

第三章 車牌定位

一個完整的車牌辨識系統的主要部分為車牌定位系統、字元切割、字元辨識三部分。要獲得良好的辨識效果，除了良好的辨識系統外，最重要的就取決於在此之前的車牌定位與字元切割部分。因此，車牌定位對於一個車牌辨識系統來說是相當重要。

3-1 車牌影像定位的難題

一個車牌定位系統通常包括影像前處理、車牌區域搜索、車牌定位與字元分割等部分。車牌取像常受以下因素，而使得不容易完成車牌定位：

- (1) 擷取圖像受環境因素干擾（環境光照不均勻造成反光、或光源不足等），照片品質難以保證。
- (2) 其他字元或符號區域干擾，難以準確定位。
- (3) 車牌出現污點、變髒、模糊、褪色等。
- (4) 車牌被部分遮擋或車牌與車子顏色相同。
- (5) 車牌破碎或被反折。
- (6) 運動（高速行駛）圖像的模糊失真，形成鋸齒等。
- (7) 攝影器材的視角與放大倍率。

在以上所列的困難條件下，如何利用有限的影像資訊，去做車牌定位是相當困難的一件事。若以車牌辨識成功率取決於車牌定位成功率，則車牌定位成功率為研究方法的強健性為第一重要性外，第二個重要性就是前處理的影像處理。所以針對影像受環境因素干擾部分，本章的夜間車牌定位演算法是利用 BBHE 影像增強方法來先對影像作增強，減少因影像亮度不均勻下對影像處理時所遺失的車牌區域，提高處理效果及成功率。

3-2 車牌定位系統架構

目前車牌定位與車牌辨識系統的文獻，多是停車場入口的車牌定位與辨識，在其背景光源充足的情況下，其前處理鮮少做影像強化的動作，就直接進入車牌搜尋法則，因此，定位成功率很高。但在室外的光源不定及光源方向不定下，此類搜尋法則的車牌判別成功率就相當得低。

所以前處理以單純的影像轉成灰階影像並不可滿足我們對影像處理的要求，因此在本章提出兩種演算法，即日間演算法與夜間演算法來做車牌偵測，本系統架構流程圖如圖 3.1 所示。

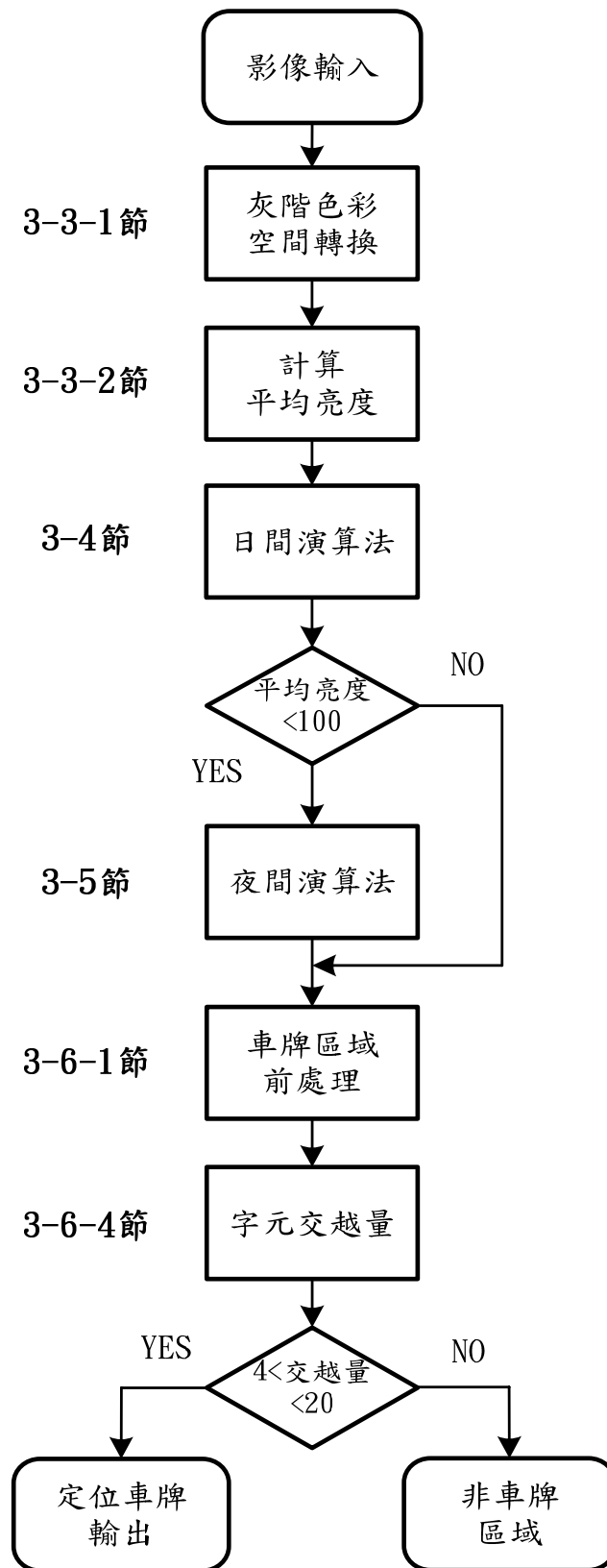


圖 3.1 車牌定位系統架構流程圖

3-3 前處理

在進入車牌偵測前，為了增加處理速度及針對影像特性的處理，在前處理部分分為兩部分：3-3-1 節灰階色彩空間轉換及 3-3-2 計算平均亮度。其中，統計平均亮度有兩種功用，一是為了區分日、夜間演算法的判斷閾值，二是當作日夜間二值化之參考閾值。

3-3-1 灰階色彩空間轉換

由於本文採用彩色數位相機拍攝，所擷取的原始影像為 RGB 色彩空間影像，影像中含有 R、G、B 三種色彩的資訊。在本系統中，基於系統整體速度的提升（利用灰階影像可在定位車牌部分可節省一半時間）與本系統對於色彩資訊不是非常需求，因人類眼睛對於影像中的亮度（Luminance，如 YIQ 色彩空間的 Y）感受比較強烈，所以在進入車牌偵測前先做影像色彩灰階轉換。

彩色影像的每一個像素利用三個位元組來記錄色階強度即 R、G、B 等 3 個分量，其中每個像素佔一個位元組（0~255）的平面。對於任何彩色空間分量 R、G、B 轉換為灰階平面的公式，可由 NTSC（National Television System Committee，美國聯邦電視規格委員會）的標準對亮度分量求得，其定義如下：

$$Y = 0.299R + 0.5817G + 0.144B \quad (3-1)$$

其中 Y 為亮度資訊，R、G、B 各代表三原色。

3-3-2 計算平均亮度

為了增加處理速度，避免不必要的處理，本節提出計算平均亮度做為處理亮度過暗或不均的情況時，能自動區分日、夜間演算法。其作法為對整張

原始灰階影像作亮度統計，再以總亮度數除以總像素數，即為平均亮度。在統計 40 次平均亮度的實驗結果中，在白天影像部分其平均亮度遠高於 100，其統計次數如圖 3.2 所示。而亮度過暗的影像平均亮度都遠小於 70，其統計次數如圖 3.3 所示。因此，設定判斷平均亮度的閾值為 100。在原始影像的平均亮度高於 100 時使用白天車牌偵測演算法，而低於 100 時則是夜間演算法與日間演算法綜合使用。

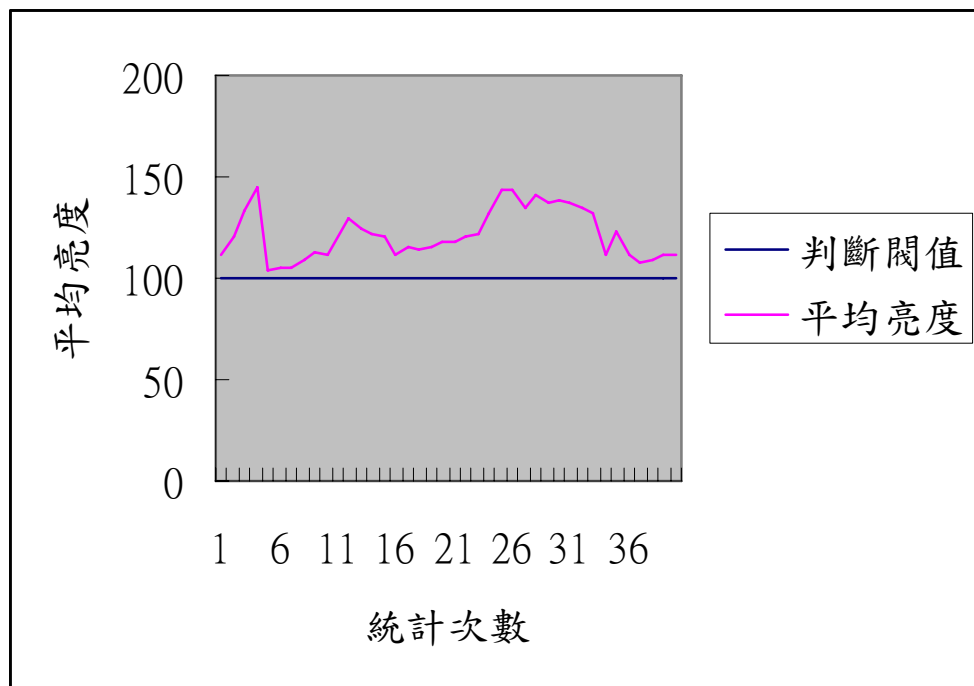


圖 3.2 白天平均亮度統計圖

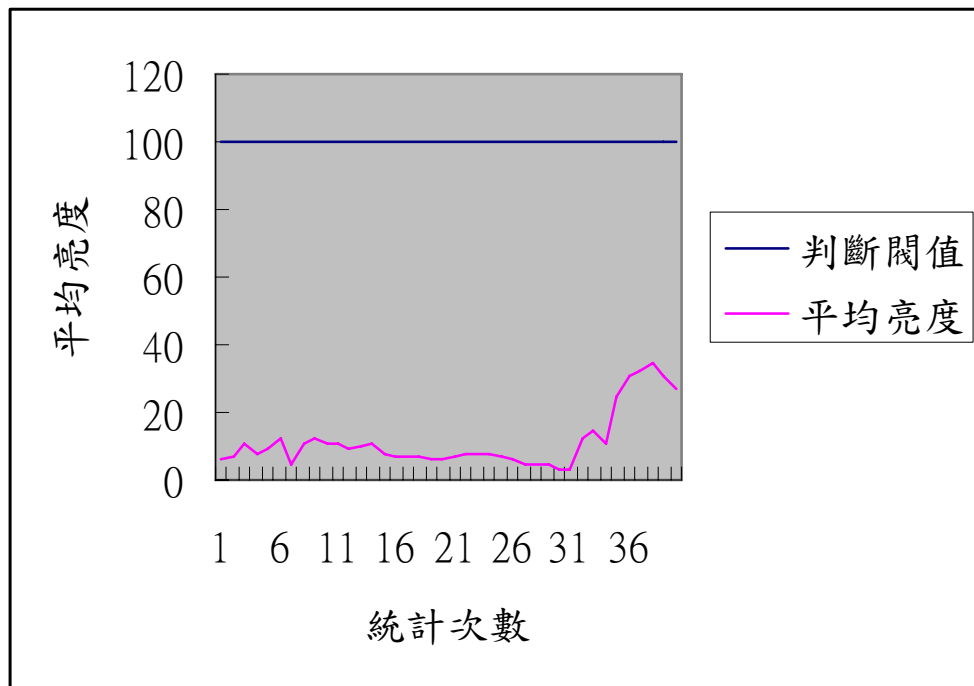


圖 3.2 夜間平均亮度統計圖

3-4 日間車牌偵測演算法

3-4-1 日間演算法架構

日間演算法架構流程圖如下圖 3.4 所示，其中方塊底色較深為本文作法與[17][20]不同的地方，詳細作法將於之後各節做詳細介紹。

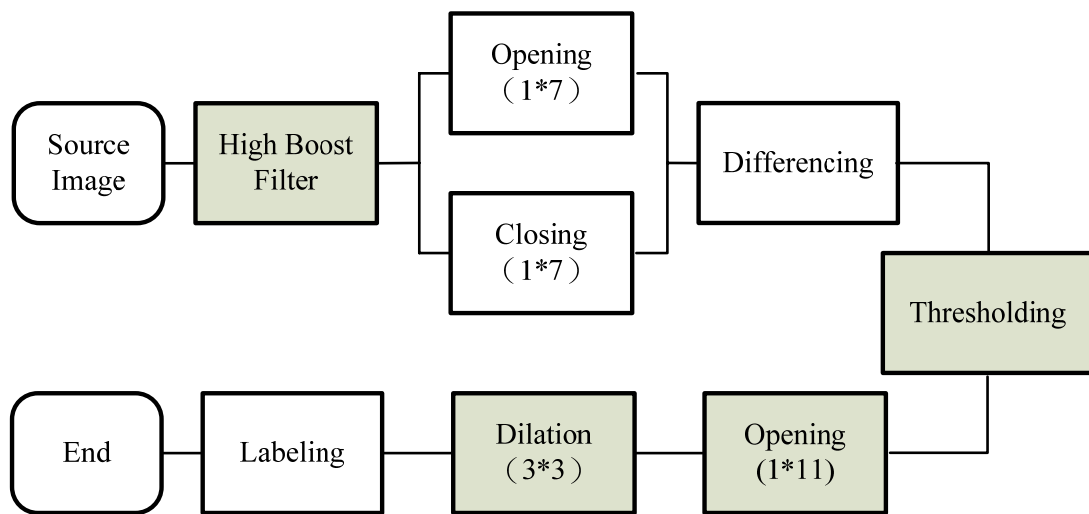


圖 3.4 日間演算法架構流程圖

3-4-2 High Boost Filter

為了使影像邊緣更明顯銳利，所以需先增強對比，增強對比的方法有很多，如[20]所用的 BottomHat 即是一種很好的增強對比的方法，但考量效果與速度下，本研究選用 High Boost Filter 來增強影像對比，其速度跟 BottomHat 相比下，效能並不會太差，而速度卻快上許多。

利用 2-1-2 節 High Boost Filter 基本原理的方法，經實驗比較後，在考量速度與效果下，我們設定此低通濾波遮罩大小為 3×3 。在考量高增幅係數之大小，因為高增幅係數若太低如係數為 1、2，則影像對比並無太大增強，高增幅係數設定太高如係數為 6、7，則會出現過多的大顆雜訊，因此經實驗後，以高增幅係數 α 設為 4 的效果比較好，其中 I 為原始影像， I_{Lp} 為經低

通濾波後的模糊化影像，其高增幅影像 I_{HB} 定義如下：

$$I_{HB} = \alpha \times (I) - I_{LP} \quad (3-2)$$

處理效果如圖 3.5 所示，可發現在具有車牌特徵的區域對比，因此大大增強，經過增強對比後可以方便後續的處理。



圖 3.5 原圖灰階影像經過 High Boost Filter (a) 原影像，(b) 影像增強後

圖 3.5 (a) 為原圖影像，圖 3.5 (b) 為圖 3.5 (a) 轉換為灰階影像經 High Boost Filter 影像增強後的結果。

3-4-3 形態梯度

二值數學形態學在 1964 年 Matheron 與 J. Serra 提出時，只適用二值影像 (Binary image)。Serra 在 1982 年提出灰階數學形態學的原理和方法，再經 S. R. Sternberg 提出灰階數學形態學的影像處理應用，形態學才開始有了更多多采多姿的應用變化。

有了 2-1-3 節形態梯度原理的概念後，在此我們不用二值形態學來做運

算，因為二值化後的影像其判斷不是黑就是白，會使得其後的運算損失過多特徵，使得其運算後的影像並非理想的結果。因此，我們利用灰階數學形態學來做灰階形態梯度的運算，即可改善此一缺點。

灰階形態學運算原理跟二值形態學非常近似，其中 A 為原灰階影像， B 為結構元件，其定義如下：

$$\text{膨脹 (Dilation): } D(A, B) = \underset{B}{\text{Max}}(A) \quad (3-3)$$

$$\text{侵蝕 (Erosion): } E(A, B) = \underset{B}{\text{Min}}(A) \quad (3-4)$$

$$\text{開運算 (Opening): } O(A, B) = \underset{B}{\text{Max}}(\underset{B}{\text{Min}}(A)) \quad (3-5)$$

$$\text{閉運算 (Closing): } C(A, B) = \underset{B}{\text{Min}}(\underset{B}{\text{Max}}(A)) \quad (3-6)$$

在此條件下，灰階形態學運算就成為 Maximum filter 和 Minimum filter。因此，在灰階影像上進行膨脹運算後，輸出影像會比輸入影像明亮，即影像中較為黑暗的細節部分會因結構元件的值與形狀減少或消除。在灰階影像上進行侵蝕運算後，輸出影像比輸入影像為暗，即影像中較為明亮的細節部分會因結構元件的值與形狀減少或消失。

灰階形態梯度 (Gray Morphological Gradient) 為典型的梯度濾波器，可產生一個包含強度 (magnitude) 和方向 (orientation) 的向量，使輸入影像中具有尖銳灰階變化部分變的更為明亮。其灰階形態梯度定義如下：

$$G(A, B) = \underset{B}{\text{Max}}(A) - \underset{B}{\text{Min}}(A) \quad (3-7)$$

因為灰階形態梯度在我們系統中處理效果並不明顯，所以我們利用此原

理，將其修改為原圖經閉運算的結果減去原圖經開運算的結果，其灰階形態梯度影像 I_G 定義如下：

$$I_G = \text{Min}_B(\text{Max}_B(I_{HB})) - \text{Max}_B(\text{Min}_B(I_{HB})) \quad (3-8)$$

考量影像大小 640×480 pixel 在雙車道情況下，其車牌大小最大為 40×20 pixel，其一個車牌字元寬度不超過 7 pixel，所以在此我們定義結構元件 B 為 1×7 的遮罩，其灰階形態梯度影像 I_G 如圖 3.6 所示：

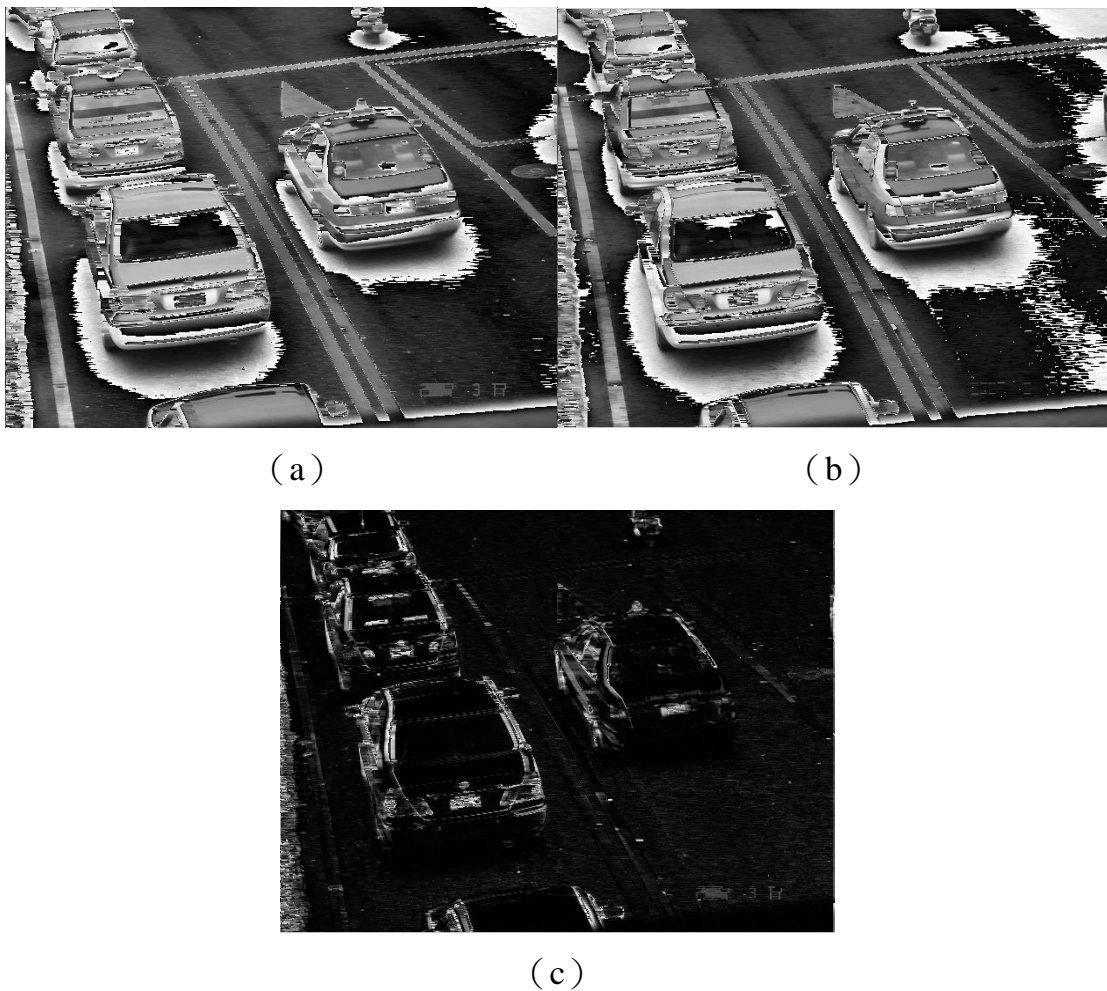


圖 3.6 經過形態梯度及相減後之影像 (a) 閉運算後，(b) 開運算後，(c) 梯度影像 I_G

圖 3.6 (a) 為圖 3.5 (b) 經過閉運算後的影像，圖 3.6 (b) 為圖 3.5 (b) 經過開運算後的影像，圖 3.6 (c) 梯度影像 I_G 為圖 3.6 (a) 減去圖 3.6 (b) 的差值影像。

3-4-4 日間二值化閾值

經過差值運算後，必須二值化影像以利後面車牌候選區的框選。在經過實驗發現，影像的明暗與平均亮度有相當密切的關係，因此我們利用平均亮度乘以設定的常數，即可得到二值化閾值，在此我們得出日間演算二值化臨界值為 0.9 倍的平均亮度值為最佳閾值，其處理效果如圖 3.7 所示：



圖 3.7 經過二值化後的影像

在本文系統架構相同情況下，二值化方法分別使用平均亮度乘上一常數來二值化與使用 Otsu 二值化進行比較，發現以本文所提平均亮度乘上一常數來進行二值化後，其較暗處的車牌仍能框選出來，而 Otsu 二值化則會因閾值過高而遺失車牌特徵，未框選到車牌。其框選比較如圖 3.8 所示。

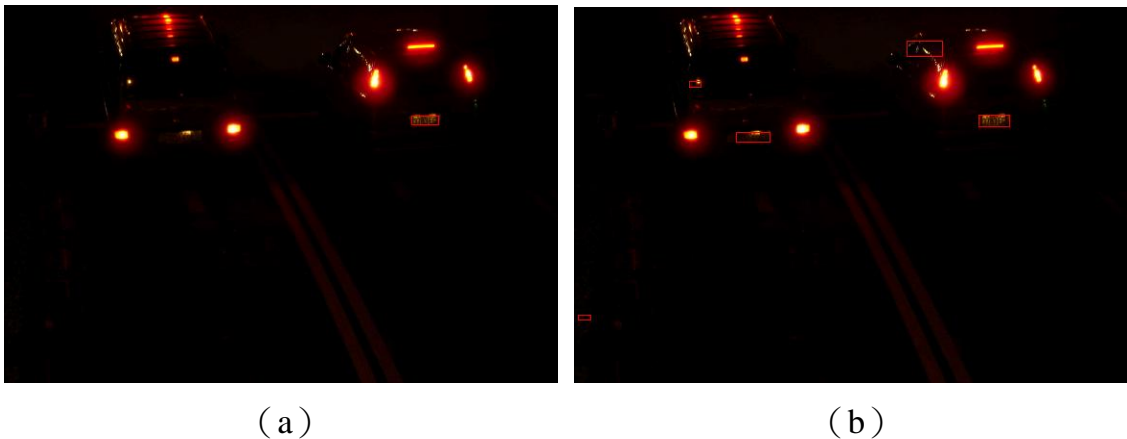


圖 3.8 經 Otsu 與本文的二值化比較圖 (a) Otsu 二值化，(b) 本文二值化

圖 3.8 (a) 經 Otsu 二值化後所框選的候選區只有一個，圖 3.8 (b) 為經本文提出的二值化方法，所框選的候選區有兩個，顯示本方法比較適合用於夜間二值化。

3-4-5 Opening

經過二值化後可看出具有車牌特徵的候選區都顯現出來，但在某些情況下（如夜間有後照光源照射車牌候選區）車牌邊緣會與車體相連，造成車牌候選區過大，因此必須再經過一個開運算運算，斷開車牌與車體相連部分，順便也可去除部分雜訊。

在實驗中，因在某些車輛具有其車子的標誌（如 TOYOTA 為一個圈圈內似廿的標誌）具有車牌字元交越特性，所以也會進入候選區，因我們拍攝

角度是以上往下拍，所以標誌距離車牌間格更加縮小，進而跟車牌相連。其車輛標誌與車牌相連影像如圖 3.9 所示。

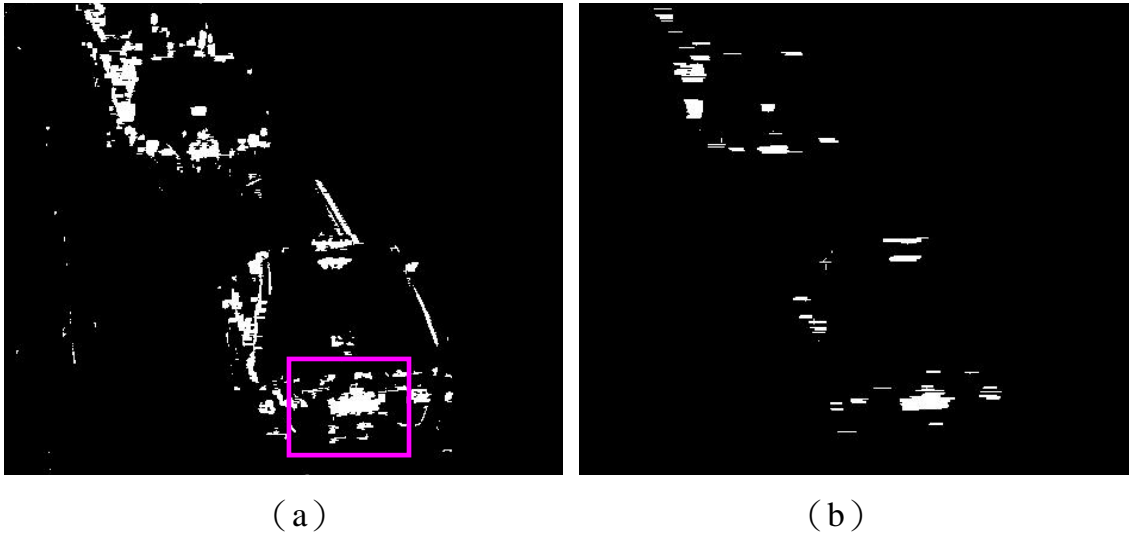


圖 3.9 車輛標誌與車牌相連的處理 (a) 原圖 (b) 經開運算後

圖 3.9 (a) 為車牌與車體標誌相連的影像，圖 3.9 (b) 為經過開運算後的影像，可看出已經斷開相連部分，也濾除不少雜訊。

經過實驗後，發現其標誌大小大約為 7~10 pixel，所以我們定義最佳大小為 1×11 的遮罩來進行斷開車牌與標誌。其中二值化後影像為 I_B ，經過開運算影像 I_O 的定義如下：

$$I_O = I_B \circ B_{m \times n} \quad (3-9)$$

經過開運算後，可發現斷開車牌與車體標誌，另外，整張影像也濾除許多雜訊，剩下的白色區域為可能車牌候選區，其中可很明顯發現較大的白色候選區就是車牌可能的位置，其處理效果如圖 3.10 所示：

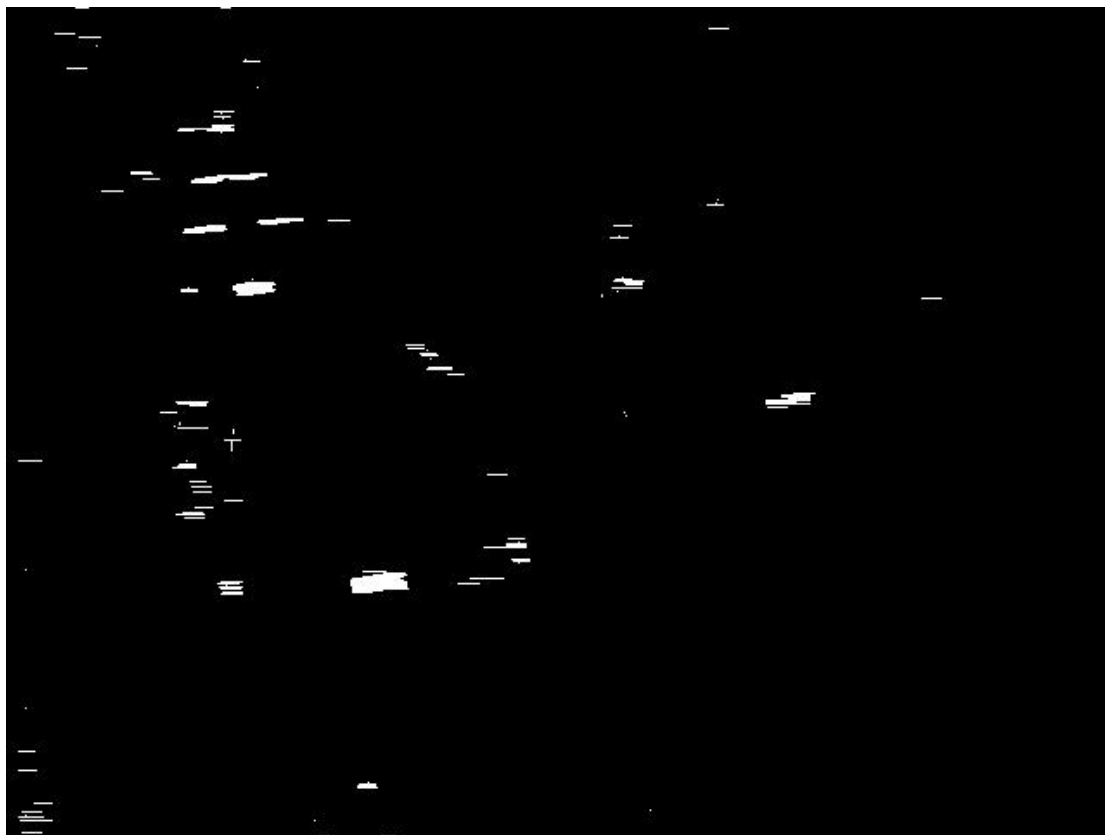


圖 3.10 經過開運算後的影像

3-4-6 Dilation

在 3-4-5 節的實驗中，因經過開運算後其候選區域會變小，若直接進行框選，其框選位置會比原車牌位置小，在判斷車牌物理特徵時，會遺失部分車牌，所以需再一次膨脹，使其回復為正常比例。在此考量膨脹遮罩大小不能太大，避免先前開運算處理時，分割車牌與標誌部分又因遮罩過大而相連。在此我們使用結構元件 B 為 3×3 的遮罩，其中經開運算後的影像為 I_O ，經過膨脹運算影像為 I_D ，因本節的影像為二值影像，所以利用 2-1-4 節的二值形態學中的膨脹運算，其定義如下：

$$I_D = I_O + B_{m \times n} \quad (3-10)$$

經過二值形態學的膨脹處理後，可發現符合車牌長寬比例的候選區，已經明顯凸顯出來，其處理效果如下圖 3.11 所示：



圖 3.11 經過 Dilation 後的影像

3-4-7 Labeling

經過膨脹運算後，這時我們利用車牌的物理特徵的先備知識（如車牌寬、車牌高、車牌寬高比例、車牌面積…等），來對車牌候選區進行框選。現行台灣標準車牌其寬高比例如表 3.1 所示：

表 3.1 現行車牌規格表

種類 \ 比例	寬 度 (L_W)	高 度 (L_H)	寬高比 ($Ratio$)
汽 車	32 公分	15 公分	2.133
機 車	25 公分	14 公分	1.785

因此我們可以利用車牌的寬度 (L_W)、高度 (L_H)、寬高比 ($Ratio$)、面積 ($Area$) 等條件，來判別是否為車牌影像，經實驗統計，我們可定義出三條較為寬鬆的判別式，避免因選取過高的判別臨界值，而遺失部分具有車牌特徵的區域。判別式表示如下：

1. $10 \leq L_w \leq 80$
2. $5 \leq L_H \leq 30$
3. $1.5 \leq Ratio \leq 6$

其判別式的單位為 Pixel，車牌寬高比的定義如下：

$$Ratio = \frac{L_W}{L_H} \quad (3-11)$$

最後利用連通成分分析法，將我們所定義的判別式對圖 3.11 進行標誌 (Labeling)，符合我們判別式定義的區域才留下。最後，將符合判別式的區域位置帶回原圖，進行原圖的車牌候選區的框選，其結果如圖 3.12 所示：



圖 3.12 經過 Labeling 後可框出車牌候選區

3-5 夜間車牌偵測演算法

3-5-1 夜間車牌偵測演算法架構

夜間車牌偵測演算法為先對原影像做日間演算法後，框出車牌候選區域，再對原影像做夜間演算法，框出車牌候選區，綜合兩個演算法的候選區即為夜間車牌偵測演算法的車牌候選區。

3-5-2 夜間演算法架構

因夜晚亮度不夠或不均勻，常會造成定位失敗，必須調整原始影像的整體亮度。其架構流程圖如圖 3.13 所示，其中方塊底色較深為本文提出改良的作法與[17][20]不同的地方。

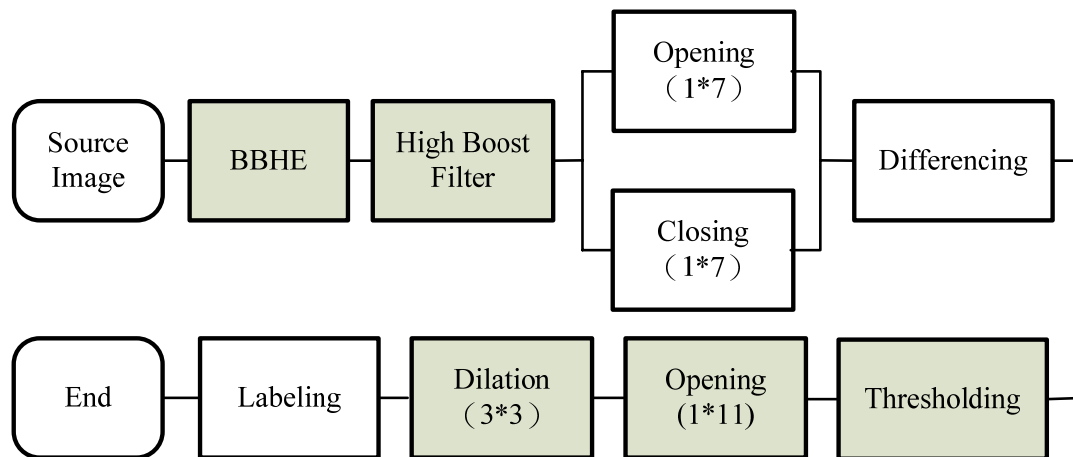


圖 3.13 夜間演算法架構流程圖

夜間演算法是利用日間演算法為架構，在進行 High Boost Filter 前先提高整張亮度，我們選用 BBHE 來做提高整體亮度的方法。而夜間演算法的二值化閾值也跟日間演算法有所不同。

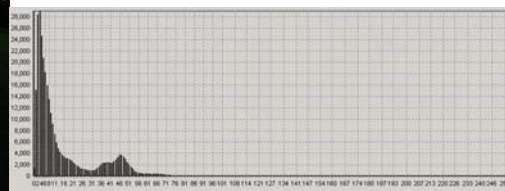
3-5-3 BBHE 演算法

對於提高整體亮度的方法，有相當多文獻提出相當不錯的方法，例如最普遍的 HE 即為我們常用的方法之一。本系統使用為 BBHE，其詳細作法可參考 2-1-3 節。

以下各圖為經 HE 及 BBHE 處理後轉成灰階之直方圖的比較，如圖 3.14 所示：



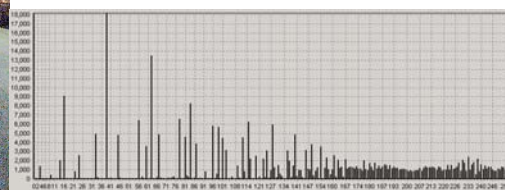
(a)



(b)



(c)



(d)

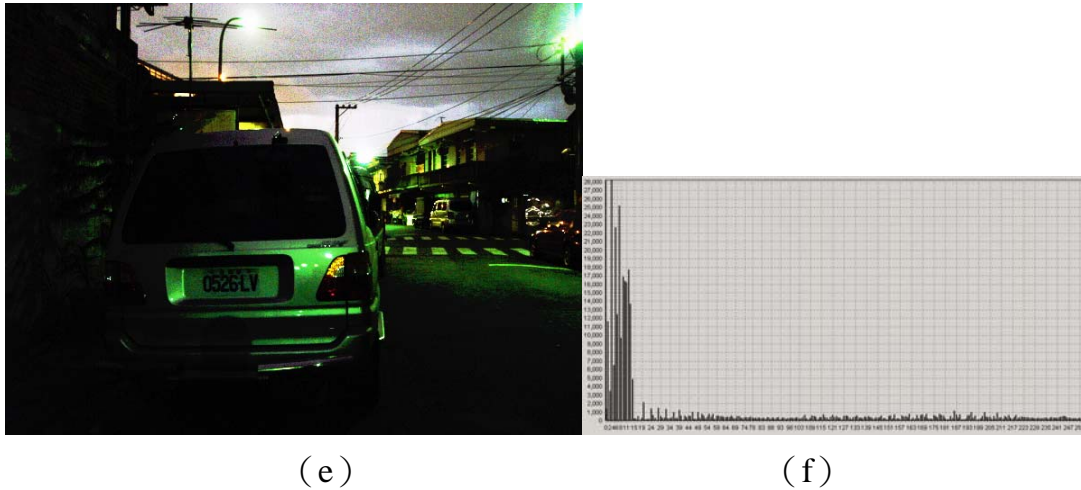
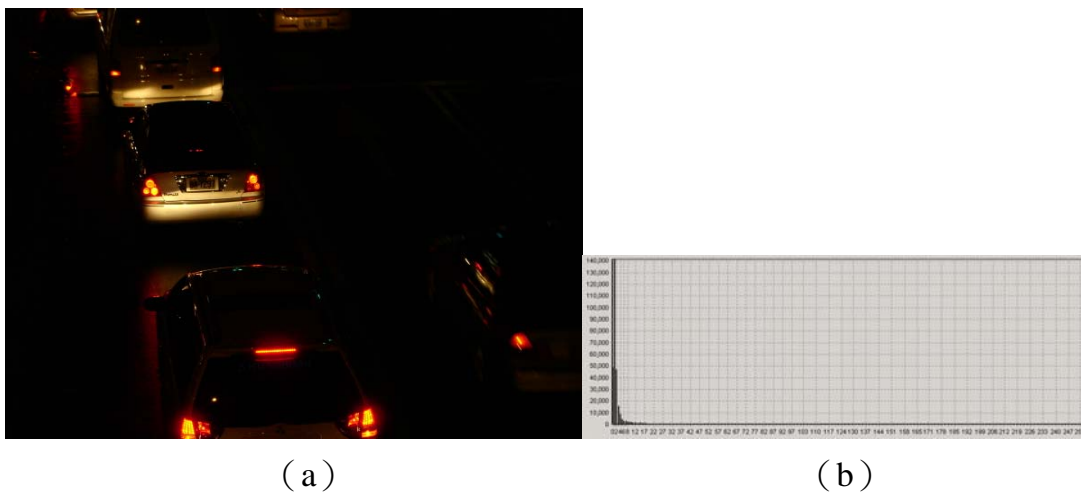


圖 3.14 亮度分佈均勻經 HE 及 BBHE 處理之比較圖 (a) 原圖，(b) 原圖之直方圖，(c) HE 處理後，(d) HE 處理後之直方圖，(e) BBHE 處理後，(f) BBHE 處理後之直方圖

圖 3.14 (a) 原圖，圖 3.14 (b) 原圖之灰階直方圖，圖 3.14 (c) 原圖經過 HE 處理後之影像，圖 3.14 (d) 原圖經過 HE 處理後之灰階直方圖，圖 3.14 (e) 原圖經過 BBHE 處理後之影像，圖 3.14 (f) 原圖經過 BBHE 處理後之灰階直方圖。

由圖 3.14 比較得知，若在夜間影像中使用 HE 來提高亮度，經由實驗中發現，若影像的直方圖其分佈較為均勻時，其效果相當好。



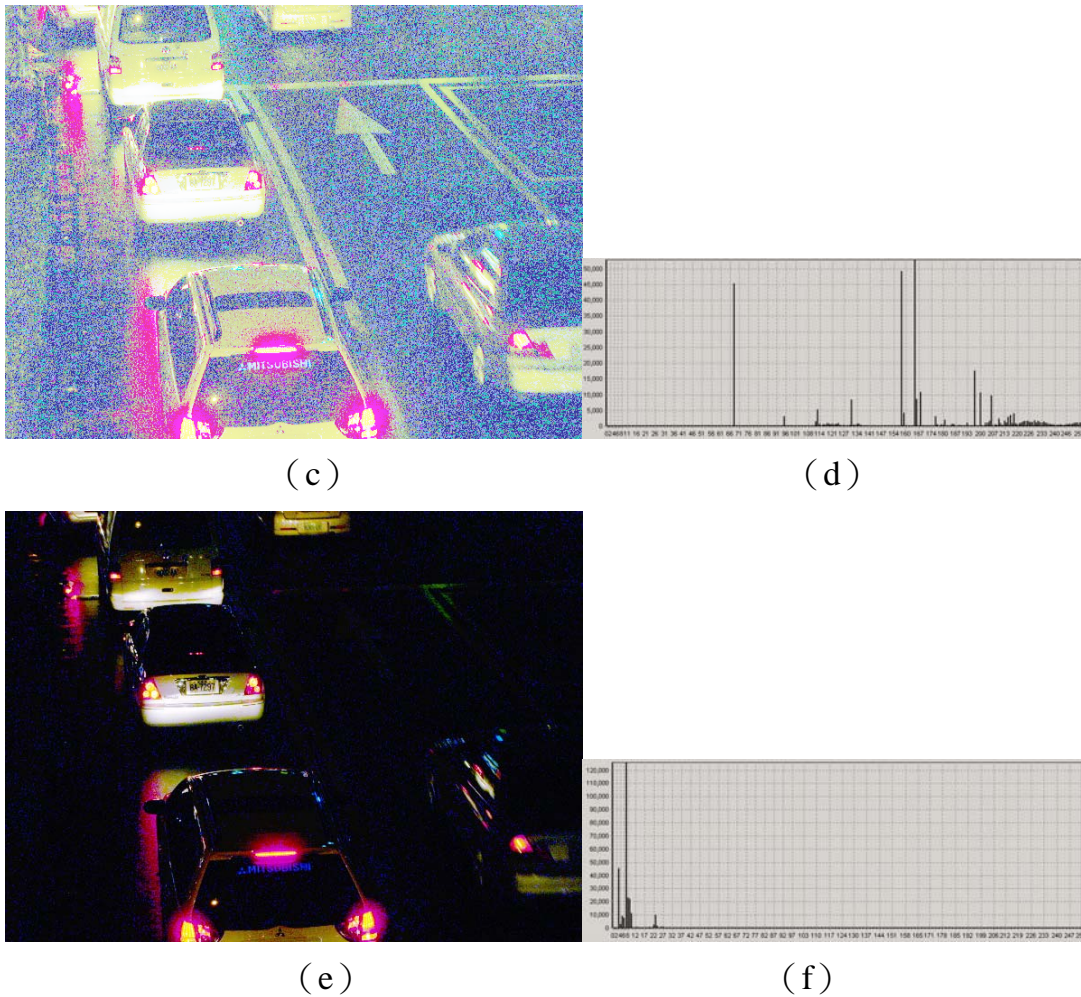


圖 3.15 亮度分佈不均勻經 HE 及 BBHE 處理之比較圖 (a) 原圖，(b) 原圖之直方圖，(c) HE 處理後，(d) HE 處理後之直方圖，(e) BBHE 處理後，(f) BBHE 處理後之直方圖

圖 3.15 (a) 原圖，圖 3.15 (b) 原圖之灰階直方圖，圖 3.15 (c) 原圖經過 HE 處理後之影像，圖 3.15 (d) 原圖經過 HE 處理後之灰階直方圖，圖 3.15 (e) 原圖經過 BBHE 處理後之影像，圖 3.15 (f) 原圖經過 BBHE 處理後之灰階直方圖。

由圖 3.15 比較得知，若影像的直方圖分佈比較集中在較低的像素值時（如多分佈於 10 pixel 以下時），此時 HE 提高亮度的效果相當不好，因在車輛部分有可能被後照光源照射，使得車牌部分比整張影像明亮。在進行均化

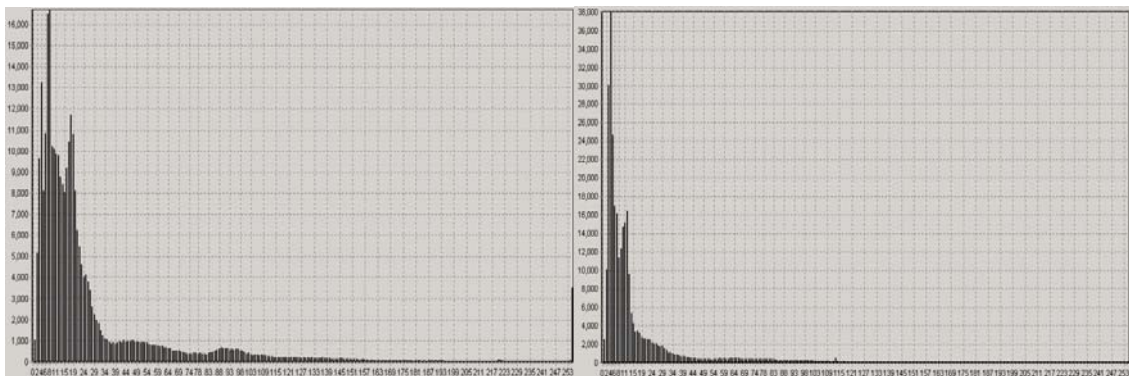
時，會將原直方圖最高像素拉伸至 250 以上，此時整張影像在均化（提高亮度）後會變成亮度過飽和，而損失車牌特徵。而在夜間影像中，大都為較暗且其直方圖的像素值多集中在較低的像素值，所以 HE 的提高亮度方法不符我們所需。

當然除了我們所提的 BBHE 外，另外有很多文獻也是架構在 HE 方法之上，如 DSIHE[33]就是四分亮度均化法，其作法即是二分 HE 後再各自二分亮度。但考量其效果與速度，因此使用 BBHE 的方法較為適合本夜間演算法所需。

在本文中，我們採用 BBHE 對 RGB 影像中各成分進行均化，不直接對灰階影像作均化，因實驗證明，單單只取灰階亮度值的均化結果，在某些情況下並不如預期。以灰階影像經過 BBHE 後與 RGB 影像各成分進行 BBHE 後再轉成灰階影像，其直方圖就可以看出差異，差別示意圖如圖 3.16 所示。



(a)



(b)

(c)

圖 3.16 灰階影像與 RGB 影像經過 BBHE 之差異圖 (a) 原始影像，(b) 灰階影像經 BBHE 後之直方圖，(c) 各自 BBHE 後轉灰階影像之直方圖

圖 3.16 (a) 為原始影像，圖 3.16 (b) 為 RGB 影像經過轉成灰階影像後，再經過 BBHE 後之直方圖，圖 3.16 (c) 為 RGB 影像各成分進行 BBHE 後，再轉成灰階影像之直方圖。

可發現直接灰階影像經過 BBHE 後，其像素分佈均提高不少亮度。而在夜間車牌有可能因後照光源之影響，其亮度值已經夠大，此時又提高亮度，此現象將會破壞我們的車牌特徵。

因此，我們以 RGB 各成分各自二分亮度均化後的影像，再將此三分量的影像轉至灰階影像，其均化效果符合我們對影像增強的要求。

3-5-4 夜間二值化閾值

經過差值運算後，必須二值化影像以利後面車牌候選區的框選，在經過實驗發現，我們得出夜間演算二值化臨界值為 2 倍的平均亮度值為最佳閾值。

3-5-5 夜間車牌偵測演算法

首先，我們考量夜間影像的車牌區域可能因後照光源的影響，其具有一定亮度值，甚至會有過亮的情況。若我們直接使用夜間車牌偵測演算法，則會使得車牌候選區域過亮，而喪失車牌特徵，造成車牌偵測失敗。

所以我們先利用日間演算法去偵測這些夠亮的車牌候選區後，將其車牌候選區域的位置保留，再利用夜間演算法去做原影像過暗的車牌候選區域。

其利用日間演算法進行候選區框選的影像如圖 3.17 所示：

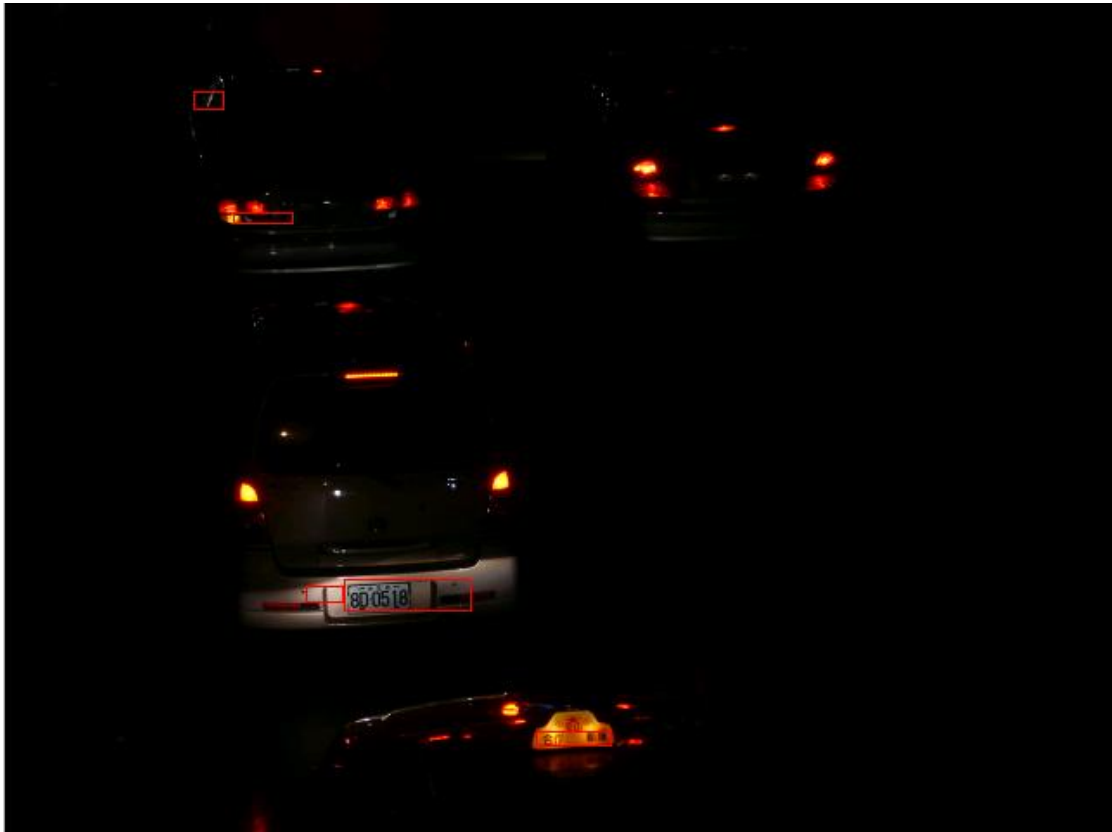


圖 3.17 夜間影像經日間演算法後的影像

在經過夜間演算法後，綜合兩部分的候選區位置即為夜間車牌偵測演算法的候選區，其夜間車牌偵測演算法結果如圖 3.18 所示：

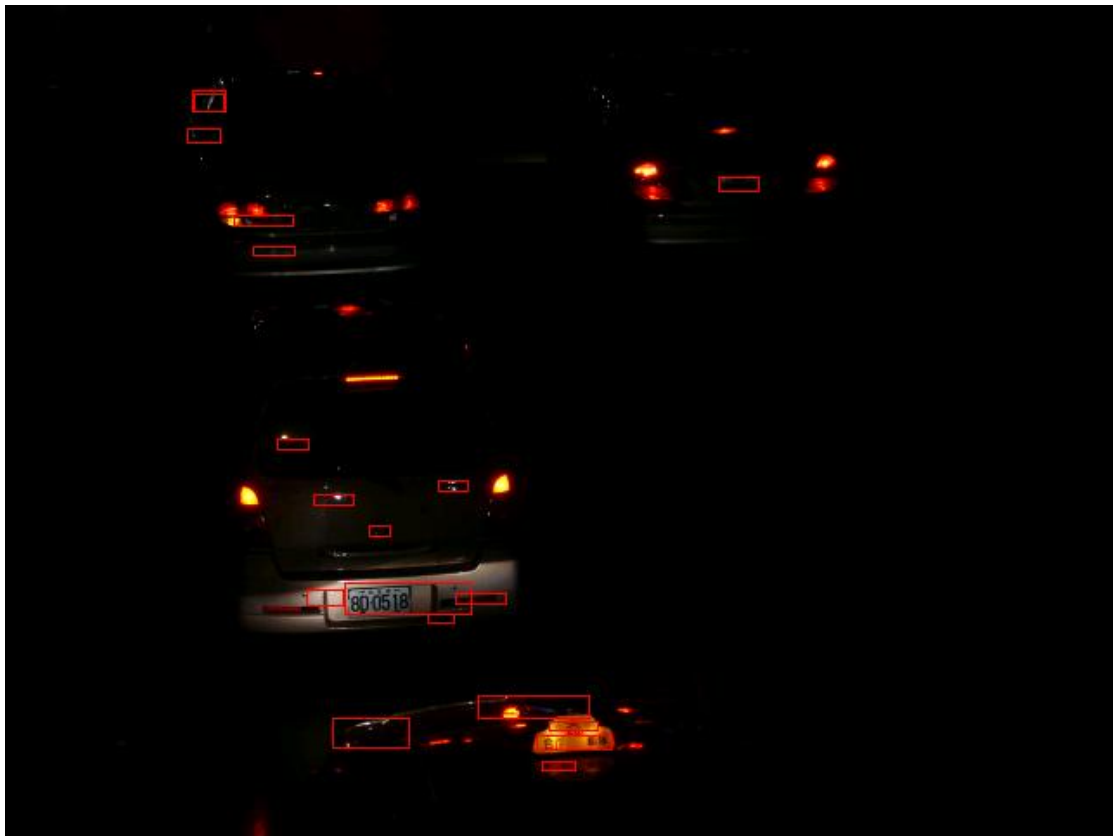


圖 3.18 綜合日、夜間演算法候選區後的影像

3-6 車牌定位

在獲得車牌候選區域位置後，利用三條字元交越量來判斷是否為車牌，但在進行三條字元交越量之前需先經過前處理，才能利用垂直投影法去判斷是否有交越量。

3-6-1 前處理架構

為了增強車牌亮度不均的情況下，使用 BBHE 演算法進行亮度均化，再經過改良式二值化後，其車牌字元二值影像能最佳化。其前處理流程圖如圖 3.19 所示，其詳細作法如後各小節所述：

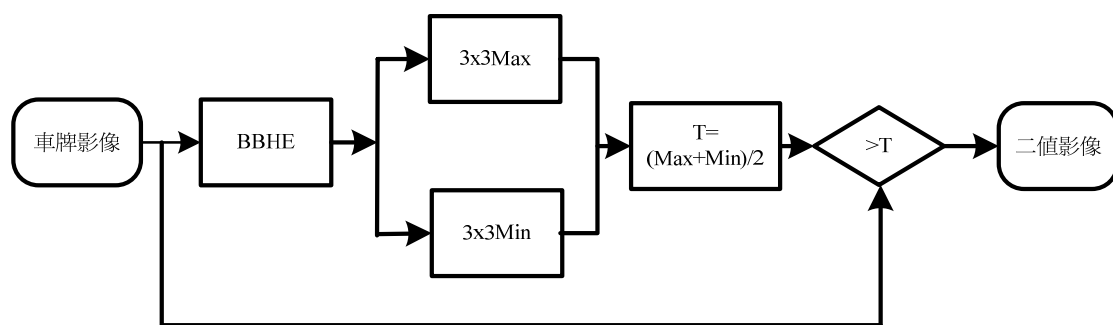


圖 3.19 前處理架構流程圖

3-6-2 BBHE

在此我們仍用 2-1-3 節的定義來做影像亮度提升，唯一要注意的是我們是以原圖的影像做 BBHE 的亮度提升，而不是用灰階影像做亮度提升。因灰階影像經過 BBHE 在某些情況下，並沒有提升的很好，所以必須以原 RGB 影像做 BBHE 的亮度提升。亮度提升完後再進行灰階轉換，以便進行下一節的二值化。

3-6-3 改良式二值化

此步驟是要將上一小節所取得的灰階圖像轉換成二值的影像，即設定一門檻值，將影像中所有灰階值大於門檻值的全部設為 1 (白色)，反之設為 0 (黑色)，它最主要的功能是區分出影像中的物件和背景，以獲取物件影像資訊。二值化的一般式表示如下：

$$f_{binary}(x, y) = \begin{cases} 0 & , f(x, y) \geq T \\ 1 & , f(x, y) < T \end{cases} \quad (3-12)$$

其中， $f(x, y)$ 為影像在原始影像的灰階值， $f_{binary}(x, y)$ 為影像經過二值化的灰階值， T 為門檻值。

影像二值化的方式很多，一般的門檻值多以灰階統計圖 (如 Otsu 演算法) 來做二值化，但它常做不出很好的效果。對本系統而言，車牌會因光線、環境以及車子本身狀況不同，使得影像灰階分布變異很大，因此一般的二值化方法並不能滿足我們系統的要求，所以我們必須要動態的選取二值化的門檻值。

由於簡單通用的二值化方法，雖然計算少，速度快，也因過於簡單，所以不適用於大部份情況，所以我們提出一個改良的方法，其概念如圖 3.20 所示：

57	54	45	57	54	45
55	33	43	55	43	43
56	30	46	56	40	42
57	21	45	57	54	45
55	32	43	55	43	43
56	50	32	56	50	52

圖 3.20 像素模擬示意圖

因人的眼睛對亮度並非決定於絕對亮度變化，而是決定於相對亮度變化，所以，雖然整體灰階度變化不大，但當我們看到圖 3.17 時，即仍可以看到左半部份的像素較為暗的部分（字元反白部分）。

所以利用此原理，我們的作法是對一個影像進行改良式二值化。其運算方法類似灰階形態學的膨脹演算法與侵蝕演算法。其中 I 為原始影像， B 為 3×3 遮罩，因車牌影像過小，所以遮罩大小不能過大，避免喪失其區域二值化的效果，所以定義遮罩為 3×3 大小為最適合。

$$\text{Max Value : } MAXV = \underset{B}{Max}(I) \quad (3-13)$$

$$\text{Min Value : } MINV = \underset{B}{Min}(I) \quad (3-14)$$

其最大值遮罩跟最小值遮罩分別對原影像的重點像素進行八連通的最大及最小的像素選取，選取完後將最大值及最小值相加，再取其平均值，即可得這個重點像素的門檻值，其中 T 為二值化閾值定義如下，











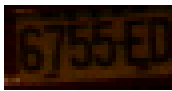




$$T = \frac{MAXV + MINV}{2} \quad (3-15)$$

因此，我們可以定義二值化閾值，其大於此門檻值即為 255（白），小於此門檻值即為 0（黑），以上圖舉例來說，對 33 這個像素值，我們取到的 $MAXV$ 為 57， $MINV$ 為 30，所以 T 為 43.5，再跟 33 這點像素比較下，這個重點像素小於我們的 T 值 43.5，所以 33 這個重點像素值，我們就改為 0，以此類推。

在國內外文獻中大都使用 Otsu 來進行二值化，這在光源穩定下，可有相當好的結果，但因我們實驗的環境為不穩定光源下，此時以 Otsu 對整張影像進行二值化，會因光源照射的位置，而二值化失敗。因此，必須加以改良運用才能得到較好的結果。所以本實驗嘗試以較小區塊來各自進行 Otsu 二值化，在考量速度與運算效果下，嘗試以四分 Otsu 二值化及六分 Otsu 二值化來進行實驗。其中四（六）分 Otsu 二值化為四（六）分區塊各自使用 Otsu 二值化為將車牌影像等比例分割為四（六）等分，再各自進行 Otsu 二值化。

以下表 3.2 為在光源不穩定下，使用 Otsu 二值化、四分 Otsu 二值化與六分 Otsu 二值化及改良式二值化的比較，可輕易發現改良式二值化後的結果遠優於 Otsu 二值化。

表 3.2 各種二值化後的比較圖

	原圖	Otsu	四分 Otsu	六分 Otsu	本方法
比較圖一					
比較圖二					
比較圖三					

我們所提的方法是對每個像素進行兩個 3×3 的遮罩，所以，若以平常 640×480 的大張影像來運算的話，本方法執行時間是 Otsu 演算法的四倍，但若以我們要二值化的車牌影像，其影像大小多為 40×20 為最大，因此本方法的速度反而為 Otsu 演算法的一半而已，且其執行效果遠比 Otsu 為佳。

3-6-4 三交越量

簡單來說，我們最直接的在影像中觀察車牌明顯的特徵就是具有密集性，在車牌上進行掃瞄線的黑白跨越計算，稱做零交越次數 (Zero Crossing) 或交越量。我們利用這個特性對來對車牌進一步的掃瞄處理，由於影像中只有車牌部分具有黑白點數的變化，利用三條掃瞄線去做交越次數統計可以提高定位正確率。

車牌有字元組成，字元筆畫粗係固定，其間格寬度也固定，對一條掃過車牌的水平線而言，可如圖 3.21 發現掃瞄線穿過物件，也就是交越次數有一定的數量。

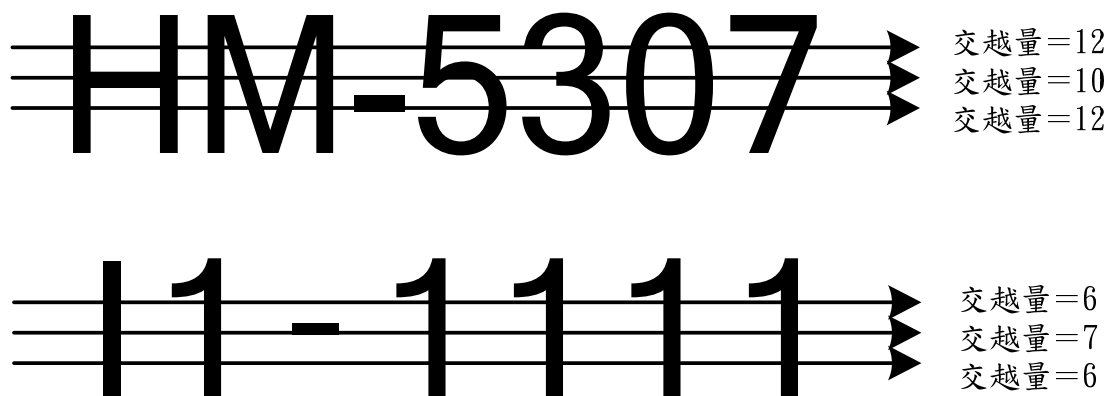


圖 3.21 車牌交越量示意圖

由此可知，若掃瞄線經過車牌後，交越次數最少為 6 次，最多 17 次（對字母而言，W、M 最多的交越次數為 4 次；數字部分 5 跟 0 則最多可能有 2 次，故最大的交越次數為 $4 \times 2 + 1 + 2 \times 4 = 17$ 次）。不過在實際情況中，車牌可能因陰影、污損、歪斜、車框的影響，導致車輛影像解析度的不同，使得實際交越次數增加。因此有必要將交越次數的上限加大，所以我們定義車牌交越次數最大值為 20，最小值為 4。若符合我們限定的交越次數，即為車牌候選區。