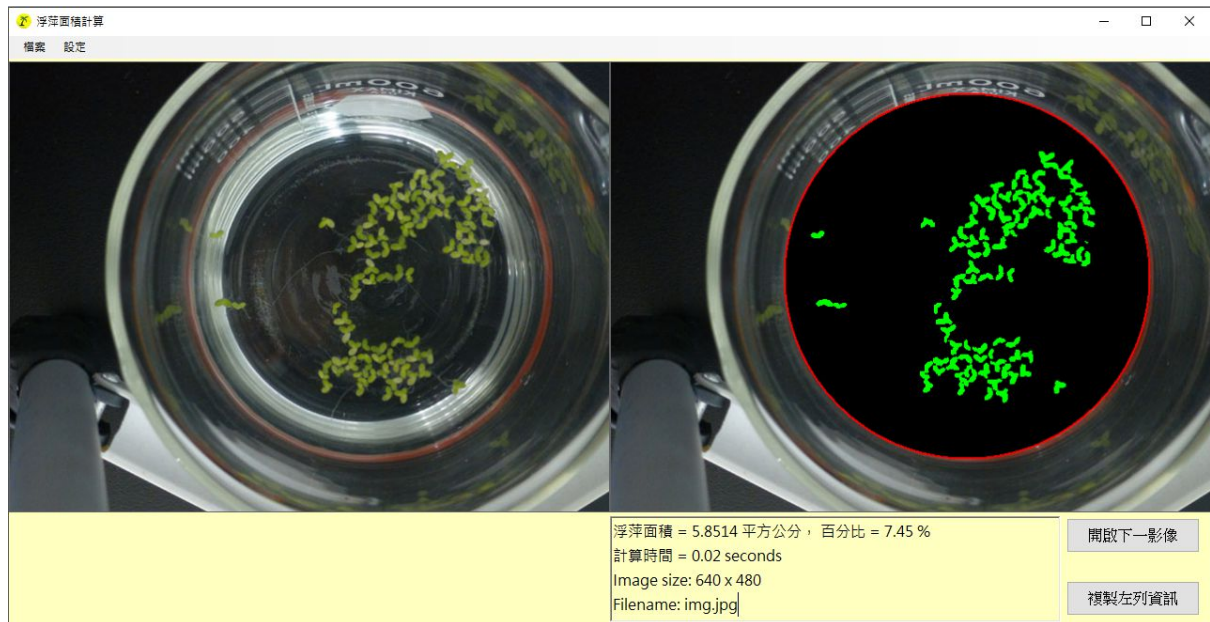


第 11 章 影像分割可以很簡單

本書以車牌辨識為例，談了很多以 OCR 為基礎與目的的影像辨識技術，但其實影像辨識並不能以 OCR 涵蓋全部，本章我們試著舉出一個性質較為不同的影像辨識案例，讓大家知道即使是看起來目標較不明確的影像辨識也可以使用傳統的影像辨識輕鬆作到，並不一定需要用到特殊的數學技巧，或冗長的機器學習訓練過程。

11-1 一個辨識浮萍面積的案例

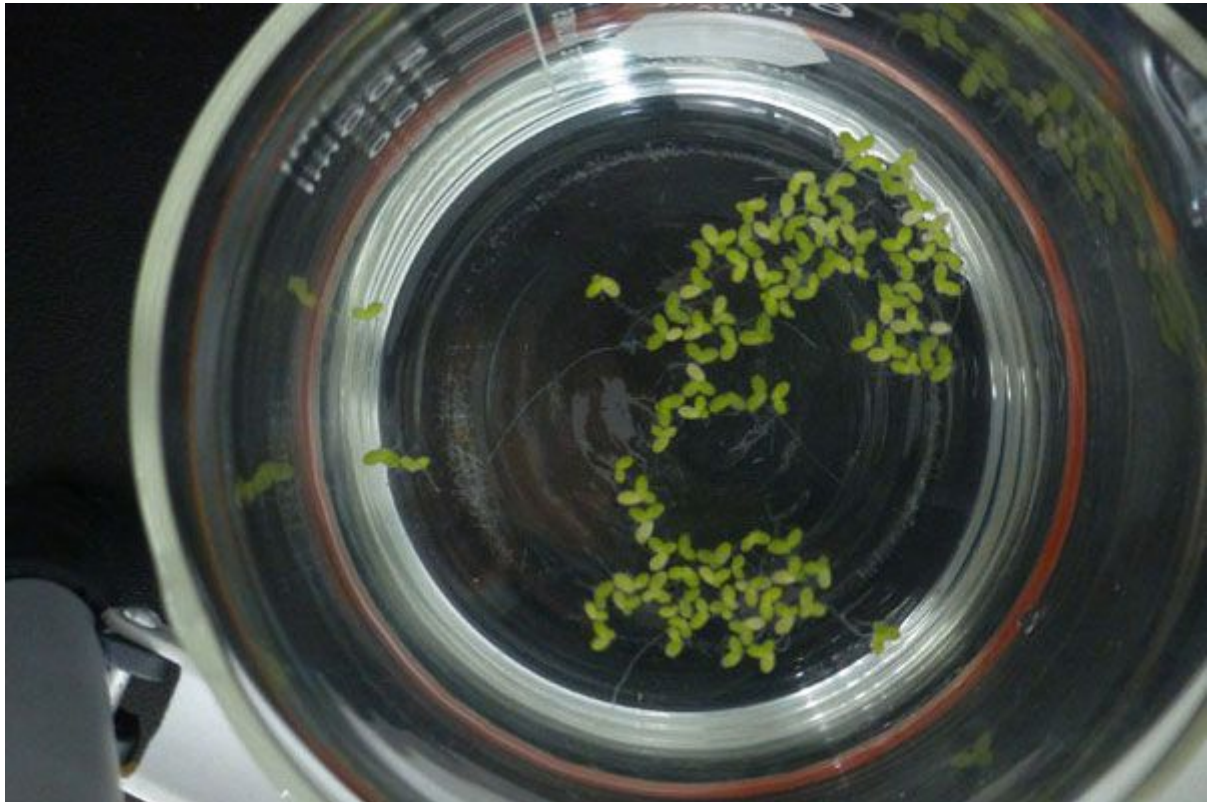


上圖是一個經過影像辨識燒杯中培養的浮萍每日增生面積的專案，乍看之下浮萍與背景的颜色對比並不明顯，燒杯壁上還有反射的浮萍倒影，那當然是不能加入計量的雜訊。因為要準確辨識浮萍佔據「水面」的面積，水與燒杯玻璃卻都是透明的，很難辨識出水面邊界，輔助方式是以紅色橡皮筋環繞套住燒杯，讓它與水面同高，希望經過辨識紅色圓圈能定義出水面的範圍。

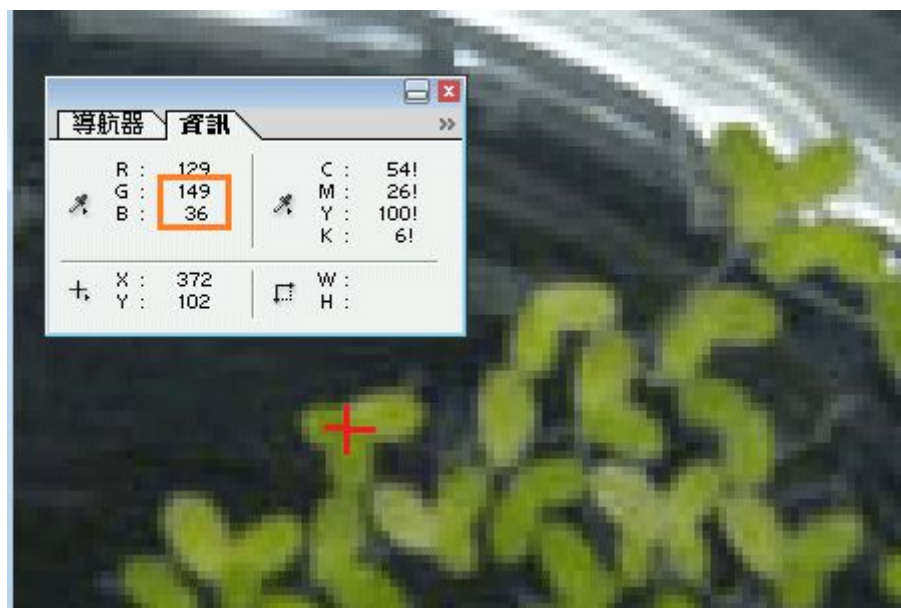
這些對比不強不太清晰的目標，看似很難辨識，如果你學過正統的影像辨識課程就會開始煩惱如何精確地將目標與背景分割(Segmentation)出來了！要不要用到複雜的影像特徵分類統計方法呢？辨識結果能不能達到收斂穩定呢？事實上只要善用常識，設計極簡單的演算法就可以作好這個辨識，本章內容之簡單直覺應該會讓你感到驚訝！

首先請複製本書第 4 章的專案程式至此修改，主功能表按鈕設計改為：Open, Leaf, Rim, Rim Tg 與 Leaf_Rim。功能表與使用範例影像如下：





11-2 辨識浮萍目標



一如之前的範例，Open 影像後會將影像的 RGB 強度拆解為 Rv, Gv 與 Bv 三個陣列，要辨識浮萍目標的特徵，很簡單的想法就是顏色偏綠， Gv 會有一定的強度，而且一定比 Rv 及 Bv 明顯要強！我們用影像處理軟體(如 PhotoShop)作了一些實驗觀察，發現綠與藍光差距較大時最能凸顯出浮萍的特徵。所以 Leaf 功能按鍵設計如下：

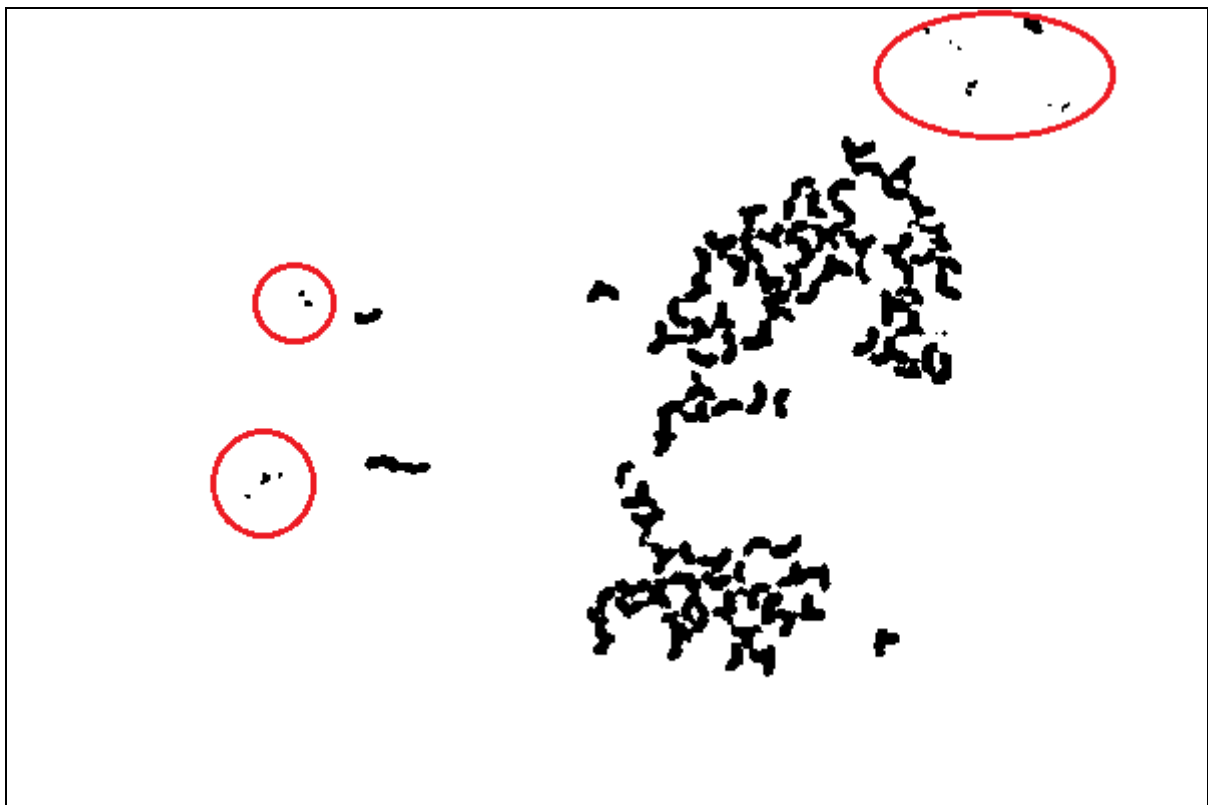
浮萍二值化

```

Private Sub LeafToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs) _
    Handles LeafToolStripMenuItem.Click
    ReDim d(nx - 1, ny - 1) '浮萍目標點陣列
    For i As Integer = 1 To nx - 2
        For j As Integer = 1 To ny - 2
            If Gv(i, j) > 70 And CInt(Gv(i, j)) - Bv(i, j) > 50 Then
                d(i, j) = 1 : Continue For '浮萍(偏綠色點)
            End If
        Next
    Next
    Dim bmp As Bitmap = BWImg(d) '建立浮萍二值化圖
    PictureBox1.Image = bmp '顯示
End Sub

```

執行後結果影像如下，比照原圖可知，紅色圈選部位是燒杯壁的浮萍倒影，也是我們接下來要設法排除的雜訊。怎麼知道他們是雜訊？當然就是要辨識水面的邊緣，紅色橡皮筋的位置了！



11-3 辨識水面邊界

如前所述，我們以紅色橡皮筋標示水面位置，它的特徵應該是紅色光明顯大於藍綠色光，所以 Rim 功能鍵的程式設計如下：

```

'水面圓周邊界二值化
Private Sub RimToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs) _

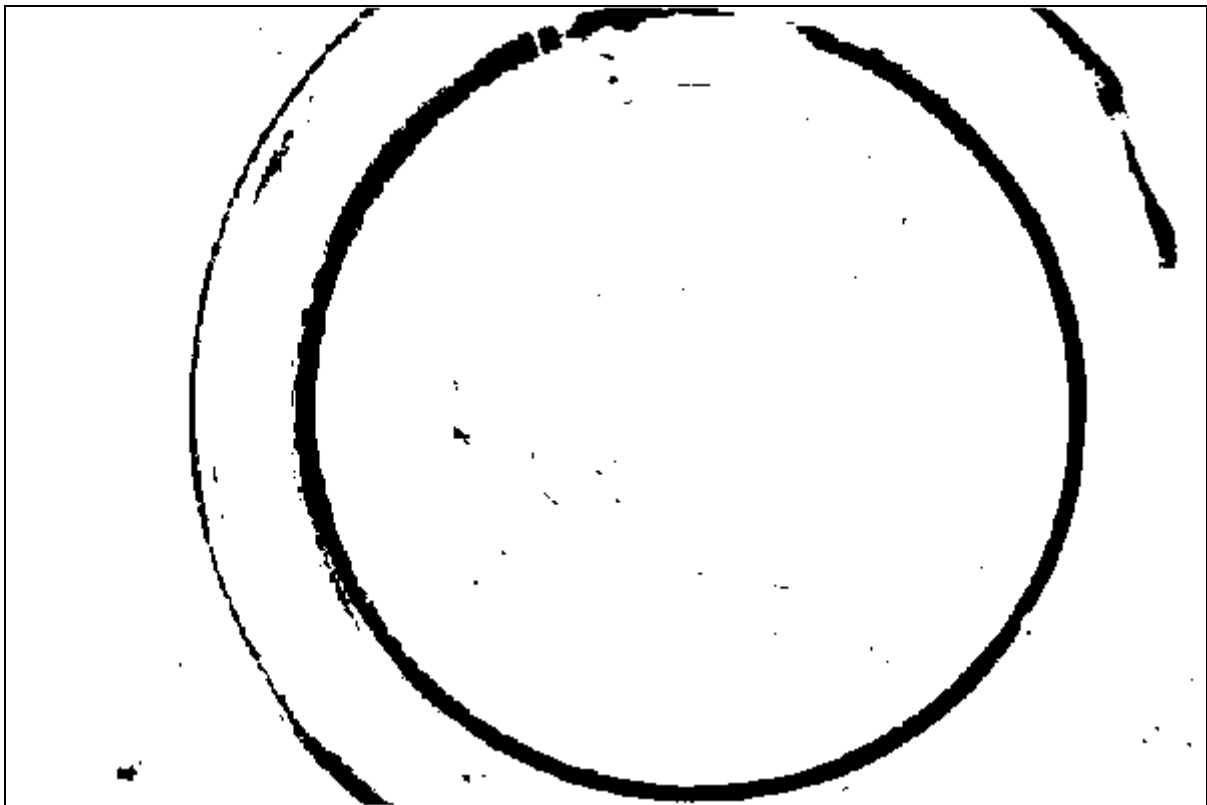
```

```

Handles RimToolStripMenuItem.Click
ReDim c(nx - 1, ny - 1) '圓周標記目標點陣列
For i As Integer = 1 To nx - 2
    For j As Integer = 1 To ny - 2
        If Rv(i, j) > 70 And Rv(i, j) > Gv(i, j) And Rv(i, j) > Bv(i, j) Then '偏紅色點
            c(i, j) = 1 '邊界
        End If
    Next
Next
Dim bmp As Bitmap = BWImg(c) '建立邊界二值化圖
PictureBox1.Image = bmp '顯示
End Sub

```

執行結果如下：



比照原圖我們知道，中心較完整的圓圈是真正橡皮筋的位置，外圈則是它的倒影，還有一些意外偏紅色的雜點，這些當然都是我們不要的雜訊了！該如何將這個結果變成一個簡單的圓形邊界資訊呢？以本書第 4 章建立獨立目標物件的方式，我們可以將上圖的幾個區塊變成目標物件，中央最清晰的目標必然會變成最大，點數也最多的目標，只留下它就是清除雜訊了！這就是 Rim Tg 的功能，程式碼如下：

```

'取得邊界(紅色圓圈)目標
Private Sub RimTgToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs) _
    Handles RimTgToolStripMenuItem.Click
    Dim T As TgInfo = getMaxTg(c) '取得最大紅色區塊目標

```

```

'繪製圓周目標
Dim bmp As New Bitmap(nx - 1, ny - 1)
For k As Integer = 0 To T.P.Count - 1
    Dim pt As Point = T.P(k)
    bmp.SetPixel(pt.X, pt.Y, Color.Black)
Next
PictureBox1.Image = bmp '顯示
End Sub
'取得最大目標
Function getMaxTg(ByVal q(.) As Byte) As TgInfo
    Dim mx As Integer = 0 '最多的目標點數
    RTg = New TgInfo '最大的圓周目標
    Dim b(.) As Byte = q.Clone '建立目標點陣列副本
    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.np = 1 : 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
            b(i, j) = 0 '清除此起點之標記
            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 區域內找目標點
                    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 '非目標點忽略
                            Dim k As New Point(ii, jj) '建立點物件
                            nc.Add(k) '本輪搜尋新增的目標點
                            G.P.Add(k)
                            G.np += 1 '點數累計
                            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 '此輪搜尋有新發現目標點時繼續搜尋
            If q(i - 1, j) = 1 Then Continue For '排除白色區塊的負目標，起點左邊是黑點
            If G.np > mx Then '發現更大目標時
                mx = G.np
            End If
        Next
    Next
End Function

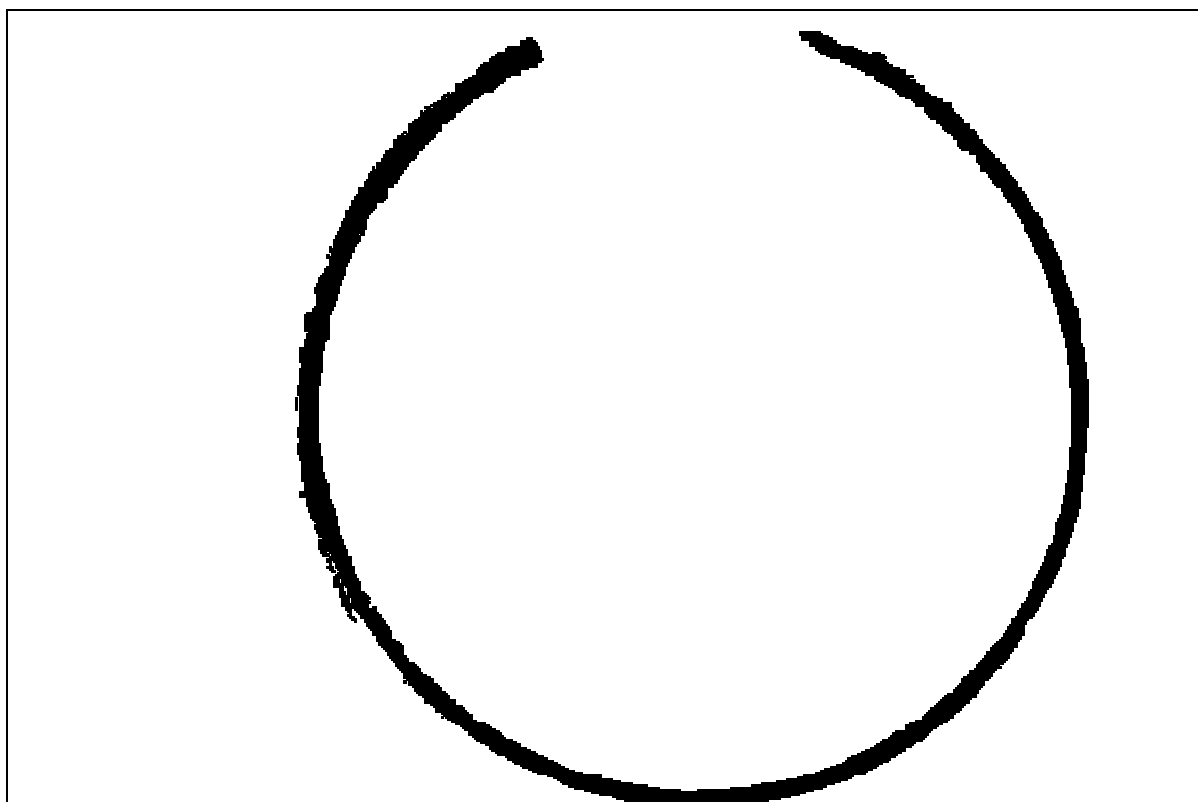
```

```

        G.width = G.xmx - G.xmn + 1 : G.height = G.ymx - G.ymn + 1
        G.cx = (G.xmn + G.xmx) / 2 : G.cy = (G.ymn + G.ymx) / 2
        RTg = G
    End If
Next
Next
Return RTg '回傳最大目標
End Function

```

getMaxTg 副程式是前面幾章中 getTargets 副程式的稍微簡化版，指回傳最大的一個目標，而不是所有合格目標的集合。執行結果就會像這樣：



11-4 建立邊界計算浮萍面積

面對上節的辨識結果，我們已經非常接近能夠建立一個應該為圓形的水面邊界，只是最上方因為燒杯上有塊不透明的商標遮住了橡皮筋造成一個缺口，但是不難推算補足。重點是要如何「回歸」計算出一個圓形的「方程式」？就想我們作虎克定律的實驗時要回歸計算出一個直線方程式一樣？

如果你從這個方向思考還真的是陷入了一個數學的難題，但從另一角度看，我們上一節找出的目標物件屬性中已經有**概略**的「圓心」了！就是目標的中心點，除非圓弧的缺口實在太大，否則把它當作原因的誤差是很小的。如果我們計算此目標內所有已知的偏紅目標點與此圓心的距離作個平均，我們就有此邊界圓形概略的半徑了！

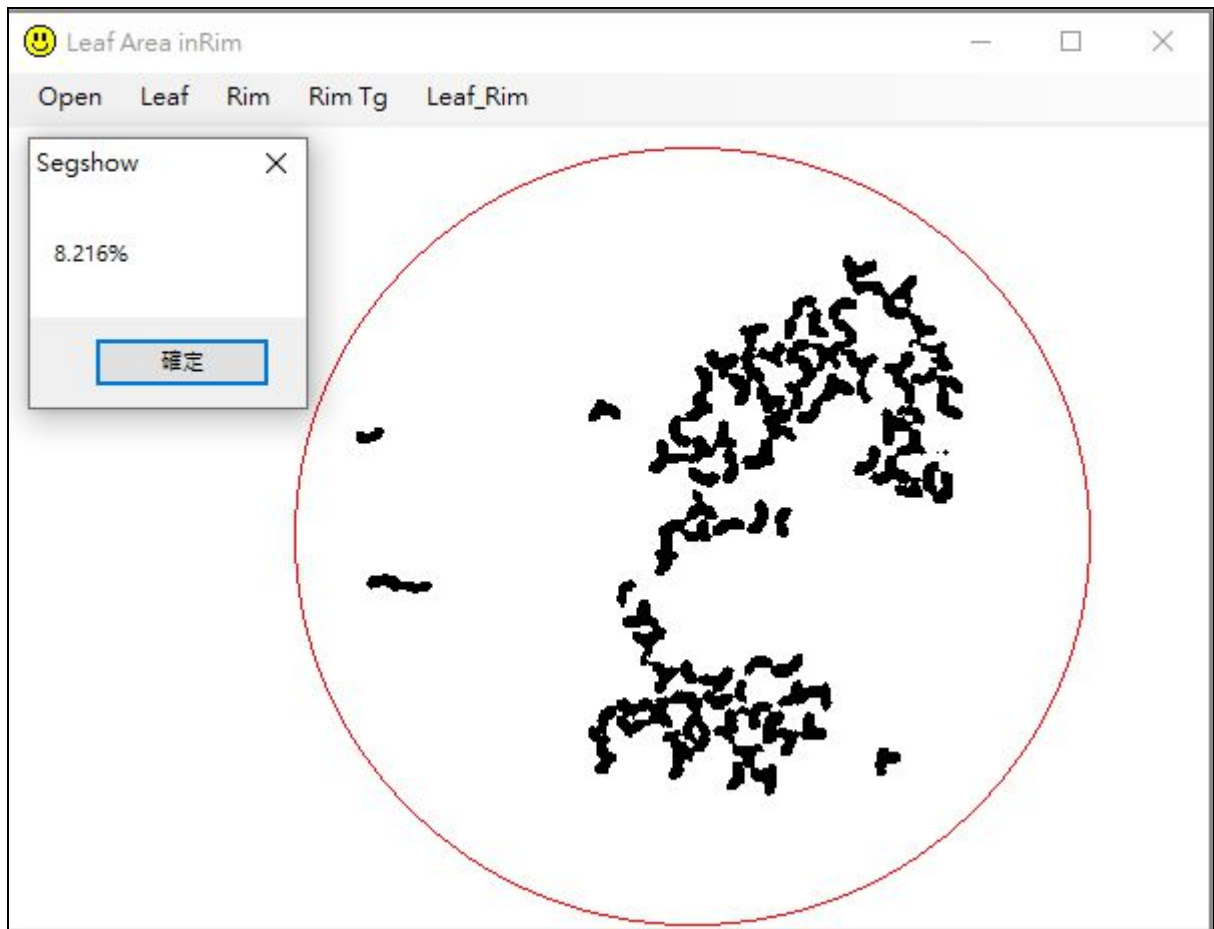
有圓心也有半徑，我們就可以知道原始影像上，所有「可能」是浮萍的偏綠目標點

哪些是在水面邊界的內部？哪些是在外部？在內的就是真的浮在水面的浮萍，在外的就是燒杯壁上的倒影了！接下來只要統計真實浮萍目標點的點數，除以圓形邊界包圍的區域面積就可以得到正確的覆蓋率了！Leaf_Rim 程式碼如下：

計算與繪製結果→圓周內之浮萍面積

```
Private Sub LeafRimToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs) _
    Handles LeafRimToolStripMenuItem.Click
    Dim r As Double = 0 '邊界圓圈之半徑
    For k As Integer = 0 To RTg.P.Count - 1
        Dim p As Point = RTg.P(k)
        Dim rr As Double = Math.Sqrt((p.X - RTg.cx) ^ 2 + (p.Y - RTg.cy) ^ 2)
        r += rr
    Next
    r /= RTg.P.Count '平均半徑
    Dim A(nx - 1, ny - 1) As Byte, nLf As Integer = 0
    For i As Integer = RTg.xmn To RTg.xmx
        For j As Integer = RTg.ymn To RTg.ymx
            If d(i, j) = 0 Then Continue For
            Dim rr As Double = Math.Sqrt((i - RTg.cx) ^ 2 + (j - RTg.cy) ^ 2) '浮萍目標到中心距離
            If rr > r Then Continue For '目標點在監視區外
            A(i, j) = 1 : nLf += 1 '監視範圍內→計量
        Next
    Next
    Dim bmp As Bitmap = BWImg(A) '建立監視區內浮萍二值化圖
    Dim G As Graphics = Graphics.FromImage(bmp)
    G.DrawEllipse(Pens.Red, RTg.xmn, RTg.ymn, RTg.width, RTg.height) '繪製監視區邊界圓周
    PictureBox1.Image = bmp '顯示
    Dim Area As Double = Math.PI * r ^ 2 '監視總面積→畫素單位
    Dim pc As Double = nLf * 100 / Area '浮萍面積百分比
    MsgBox(Format(pc, "###.###") + "%")
End Sub
```

執行結果如下：



11-5 其他影像辨識書沒說的重點

我是真正從事影像辨識產品開發的業者，從學習理論到實務開發多年間最大的感觸是：影像辨識教材內容多半著重在特徵的辨識擷取，而且使用的數學方法都太過繁複，誤導學習者以為將影像分割擷取必須使用很複雜的數學概念？但是實務上，即使如本章案例看起來是不太容易辨識的自然影像，還是可以用簡單的常識概念就能達到正確切割目標的目的，書上介紹的**複雜影像分類演算法**其實極少必須使用！

另一方面，要完成任何影像辨識的完整目標，幾乎都無法避免會用到很多**幾何學**的概念技術，譬如本章案例辨識出紅色橡皮筋的概略位置之後，必須結合幾何概念來建立邊界資訊排除浮萍倒影的雜訊。又如前面的車牌辨識，很多演算法必須結合字元排列與分佈的幾何關係來設計，目標不會永遠清晰，目標切割不會永遠完美，如果遇到不完美的狀況，我們都需要目標群組之間的幾何關係來補救，譬如車牌字元之間有太大的空隙就必須補字作強迫性的辨識之類的！

這些幾何運作的資訊假設也不是都來自影像本身，譬如我們為何認定本案例的水面是圓形的？其實是來自我們對於燒杯形狀的認知！所以我始終強調：**影像辨識不只是辨識影像**！要達到辨識的目的，我們必須加入很多影像之外的先備資訊，這些資訊可以幫助我們更容易，也更明確的完成任務！事實上，實務開發影像辨識產品時，很少需要動

用到機率統計學的！碰到任何影像辨識工作，二話不說就開始啟動機器學習或深度學習的工作流程，讓電腦自行進行大量的統計與學習，其實就是盲目摸索嘗試錯誤，那是非常沒有效率的做法。

完整專案

Public Class Form1

Dim c(,) As Byte '監視區圓周標記點

Dim d(,) As Byte '浮萍目標點

Dim RTg As TgInfo '邊界標記之目標物件

'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 width As Integer, height As Integer '寬與高

Dim cx As Integer, cy As Integer '中心點座標

End Structure

'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) '讀取 RGB 亮度陣列

PictureBox1.Image = bmp '顯示

End If

End Sub

'浮萍二值化

Private Sub LeafToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs) _

Handles LeafToolStripMenuItem.Click

ReDim d(nx - 1, ny - 1) '浮萍目標點陣列

For i As Integer = 1 To nx - 2

For j As Integer = 1 To ny - 2

If Gv(i, j) > 70 And CInt(Gv(i, j)) - Bv(i, j) > 50 Then

d(i, j) = 1 '浮萍(偏綠色點)

End If

Next

Next

PictureBox1.Image = BWImg(d) '建立浮萍二值化圖

End Sub

'水面圓周邊界二值化

Private Sub RimToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs) _

Handles RimToolStripMenuItem.Click

ReDim c(nx - 1, ny - 1) '圓周標記目標點陣列

For i As Integer = 1 To nx - 2

For j As Integer = 1 To ny - 2

If Rv(i, j) > 70 And Rv(i, j) > Gv(i, j) And Rv(i, j) > Bv(i, j) Then '偏紅色點

c(i, j) = 1 '邊界

End If

Next

Next

Dim bmp As Bitmap = BWImg(c) '建立邊界二值化圖

PictureBox1.Image = bmp '顯示

End Sub

'取得邊界(紅色圓圈)目標

Private Sub RimTgToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs) _

Handles RimTgToolStripMenuItem.Click

Dim T As TgInfo = getMaxTg(c) '取得最大紅色區塊目標

'繪製圓周目標

Dim bmp As New Bitmap(nx - 1, ny - 1)

For k As Integer = 0 To T.P.Count - 1

```

        Dim pt As Point = T.P(k)
        bmp.SetPixel(pt.X, pt.Y, Color.Black)
    Next
    PictureBox1.Image = bmp '顯示
End Sub
'取得最大目標
Function getMaxTg(ByVal q(,) As Byte) As TgInfo
    Dim mx As Integer = 0 '最多的目標點數
    RTg = New TgInfo '最大的圓周目標
    Dim b(,) As Byte = q.Clone '建立目標點陣列副本
    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.np = 1 : 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
            b(i, j) = 0 '清除此起點之標記
            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 區域內找目標點
                    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 '非目標點忽略
                            Dim k As New Point(ii, jj) '建立點物件
                            nc.Add(k) '本輪搜尋新增的目標點
                            G.P.Add(k)
                            G.np += 1 '點數累計
                            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 '此輪搜尋有新發現目標點時繼續搜尋
            If q(i - 1, j) = 1 Then Continue For '排除白色區塊的負目標 · 起點左邊是黑點
            If G.np > mx Then '發現更大目標時
                mx = G.np
                G.width = G.xmx - G.xmn + 1 : G.height = G.ymx - G.ymn + 1
                G.cx = (G.xmn + G.xmx) / 2 : G.cy = (G.ymn + G.ymx) / 2
                RTg = G
            End If
        Next
    Next
    Return RTg '回傳最大目標
End Function
'計算與繪製結果→圓周內之浮萍面積
Private Sub LeafRimToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs) _
    Handles LeafRimToolStripMenuItem.Click
    Dim r As Double = 0 '邊界圓圈之半徑
    For k As Integer = 0 To RTg.P.Count - 1
        Dim p As Point = RTg.P(k)
        Dim rr As Double = Math.Sqrt((p.X - RTg.cx) ^ 2 + (p.Y - RTg.cy) ^ 2)
    Next

```

```

        r += rr
    Next
    r /= RTg.P.Count '平均半徑
    Dim A(nx - 1, ny - 1) As Byte, nLf As Integer = 0
    For i As Integer = RTg.xmn To RTg.xmx
        For j As Integer = RTg.ymn To RTg.ymx
            If d(i, j) = 0 Then Continue For
            Dim rr As Double = Math.Sqrt((i - RTg.cx) ^ 2 + (j - RTg.cy) ^ 2) '浮萍目標到中心距離
            If rr > r Then Continue For '目標點在監視區外
            A(i, j) = 1 : nLf += 1 '監視範圍內→計量
        Next
    Next
    Dim bmp As Bitmap = BWImg(A) '建立監視區內浮萍二值化圖
    Dim G As Graphics = Graphics.FromImage(bmp)
    G.DrawEllipse(Pens.Red, RTg.xmn, RTg.ymn, RTg.width, RTg.height) '繪製監視區邊界圓周
    PictureBox1.Image = bmp '顯示
    Dim Area As Double = Math.PI * r ^ 2 '監視總面積→畫素單位
    Dim pc As Double = nLf * 100 / Area '浮萍面積百分比
    MsgBox(Format(pc, "##.###") + "%")
End Sub
'儲存影像
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
End Class

```