影像處理第三次作業

班級：資工碩士一年級
學號：7109056258
學生：陳峻郁

目錄

形態學:邊界抽取與區域填充 (Morphology:Boundary Extraction & Region Filling)

1.題目說明

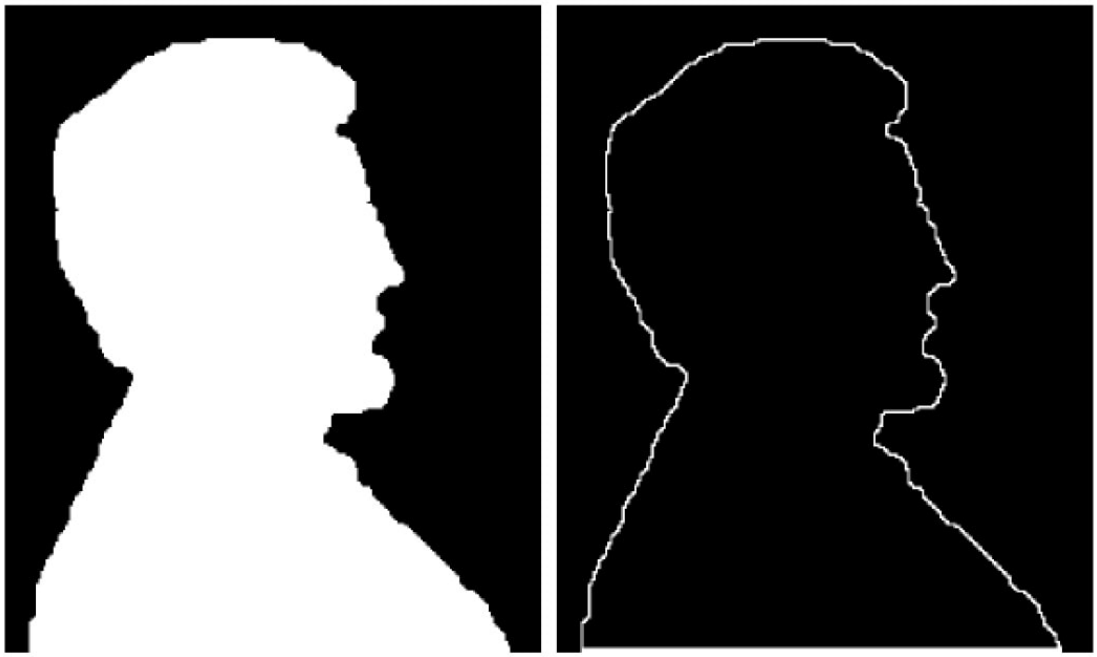
(a) Boundary Extraction

邊界抽取，與我們在第二次作業中所做的內容十分相似，laplace、sobel 等等，但是那些使用微分的方法通常計算複雜度都偏難，而用 Morpholohy 方法做這件事情的計算複雜度是較低的，加上可以透過調整 Structures Element 來決定你的邊界要多寬，這是那些用微分找邊緣的方法不容易辦到的。

Boundary Extraction 的數學公式如下：

$$\beta(A) = A - (A \ominus B)$$

公式中 A 為你打算做 Boundary Extraction 的目標 (Source Image)，B 為你設定的 Structures Element。簡單來說就是用 A 跟 B 做 Erosion 會得到一個比較小的 A，再用原來的 A 減掉比較小的 A，那麼多出來的地方就剛好是邊界了。



圖(1) 邊界抽取結果圖

(b)Region Filling

它具體的功能可以用來填色，把相鄰同色區域改填成另一種顏色，就是可以將黑色圓形區域填成白色。

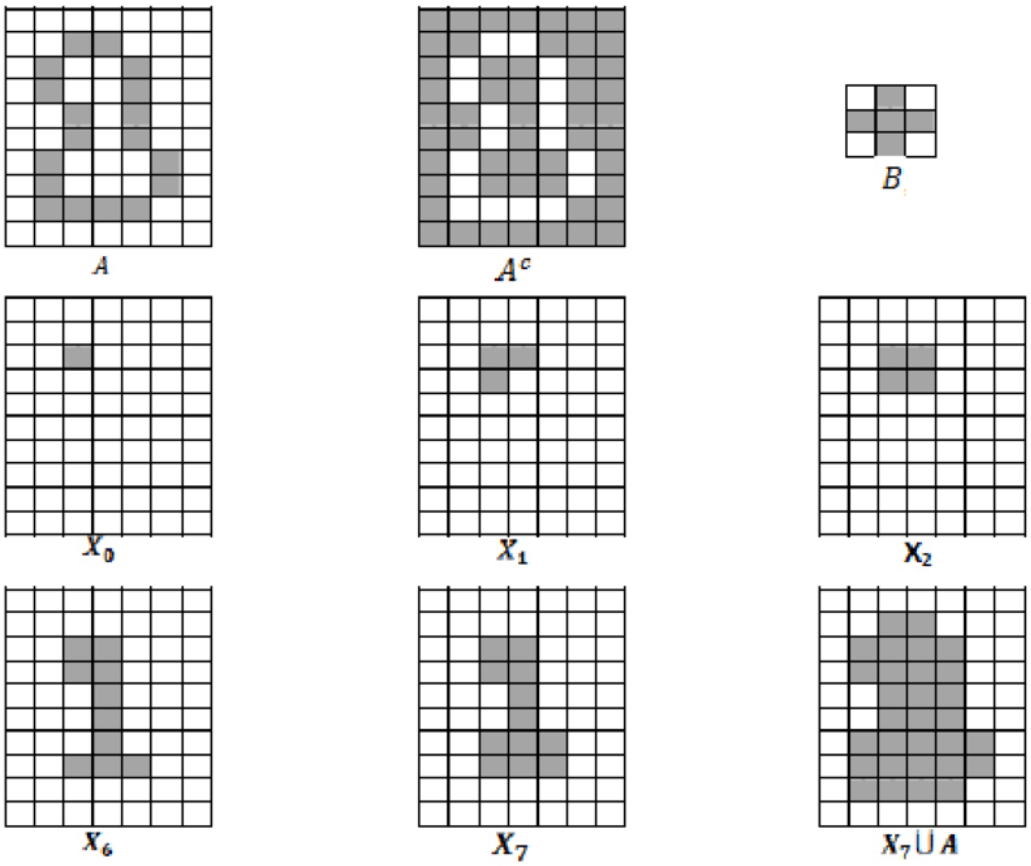
Region Filling 的數學公式如下

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1,2,3,\cdots$$

Xk 就是今天你要填充區域裡面的 Element，B 為你要拿來做 Dilation 的 Structures Element，配合後面跟 A 的補集做 AND 的條件，然後就會從 k = 1、2、3、... 開始做，做到被 A 包住的整個範圍都做完以後結束。

0	1	0
1	1	1
0	1	0

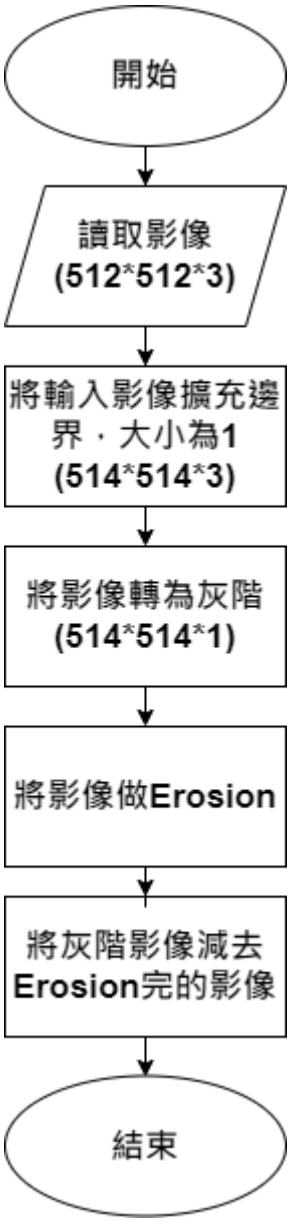
圖(2) Structure Element



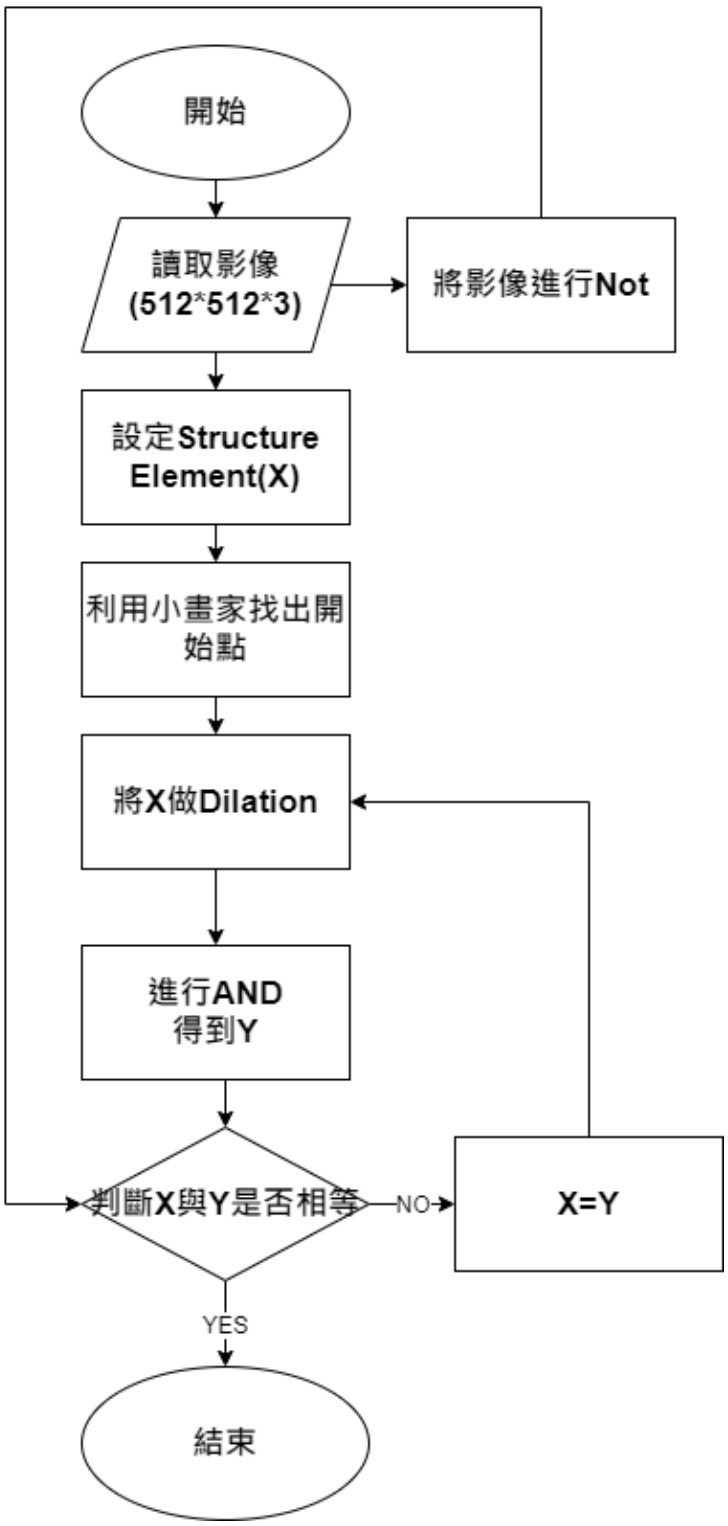
圖(3) Region Filling 範例

2. 演算法流程圖

(a) Boundary Extraction



(b) Region Filling



3. 程式碼與步驟

(a) Boundary Extraction

step1:引入函示庫

引入必要函式庫 ¶

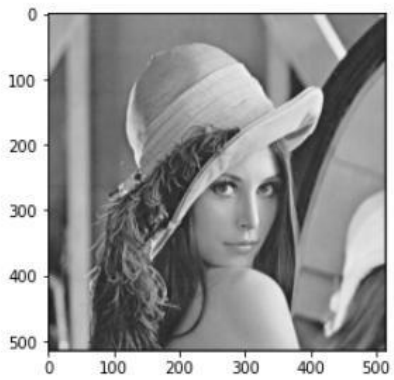
In [1]:
```
import sys
import cv2
import numpy as np
import matplotlib.pyplot as plt
import math
```

Step2:讀取影像(512*512*3)，並將影像進行邊緣擴充(514*514*3)，因為要處理卷積，之後將影像轉成灰階(514*514*1)。

讀取照片

In [2]:
```
img=cv2.imread("512.jpg")
print(img.shape)
img=cv2.copyMakeBorder(img,1,1,1,1,cv2.BORDER_CONSTANT,value=0)
imgInfo = img.shape
height = imgInfo[0]
width = imgInfo[1]
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(gray,cmap ='gray')
plt.show()
print(img.shape)
```

```
(512, 512, 3)
```

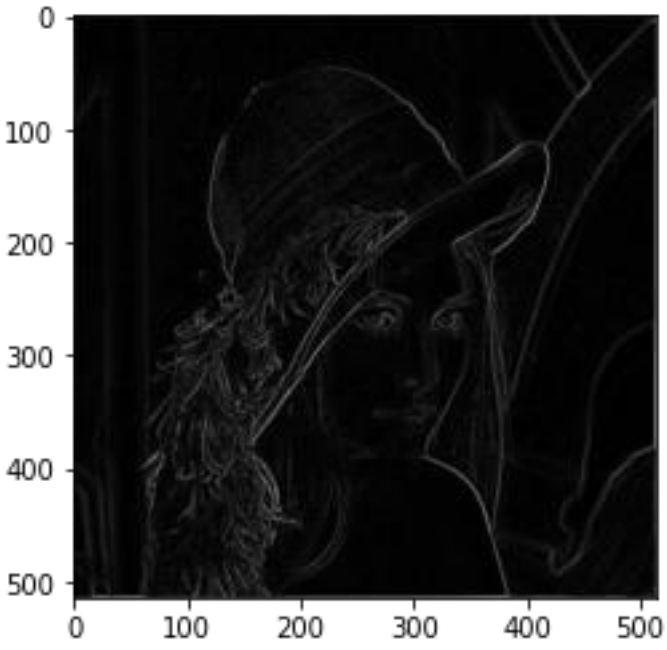



```
(514, 514, 3)
```

Step3:對灰階影像進行 Erosion，就是將影像進行收縮，將影像的每個點，以他為中心，將周圍的一圈的點(包含自己)(3*3)取最小值

Erosion

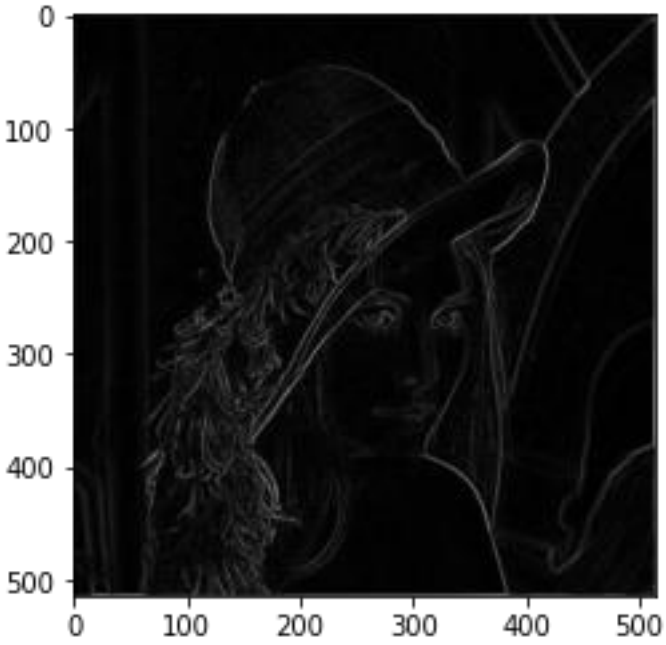
In [4]:
```
erosion = np.zeros((height,width))
for i in range(0,height-2):
    for j in range(0,width-2):
        erosion[i,j]=min(gray[i-1][j-1],gray[i][j-1],gray[i+1][j-1],
                         gray[i-1][j],gray[i][j],gray[i+1][j],
                         gray[i-1][j+1],gray[i][j+1],gray[i+1][j+1])
result=gray-erosion
plt.imshow(result,cmap ='gray')
plt.show()
print(img.shape)
```



Step4:最後再將原圖灰階影像減去 erosion 完的影像，因為 erosion 完，圖片會收縮，再利用原圖減去，即可找到微小的差距，也就是他的邊界。

Result

```
In [5]: result=gray-erosion
        plt.imshow(result,cmap ='gray')
        plt.show()
        print(img.shape)
```



(b) Region Filling

Step1:引入函示庫

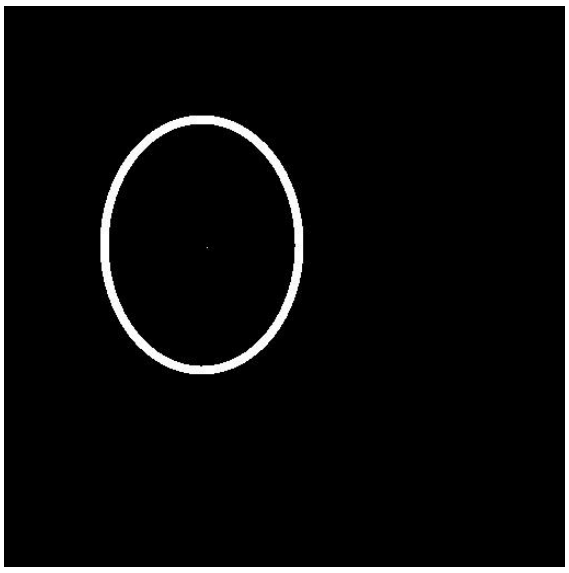
引入必要函式庫 ¶

```
In [1]: import sys
        import cv2
        import numpy as np
        import matplotlib.pyplot as plt
        import math
```

Step2: 將欲處理的圖片載入

載入圖片

```
img=cv2.imread("circle.jpg",0)
imgInfo = img.shape
height = imgInfo[0]
width = imgInfo[1]
print(img.shape)
kernel=np.array([[0,1,0],[1,1,1],[0,1,0]],np.uint8)
not_img = cv2.bitwise_not(img)
#185,220
print(height,width)
```



Step3: 利用小畫家，找出起始的那個座標，也就是白色的點，我找出來後是(185,220)

利用小畫家找出起始座標(白點)

```
print(img[185][220])
sta_x=185
sta_y=220
x = np.zeros((512,512),np.uint8)
x[sta_x][sta_y] = 255
```

Step4: 接下來要對找出來的座標矩陣 x 與 Structures Element 做 dilation，dilation，因為我的 Structures Element 為 $\begin{bmatrix}0&1&0\\1&1&1\\0&1&0\end{bmatrix}$，要對這個做 dilation，而 dilation 就是取 9 個點當中的最大值，由於其他點沒有用到，只要寫出與中心距離唯一的其他 4 個點就好

dilation函式 ¶

```
def dilation(x):
    result = np.zeros((514,514),np.uint8)
    x = cv2.copyMakeBorder(x,1,1,1,1, cv2.BORDER_CONSTANT,value=[0,0,0])
    for i in range(1,513):
        for j in range(1,513):
            result[i][j] = max(x[i-1][j],x[i][j-1],x[i][j],x[i][j+1],x[i+1][j])
    return result[1:513,1:513]
```

Step5: 之後將 dilation 完的 x 和原圖片的補集合做 AND，並取名叫做 y。之後判斷 x 與 y 是否相等，

相等:代表已經填滿

不相等:尚有未填滿的部分，並且讓 x=y，也就是讓填滿完的部分=x，在與 Structures Element 進行 dilation，在跟原圖片的補集合做 AND，直到 x=y 為止。

將Structures Element與照片的補集合做and

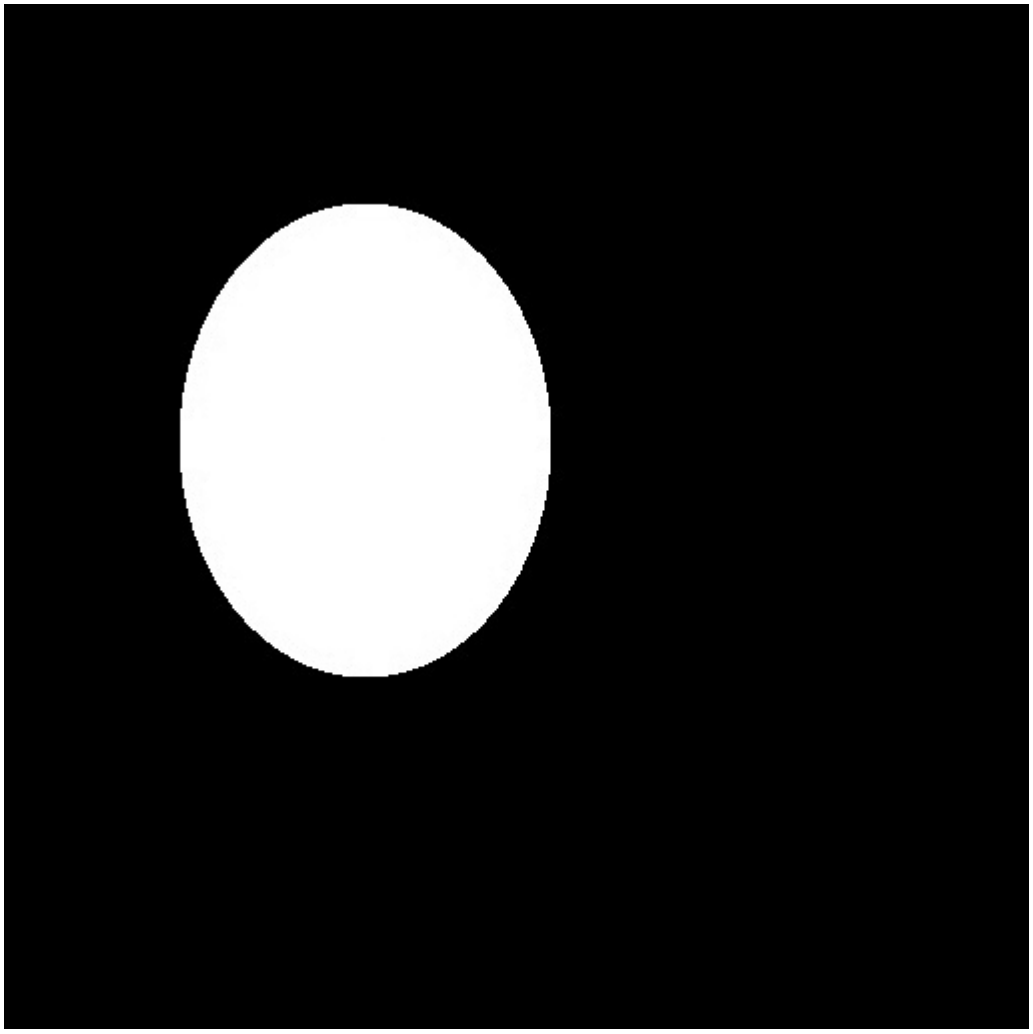
```
y = cv2.bitwise_and(dilation(x),not_img)
while ~((x==y).all()):
    x = y
    y = cv2.bitwise_and(dilation(x),not_img)
y = cv2.bitwise_or(y,img)
```

Step6: 將圖片儲存

儲存圖片

```
cv2.imshow('y',y)
cv2.waitKey(0)
cv2.destroyAllWindows()
cv2.imwrite('output.png',y)
```

Step6: 結果



4.心得

這次的作業是做 Morphology 裡的邊界抽取與區域填充，一開始看到，就想說邊界抽取不就跟作業二所做的事情一樣嗎?後來才發現，原來 Morphology 裡的方法，計算量比 sobel 小很多，可以增加運算速度。在還沒碰 Morphology 之前，就有聽同學說過 dilation 和 erosion，但是都不知兩個方法的定義和他是怎麼做的，上完課之後，稍微有一點概念，在做作業時，上網查了很多資料，才了解到他為什麼要這麼做，其實概念很簡單，實作起來也不難，簡單來說就是取 max 跟 min 而已。但是這只是 Morphology 的基礎，要做到其他事情，還是要搭配到其他的運算方式，才能做其他的應用。在做邊界抽取時，沒什麼問題，但是在做區域填充時，就因為不知道要怎麼找出起始點而卡住，跟同學討論也討論不出所以然，之後去問老師，老師說直接用小畫家隨便點的點就好!，最後就找一個點當作起始點去做，就順利的做出來。這次作業我也學到很多，最重要的就是 Morphology 的根本，dilation 和 erosion!