(1) 下載和安裝 SVM

https://github.com/cjlin1/libsvm

這裡有不同版本的 SVM code

(2) 從以下的雲端連結

來將 Berkeley dataset (部分的圖是我們自己加進去的)的影像下載

https://drive.google.com/file/d/1z4VLBE08GuP2KVW77POQYG8khZWJVw7s/view?usp=sharing

或者也可以由以下連結下載

http://djj.ee.ntu.edu.tw/TestImagesforPrograms.zip

(3) 從圖上來選取120個 training points

(60個來自於人臉，60個來自於非人臉

為了讓 training data 比較 robust， training points 要來自於不同的影像)

並且記錄這些點的 RGB 值，以及哪些是人臉的點，哪些是非人臉的點

註：可以用 ginput 的指令協助

(4) 將這些 training points 的 RGB 值轉換成 YCbCr 值



(5) 將 Y, Cb, Cr 值做 normalization

 where 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 150.87295 | -8.95915 | 11.4630917 | 20.6413177 | 13.3093015 | 14.2372615 |

(1) 將 training data 做為 SVM 的輸入

總共120筆資料

每一筆資料有3個 features (即已經做好normalization 的 Y，Cb, Cr)

Label 有兩個 (label 0代表non-face，label 1代表face)

輸入資料要打亂會比較好

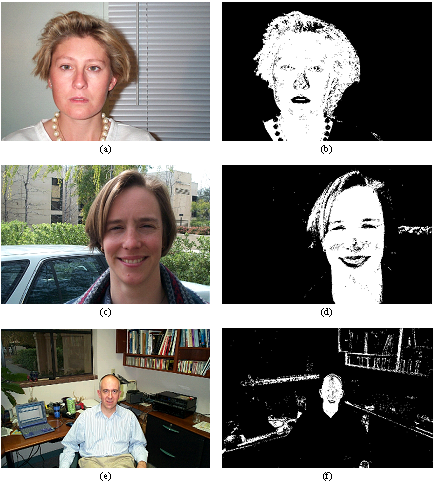
(避免前60點皆為人臉，後60點皆不為人臉

最好人臉非人臉的點交錯分佈)

(2) 得出 SVM 的 training model

(3) 拿 TestImagesForPrograms 當中其中幾個影像，做為 test data，將影像當中的每個點將放進 SVM，來得出 test output

(4) 將結果顯示出來，將被判斷是人臉的點設為白色 (gray level = 255)，被判斷是非人臉的點設為黑色 (gray level = 0)，例如左圖是原圖，右圖是 SVM 做 skin filter 的結果



(5) 找出判別錯誤的點 (從右圖可以看出，有部分人臉的點被判斷成是非人臉，有部分非人臉的點被判斷成是人臉)

(6) 選擇其中40個判斷錯誤的點 (要來自不同的影像，人臉點和非人臉點數量不需平衡)，來加入 training data 當中，將 training data 的數量增加40個

回到第 (1) 步

(7) 以上的流程，反覆進行二至三次(training points 增加到 200或240個)，通常就可以得出還算不錯的人臉偵測結果

Skin Filter 的後處理

假設人臉偵測的輸出為 F[m, n] (F[m, n] = 1 for face, F[m, n] = 0 for non-face)

(1) 做 Opening Operation

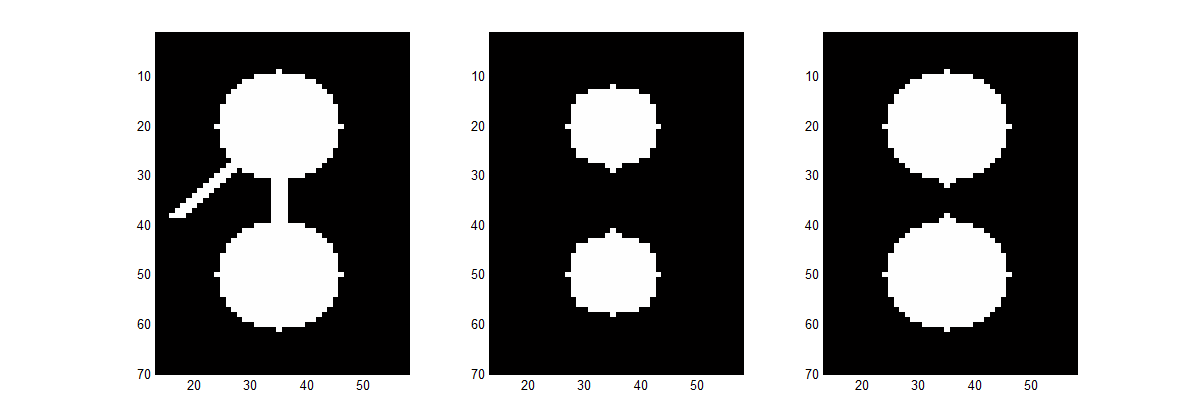
**Opening = Erosion連做 k 次 加上 Dilation連做k 次**

(k = N/100, MxN 是影像的 size)

Erosion 的運算為 F[m, n] = min(F[m, n], F[m-1, n], F[m+1,n], F[m, n-1], F[m,n+1])

Dilation 的運算為 F[m, n] = Max(F[m, n], F[m-1, n], F[m+1,n], F[m, n-1], F[m,n+1])

Opening 在影像處理當中，有移除雜點雜訊，和讓不小心連起來的區域分開的效用，如下圖



(a) 原圖 (b) erosion k 次 (c) 再做dilation k 次

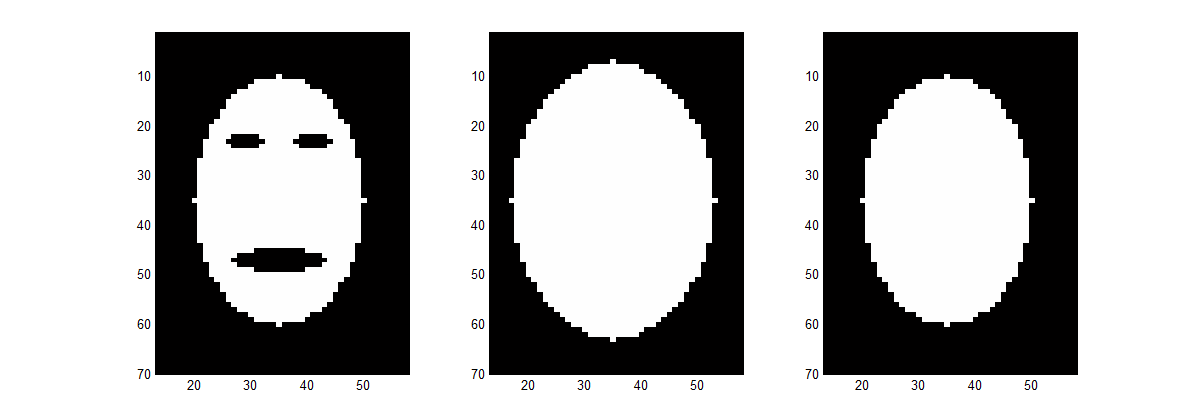
(2) 做 Closing Operation

**Closing = Dilation連做 k 次 加上 Erosion 連做k 次**

k 通常選為 min(M, N)/100, MxN 是影像的大小

Closing 在影像處理當中，有填孔的效果，dilation k 次之後，只要 k 夠大，中間的缺孔會被填起來，再做 erosion 之後，外面的邊界會回到原來的地方，但中間的缺孔不會再出現，如下圖

(註：也有部分同學認為，做了 closing operation 反而效果未必較好)



(a) 原圖 (b) dilation k 次 (c) 再做 erosion k 次

(3) 將剩下的人臉區域，用 binary segmentation 的方式來做分割

將等於1且相鄰的區域編為同一個編號

等於1 而不相鄰的區域編為不同的個編號

例如，假設人臉偵測結果為



經過二值化分割後，結果變成



相鄰的等於1的點編為同一個編號

若使用 Matlab, binary segmentation 的指令為

B = bwlabel(A)

若使用 Python, binary segmentation 的指令為

from skimage import measure

B = measure.label(A)

(4) 計算每一個區域的面積

例如，上面的例子，若要計算第二區的面積，可使用

Area(2) = sum(B==2, ‘all’);

(5) 假設人臉再怎麼小，也會大於整張圖的面積的1/1000

所以，若 Area(n) < MN/1000

則將這個區域排除

if Area(B[m, n])< M\*N/1000

A[m, n] = 0

B[m, n] = 0