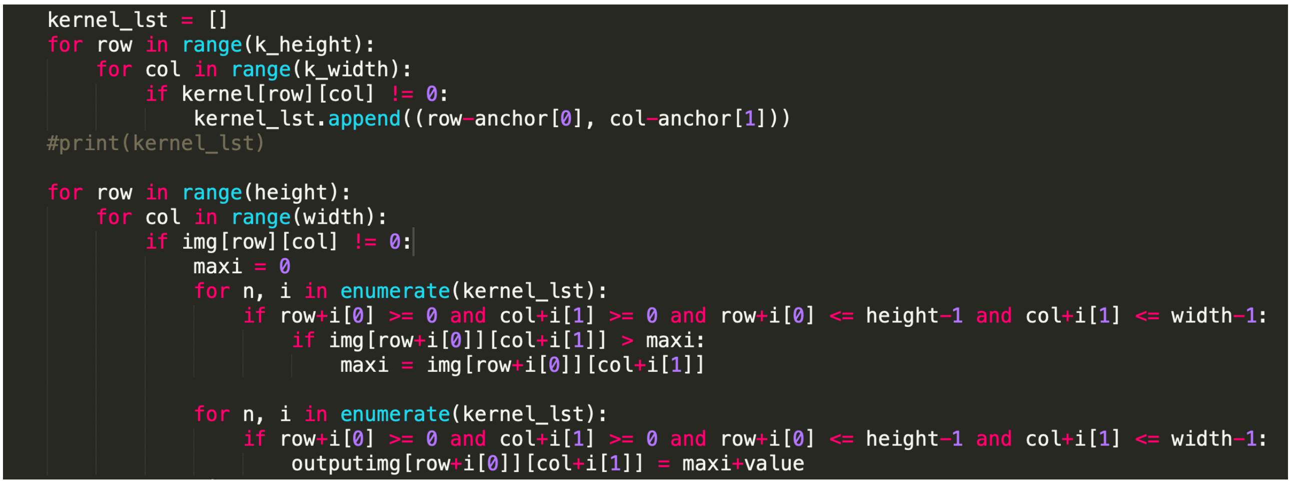
Computer Vision 2018Fall
HW05
電信所碩一 謝硯澤 R07942115

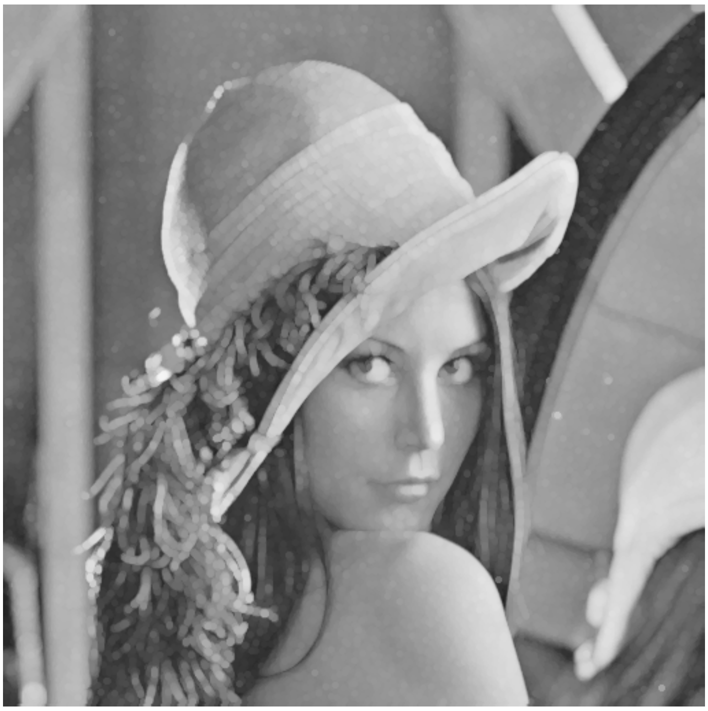
使用環境說明：

```
#macOS Majave 10.14
#Python 3.7.0
#openCV 3.4.2
#PIL 5.2.0
#matplotlib 3.0.0
```

(a)Gray scale Dilation

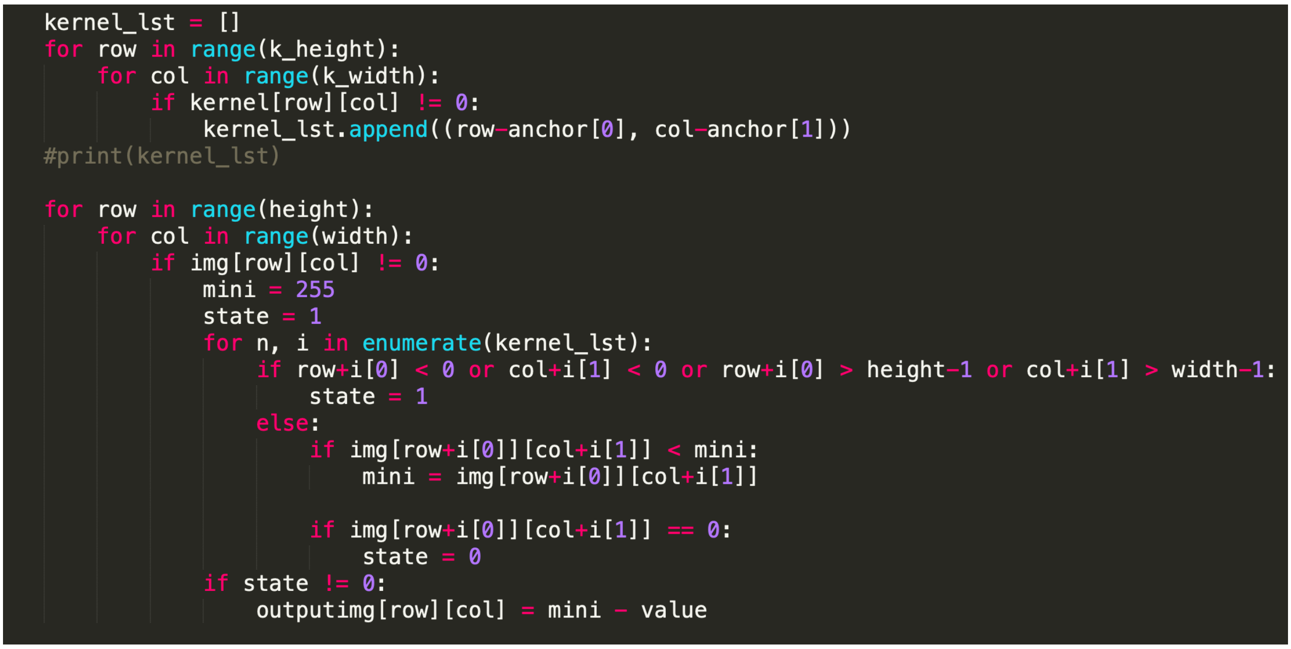


```python
kernel_lst = []
for row in range(k_height):
    for col in range(k_width):
        if kernel[row][col] != 0:
            kernel_lst.append((row-anchor[0], col-anchor[1]))
#print(kernel_lst)

for row in range(height):
    for col in range(width):
        if img[row][col] != 0:
            maxi = 0
            for n, i in enumerate(kernel_lst):
                if row+i[0] >= 0 and col+i[1] >= 0 and row+i[0] <= height-1 and col+i[1] <= width-1:
                    if img[row+i[0]][col+i[1]] > maxi:
                        maxi = img[row+i[0]][col+i[1]]

            for n, i in enumerate(kernel_lst):
                if row+i[0] >= 0 and col+i[1] >= 0 and row+i[0] <= height-1 and col+i[1] <= width-1:
                    outputimg[row+i[0]][col+i[1]] = maxi+value
```

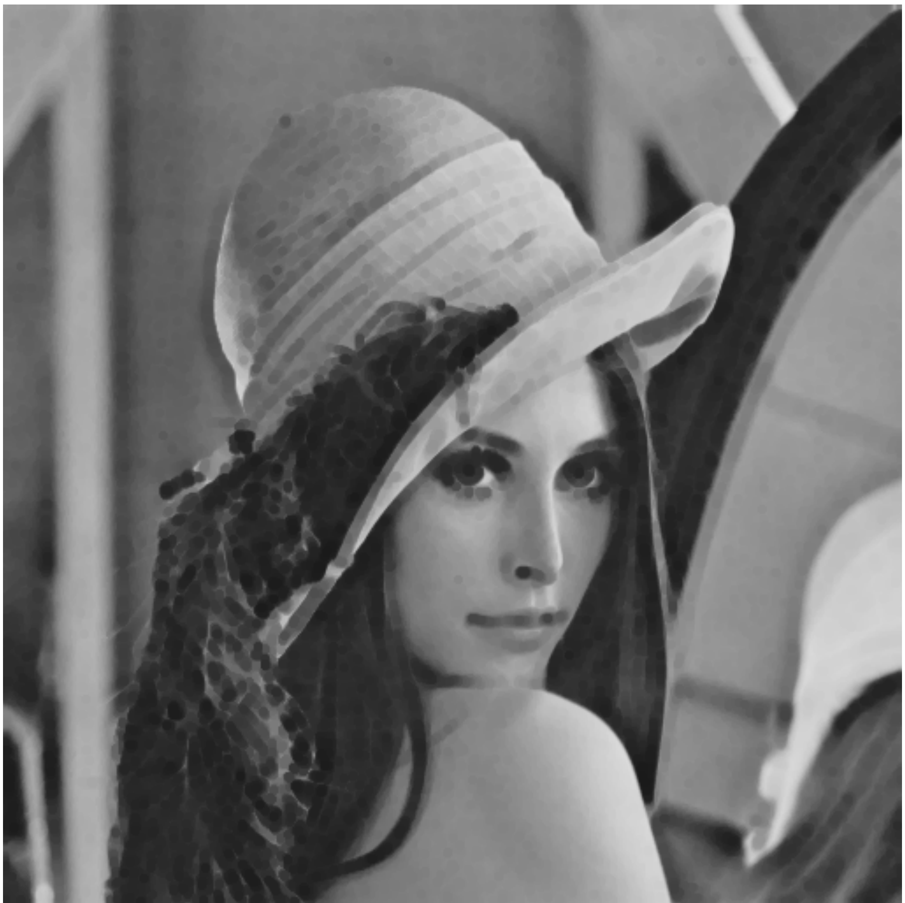


Dilation:先去看kernel中哪些位子的值為1，把它存成一個陣列，方便讀取。再把每一個像素掃過，當中心像素有值時，把kernel映射上去，去看每一個周圍像素的值，並根據ppt中的演算法做更改。

(b)Gray scale Erosion

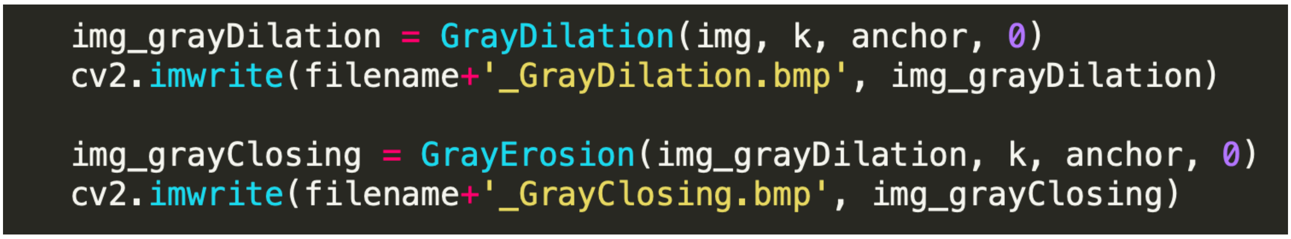


```
kernel_lst = []
for row in range(k_height):
    for col in range(k_width):
        if kernel[row][col] != 0:
            kernel_lst.append((row-anchor[0], col-anchor[1]))
#print(kernel_lst)

for row in range(height):
    for col in range(width):
        if img[row][col] != 0:
            mini = 255
            state = 1
            for n, i in enumerate(kernel_lst):
                if row+i[0] < 0 or col+i[1] < 0 or row+i[0] > height-1 or col+i[1] > width-1:
                    state = 1
                else:
                    if img[row+i[0]][col+i[1]] < mini:
                        mini = img[row+i[0]][col+i[1]]

                    if img[row+i[0]][col+i[1]] == 0:
                        state = 0
            if state != 0:
                outputimg[row][col] = mini - value
```

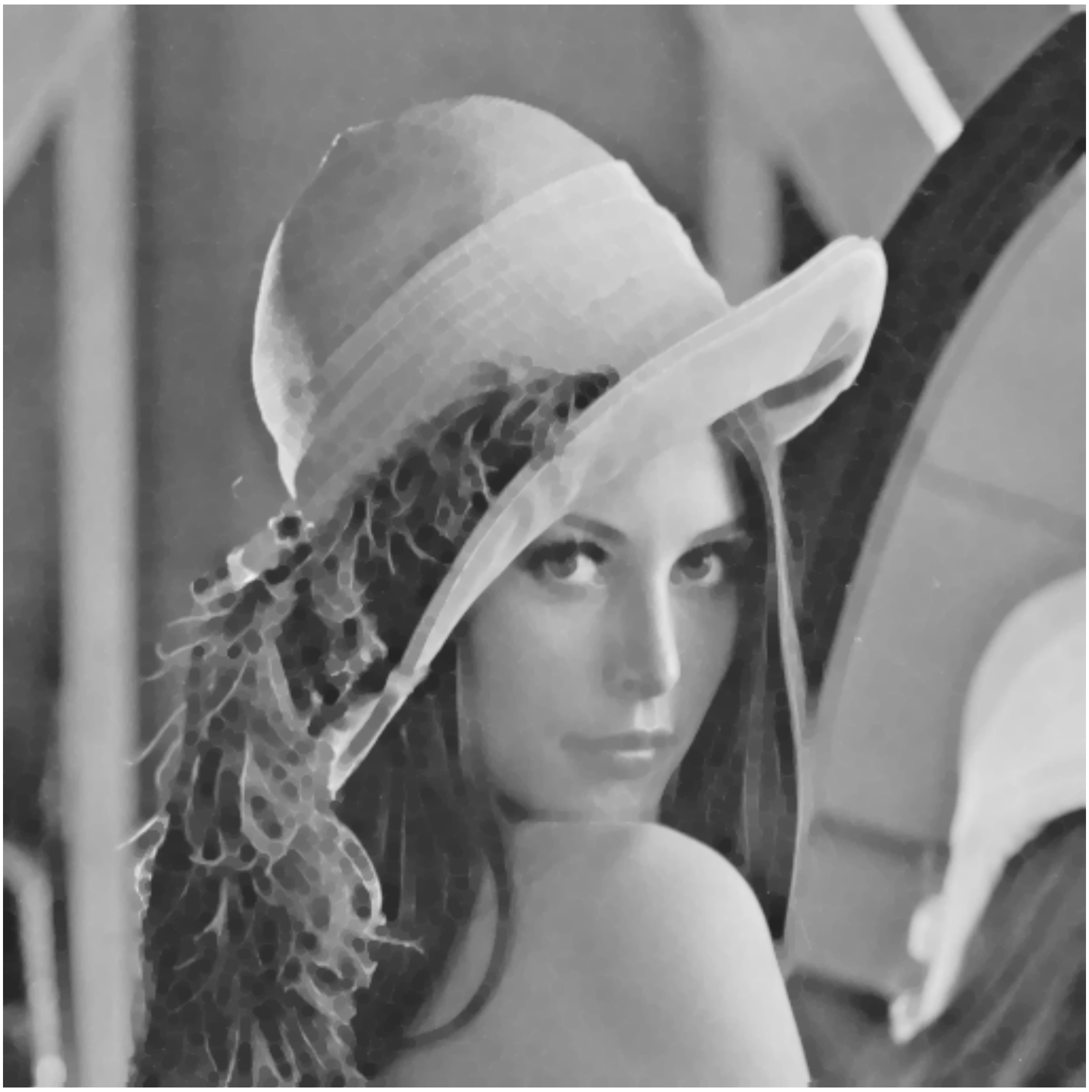


Erosion:先去看kernel中哪些位子的值為1，把它存成一個陣列，方便讀取。再把每一個像素掃過，當中心像素有值時，把kernel映射上去，如果對應的像素值為零的話，直接輸出該點像素為0，再根據ppt的演算法去做計算。

(c)Gray scale Closing

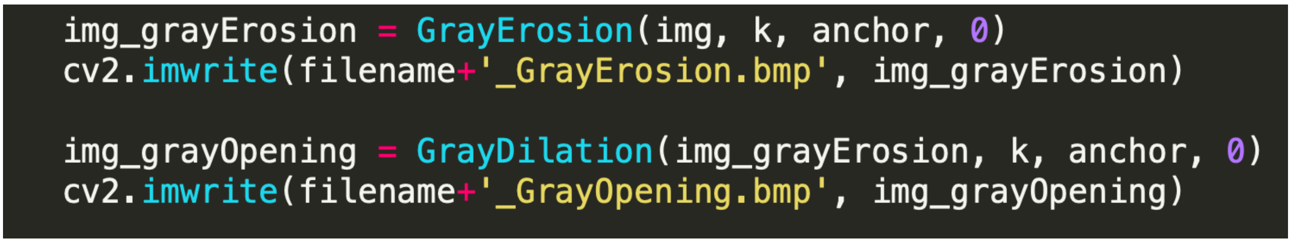


```
    img_grayDilation = GrayDilation(img, k, anchor, 0)
    cv2.imwrite(filename+'_GrayDilation.bmp', img_grayDilation)

    img_grayClosing = GrayErosion(img_grayDilation, k, anchor, 0)
    cv2.imwrite(filename+'_GrayClosing.bmp', img_grayClosing)
```

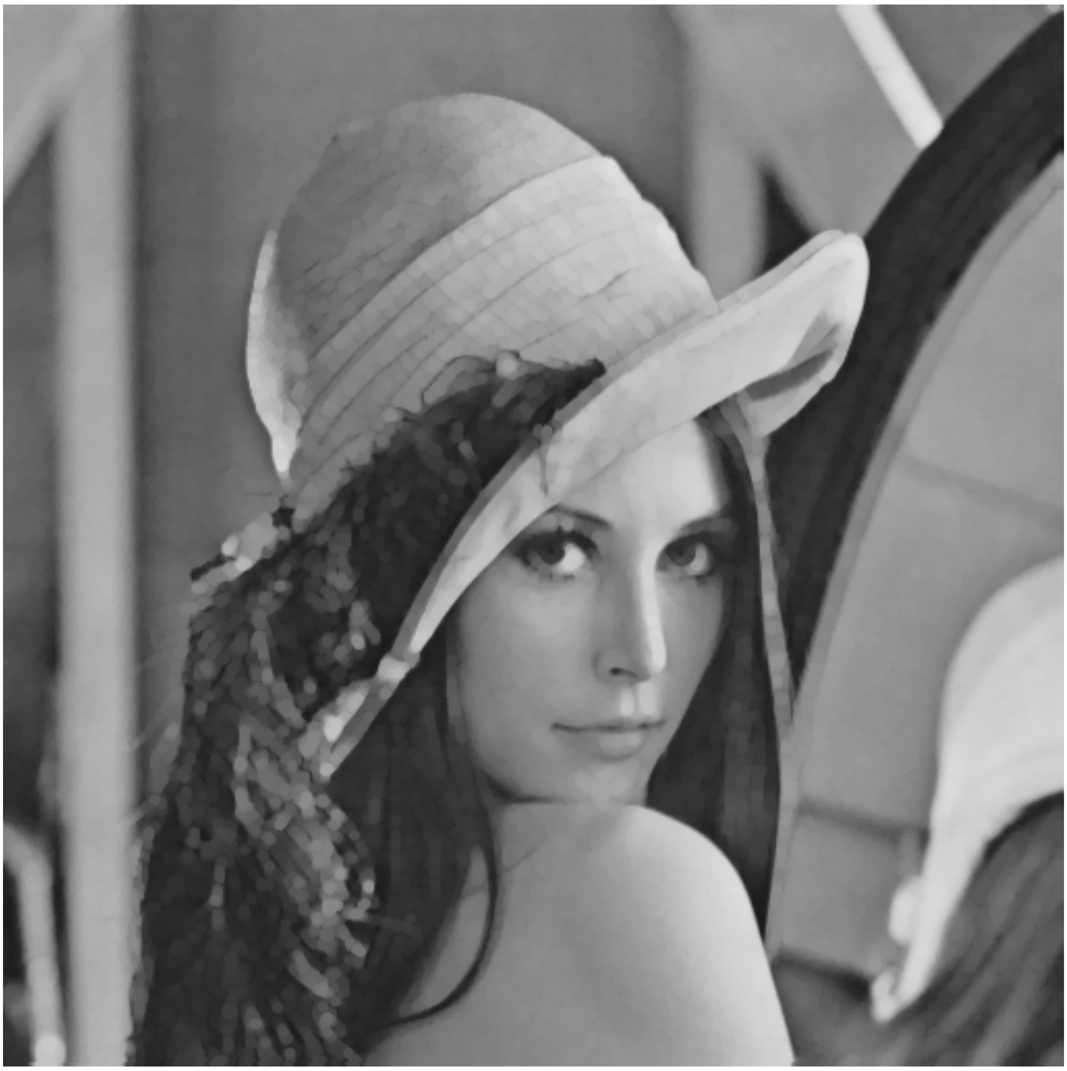


Gray scale Closing：先做Dilation再做Erosion

(d)Gray scale Opening



```
img_grayErosion = GrayErosion(img, k, anchor, 0)
cv2.imwrite(filename+'_GrayErosion.bmp', img_grayErosion)

img_grayOpening = GrayDilation(img_grayErosion, k, anchor, 0)
cv2.imwrite(filename+'_GrayOpening.bmp', img_grayOpening)
```



Gray scale Opening：先做Erosion再做Dilation