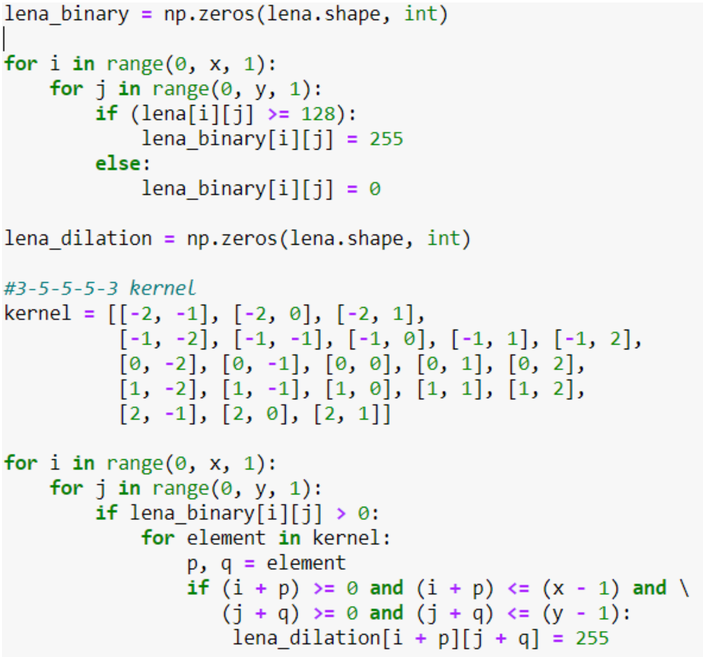
Computer Vision HW4

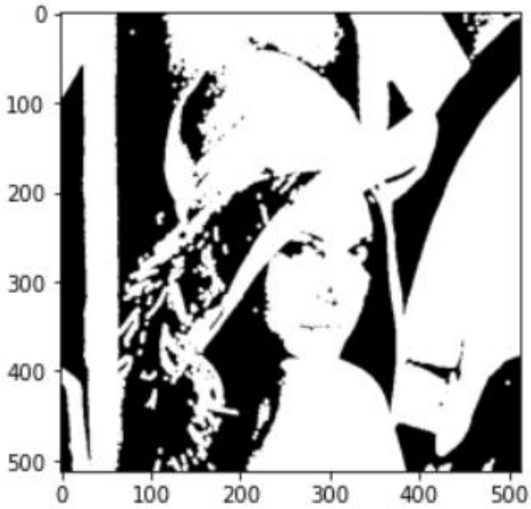
1.

先變成 binary 然後跟八角形比

$$A \oplus B = \{c \in E^N \mid c = a + b \text{ for some } a \in A \text{ and } b \in B\}$$



```
lena_binary = np.zeros(lena.shape, int)

for i in range(0, x, 1):
    for j in range(0, y, 1):
        if (lena[i][j] >= 128):
            lena_binary[i][j] = 255
        else:
            lena_binary[i][j] = 0

lena_dilation = np.zeros(lena.shape, int)

#3-5-5-5-3 kernel
kernel = [[-2, -1], [-2, 0], [-2, 1],
          [-1, -2], [-1, -1], [-1, 0], [-1, 1], [-1, 2],
          [0, -2], [0, -1], [0, 0], [0, 1], [0, 2],
          [1, -2], [1, -1], [1, 0], [1, 1], [1, 2],
          [2, -1], [2, 0], [2, 1]]

for i in range(0, x, 1):
    for j in range(0, y, 1):
        if lena_binary[i][j] > 0:
            for element in kernel:
                p, q = element
                if (i + p) >= 0 and (i + p) <= (x - 1) and \
                   (j + q) >= 0 and (j + q) <= (y - 1):
                    lena_dilation[i + p][j + q] = 255
```



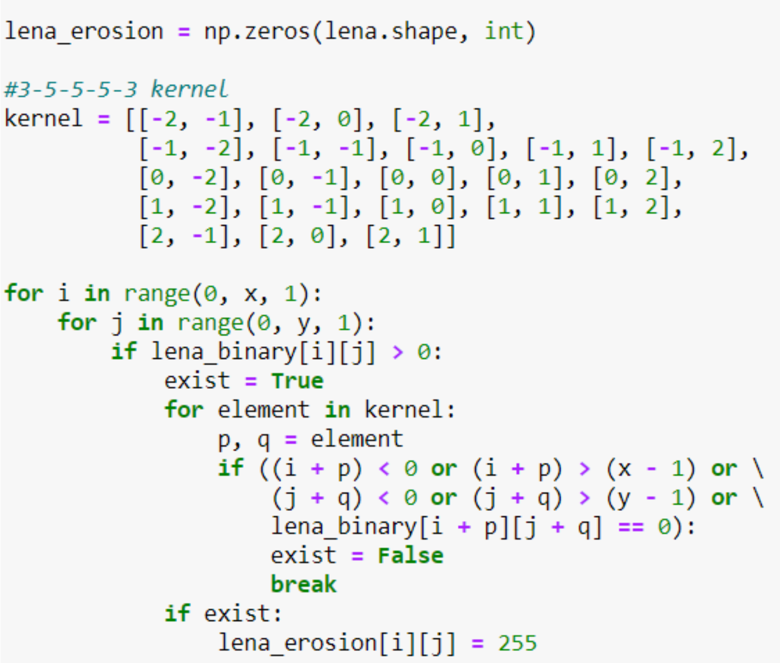
2.

定義 erosion 函數，erosion 函數是指將 kernel 與二值圖的位置相

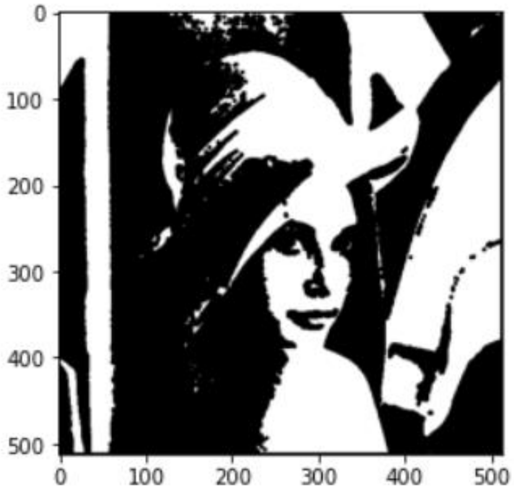
加，如二值圖的位置的值是 255，符合這些條件，與 kernel 相加後

周圍都還在矩陣內，且周圍的值都是 255，那此位置的值為 255。

$$A \ominus B = \{x \in E^N | x + b \in A \text{ for every } b \in B\}$$



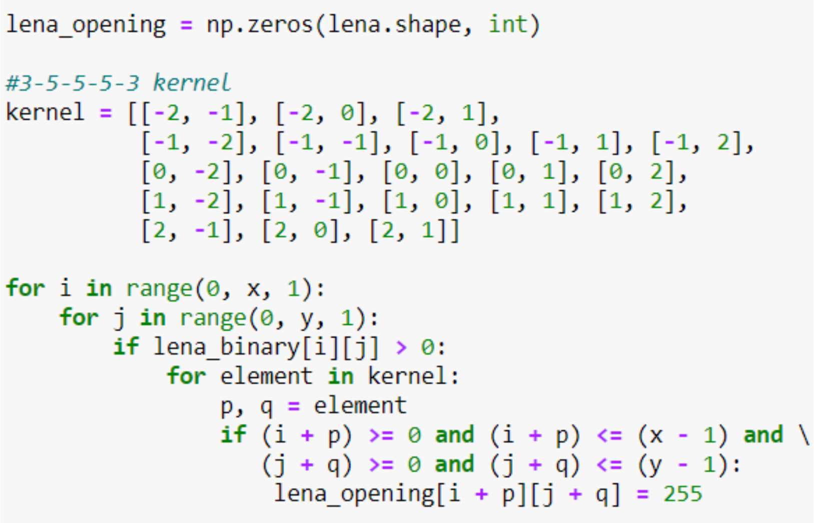
```python
lena_erosion = np.zeros(lena.shape, int)

#3-5-5-5-3 kernel
kernel = [[-2, -1], [-2, 0], [-2, 1],
          [-1, -2], [-1, -1], [-1, 0], [-1, 1], [-1, 2],
          [0, -2], [0, -1], [0, 0], [0, 1], [0, 2],
          [1, -2], [1, -1], [1, 0], [1, 1], [1, 2],
          [2, -1], [2, 0], [2, 1]]

for i in range(0, x, 1):
    for j in range(0, y, 1):
        if lena_binary[i][j] > 0:
            exist = True
            for element in kernel:
                p, q = element
                if ((i + p) < 0 or (i + p) > (x - 1) or \
                    (j + q) < 0 or (j + q) > (y - 1) or \
                    lena_binary[i + p][j + q] == 0):
                    exist = False
                    break
            if exist:
                lena_erosion[i][j] = 255
```



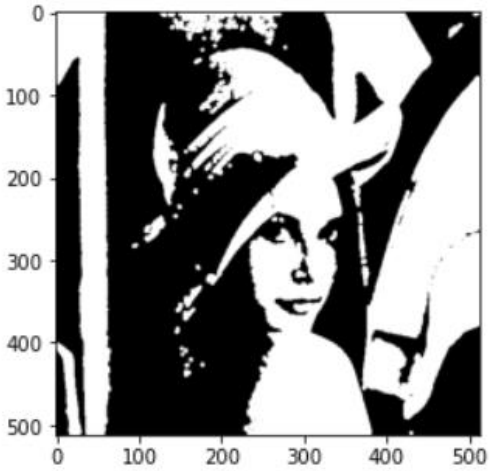
3 .

先縮小再擴大

dilation(erision(lena_binary, kernal))

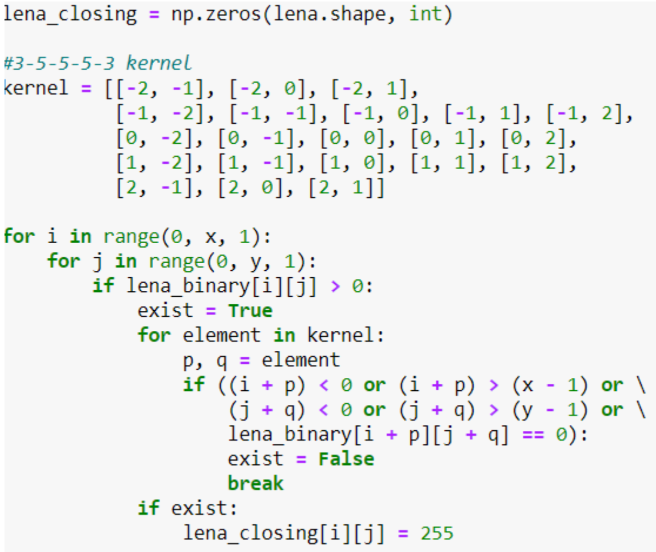


```
lena_opening = np.zeros(lena.shape, int)

#3-5-5-5-3 kernel
kernel = [[-2, -1], [-2, 0], [-2, 1],
          [-1, -2], [-1, -1], [-1, 0], [-1, 1], [-1, 2],
          [0, -2], [0, -1], [0, 0], [0, 1], [0, 2],
          [1, -2], [1, -1], [1, 0], [1, 1], [1, 2],
          [2, -1], [2, 0], [2, 1]]

for i in range(0, x, 1):
    for j in range(0, y, 1):
        if lena_binary[i][j] > 0:
            for element in kernel:
                p, q = element
                if (i + p) >= 0 and (i + p) <= (x - 1) and \
                   (j + q) >= 0 and (j + q) <= (y - 1):
                    lena_opening[i + p][j + q] = 255
```

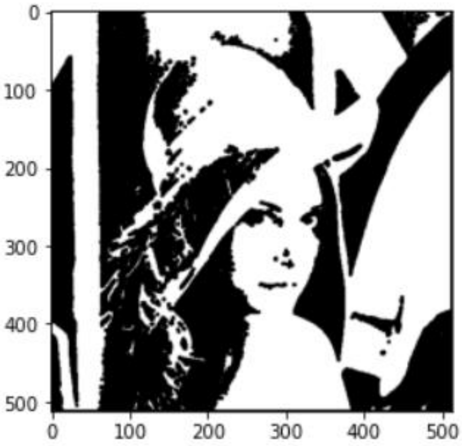


4． 先擴大再縮小

erosion(dilation(lena_binary, kernel))

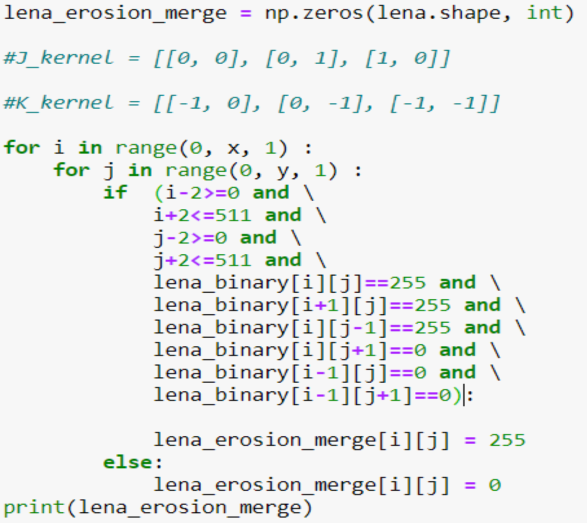


```
lena_closing = np.zeros(lena.shape, int)

#3-5-5-5-3 kernel
kernel = [[-2, -1], [-2, 0], [-2, 1],
          [-1, -2], [-1, -1], [-1, 0], [-1, 1], [-1, 2],
          [0, -2], [0, -1], [0, 0], [0, 1], [0, 2],
          [1, -2], [1, -1], [1, 0], [1, 1], [1, 2],
          [2, -1], [2, 0], [2, 1]]

for i in range(0, x, 1):
    for j in range(0, y, 1):
        if lena_binary[i][j] > 0:
            exist = True
            for element in kernel:
                p, q = element
                if ((i + p) < 0 or (i + p) > (x - 1) or \
                    (j + q) < 0 or (j + q) > (y - 1) or \
                    lena_binary[i + p][j + q] == 0):
                    exist = False
                    break
            if exist:
                lena_closing[i][j] = 255
```



5.點出 j-kernel 與 k-kernel 的設為 255 其餘為 0，

$$A \otimes (J, K) = (A \ominus J) \cap (A^c \ominus K)$$



```
lena_erosion_merge = np.zeros(lena.shape, int)

#J_kernel = [[0, 0], [0, 1], [1, 0]]

#K_kernel = [[-1, 0], [0, -1], [-1, -1]]

for i in range(0, x, 1) :
    for j in range(0, y, 1) :
        if  (i-2>=0 and \
            i+2<=511 and \
            j-2>=0 and \
            j+2<=511 and \
            lena_binary[i][j]==255 and \
            lena_binary[i+1][j]==255 and \
            lena_binary[i][j-1]==255 and \
            lena_binary[i][j+1]==0 and \
            lena_binary[i-1][j]==0 and \
            lena_binary[i-1][j+1]==0):

            lena_erosion_merge[i][j] = 255
        else:
            lena_erosion_merge[i][j] = 0
print(lena_erosion_merge)
```