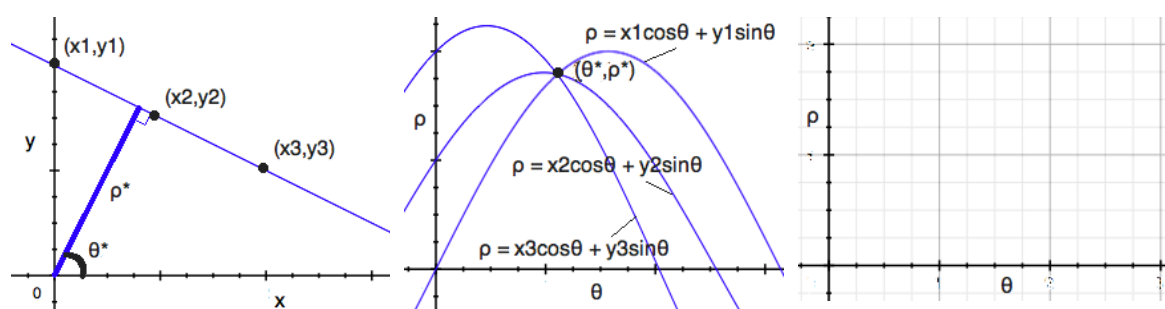


圖 3：(a)在 x-y 空間的直線。(b)將 (x_1, y_1) , (x_2, y_2) , (x_3, y_3) 三點轉換至 ρ - θ 參數空間的弦波曲線，可發現集中在一點。(c)將 ρ - θ 空間切割成累積單元來「投票」，真正的 (ρ, θ) 解可獲得高票數。

為了求得此直線真正的 (ρ, θ) 解，將 ρ - θ 空間切割成許多的累積單元 (accumulator cells)，如圖 3(c)，並將上一步在參數空間畫曲線的 (ρ, θ) 座標，利用類似「投票」的方式，在各種 (ρ, θ) 組合累積票數。 ρ 的範圍是 $-D \leq \rho \leq D$ ， θ 的範圍是 $0^\circ \leq \theta \leq 180^\circ$ ，其中 D 代表影像的對角線長度。理論上正確 (ρ, θ) 值附近會累積到最多的票數，若將(1)式改寫成：

$$y = -\left(\frac{\cos \theta}{\sin \theta}\right)x + \frac{\rho}{\sin \theta} = -(\cot \theta)x + \rho \csc \theta \quad (2)$$

即可用(2)式的 x-y 平面點斜式將原本的直線重建出來。

(2) 來車偵測 -- 使用 {REC, EOH, ED} 特徵配上 Cascaded AdaBoost

(2.1) 特徵使用

對於影像中的任一矩形區塊(block)，文獻 [6] 提出的特徵組合包含三種類型：1. Rectangle (REC)、2. Edge Orientation Histogram (EOH)、3. Edge Density (ED)，分述如下：

- Rectangle 特徵 (REC)：

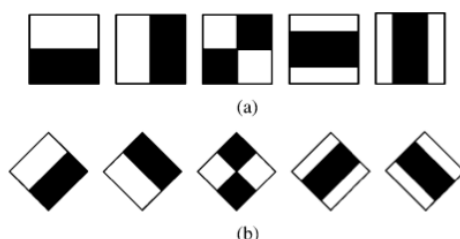


圖 3：Rectangle 特徵(REC)示意圖。(a)正立矩形，(b)45°傾斜版本的矩形。

如圖 3 所示，REC 特徵計算區域間影像亮度值的加減，例如白色區域減黑色區域，寫成數學型式可表示為式(3)：

$$F_{i,r}^{REC} = \text{White pixel sum} - \text{Black pixel sum} \quad (3)$$

下標 i 代表區塊 B_i ，下標 $r = 1 \sim 10$ 代表上圖 REC feature 的子類型。(a)部分的五種特徵和 Viola 與 Jones 所使用的差不多，而(b)部分的五種是 Lienhart 與 Maydt [9] 提出的 45°傾斜版本，有助於偵測的準確率。計算像素的亮度值和可使用積分影像(integral image)的技巧來加速，針對(a)的一般矩形和(b)的傾斜矩形都有其計算法。

- Edge Orientation Histogram (EOH)：

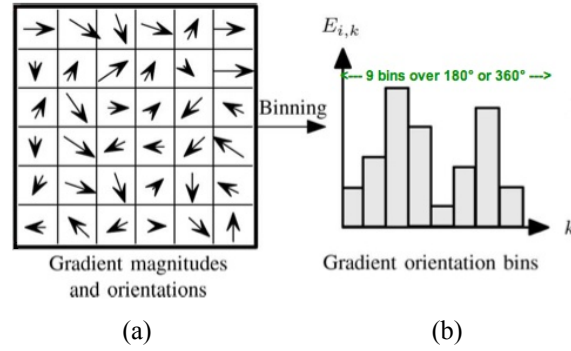


圖 4：Edge Orientation Histogram 特徵(EOH)示意圖。(a)區塊內各個像素的梯度量值與方向，(b)將梯度資訊依方向角落在的 bin 統計出成直方圖。

對於區塊 B_i ，我們可算出各點的水平方向 G_x 和垂直方向 G_y 的梯度值，再透過式(4)、式(5)的關係式得到如圖 4(a)的梯度量值 m 和方向角 θ ：

$$m(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \quad (4)$$

$$\theta(x, y) = \tan^{-1} \frac{G_y(x, y)}{G_x(x, y)} \quad (5)$$

再將 $0^\circ \sim 360^\circ$ 的 θ 角度範圍分成 K 個 bin，例如 $K = 9$ ，每個 bin 40 度，統計出直方圖(histogram)如圖 4(b)。

EOH 所衡量的是在 B_i 中每個梯度角度 bin 值 ($E_{i,k}$)，佔所有 bin 值總和的比例，如式(6)所示：

$$F_{i,k}^{EOH} = \frac{E_{i,k}}{\sum_{j=1}^K E_{i,j} + \epsilon} \quad (6)$$

下標 $k = 1 \sim K$ 代表直方圖的第 k 個 bin， ε 代表防止除數為 0 所加上的小正數。類似積分影像法，EOH 特徵也可用積分直方圖(integral histogram)的技巧來加速計算。

- Edge Density (ED)：

對於區塊 B_i ，如同圖 4(a) 計算出梯度量值後，ED 特徵所衡量的是 B_i 範圍內的平均梯度量值，而不在乎方向，如式(7)：

$$F_i^{ED} = \frac{\sum_{(x,y)} m(x,y)}{\text{area}(B_i)} \quad (7)$$

如同 REC 特徵，ED 特徵也可使用積分影像來加速計算。

總計對於一個區塊，在上述例子中可取出 10 個 REC，9 個 EOH，和 1 個 ED 共 20 個特徵值。對於整張影像，我們嘗試所有可能的區塊大小和區塊位置，每次都去擷取該區塊的 20 個特徵值。以文獻 [6] 為例，一張影像共有 $6948 \times 20 = 138,960$ 個特徵值。雖然維度非常高，但重點在於接下來的訓練過程「一次只使用一個特徵」，也就是 AdaBoost 弱分類器是做一維的分類。我的實作方式是用最簡單的分別對兩類資料取平均，再將決策邊界(decision boundary)訂為兩個平均值的平均： $(\text{pos_mean} + \text{neg_mean}) / 2$ 。

(2.2) 訓練/偵測架構

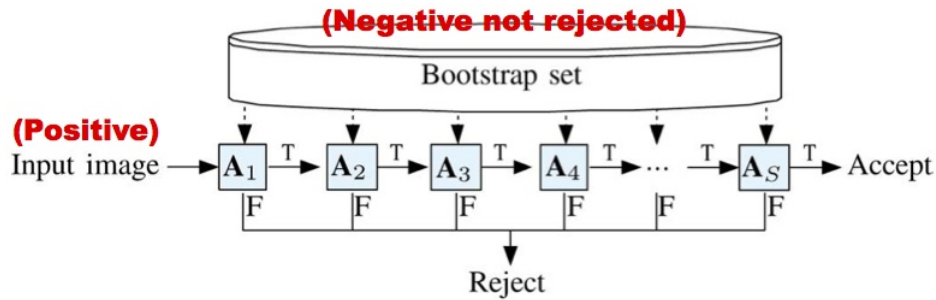


圖 5：訓練/偵測架構示意圖。

圖 5 為訓練和偵測的架構圖，是一個決策樹。在滑動視窗尋找車輛的應用中，大部分的視窗都是否定資料(negative data)，即無車輛，因此我們的目標是在這些時候能快速淘汰掉(reject)否定資料，也就是在串接(cascade)的前面幾個階段就能判斷出是否否定；而佔少數的肯定資料(positive data)則必須通過所有的方框才能被接受(accept)。以下分作訓練和偵測兩部分來說明：

- 訓練：

圖 5 中的每一個方框 A_i 都代表一個 AdaBoost 的強分類器(AdaBoost 階段)，左方代表肯定資料，上方的 bootstrap set 代表否定資料。此種訓練架構需要非常大量的否定資料，假設肯定資料量 N_1 ，否定資料量 N_2 ，則 N_2 必須是 N_1 的十多倍至數十倍，例如 $N_1 = 500$ ， $N_2 = 10,000$ 。

每一個 AdaBoost 階段都會訓練出一個 AdaBoost 強分類器。在每一個階段，肯定的 N_1 筆資料都會使用到，否定部分則只隨機選出 N_1 筆「尚未被淘汰掉(not rejected)」的資料來訓練，來確保肯定、否定的資料量一致。強分類器訓練完後，能成功被正確分類的真否定(true negative)資料將從 bootstrap 中被「淘汰(reject)掉」，之後的 AdaBoost 階段將不再被列為候選。這樣做的道理，是因為能在前面的訓練階段就被分類正確否定資料，代表「已經被學會了」，是相對簡單的例子。後面的訓練階段會針對還學不會的相對「困難」例子來加強。

訓練過程會一直持續到 N_2 筆否定資料幾乎被淘汰完，以致於剩不到 N_1 筆可被選出為止；或是因整體的偽否定率(false positive rate)達到所設定的目標 F_target 而結束。

- 偵測：

如同本節一開始的敘述，測試資料從 A_1 開始經過一連串 AdaBoost 強分類器的判斷。只要其中一個 A_i 判斷為否定，程式就判斷該張影像為否定並結束（較快）；只有在所有的 A_i 都判斷為肯定時，程式才將該張影像判斷為肯定（較慢）。

(2.3) 訓練演算法

圖 6 中的演算法有幾個主要的可調參數和範例值： $F_target = 1e-20$ 是整體(overall)偽否定律的目標值； $d_minA = 99.9\%$ 是每階段最小可接受的偵測率(detection rate)，差 0.1% 是預防極端值； $f_maxA = 50.0\%$ 是每階段最大可接受的局部(local)偽否定律。演算法如圖 6 表示：

```
while (F_current > F_target) do {
  (1) Learn AdaBoost stage A(i,j) {
    Randomly select  $N_1$  non-rejected negatives from bootstrap set;
    Set f_local = 1.0;
    while (f_local > f_maxA) do {
      [1] Add the weak classifier to A(i,j) that minimizes error;
      [2] If necessary, modify threshold of A(i,j) to fulfill d_minA;
      [3] Update f_local of A(i,j);
    }
    F_current = F_current * f_local;
    Reject the negatives that were correctly classified;
  }
  (2) Learn Meta stage M(i) ... skipped
```

}

圖 6：訓練演算法。

f_local 代表在該 AdaBoost 階段的局部偽否定率，必須小於等於 f_maxA 才會結束此階段。

添加弱分類器 h 的部分，我的實作和傳統的 discrete AdaBoost [10] 一樣，有各筆的資料權重 $D(i)$ ，和「加權錯誤率(weighted error rate)」 ϵ ，如式(8)：

$$\epsilon = \sum_{i=1}^N D(i) \cdot [y(i) \neq h(x_i)] \quad (8)$$

當資料 x_i 的弱分類器結果 $y(i)$ 不等於真實類別 $y(i)$ 時，將資料權重累積至 ϵ 。

每次添加 h 時選出有最小 ϵ ，也就是表現最佳的(區塊 id, 特徵 id)加入強分類器，接著依照傳統方法 [10] 算出該弱分類器的權重 α ，並更新資料權重。最後的強分類器判斷結果可視為各個弱分類器結果 $h_t(x)$ 的「加權投票」如式(9)：

$$H(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x) - \text{threshold}) \quad (9)$$

threshold 代表一個調整結果的門檻值，每個強分類器都有一個 threshold。

接著說明演算法中「調整 threshold 使滿足 d_minA 」的意義。若去掉式(9)的 threshold，一維資料分布及強分類器邊界（棕線）可能如圖 7(a)：

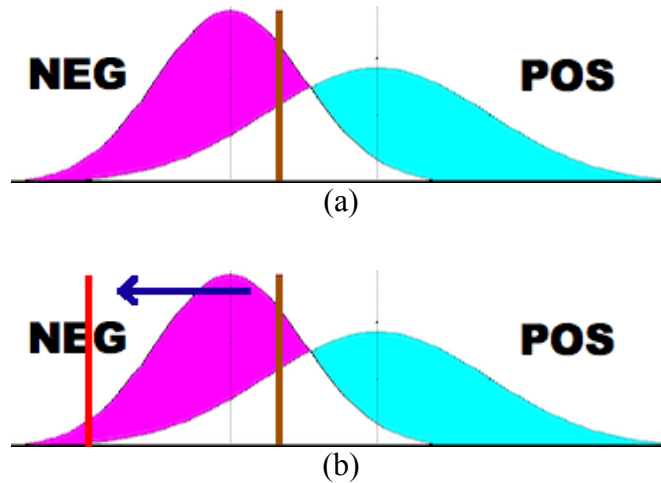


圖 7：扣掉 threshold 以調整強分類器的結果。(a)調整前，(b)調整後。

POS(淺藍色)表示肯定資料，NEG(粉紅色)表示否定資料。

棕線代表的數值是 0，此時肯定資料無法被分到 99.9% 正確。於是我們調整強分類器的判斷結果，將決策界線從 0 移至紅線，使得肯定資料可以 99.9% 被分對（滿足 d_minA ），如圖 7(b)。這個調整的量（負值）就是該強分類器的

threshold，偵測的時候該強分類器也是依照 threshold 調整判斷結果。

因為時間關係，演算法中細節複雜的 Learn Meta stages 被我省略掉了。其實文獻 [6] 中所訓練的不僅有 AdaBoost 階段，也有在中間安插「Meta 階段」。而其採用的 AdaBoost 不是傳統的 discrete AdaBoost，而是 real AdaBoost [11]，並根據此算出強分類器的信賴值(confidence value)。Meta 階段就是根據之前數個（例如 2 個）階段的信賴值（每筆資料有 2 個信賴值，組成一個特徵向量），然後用 SVM 訓練出一個 2-D 分類器。相關的概念如圖 8：

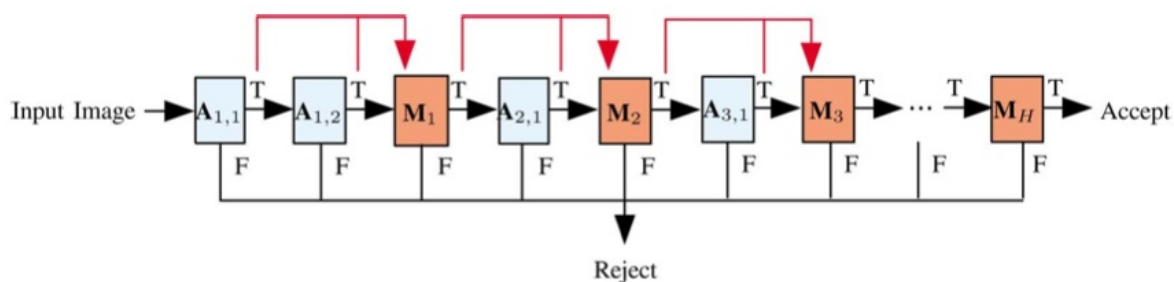


圖 8：於訓練/偵測架構中安插 Meta 階段分類器。

(五)實驗成果與討論

本計畫的程式碼、演示影片、影像資料、報告等檔案可於 Google Code 的專案托管網頁 [12] 下載或觀看。

(1)車道標線與人行道邊界

此為計畫前期的實作目標，使用無任何函式庫的 C 語言在 Intel Cuo 2 Duo 2.16GHz CPU, 3GB RAM 的 MacBook 筆記型電腦上，搭配 Mac OS X 10.6 作業系統與內建編譯器 gcc 4.2 執行。目前僅實作出直線偵測，並沒有做任何其他判斷與引導使用者。其中兩張影像的偵測結果如圖 9：

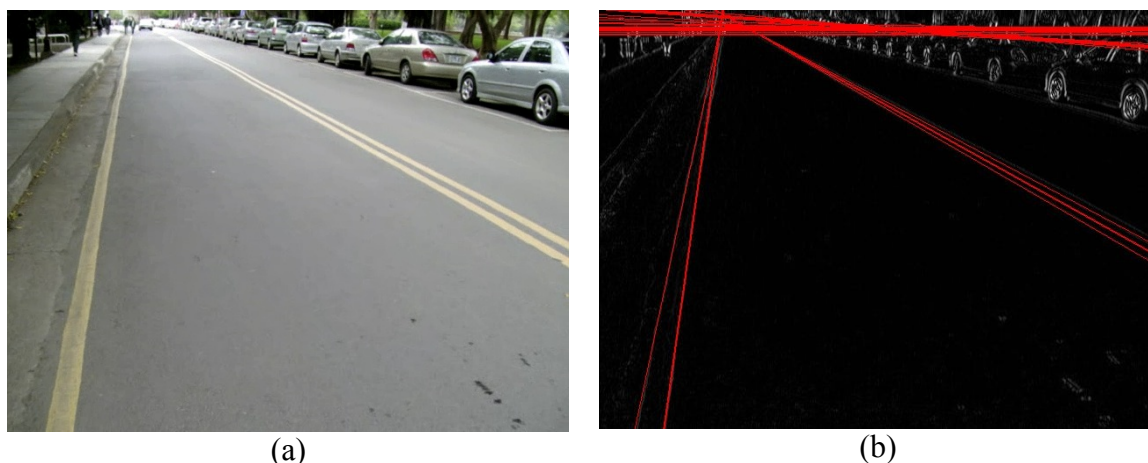




圖 9：車道標線、人行道邊界的辨識結果。

(a), (c)為原圖，(b), (d)為 Hough 轉換的直線偵測結果。

直線偵測的結果看似不錯，不過目前每一張影像都必須人為介入設定參數，才能偵測到適當數量的直線。指導教授建議不宜輕易主觀判斷，設定一些門檻值就將資訊丟棄，而是要盡量保留任何可能有用的資訊，交給演算法本身的「投票機制」來決定。

由於計畫後期專注在來車偵測的部分，直線偵測的程式碼已不再維護，也不再改進功能。

(2) 來車偵測

此為計畫後期的實作目標，使用包含 OpenCV 2.0, C interface 函式庫 [13] [14] 的 C++ 語言，在相同的 MacBook 筆記型電腦上，搭配 Mac OS X 10.6 作業系統與 Xcode 3.2 的 IDE 執行。

(2.1) 影像資料(dataset)

由於開走的車和橫向經過的車對於行人威脅不大，不太符合視覺輔助的目的，此計畫特別強調是「來車」的偵測。又為了針對國立清華大學的校園，所有的車輛、非車輛照片皆由自己拍攝，車輛一律為「正面」或是「有斜角的正面」。目前共收集 342 張車輛，199 張非車輛影像，並用 DV 拍下數十分鐘的校園行走影片（尚未使用）。偵測視窗大小定為 128 x 128，因此車輛影像皆手動切為 128 x 128，並用自己的程式把 199 張非車輛影像切成 9355 張 128 x 128 的小影像。

目前訓練資料共有 300 張肯定，9000 張否定影像，測試資料共有 42 張肯定，355 張否定影像。影像資料可於專案網頁下載，數張範例如圖 10：



圖 10：國立清華大學校園影像範例。

(a), (b), (c), (d) 為車輛資料，(e), (f), (g), (h) 為非車輛資料。

(2.2) 程式實作

由於時間和人力因素，實作時將文獻 [6] 做了許多簡化：特徵種類只用了 5 個正立矩形的 REC 特徵，故方法近似於 Viola & Jones 演算法 [3]；且訓練架構是使用 discrete AdaBoost 而不是文獻 [6] 用的 real AdaBoost；另外，也沒有像文獻 [6] 使用信賴值的概念，插入 Meta 階段分類器加強結果。

執行介面部分，目前只完成對於單一影像/無滑動視窗的車輛分類器，可使用 GUI 一次分類一張或 CLI 批次分類整個目錄。而單一影像/有滑動視窗，影片輸入，甚至是攝影機輸入等功能尚未完成。

訓練演算法的參數： F_{target} 設為一極小正數，確保否定資料能盡量用完； d_{minA} 設為 100%，使得調整 threshold 之後所有肯定資料都必須被分對； f_{maxA} 設為 50%，使得每一訓練階段皆可淘汰 $(N_1 / 2)$ 以上筆否定資料。

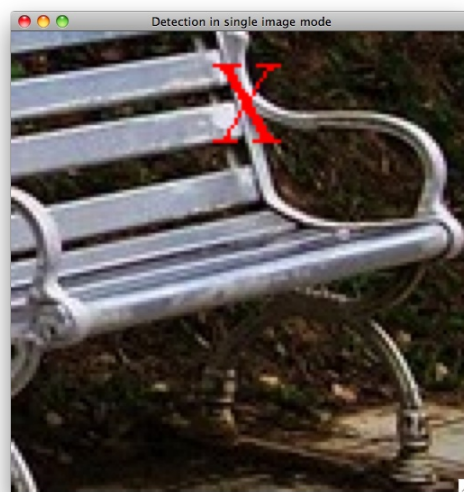
(2.3) 偵測結果

GUI 程式執行介面如圖 11，YouTube 演示影片可於專案網頁觀看：



[+]: Score = 75.057

(c)



[-]: Rejected at stage 8

(d)

圖 11：來車偵測的程式執行畫面。

(a), (c)為肯定資料成功偵測的畫面，(b), (d)為否定資料成功拒絕的畫面。

對於判斷為肯定的影像，在命令列上會顯示一個參考的分數（定為所有 $H(x)$ 的和）；判斷為否定的影像，會顯示出是在第幾個階段被拒絕掉。

實驗共訓練出四個模型(model)檔，記錄所有強、弱分類器的資訊。模型檔的檔名取為 model_xxx_yyyy，xxx 為肯定資料量 N_1 ，yyyy 為否定資料量 N_2 ，因 N_2 過大時程式有 runtime error 尚未解決，目前最大只測到 $N_2 = 3,000$ ，而不是 9,000。數據如圖 12：

| 模型檔名 | 階段數 | 整體訓練偽否定率 | 訓練情形 | | 偵測情形 | |
|----------------|-----|----------|----------------------|---------------------|---------------------|--------------------|
| | | | 偽肯定率 | 偽否定率 | 偽肯定率 | 偽否定率 |
| model_100_1000 | 16 | 1.00e-06 | 4.1% (41 / 1000) | 8.0% (8 / 100) | 4.5% (16 / 355) | 40.5% (17 / 42) |
| model_100_3000 | 47 | 6.85e-21 | 1.8% (54 / 3000) | 14.0% (14 / 100) | 0.8% (3 / 355) | 54.8% (23 / 42) |
| model_200_3000 | 23 | 2.19e-10 | 4.6% (137 / 3000) | 7.0% (14 / 200) | 5.9% (21 / 355) | 19.0% (8 / 42) |
| model_300_3000 | 15 | 4.65e-07 | 9.5% (284 / 3000) | 3.0% (9 / 300) | 15.2% (54 / 355) | 7.1% (3 / 42) |

圖 12：來車偵測的實驗數據。

其中整體訓練偽否定率即為演算法中的 $F_{current}$ 。

(2.4)結果討論

速度方面，批次處理平均下來，一張影像(無滑動視窗)的偵測時間不到 0.005 秒，也就是一秒鐘可以處理 200 張。以滑動視窗搜尋車輛的需求來說，速度已經足夠；但影片輸入和攝影機輸入要求更快速，200 張/秒未必足夠。

錯誤率方面，由圖 12 的數據可看出，當 N_2/N_1 的比值越大，AdaBoost 階段越多，整體訓練偽肯定率 $F_{current}$ 越低，系統越容易「拒絕」送進來的資料。又隨著資料量 N_1 的增加，表現也越好，偵測的偽否定率從 54.8%大降至 7.1%，可惜也犧牲了偽肯定率，從 0.8%上升至 15.2%。由圖 13 負斜率的偵測錯誤取捨曲線(Detection Error Tradeoff, DET)可看出，偽肯定率與偽否定率之間，確實存在著取捨。

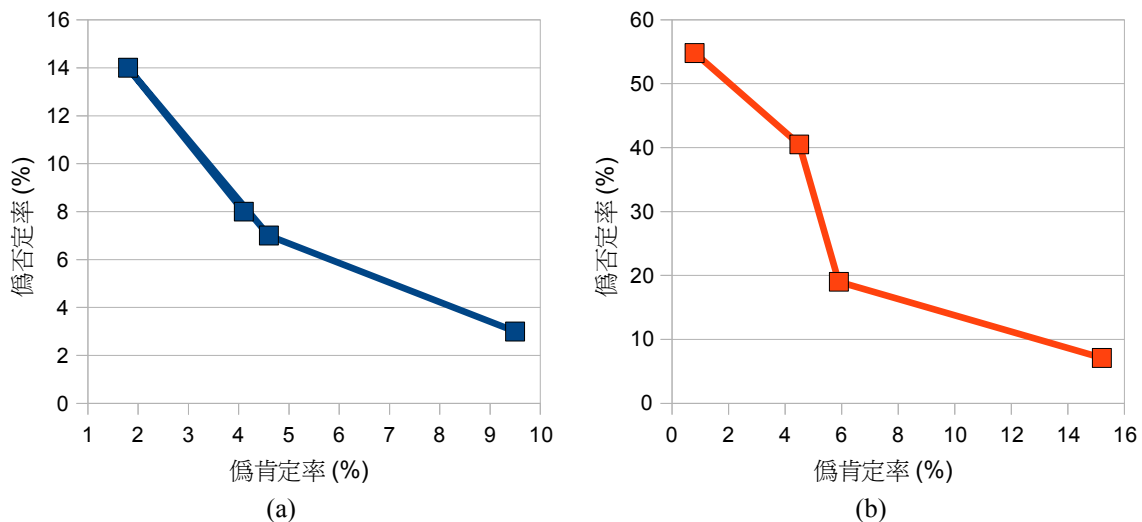


圖 13：偵測錯誤取捨曲線。(a)為訓練情況，(b)為偵測情況。

弱分類器的使用數，以模型 model_300_3000 為例，15 個 AdaBoost 階段分別需要[6, 4, 6, 4, 4, 4, 4, 3, 3, 3, 6, 6, 4, 6, 6]個弱分類器，並不像文獻 [6] 的結果如圖 14：由於否定資料越來越「困難」，每次使用的弱分類器越來越多。

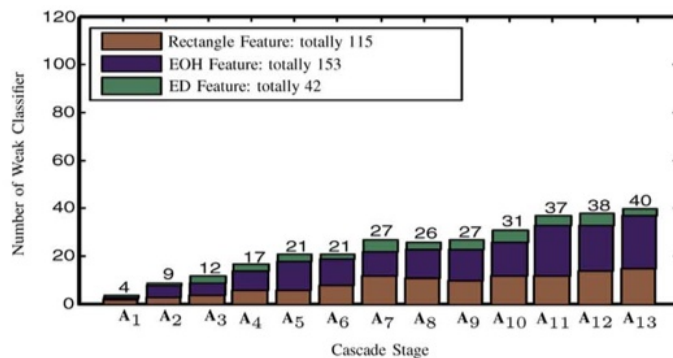


圖 14：文獻 [6] 中每個 AdaBoost 階段所需要的弱分類器個數，有越用越多的趨勢。

推測原因可能是因為訓練架構使用的是 discrete AdaBoost，使得由 T 個弱分類器所構成的強分類器，如式(9)所加權出來的結果 $H(x)$ 只集中在 2^T 種可能。不像文獻 [6] 使用的 real AdaBoost， $H(x)$ 的可能結果分布於整個實數軸，而影響弱分類器的需要個數。

由於時間及人力因素，簡化了文獻 [6] 的許多細節，實驗結果其實還有很大的進步空間。以下列舉一些可能的改良，供未來參考：

1. 補上EOH、ED、45°傾斜的REC特徵，並配合積分立方圖、積分影像的技巧加速。這會是很顯著的改良，至少EOH特徵的準確度是很不錯的。
2. 訓練架構改用real AdaBoost，並計算信賴值特徵，再用SVM訓練出Meta階段分類器。這樣可符合文獻 [6] 所強調的分類器改良。
3. 在單張影像/無滑動視窗的分類夠精準以後，增加單張影像/有滑動視窗，影片輸入，攝影機輸入等使用者介面。這是當初的理想操作情況，最能模擬視障者配戴智慧型裝置的情形。
4. 目前的實驗原理皆只用到機器學習領域的知識來分類單張影像，但由於電子式行進輔具的輸入資料往往是一連串의 影片，如果能結合電腦視覺領域的「物件追蹤(object tracking)」，對於車輛的動向將能更有效地掌握。

(六)參考文獻

- [1] http://blindconfidential.blogspot.com/2007_04_01_archive.html
- [2] 孫中麒，低價位之導盲系統，國立中央大學碩士論文，2005。
- [3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE Proc. International Conference on Computer Vision and Pattern Recognition, 2001*, vol. 1, pp. 511–518.
- [4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proc. IEEE International Conference on Computer Vision and Pattern Recognition, 2005*, vol. 1, pp. 886–893.
- [5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [6] Y. T. Chen(陳昱廷) and C. S. Chen(陳祝嵩), "Fast human detection using a novel boosted cascading structure with meta stages," *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1452–1464, 2008.
- [7] Duda, R. O. and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Comm. ACM*, vol. 15, pp. 11–15, 1972.
- [8] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," 3rd Ed., pp. 733–788, Prentice Hall, 2008.
- [9] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," *Proc. IEEE International Conference on Image Processing, 2002*, vol. 1, pp. 900–903.
- [10] <http://en.wikipedia.org/wiki/Adaboost>

- [11] R. E. Schapire and Y. Singer, “Improved boosting algorithms using confidence-rated predictions,” *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.
- [12] 本計畫專案網址：<http://code.google.com/p/candy2009>。
- [13] <http://opencv.willowgarage.com/wiki>
- [14] G. Bradski and A. Kaehler, “*Learning OpenCV: Computer Vision with the OpenCV Library*,” 1st Ed., *O'Reilly Media*, 2008.