數位視訊技術

105323031 蔣圳嵐

原理:

Guassian smooth:利用 guassian distribution 做出一個 mask，再拿去和原圖像 convolution，就可以得到一個較模糊的照片，可以將圖像中的一些雜訊濾掉。

Edge detection:我是利用 Sobel 的方式，也同樣是把一個矩陣拿去和原圖像 convolution，而我會選擇 Sobel 的原因是，Sobel 縱橫的方向都可以同時取得。

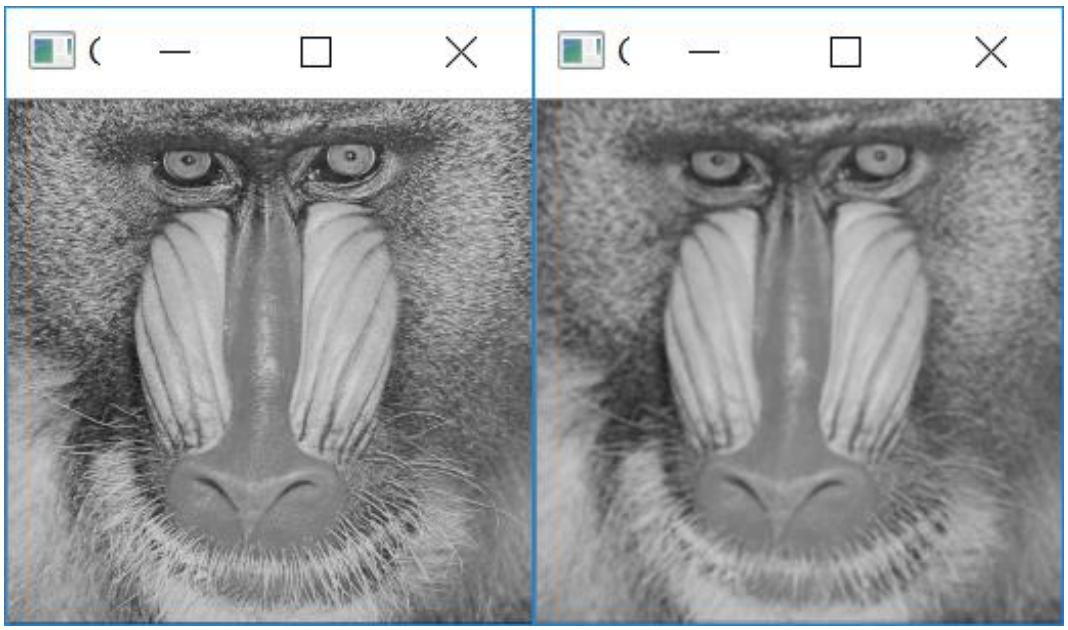
Opening and Closeing:首先要先明白 Dilation(擴張)和 erosion(侵蝕)，顧名思義前者會把原圖像變的肥胖，而後者則是淘汰掉小於輸入矩陣大小的影像，因此叫侵蝕。而 Opening 為先做 erosion 再做 dilation，Closing 為先做 dilation 再做 erosion。

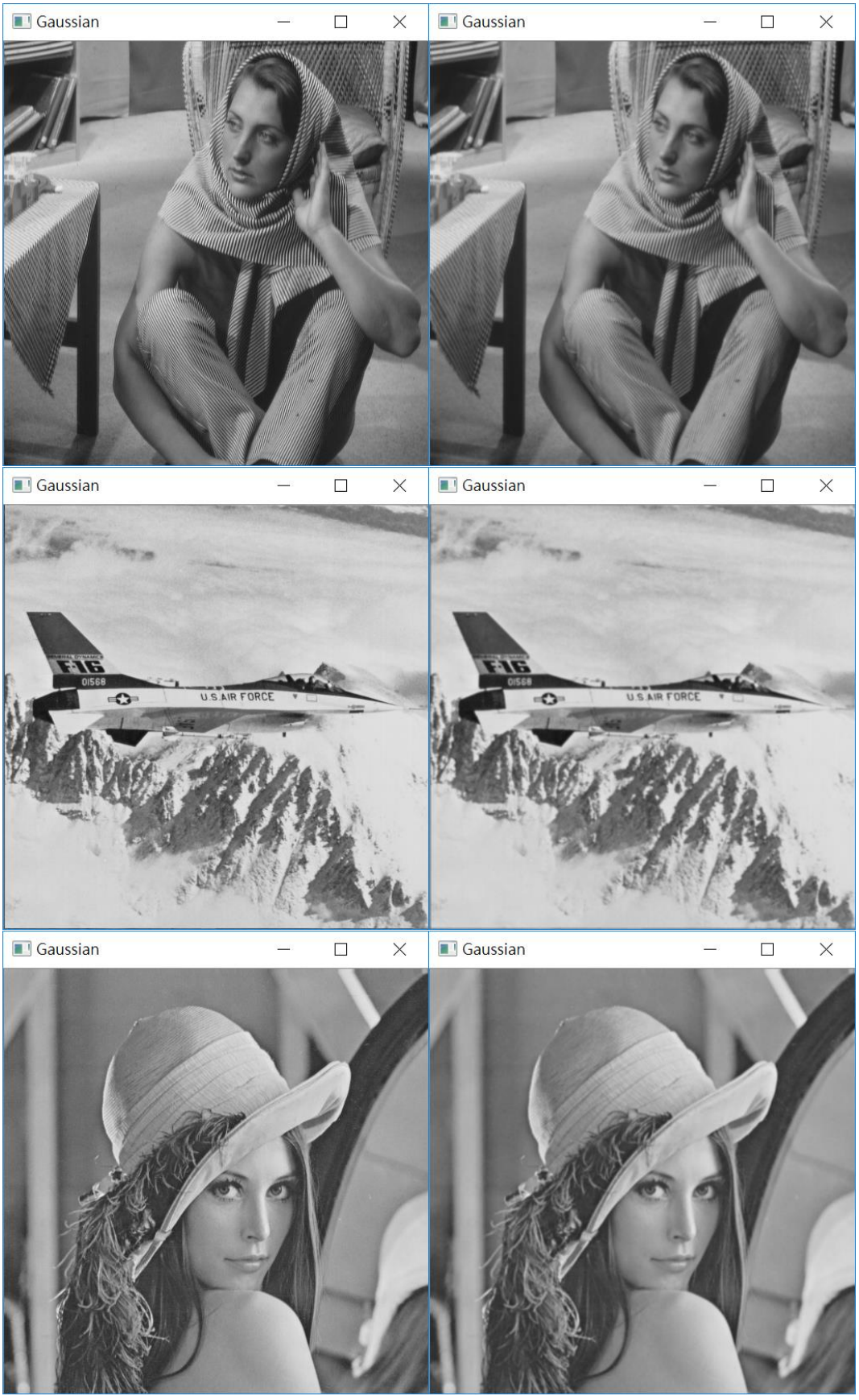
程式碼和結果:

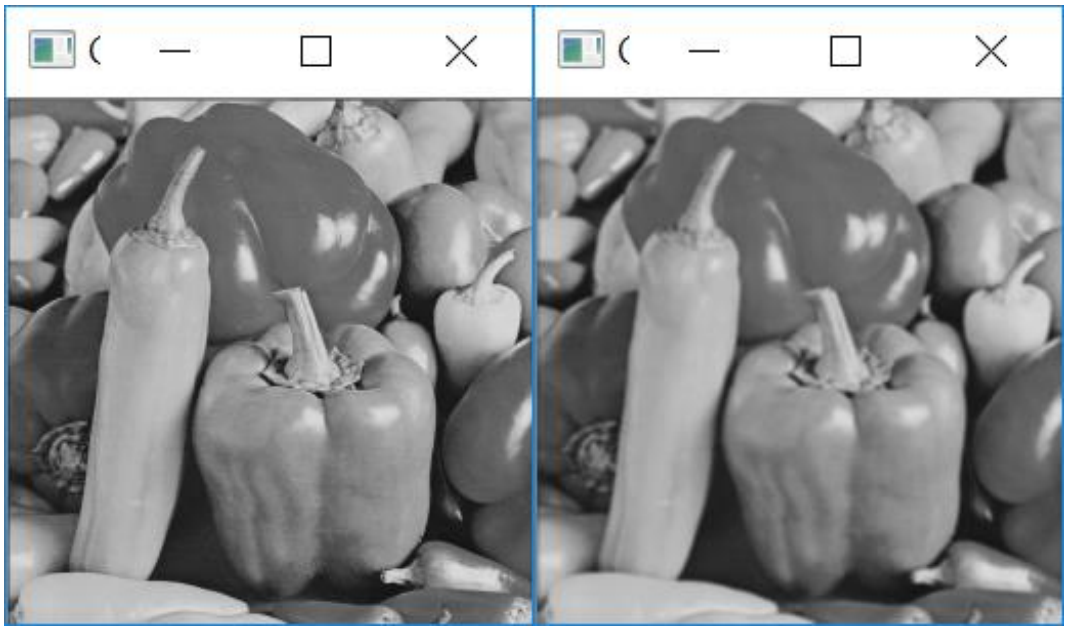
Guassian Smooth:

```python
import numpy as np
import cv2
a = cv2.imread("pepper256.bmp")
blur = cv2.GaussianBlur(a, (3, 3), 0)
cv2.imshow("Gaussian" , blur)
cv2.waitKey(0)
```

利用 opencv 內建的一個函式 GaussianBlur 就是 gussian distribution 的一個 mask，後面的參數分別為輸入圖像、mask 大小、顏色標準差，顏色標準差設為 0 則是讓程式自己運算合適的值，因此我設定為 0。



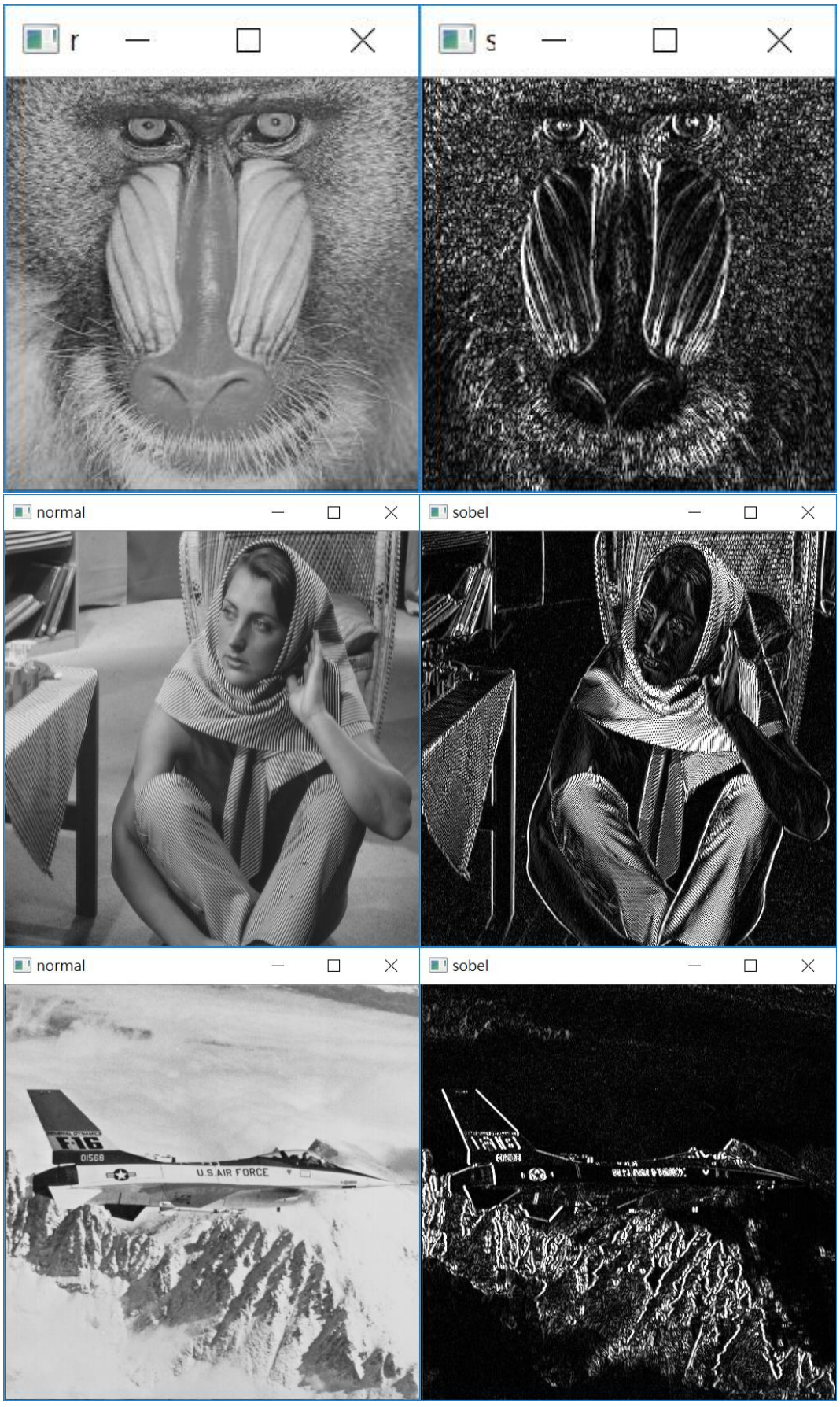

Gaussian
Gaussian
Gaussian
Gaussian
Gaussian
Gaussian
F16
U.S.AIR FORCE
F16
U.S.AIR FORCE

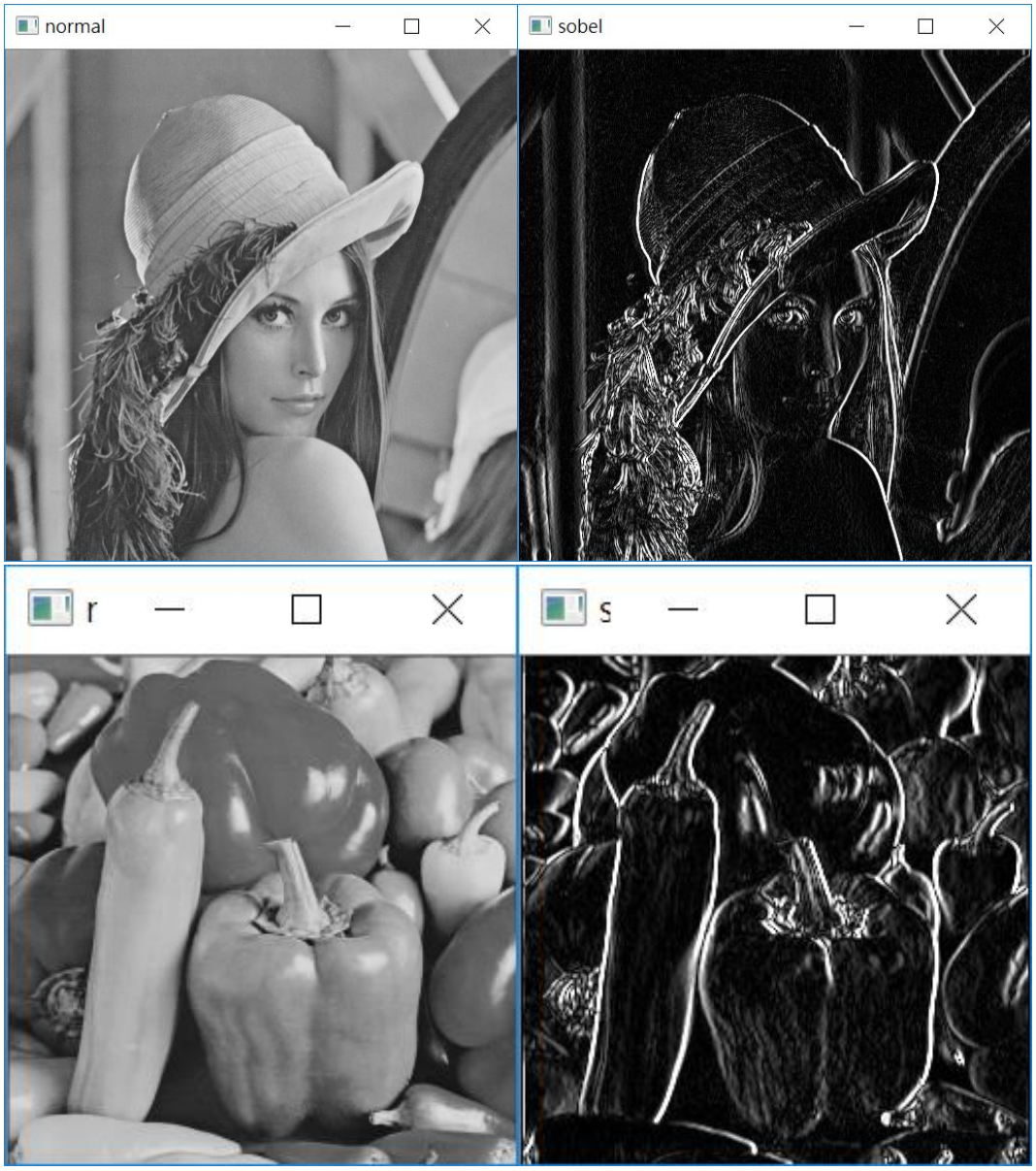


Edge detection:

```
1 import numpy as np
2 import cv2
3 a = cv2.imread("baboon256.bmp")
4 b = cv2.Sobel(a,cv2.CV_16S,1,0)
5 c = cv2.convertScaleAbs(b)
6 cv2.imshow("sobel" , c)
7 cv2.waitKey(0)
```

邊緣偵測我使用了 Sobel 這個方法，而指令則是使用 cv2.Sobel，他有四個參數分別為:輸入圖片、圖像深度、dx,dy 的求導皆數。第二個參數我使照片轉為 16 位元，因此最後要再使用 cv2.convertScaleAbs 這個函式將輸入圖像轉變成 8 位元，最後才能正常顯示。


r
s
normal
sobel
normal
sobel
F16
U.S.AIR FORCE

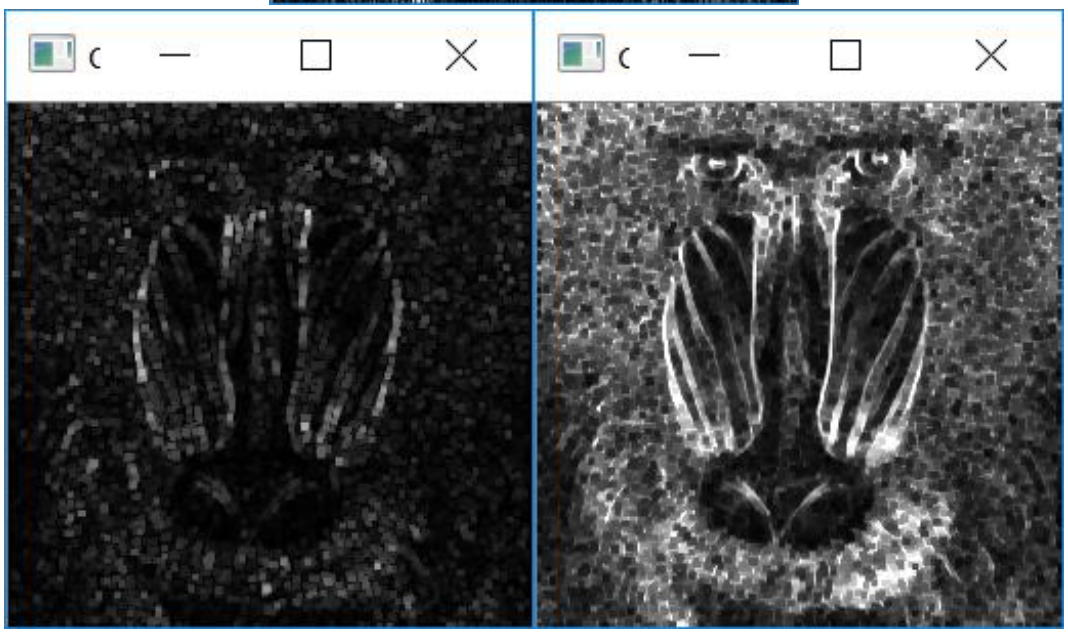


Opening and Closeing:

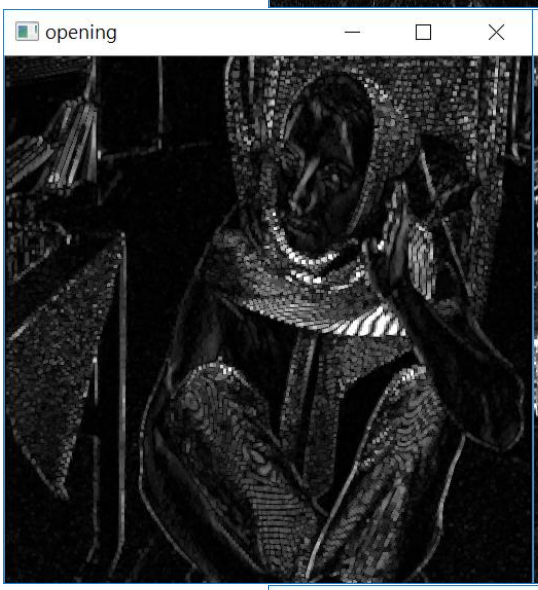
```
import numpy as np
import cv2
a = cv2.imread("baboon256.bmp")
b = cv2.Sobel(a,cv2.CV_16S,1,0)
c = cv2.convertScaleAbs(b)
kernel = np.ones((3,3),np.uint8)
opening = cv2.morphologyEx(c, cv2.MORPH_OPEN , kernel)
cv2.imshow("opening" , opening)
cv2.waitKey(0)
```

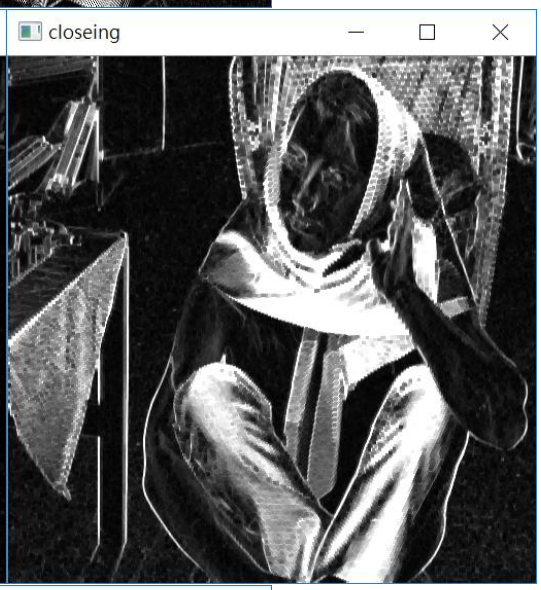
```python
import numpy as np
import cv2
a = cv2.imread("baboon256.bmp")
b = cv2.Sobel(a,cv2.CV_16S,1,0)
c = cv2.convertScaleAbs(b)
kernel = np.ones((3,3),np.uint8)
closeing = cv2.morphologyEx(c, cv2.MORPH_CLOSE , kernel)
cv2.imshow("closeing" , closeing)
cv2.waitKey(0)
```

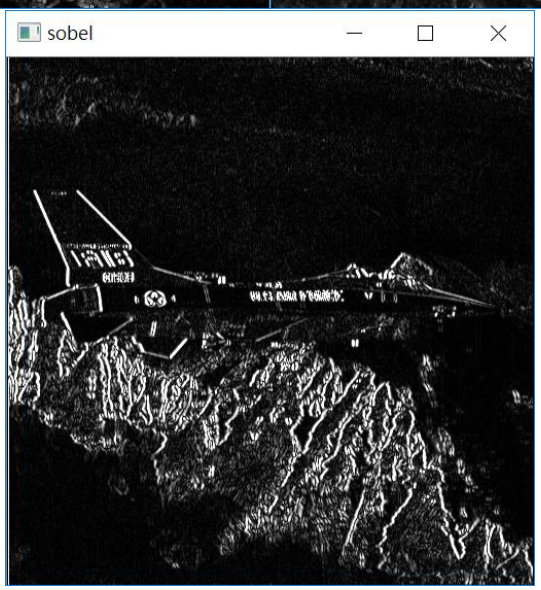
opening 和 closeing 則是拿前面做好 Sobel 的圖像再宣告一個 3x3 的矩陣叫做 kernel，然後再使用 cv2.morphologyEx 的函式，他也一樣有三個參數:輸入圖像、要做 OPEN 還是 CLOSE、矩陣大小。



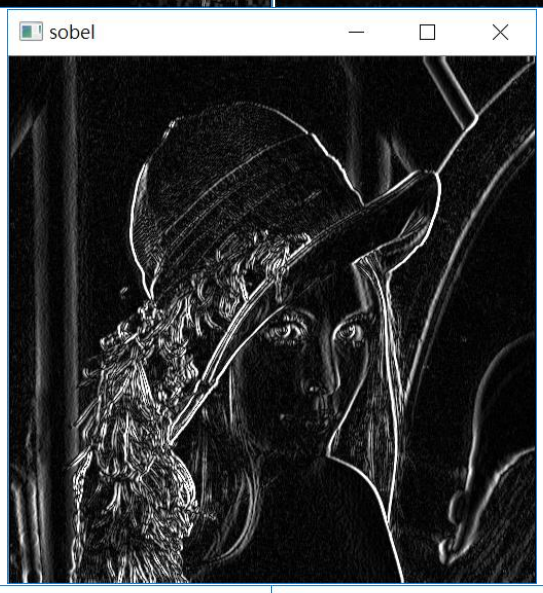

sobel

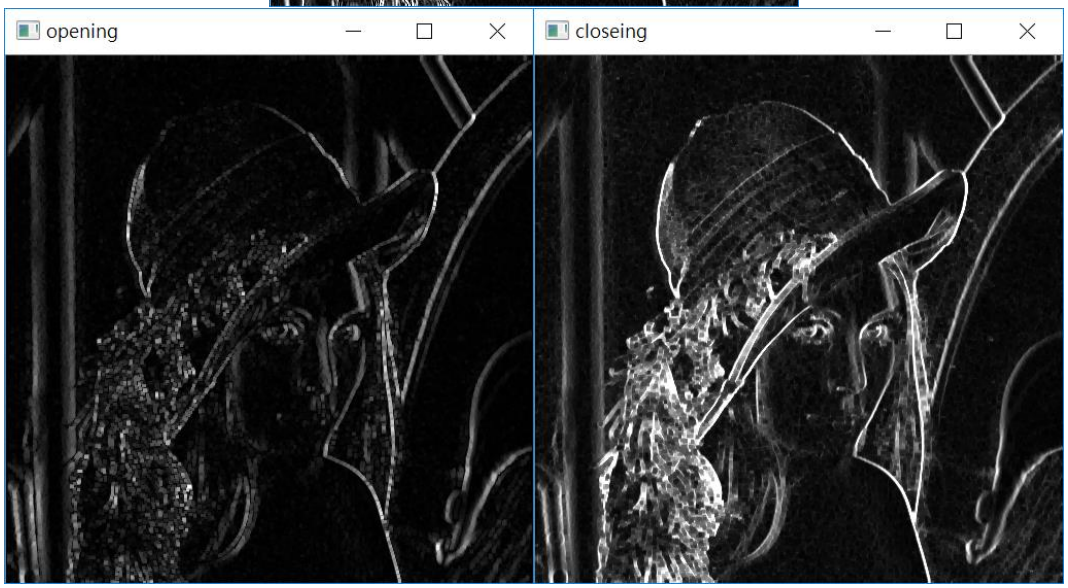

opening

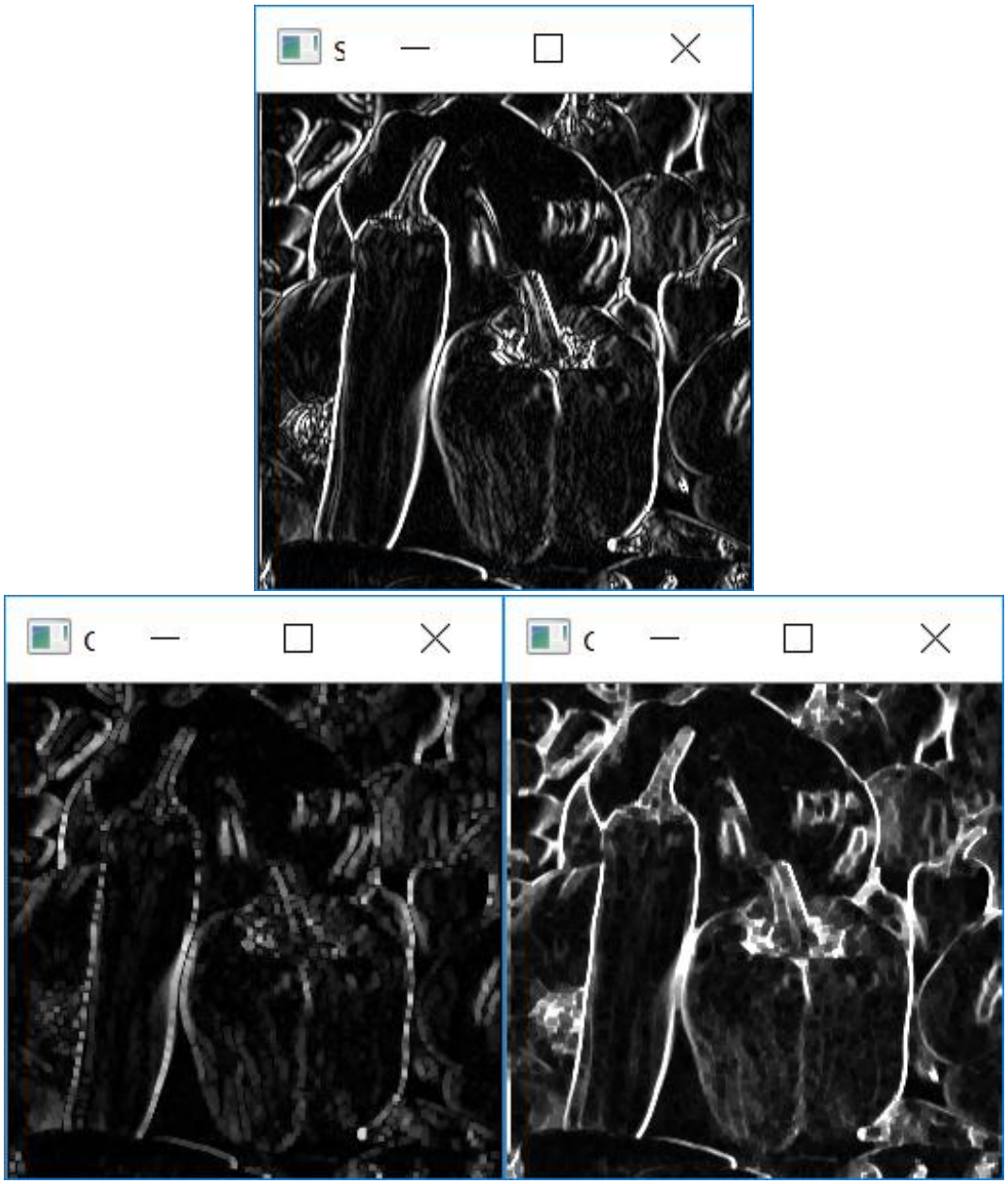

closeing


sobel


opening
closeing


sobel


opening
closeing


s
c
c

結論:

Guassian smooth:因為我沒有在圖像中加入雜訊，因此 Guassian smooth 的效果沒那麼顯著，只是單純把圖像做模糊的狀況。但這個也告訴我們世界上沒有一個事物是完美的，要濾雜訊就必須犧牲圖像的清晰度，是一個雙面刃。

Edge detection:邊緣偵測是我認為做最有感的一個，因為之前我總是很好奇邊緣到底是如何找到的，沒想到只要設計好矩陣再拿去和圖像 convolution 就可以得到了。這樣以後可以拿這些處理過後的圖片來 train 就可以讓神經網路學習我們想要他們注意的地方。

Opening and Closeing:Opening 的效果不是很顯著，通常都會讓影像的邊緣變得更糊，我想這個的應用應該是把影像內部一些小小的邊緣都模糊掉，只留下邊緣較粗的紋路。而 Closeing 就是把整個的邊緣紋路再更加深，讓整個效果更明顯更好辨識。