第 9 章　負片與黯淡車牌的辨識

9-1 如果你的車牌不是最耀眼的明星怎麼辦？

截至目前為止，我們已經將一個完整的車牌辨識主流程介紹完畢。基本假設是找白底黑字的目標，而且是從對比度最高的目標找起。那如果最明亮顯眼的那組目標其實不是車牌呢？像上圖一樣，我們一定會先將那幾個圖案的組合當車牌，但也預期結果一定是無意義的答案，那時我們就必須繼續搜尋下一組目標了！當車牌本身在全景中較黯淡，背景又很複雜時，可能要找很多次才能找到真的車牌。

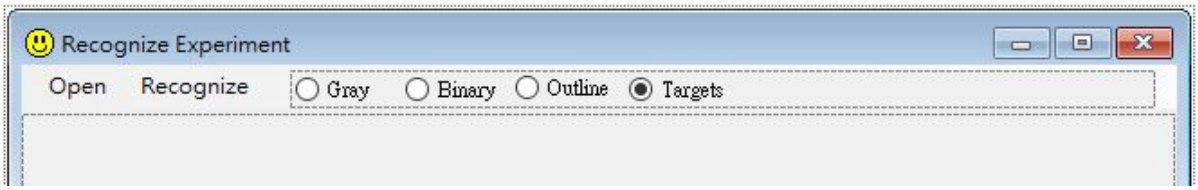
另一方面，車牌並不都是淺色背景深色字元的，如果是綠底白字的工程車，或紅底白字的遊覽車或重機等等，一開始就違反了我們的目標定義條件，當然是不可能辨識出車牌的！所以必須以負片方式再作一次辨識，才能套用前面幾章的技術內容找到白字車牌。以上兩點這些就是本章要介紹的主題了！

但是前面的那些章節滾動式的介紹方式，主要是讓讀者可以充分了解每一個辨識步驟的原理、事實現象與實作的方法。以程式寫作的角度來說是很不緊密簡潔，不太容易繼續衝建構更大規模程式的！所以我們會先作個整理。

9-2 程式架構整理

這一節文字內容會很長，但多數內容前面都介紹過，只是微調程式結構而已。請先建立如下的程式專案介面，主功能表只有 Open 與 Recognize 兩個按鍵，但為了顯示辨識過程作了一個 Panel 物件，裡面有：Gray, Binary, Outline 與 Target 等幾個單選按鈕，

工作區當然是一個 PictureBox1。



完整的全域變數宣告如下，之前專案沒有的是 skp 這個陣列，用於標記已經處理(辨識)過的目標，這樣在全圖中找最明亮目標時才不會一直重複找到同一組：

```
Dim B(,) As Byte '灰階陣列
Dim Z(,) As Byte '全圖二值化陣列
Dim Q(,) As Byte '輪廓線陣列
Dim minHeight As Integer = 20, maxHeight As Integer = 80 '有效目標高度範圍
Dim minwidth As Integer = 2, maxWidth As Integer = 80 '有效目標寬度範圍
Dim Tgmax As Integer = 20 '進入決選範圍的最明顯目標上限
Dim C As ArrayList, CA As ArrayList '目標物件集合
Dim Pw As Integer = 25, Ph As Integer = 50 '字模的寬與高
Dim P(1, 35, Pw - 1, Ph - 1) As Byte '六七碼車牌所有英數字二值化陣列
Dim P69(1, Pw - 1, Ph - 1) As Byte '變形的 6 與 9
Dim MC() As Array '正規化完成後的字元二值化陣列
Dim skp() As Boolean '已檢視目標註記
Dim Inc As Double '車牌傾斜角度(>0 為順時針傾斜)
'字元對照表
'0-9→0-9
'10→A，11→B，12→C，13→D，14→E，15→F，16→G，17→H，18→I，19→J
'20→K，21→L，22→M，23→N，24→O，25→P，26→Q，27→R，28→S，29→T
'30→U，31→V，32→W，33→X，34→Y，35→Z
Dim Ch() As Char = {"0"c, "1"c, "2"c, "3"c, "4"c, "5"c, "6"c, "7"c, "8"c, "9"c, _
                    "A"c, "B"c, "C"c, "D"c, "E"c, "F"c, "G"c, "H"c, "I"c, "J"c, _
                    "K"c, "L"c, "M"c, "N"c, "O"c, "P"c, "Q"c, "R"c, "S"c, "T"c, _
                    "U"c, "V"c, "W"c, "X"c, "Y"c, "Z"c}
```

下面是目標(TgInfo)、字元(ChInfo)與車牌(LPInfo)的三個資料結構(Structure)，前兩者與之前專案大致一樣，車牌結構是用來記錄有關整張車牌辨識結果的資訊：

```
'目標物件結構
Public Structure TgInfo
    Dim P As ArrayList '目標點的集合
    Dim xmn As Short, xmx As Short, ymn As Short, ymx As Short '四面座標極值
    Dim cx As Integer, cy As Integer '目標中心點座標
    Dim width As Integer, height As Integer '寬與高
    Dim pm As Integer '目標與背景的對比強度
    Dim ID As Integer
End Structure
```

```
'字元結構
Public Structure ChInfo
    Dim Ch As Char '最符合字元
    Dim ft As Integer '符合度評分
End Structure
'車牌資料結構
Public Structure LPInfo
    Dim N As Integer '車牌字元目標個數
    Dim A As String '車牌號碼
    Dim Sc As Integer '符合度
    Dim kind As Integer '六或七碼字型
    Dim xmn As Integer, ymn As Integer, xmx As Integer, ymx As Integer '車牌四邊極值
    Dim width As Integer, height As Integer '車牌寬與高
    Dim cx As Integer, cy As Integer '車牌中心點座標
End Structure
```

主功能表中的 Open 及 Form_Load 事件時匯入字模資料的程式與前一章相同，可直接複製無須修改。Recognize 按鍵功能是之前由影像到車牌整個辨識程序的整合，程式碼如下：

```
'辨識整個車牌
Private Sub RecognizeToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) _
    Handles RecognizeToolStripMenuItem.Click
    Z = DoBinary(B) '二值化
    Q = Outline(Z) '建立輪廓點陣列
    CA = getTargets(Q) '建立所有目標物件集合
    ReDim skp(CA.Count - 1)
    Dim R As New LPInfo
    R = getLP(CA)
    Me.Text = R.A + "," + R.Sc.ToString + "," + R.cx.ToString + "," + R.cy.ToString
End Sub
```

幾個影像前處理程序，到 getTargets 為止都完全沒變，增加的是宣告 skp 陣列與目標數目同大，稍後被處理過的目標要在此陣列中註記為 True，以及將前幾章的辨識程序進一步封裝為一個 getLP 的副程式，程式碼如下：

```
'依據可能目標組合辨識車牌
Private Function getLP(ByVal A As ArrayList) As LPInfo
    Dim R As New LPInfo '建立車牌資訊物件
    C = AlignTgs(A) '找到最多七個的字元目標組合
    Dim n As Integer = C.Count '目標個數
    For i As Integer = 0 To n - 1
        skp(C(i).ID) = True '標示目標已處理
    Next
    If n < 4 Then Return R '目標數目不足以構成車牌
    R.N = n '車牌目標個數
```

```
'末字中心點與首字中心點的偏移量，斜率計算參數
Dim dx As Integer = C(n - 1).cx - C(0).cx
Dim dy As Integer = C(n - 1).cy - C(0).cy
Inc = Math.Atan2(dy, dx) '字元排列傾角
'旋轉所有目標
Dim T(n - 1) As TgInfo, M(n - 1) As Array, w(n - 1) As Integer, h(n - 1) As Integer
R.xmn = nx : R.xmx = 0 : R.ymn = ny : R.ymx = 0 '車牌四面極值
For k As Integer = 0 To n - 1
    Dim G As TgInfo = C(k)
    M(k) = Tg2Bin(G) '建立單一目標的二值化矩陣
    M(k) = RotateTg(M(k), G, Inc) '旋轉目標，G 為 ByRef
    T(k) = G '儲存旋轉後的目標物件
    w(k) = G.width '寬度陣列
    h(k) = G.height '高度陣列
    If G.xmn < R.xmn Then R.xmn = G.xmn
    If G.xmx > R.xmx Then R.xmx = G.xmx
    If G.ymn < R.ymn Then R.ymn = G.ymn
    If G.ymx > R.ymx Then R.ymx = G.ymx
Next
R.width = R.xmx - R.xmn + 1 : R.height = R.ymx - R.ymn + 1 '車牌寬高
R.cx = (R.xmn + R.xmx) / 2 : R.cy = (R.ymn + R.ymx) / 2 '車牌中心點
Array.Sort(w) '寬度排序小到大
Array.Sort(h) '高度排序小到大
Dim mw As Integer = w(n - 2) '取第二寬的目標為標準，避開意外沾連的極端目標
Dim mh As Integer = h(n - 2) '取第二高的目標為標準，避開意外沾連的極端目標
'目標正規化→寬高符合字模
ReDim MC(n - 1) '正規化後之字元二值化陣列
For k As Integer = 0 To n - 1
    MC(k) = NmBin(T(k), M(k), mw, mh) '個別字元正規化矩陣
Next
'計算最大字元間距與位置
Dim dmx As Integer = 0, mi As Integer = 0
For i As Integer = 0 To n - 2
    Dim d As Integer = (C(i + 1).cx - C(i).cx) ^ 2 + (C(i + 1).cy - C(i).cy) ^ 2
    If d > dmx Then dmx = d : mi = i
Next
For i As Integer = 0 To MC.Count - 1
    Dim D As ChInfo = BestC(MC(i))
    R.A += D.Ch '車牌字串累加
    R.Sc += D.ft '字源符合度累加
    If i = mi Then R.A += "-"c
Next
R.Sc /= MC.Count
R = ChkLP(R) '檢查是否為合格車牌
R = ChkED(R) '修正英數字
Return R '回傳車牌資料
```

End Function

此副程式的功能就是饋入所有的目標物件集合，傳回一個車牌資訊物件。首先是找出可能的車牌目標群組，過程中會忽略已處理過的目標，所以可以重複使用，譬如第一組車牌答案無意義就可以繼續用此副程式搜尋第二組可能的車牌，但是 skp 的資訊必須持續。

找到可能的目標群組之後就是在此進行：目標二值化(Tg2Bin)→旋轉(RotateTg)→正規化(NmBin)→辨識字元(BestC)→組織車牌資訊→檢查車牌合理性(ChkLP)→依照車牌格式強制修正英數字(ChkED)，最後回傳車牌資訊物件。其中 Tg2Bin，RotateTg 與 NmBin 副程式的內容都沒改變與前面章節一樣，但是 BestC 則有比較大的改變，程式碼如下：

```
'最佳字元
Private Function BestC(ByVal A(,) As Byte) As ChInfo
    Dim C As New ChInfo '辨識字元資料結構
    For m As Integer = 0 To 1
        For k As Integer = 0 To 35
            For x As Integer = -1 To 1
                For y As Integer = -1 To 1
                    Dim n0 As Integer = 0, nf As Integer = 0 '字模黑點數，符合點數
                    For i As Integer = 0 To Pw - 1
                        Dim ix As Integer = i + x
                        If ix < 0 Or ix > Pw - 1 Then Continue For
                        For j As Integer = 0 To Ph - 1
                            Dim jy As Integer = j + y
                            If jy < 0 Or jy > Ph - 1 Then Continue For
                            If P(m, k, i, j) = 0 Then
                                If A(ix, jy) = 1 Then nf -= 1 '目標與字模不符合點數-1
                            Else
                                n0 += 1 '字模黑點數累計
                                If A(ix, jy) = 1 Then nf += 1 '目標與字模符合點數+1
                            End If
                        Next
                    Next
                    Dim v As Integer = nf * 1000 / n0 '符合點數百分比
                    If v > C.ft Then '符合度最高字元
                        C.ft = v : C.Ch = Ch(k)   '符合度，字元
                    End If
                Next
            Next
        Next
    Next
    For k As Integer = 0 To 1
        For x As Integer = -1 To 1
```

```
                For y As Integer = -1 To 1
                    Dim n0 As Integer = 0, nf As Integer = 0
                    For i As Integer = 0 To Pw - 1
                        Dim ix As Integer = i + x
                        If ix < 0 Or ix > Pw - 1 Then Continue For
                        For j As Integer = 0 To Ph - 1
                            Dim jy As Integer = j + y
                            If jy < 0 Or jy > Ph - 1 Then Continue For
                            If P69(k, i, j) = 0 Then
                                If A(ix, jy) = 1 Then nf -= 1 '目標與字模符合點數
                            Else
                                n0 += 1 '字模黑點數累計
                                If A(ix, jy) = 1 Then nf += 1 '目標與字模符合點數
                            End If
                        Next
                    Next
                    Dim v As Integer = nf * 1000 / n0 '符合點數百分比
                    If v > C.ft Then '符合度最高字元
                        C.ft = v '符合度
                        If k = 0 Then C.Ch = "6"c Else C.Ch = "9"c
                    End If
                Next
            Next
        Next
        Return C
    End Function
```

跟前章介紹的 BestC 差別是我們加了兩層迴圈 x 與 y，讓目標上下左右各移動一個畫素的距離作比對，找出最大的符合度。這在實務上是很有用處的，因為二值化過程讓目標因為邊界的雜訊偏移一兩個畫素是常有的事，就跟瞄準插入卡榫一樣，有時偏移一個畫素符合度就大增了！尤其是原始目標畫素尺度較小時幫助更大！

檢查車牌資訊合理性的副程式 ChkLP 程式碼如下，前一章也介紹過，但此地的參數與回傳值都改成車牌資訊物件了！車牌物件的 A 屬性是車牌號碼，Sc 屬性是符合度，600 以下就是不及格，凡是不及格的就會回傳空物件，主程式就知道沒找到車牌了！

```
'檢驗車牌是否正確的程式
Private Function ChkLP(ByVal R As LPInfo) As LPInfo
    If R.Sc < 600 Then Return New LPInfo '符合度低於及格分數
    If R.A.Length < 5 Then Return New LPInfo '包含分隔線在內字數小於 5
    Dim m As Integer = R.A.IndexOf("-") '格線位置
    If m = 1 Then Return New LPInfo '沒有 1-x 的字數區段格式
    Return R '合格車牌
End Function
```

依據格式修正英數字的程式 ChkED 程式碼如下，也是輸出入都變成車牌資訊物

件，內容功能則與前一章相同，附屬的 E2D 與 D2E 副程式一起列出。

```
'嘗試依據英數字規範修改車牌答案
Private Function ChkED(ByVal R As LPInfo) As LPInfo
    If IsNothing(R.A) Then Return R '無字串
    Dim C() As Char = R.A.ToCharArray '字串轉成字元陣列
    Dim n1 As Integer = R.A.IndexOf("-") '第一區段長度
    Dim n2 As Integer = C.Length - n1 - 1 '第二區段長度
    Dim d1 As Integer = 0, d2 As Integer = 0 '數字區的起終點
    If n1 = n2 Then Return R '無法判定純數字區段(2-2 或 3-3)
    If n1 > n2 Then '第一區段較長
        d1 = 0 : d2 = n1 - 1
    Else '第二區段較長
        d1 = n1 + 1 : d2 = C.Length - 1
    End If
    '嘗試將純數字區段的英文字改成數字
    For i As Integer = d1 To d2
        C(i) = E2D(C(i))
    Next
    '如果是七碼車牌，強制將前三碼中的數字改成英文
    If n1 = 3 And n2 = 4 Then
        For i As Integer = 0 To 2
            C(i) = D2E(C(i))
        Next
    End If
    R.A = "" '重組字串
    For i As Integer = 0 To C.Length - 1
        R.A += C(i)
    Next
    Return R '回傳字串
End Function
'嘗試將英文字母變成相似的數字
Private Function E2D(ByVal C As Char) As Char
    If C = "B"c Then C = "8"c
    If C = "D"c Then C = "0"c
    If C = "O"c Then C = "0"c
    Return C
End Function
'嘗試將英文字母變成相似的數字
Private Function D2E(ByVal C As Char) As Char
    If C = "8"c Then C = "B"c
    If C = "0"c Then C = "D"c
    Return C
End Function
```

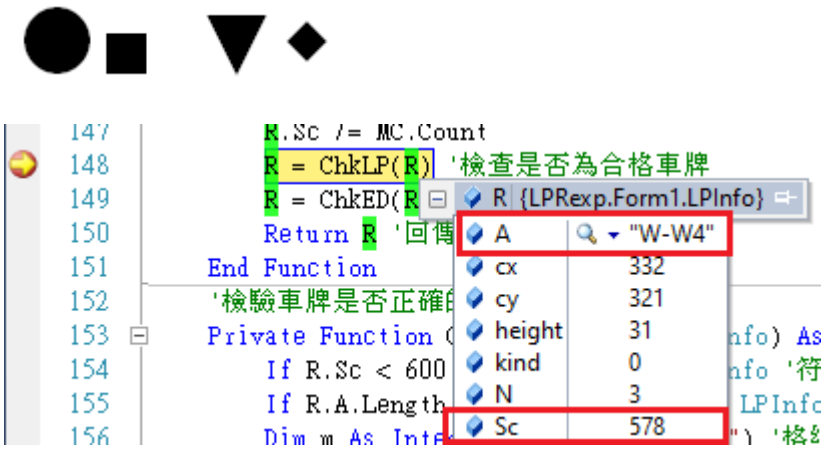
接下來我們是 Gray, Binary, Outline 與 Target 幾個圖形顯示按鈕的程式如下：

```
'灰階
Private Sub RadioButton1_CheckedChanged(ByVal sender As Object, ByVal e As EventArgs) _
    Handles RadioButton1.CheckedChanged
    If RadioButton1.Checked Then
        If IsNothing(B) Then Exit Sub
        Dim bmp As Bitmap = GrayImg(B)
        PictureBox1.Image = bmp
    End If
End Sub
'二值化圖
Private Sub RadioButton2_CheckedChanged(ByVal sender As Object, ByVal e As EventArgs) _
    Handles RadioButton2.CheckedChanged
    If RadioButton2.Checked Then
        If IsNothing(Z) Then Exit Sub
        Dim bmp As Bitmap = BWImg(Z)
        PictureBox1.Image = bmp
    End If
End Sub
'輪廓線
Private Sub RadioButton3_CheckedChanged(ByVal sender As Object, ByVal e As EventArgs) _
    Handles RadioButton3.CheckedChanged
    If RadioButton3.Checked Then
        If IsNothing(Q) Then Exit Sub
        Dim bmp As Bitmap = BWImg(Q)
        PictureBox1.Image = bmp
    End If
End Sub
'所有合格目標
Private Sub RadioButton4_CheckedChanged(ByVal sender As Object, ByVal e As EventArgs) _
    Handles RadioButton4.CheckedChanged
    If RadioButton4.Checked Then
        If IsNothing(CA) Then Exit Sub
        Dim bmp As New Bitmap(nx, ny)
        For k As Integer = 0 To CA.Count - 1
            Dim T As TgInfo = CA(k)
            For m As Integer = 0 To T.P.Count - 1
                Dim p As Point = T.P(m)
                bmp.SetPixel(p.X, p.Y, Color.Black)
            Next
        Next
        PictureBox1.Image = bmp
    End If
End Sub
```

這些介面程式在我們研究辨識過程時非常重要！從一張影像到辨識出答案的過程非常複雜，如果我們只知道「**無法辨識**」的結果，卻說不出是在哪一個步驟凸槌？就會顯得很不專業了！真正影像辨識的「**原廠**」一定可以替客戶解釋這些錯誤原因，也可以據此思考讓演算法調整精進。但機器學習作出的軟體就很難這麼作了！因為連「設計者」都不知道電腦為何要這麼辨識？那是電腦自行統計出來的演算法。

9-3 如何完成多組車牌辨識的嘗試？

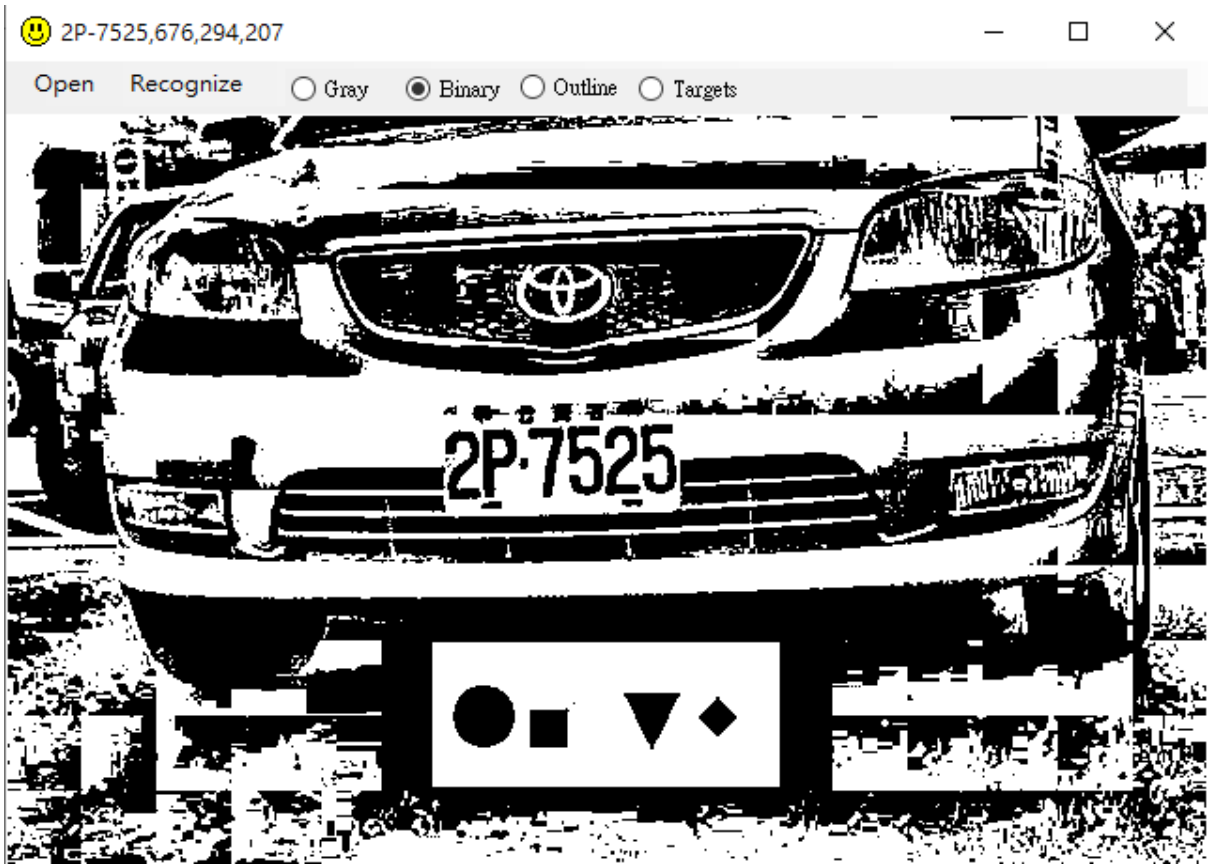
做好以上的整理工作之後，我們先用本章首頁的加工誤導的影像以程式現狀辨識，結果一定是沒有車牌的！因為它一定是辨識下面這個東西，如果你用中斷點看，答案一定很奇怪，符合度分數也很低！



這樣的不合理答案當然會被 ChkLP 所封殺，我們只要讓它用既有資料排除這幾個目標再找一次應該就可以找到正確的車牌了！請將 Recognize 主程式改成這樣：

```
Private Sub RecognizeToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) _
    Handles RecognizeToolStripMenuItem.Click
    Z = DoBinary(B) '二值化
    Q = Outline(Z) '建立輪廓點陣列
    CA = getTargets(Q) '建立所有目標物件集合
    ReDim skp(CA.Count - 1)
    Dim R As New LPInfo
    '目標搜尋辨識
    For k As Integer = 0 To 2
        R = getLP(CA)
        If R.Sc > 0 Then Exit For '有合理答案立即跳出迴圈
    Next
    Me.Text = R.A + "," + R.Sc.ToString + "," + R.cx.ToString + "," + R.cy.ToString
End Sub
```

讓 getLP 副程式的執行變成迴圈，如果某回合出現合理的答案就跳出迴圈結束辨識，以此例來說當然第二次就會辨識到了！並在標題列顯示結果，畫面如下圖。前面辨識過的目標在此都會被 skp 陣列標示，後續辨識時就不會重複了！



或許有人會說為何不用 Do While 迴圈？就是不限次數的找到有車牌為止？這其實是不好的設計！因為消極面來說車牌通常不會是很不清楚的目標，頭兩三次找不到通常就表示沒有車牌，不必找了！如果硬要讓所有目標都試過，碰上目標太多的無車牌影像就會很浪費時間了！根本沒車牌你還找得特別久？客戶就會說你的軟體很笨了！

9-4 負片辨識

根據現有的程式，面對如上的工程車，即使你讓辨識迴圈跑到天荒地老也是不會有答案的！因為車牌字是白色的，在建立目標的階段就不會被收集到，後續的目標組合當然不會有那幾個字！所以必須將灰階影像變成負片後再跑所有的辨識流程才行！

雖然正片變成負片的計算並不耗時，但因為九成以上的車牌都是白底黑字，我們也不會預先知道影像上的車牌是黑字或白字？所以必須先嘗試辨識正片，沒有發現車牌時再嘗試辨識負片，這就會讓「負片辨識」感覺上比較慢！如果你直接跳過正常的正片辨識，其實辨識速度是差不多的。正負片都辨識的完整程式碼如下：

```
Private Sub RecognizeToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) _
    Handles RecognizeToolStripMenuItem.Click
    Z = DoBinary(B) '二值化
    Q = Outline(Z) '建立輪廓點陣列
    CA = getTargets(Q) '建立所有目標物件集合
    ReDim skp(CA.Count - 1)
    Dim R As New LPInfo
    '正片目標搜尋辨識
    For k As Integer = 0 To 2
        R = getLP(CA)
        If R.Sc > 0 Then Exit For '有合理答案立即跳出迴圈
    Next
    '負片辨識
    If R.Sc = 0 Then
        B = Negative(B)
        Z = DoBinary(B) '二值化
```

```
            Q = Outline(Z) '建立輪廓點陣列
            CA = getTargets(Q) '建立所有目標物件集合
            ReDim skp(CA.Count - 1)
            For k As Integer = 0 To 1
                R = getLP(CA)
                If R.Sc > 0 Then Exit For
            Next
        End If
        Me.Text = R.A + "," + R.Sc.ToString + "," + R.cx.ToString + "," + R.cy.ToString
    End Sub
    '灰階正片轉成負片
    Private Function Negative(ByVal b(,) As Byte) As Byte(,)
        For i As Integer = 0 To nx - 1
            For j As Integer = 0 To ny - 1
                b(i, j) = 255 - b(i, j)
            Next
        Next
        Return b
    End Function
```

完整專案

```
Public Class Form1
    Dim B(,) As Byte '灰階陣列
    Dim Z(,) As Byte '全圖二值化陣列
    Dim Q(,) As Byte '輪廓線陣列
    Dim minHeight As Integer = 20, maxHeight As Integer = 80 '有效目標高度範圍
    Dim minwidth As Integer = 2, maxWidth As Integer = 80 '有效目標寬度範圍
    Dim Tgmax As Integer = 20 '進入決選範圍的最明顯目標上限
    Dim C As ArrayList, CA As ArrayList '目標物件集合
    Dim Pw As Integer = 25, Ph As Integer = 50 '字模的寬與高
    Dim P(1, 35, Pw - 1, Ph - 1) As Byte '六七碼車牌所有英數字二值化陣列
    Dim P69(1, Pw - 1, Ph - 1) As Byte '變形的 6 與 9
    Dim MC() As Array '正規化完成後的字元二值化陣列
    Dim skp() As Boolean '已檢視目標註記
    Dim Inc As Double '車牌傾斜角度(>0 為順時針傾斜)
    '字元對照表
    '0-9→0-9
    '10→A，11→B，12→C，13→D，14→E，15→F，16→G，17→H，18→I，19→J
    '20→K，21→L，22→M，23→N，24→O，25→P，26→Q，27→R，28→S，29→T
    '30→U，31→V，32→W，33→X，34→Y，35→Z
    Dim Ch() As Char = {"0"c, "1"c, "2"c, "3"c, "4"c, "5"c, "6"c, "7"c, "8"c, "9"c,
                        "A"c, "B"c, "C"c, "D"c, "E"c, "F"c, "G"c, "H"c, "I"c, "J"c,
                        "K"c, "L"c, "M"c, "N"c, "O"c, "P"c, "Q"c, "R"c, "S"c, "T"c,
                        "U"c, "V"c, "W"c, "X"c, "Y"c, "Z"c}
    '目標物件結構
    Public Structure TgInfo
        Dim np As Integer '目標點數
        Dim P As ArrayList '目標點的集合
        Dim xmn As Short, xmx As Short, ymn As Short, ymx As Short '四面座標極值
        Dim cx As Integer, cy As Integer '目標中心點座標
        Dim width As Integer, height As Integer '寬與高
        Dim pm As Integer '目標與背景的對比強度
        Dim ID As Integer '目標依據對比度的排序
    End Structure
    '字元結構
    Public Structure ChInfo
        Dim Ch As Char '最符合字元
        Dim ft As Integer '符合度評分
        Dim kind As Integer '六或七碼字元，0→六碼，1→七碼
    End Structure
    '車牌資料結構
    Public Structure LPInfo
        Dim A As String '車牌號碼
        Dim Sc As Integer '符合度
        Dim N As Integer '車牌字元目標個數
        Dim kind As Integer '六或七碼字型
        Dim xmn As Integer, ymn As Integer, xmx As Integer, ymx As Integer '車牌四邊極值
        Dim width As Integer, height As Integer '車牌寬與高
        Dim cx As Integer, cy As Integer '車牌中心點座標
    End Structure
    '啟動程式載入字模
    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Load
        FontLoad()
    End Sub
    Private Sub FontLoad()
        Dim q() As Byte = My.Resources.font '使用字模資源檔
```

```
        Dim n As Integer = 0
        '匯入六與七碼字模
        For m As Integer = 0 To 1
            For k As Integer = 0 To 35
                For j As Integer = 0 To Ph - 1
                    For i As Integer = 0 To Pw - 1
                        P(m, k, i, j) = q(n) : n += 1
                    Next
                Next
            Next
        Next
        '匯入六碼變形 69 字模
        For k As Integer = 0 To 1
            For j As Integer = 0 To Ph - 1
                For i As Integer = 0 To Pw - 1
                    P69(k, i, j) = q(n) : n += 1
                Next
            Next
        Next
    End Sub
    '開啟檔案
    Private Sub OpenToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs) _
        Handles OpenToolStripMenuItem.Click
        If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
            Dim bmp As New Bitmap(OpenFileDialog1.FileName)
            Bmp2RGB(bmp) '擷取影像資訊
            B = Gv.Clone '以綠光為灰階
            PictureBox1.Image = bmp '顯示
        End If
    End Sub
    '辨識整個車牌
    Private Sub RecognizeToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) _
        Handles RecognizeToolStripMenuItem.Click
        Z = DoBinary(B) '二值化
        Q = Outline(Z) '建立輪廓點陣列
        CA = getTargets(Q) '建立所有目標物件集合
        ReDim skp(CA.Count - 1)
        Dim R As New LPInfo
        '目標搜尋辨識
        For k As Integer = 0 To 2
            R = getLP(CA)
            If R.Sc > 0 Then Exit For '有合理答案立即跳出迴圈
        Next
        '負片辨識
        If R.Sc = 0 Then
            B = Negative(B)
            Z = DoBinary(B) '二值化
            Q = Outline(Z) '建立輪廓點陣列
            CA = getTargets(Q) '建立所有目標物件集合
            ReDim skp(CA.Count - 1)
            For k As Integer = 0 To 1
                R = getLP(CA)
                If R.Sc > 0 Then Exit For
            Next
        End If
        Me.Text = R.A + "," + R.Sc.ToString + "," + R.cx.ToString + "," + R.cy.ToString
    End Sub
    '負片
```

```
Private Function Negative(ByVal b(,) As Byte) As Byte(,)
    For i As Integer = 0 To nx - 1
        For j As Integer = 0 To ny - 1
            b(i, j) = 255 - b(i, j)
        Next
    Next
    Return b
End Function
'依據可能目標組合辨識車牌
Private Function getLP(ByVal A As ArrayList) As LPInfo
    Dim R As New LPInfo '建立車牌資訊物件
    C = AlignTgs(A) '找到最多七個的字元目標組合
    Dim n As Integer = C.Count '目標個數
    For i As Integer = 0 To n - 1
        skp(C(i).ID) = True '標示目標已處理
    Next
    If n < 4 Then Return R '目標數目不足以構成車牌
    R.N = n '車牌目標個數
    '末字中心點與首字中心點的偏移量，斜率計算參數
    Dim dx As Integer = C(n - 1).cx - C(0).cx
    Dim dy As Integer = C(n - 1).cy - C(0).cy
    Inc = Math.Atan2(dy, dx) '字元排列傾角
    '旋轉所有目標
    Dim T(n - 1) As TgInfo, M(n - 1) As Array, w(n - 1) As Integer, h(n - 1) As Integer
    R.xmn = nx : R.xmx = 0 : R.ymn = ny : R.ymx = 0 '車牌四面極值
    For k As Integer = 0 To n - 1
        Dim G As TgInfo = C(k)
        M(k) = Tg2Bin(G) '建立單一目標的二值化矩陣
        M(k) = RotateTg(M(k), G, Inc) '旋轉目標，G 為 ByRef
        T(k) = G '儲存旋轉後的目標物件
        w(k) = G.width '寬度陣列
        h(k) = G.height '高度陣列
        If G.xmn < R.xmn Then R.xmn = G.xmn
        If G.xmx > R.xmx Then R.xmx = G.xmx
        If G.ymn < R.ymn Then R.ymn = G.ymn
        If G.ymx > R.ymx Then R.ymx = G.ymx
    Next
    R.width = R.xmx - R.xmn + 1 : R.height = R.ymx - R.ymn + 1 '車牌寬高
    R.cx = (R.xmn + R.xmx) / 2 : R.cy = (R.ymn + R.ymx) / 2 '車牌中心點
    Array.Sort(w) '寬度排序小到大
    Array.Sort(h) '高度排序小到大
    Dim mw As Integer = w(n - 2) '取第二寬的目標為標準，避開意外沾連的極端目標
    Dim mh As Integer = h(n - 2) '取第二高的目標為標準，避開意外沾連的極端目標
    '目標正規化→寬高符合字模
    ReDim MC(n - 1) '正規化後之字元二值化陣列
    For k As Integer = 0 To n - 1
        MC(k) = NmBin(T(k), M(k), mw, mh) '個別字元正規化矩陣
    Next
    '計算最大字元間距與位置
    Dim dmx As Integer = 0, mi As Integer = 0
    For i As Integer = 0 To n - 2
        Dim d As Integer = (C(i + 1).cx - C(i).cx) ^ 2 + (C(i + 1).cy - C(i).cy) ^ 2
        If d > dmx Then dmx = d : mi = i
    Next
    For i As Integer = 0 To MC.Count - 1
        Dim D As ChInfo = BestC(MC(i))
        R.A += D.Ch '車牌字串累加
        R.Sc += D.ft '字源符合度累加
```

```
            If i = mi Then R.A += "-"c
        Next
        R.Sc /= MC.Count
        R = ChkLP(R) '檢查是否為合格車牌
        R = ChkED(R) '修正英數字
        Return R '回傳車牌資料
    End Function
    '檢驗車牌是否正確的程式
    Private Function ChkLP(ByVal R As LPInfo) As LPInfo
        If R.Sc < 600 Then Return New LPInfo '符合度低於及格分數
        If R.A.Length < 5 Then Return New LPInfo '包含分隔線在內字數小於 5
        Dim m As Integer = R.A.IndexOf("-") '格線位置
        If m = 1 Then Return New LPInfo '沒有 1-x 的字數區段格式
        Return R '合格車牌
    End Function
    '找車牌字元目標群組
    Private Function AlignTgs(ByVal C As ArrayList) As ArrayList
        Dim R As New ArrayList, pmx As Integer = 0 '最佳目標組合與最佳度比度
        For i As Integer = 0 To C.Count - 1
            If skp(i) Then Continue For '已處理目標
            Dim T As TgInfo = C(i) '核心目標
            Dim D As New ArrayList, Dm As Integer = 0 '此輪搜尋的目標集合
            D.Add(T) : Dm = T.pm '加入搜尋起點目標
            Dim x1 As Integer = T.cx - T.height * 2.5, x2 As Integer = T.cx + T.height * 2.5 '搜尋 X 範圍
            Dim y1 As Integer = T.cy - T.height * 1.5, y2 As Integer = T.cy + T.height * 1.5 '搜尋 Y 範圍
            For j As Integer = 0 To C.Count - 1
                If i = j Then Continue For '與起點重複略過
                If skp(j) Then Continue For
                Dim G As TgInfo = C(j)
                If G.cx < x1 Then Continue For
                If G.cx > x2 Then Continue For
                If G.cy < y1 Then Continue For
                If G.cy > y2 Then Continue For
                If G.width > T.height Then Continue For '目標寬度太大略過
                If G.height > T.height * 1.5 Then Continue For '目標高度太大略過
                D.Add(G) : Dm += G.pm '合格目標加入集合
                If D.Count >= 7 Then Exit For '目標蒐集個數已滿跳離迴圈
            Next
            If Dm > pmx Then '對比度高於之前的目標集合
                pmx = Dm : R = D
            End If
        Next
        '目標群位置左右排序
        If R.Count > 1 Then
            Dim n As Integer = R.Count
            For i As Integer = 0 To n - 2
                For j As Integer = i + 1 To n - 1
                    Dim Ti As TgInfo = R(i), Tj As TgInfo = R(j)
                    If Ti.cx > Tj.cx Then
                        R(i) = Tj : R(j) = Ti
                    End If
                Next
            Next
        End If
        Return R
    End Function
    '二值化
    Private Function DoBinary(ByVal b(,) As Byte) As Byte(,)
```

```
        Dim Gdim As Integer = 40 '計算區域亮度區塊的寬與高
        Dim Th(,) As Integer = ThresholdBuild(b, Gdim) '建立二值化使用之門檻值陣列
        Dim Z(nx - 1, ny - 1) As Byte '建立二值化陣列
        For i As Integer = 1 To nx - 2
            Dim x As Integer = i \ Gdim 'x 座標換算
            For j As Integer = 1 To ny - 2
                Dim y As Integer = j \ Gdim 'y 座標換算
                If b(i, j) < Th(x, y) Then Z(i, j) = 1 '低於亮度門檻設為目標點
            Next
        Next
        Return Z
    End Function
    '門檻值陣列建立
    Private Function ThresholdBuild(ByVal b(,) As Byte, ByVal Gdim As Integer) As Integer(,)
        Dim kx As Integer = nx \ Gdim, ky As Integer = ny \ Gdim
        Dim T(kx, ky) As Integer
        '累計各區塊亮度值總和
        For i As Integer = 0 To nx - 1
            Dim x As Integer = i \ Gdim
            For j As Integer = 0 To ny - 1
                Dim y As Integer = j \ Gdim
                T(x, y) += b(i, j) '亮度值累加
            Next
        Next
        '區塊亮度平均值計算
        For i As Integer = 0 To kx - 1
            For j As Integer = 0 To ky - 1
                T(i, j) /= Gdim * Gdim
            Next
        Next
        Return T
    End Function
    '建立輪廓點陣列
    Private Function Outline(ByVal b(,) As Byte) As Byte(,)
        Dim Q(nx - 1, ny - 1) As Byte '輪廓點陣列
        For i As Integer = 1 To nx - 2
            For j As Integer = 1 + 1 To ny - 2
                If b(i, j) = 0 Then Continue For '非輪廓點忽略
                If b(i, j - 1) = 0 Then Q(i, j) = 1 : Continue For '確認為輪廓點
                If b(i - 1, j) = 0 Then Q(i, j) = 1 : Continue For '確認為輪廓點
                If b(i + 1, j) = 0 Then Q(i, j) = 1 : Continue For '確認為輪廓點
                If b(i, j + 1) = 0 Then Q(i, j) = 1 '確認為輪廓點
            Next
        Next
        Return Q
    End Function
    '以輪廓點建立目標陣列，排除負目標
    Function getTargets(ByVal q(,) As Byte) As ArrayList
        Dim A As New ArrayList '目標集合物件
        Dim b(,) As Byte = q.Clone '建立輪廓點陣列副本
        For i As Integer = 1 To nx - 2
            For j As Integer = 1 To ny - 2
                If b(i, j) = 0 Then Continue For
                Dim G As New TgInfo
                G.xmn = i : G.xmx = i : G.ymn = j : G.ymx = j : G.P = New ArrayList
                Dim nc As New ArrayList '每一輪搜尋的起點集合
                nc.Add(New Point(i, j)) '輸入之搜尋起點
                G.P.Add(New Point(i, j)) : b(i, j) = 0 '清除此起點之輪廓點標記
```

```
                Do
                    Dim nb As ArrayList = nc.Clone '複製此輪之搜尋起點集合
                    nc = New ArrayList '清除準備蒐集下一輪搜尋起點之集合
                    For m As Integer = 0 To nb.Count - 1
                        Dim p As Point = nb(m) '搜尋起點
                        '在此點周邊 3X3 區域內找輪廓點
                        For ii As Integer = p.X - 1 To p.X + 1
                            For jj As Integer = p.Y - 1 To p.Y + 1
                                If b(ii, jj) = 0 Then Continue For '非輪廓點忽略
                                Dim k As New Point(ii, jj) : nc.Add(k) '本輪搜尋起點
                                G.P.Add(k) '目標物件點集合
                                If ii < G.xmn Then G.xmn = ii
                                If ii > G.xmx Then G.xmx = ii
                                If jj < G.ymn Then G.ymn = jj
                                If jj > G.ymx Then G.ymx = jj
                                b(ii, jj) = 0 '清除輪廓點點標記
                            Next
                        Next
                    Next
                Loop While nc.Count > 0 '此輪搜尋有新發現輪廓點時繼續搜尋
                If Z(i - 1, j) = 1 Then Continue For '排除白色區塊的負目標，起點左邊是黑點
                G.width = G.xmx - G.xmn + 1 : G.height = G.ymx - G.ymn + 1 '寬高度計算
                If G.height < minHeight Or G.height > maxHeight Then Continue For '太矮或太高
                If G.width < minwidth Or G.width > maxWidth Then Continue For '太窄或太寬
                G.cx = (G.xmn + G.xmx) / 2 : G.cy = (G.ymn + G.ymx) / 2 '中心點
                '計算目標的對比度
                For m As Integer = 0 To G.P.Count - 1
                    Dim pm As Integer = PointPm(G.P(m))
                    If pm > G.pm Then G.pm = pm '最高對比度的輪廓點
                Next
                A.Add(G) '加入有效目標集合
            Next
        Next
        '以對比度排序
        For i As Integer = 0 To A.Count - 2
            For j As Integer = i + 1 To A.Count - 1
                Dim T As TgInfo = A(i), G As TgInfo = A(j)
                If T.pm < G.pm Then A(i) = G : A(j) = T '互換位置，高對比目標在前
            Next
        Next
        '取得 Tgmax 個最明顯的目標輸出
        Dim C As New ArrayList
        For i As Integer = 0 To Tgmax - 1
            If i > A.Count - 1 Then Exit For '超過總目標數
            Dim T As TgInfo = A(i) : T.ID = i '建立以對比度排序的序號
            C.Add(T)
        Next
        Return C '回傳目標物件集合
    End Function
    '輪廓點與背景的對比度
    Private Function PointPm(ByVal p As Point) As Integer
        Dim x As Integer = p.X, y As Integer = p.Y
        Dim mx As Integer = 0 '周邊最亮點，依據灰階陣列 B
        If mx < B(x - 1, y) Then mx = B(x - 1, y)
        If mx < B(x + 1, y) Then mx = B(x + 1, y)
        If mx < B(x, y + 1) Then mx = B(x, y + 1)
        If mx < B(x, y - 1) Then mx = B(x, y - 1)
        Return mx - B(x, y) '最亮點與輪廓點的差值
```

```
End Function
'建立單一目標的二值化矩陣
Private Function Tg2Bin(ByVal T As TgInfo) As Byte(,)
    Dim b(nx - 1, ny - 1) As Byte '二值化陣列
    For k As Integer = 0 To T.P.Count - 1
        Dim p As Point = T.P(k) : b(p.X, p.Y) = 1 '起點
        '向右連通成實心影像
        Dim i As Integer = p.X + 1
        Do While Z(i, p.Y) = 1 And i < T.xmx
            b(i, p.Y) = 1 : i += 1
        Loop
        '向左連通成實心影像
        i = p.X - 1
        Do While Z(i, p.Y) = 1 And i > T.xmn
            b(i, p.Y) = 1 : i -= 1
        Loop
    Next
    Return b
End Function
'將單一目標轉正
Private Function RotateTg(ByVal b(,) As Byte, ByRef T As TgInfo, ByVal A As Double) As Byte(,)
    If A = 0 Then Return b '無傾斜不須旋轉
    If A > 0 Then A = -A '順或逆時針傾斜時需要旋轉方向相反，經過推導 A 應該永遠為負值
    Dim R(1, 1) As Double '旋轉矩陣
    R(0, 0) = Math.Cos(A) : R(0, 1) = Math.Sin(A) : R(1, 0) = -R(0, 1) : R(1, 1) = R(0, 0)
    Dim x0 As Integer = T.xmn, y0 As Integer = T.ymx '左下角座標
    '旋轉後之目標範圍
    Dim xmn As Integer = nx, xmx As Integer = 0, ymn As Integer = ny, ymx As Integer = 0
    For i As Integer = T.xmn To T.xmx
        For j As Integer = T.ymn To T.ymx
            If b(i, j) = 0 Then Continue For '空點無須旋轉
            Dim x As Integer = i - x0, y As Integer = y0 - j '轉換螢幕座標為直角座標
            Dim xx As Integer = x * R(0, 0) + y * R(0, 1) + x0 '旋轉後 X 座標
            If xx < 1 Or xx > nx - 2 Then Continue For
            Dim yy As Integer = y0 - (x * R(1, 0) + y * R(1, 1)) '旋轉後 Y 座標
            If yy < 1 Or yy > ny - 2 Then Continue For
            b(i, j) = 0 : b(xx, yy) = 1 '清除舊點繪製新點
            '旋轉後目標的範圍偵測
            If xx < xmn Then xmn = xx
            If xx > xmx Then xmx = xx
            If yy < ymn Then ymn = yy
            If yy > ymx Then ymx = yy
        Next
    Next
    '重設目標屬性
    T.xmn = xmn : T.xmx = xmx : T.ymn = ymn : T.ymx = ymx
    T.width = T.xmx - T.xmn + 1 : T.height = T.ymx - T.ymn + 1
    T.cx = (T.xmx + T.xmn) / 2 : T.cy = (T.ymx + T.ymn) / 2
    '補足因為旋轉運算實產生的數位化誤差造成的資料空點
    For i As Integer = T.xmn To T.xmx
        For j As Integer = T.ymn To T.ymx
            If b(i, j) = 1 Then Continue For
            If b(i + 1, j) + b(i - 1, j) + b(i, j + 1) + b(i, j - 1) >= 3 Then
                b(i, j) = 1
            End If
        Next
    Next
    Return b
```

```
End Function
'建立正規化目標二值化陣列
Private Function NmBin(ByVal T As TgInfo, ByVal M(,) As Byte,
                       ByVal mw As Integer, ByVal mh As Integer) As Byte(,)
    Dim fx As Double = mw / Pw, fy As Double = mh / Ph
    Dim V(Pw - 1, Ph - 1) As Byte
    For i As Integer = 0 To Pw - 1
        Dim sx As Integer = 0 '過窄字元的平移量，預設不平移
        If T.width / mw < 0.75 Then '過窄字元，可能為 1 或 I
            sx = (mw - T.width) / 2 '平移寬度差之一半
        End If
        Dim x As Integer = T.xmn + i * fx - sx
        If x < 0 Or x > nx - 1 Then Continue For
        For j As Integer = 0 To Ph - 1
            Dim y As Integer = T.ymn + j * fy
            V(i, j) = M(x, y)
        Next
    Next
    Return V
End Function
'最佳字元
Private Function BestC(ByVal A(,) As Byte) As ChInfo
    Dim C As New ChInfo '辨識字元資料結構
    For m As Integer = 0 To 1 '六或七碼字型
        For k As Integer = 0 To 35 '0-9，A-Z
            For x As Integer = -1 To 1 '左右偏移
                For y As Integer = -1 To 1 '上下偏移
                    Dim n0 As Integer = 0, nf As Integer = 0 '字模黑點數，符合點數
                    For i As Integer = 0 To Pw - 1
                        Dim ix As Integer = i + x
                        If ix < 0 Or ix > Pw - 1 Then Continue For
                        For j As Integer = 0 To Ph - 1
                            Dim jy As Integer = j + y
                            If jy < 0 Or jy > Ph - 1 Then Continue For
                            If P(m, k, i, j) = 0 Then
                                If A(ix, jy) = 1 Then nf -= 1 '目標與字模不符合點數-1
                            Else
                                n0 += 1 '字模黑點數累計
                                If A(ix, jy) = 1 Then nf += 1 '目標與字模符合點數+1
                            End If
                        Next
                    Next
                    Dim v As Integer = nf * 1000 / n0 '符合點數百分比
                    If v > C.ft Then '符合度最高字元
                        C.ft = v : C.Ch = Ch(k)   '符合度，字元
                    End If
                Next
            Next
        Next
    Next
    For k As Integer = 0 To 1 '變形的 6 或 9
        For x As Integer = -1 To 1 '左右偏移
            For y As Integer = -1 To 1 '上下偏移
                Dim n0 As Integer = 0, nf As Integer = 0
                For i As Integer = 0 To Pw - 1
                    Dim ix As Integer = i + x
                    If ix < 0 Or ix > Pw - 1 Then Continue For
                    For j As Integer = 0 To Ph - 1
```

```
                        Dim jy As Integer = j + y
                        If jy < 0 Or jy > Ph - 1 Then Continue For
                        If P69(k, i, j) = 0 Then
                            If A(ix, jy) = 1 Then nf -= 1 '目標與字模符合點數
                        Else
                            n0 += 1 '字模黑點數累計
                            If A(ix, jy) = 1 Then nf += 1 '目標與字模符合點數
                        End If
                    Next
                Next
                Dim v As Integer = nf * 1000 / n0 '符合點數百分比
                If v > C.ft Then '符合度最高字元
                    C.ft = v '符合度
                    If k = 0 Then C.Ch = "6"c Else C.Ch = "9"c
                End If
            Next
        Next
    Next
    Return C
End Function
'儲存目前影像
Private Sub SaveImageToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs)
_
    Handles SaveImageToolStripMenuItem.Click
    If SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
        PictureBox1.Image.Save(SaveFileDialog1.FileName)
    End If
End Sub
'灰階
Private Sub RadioButton1_CheckedChanged(ByVal sender As Object, ByVal e As EventArgs) _
    Handles RadioButton1.CheckedChanged
    If RadioButton1.Checked Then
        If IsNothing(B) Then Exit Sub
        PictureBox1.Image = GrayImg(B)
    End If
End Sub
'二值化圖
Private Sub RadioButton2_CheckedChanged(ByVal sender As Object, ByVal e As EventArgs) _
    Handles RadioButton2.CheckedChanged
    If RadioButton2.Checked Then
        If IsNothing(Z) Then Exit Sub
        Dim bmp As Bitmap = BWImg(Z)
        PictureBox1.Image = bmp
    End If
End Sub
'輪廓線
Private Sub RadioButton3_CheckedChanged(ByVal sender As Object, ByVal e As EventArgs) _
    Handles RadioButton3.CheckedChanged
    If RadioButton3.Checked Then
        If IsNothing(Q) Then Exit Sub
        Dim bmp As Bitmap = BWImg(Q)
        PictureBox1.Image = bmp
    End If
End Sub
'所有合格目標
Private Sub RadioButton4_CheckedChanged(ByVal sender As Object, ByVal e As EventArgs) _
    Handles RadioButton4.CheckedChanged
    If RadioButton4.Checked Then
```

```
            If IsNothing(CA) Then Exit Sub
            Dim bmp As New Bitmap(nx, ny)
            For k As Integer = 0 To CA.Count - 1
                Dim T As TgInfo = CA(k)
                For m As Integer = 0 To T.P.Count - 1
                    Dim p As Point = T.P(m)
                    bmp.SetPixel(p.X, p.Y, Color.Black)
                Next
            Next
            PictureBox1.Image = bmp
        End If
    End Sub
    '嘗試依據英數字規範修改車牌答案
    Private Function ChkED(ByVal R As LPInfo) As LPInfo
        If IsNothing(R.A) Then Return R '無字串
        Dim C() As Char = R.A.ToCharArray '字串轉成字元陣列
        Dim n1 As Integer = R.A.IndexOf("-") '第一區段長度
        Dim n2 As Integer = C.Length - n1 - 1 '第二區段長度
        Dim d1 As Integer = 0, d2 As Integer = 0 '數字區的起終點
        If n1 = n2 Then Return R '無法判定純數字區段(2-2 或 3-3)
        If n1 > n2 Then '第一區段較長
            d1 = 0 : d2 = n1 - 1
        Else '第二區段較長
            d1 = n1 + 1 : d2 = C.Length - 1
        End If
        '嘗試將純數字區段的英文字改成數字
        For i As Integer = d1 To d2
            C(i) = E2D(C(i))
        Next
        '如果是七碼車牌，強制將前三碼中的數字改成英文
        If n1 = 3 And n2 = 4 Then
            For i As Integer = 0 To 2
                C(i) = D2E(C(i))
            Next
        End If
        R.A = "" '重組字串
        For i As Integer = 0 To C.Length - 1
            R.A += C(i)
        Next
        Return R '回傳字串
    End Function
    '嘗試將英文字母變成相似的數字
    Private Function E2D(ByVal C As Char) As Char
        If C = "B"c Then C = "8"c
        If C = "D"c Then C = "0"c
        If C = "O"c Then C = "0"c
        Return C
    End Function
    '嘗試將英文字母變成相似的數字
    Private Function D2E(ByVal C As Char) As Char
        If C = "8"c Then C = "B"c
        If C = "0"c Then C = "D"c
        Return C
    End Function
End Class
```