

HW5

- **A discription of your homework**

Programming language used: Python 2.7

Library used: Numpy, PIL, Scipy.misc

- **Your parameters**

i: row

j: column

tem: 用於儲存每個像素的灰階數值 0~255

tem_blk: lena 圖片外增加黑色像素的邊界，邊界寬度為 2 pixels

tem_wht: lena 圖片外增加白色像素的邊界，邊界寬度為 2 pixels

kernel: octogonal 3-5-5-5-3 kernel

maxi: kernel 區域內灰階數值最大者

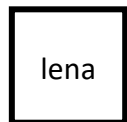
mini: kernel 區域內灰階數值最小者

- **Function**

bdry: enlarge the 512*512 lena by 4 pixels of width and length
(為計算 dilation 和 erosion 時能處理邊界的像素)

dil: dilation

ero: erosion



- **The algorithm you used**

1. Dilation

kernel 區域內灰階數值最大者存入該像素。

2. Erosion

kernel 區域內灰階數值最小者存入該像素。

3. Opening

$$B \circ K = (B \ominus K) \oplus K$$

先做 erosion，再做 dilation：把 erosion 後的結果丟到 dilation 跑一遍。

4. Closing

$$B \cdot K = (B \oplus K) \ominus K$$

先做 dilation，再做 erosion：把 dilation 後的結果丟到 erosion 跑一遍。

- Principal code fragment

```
def bdry(x):
    tem_blk = np.zeros([516,516])
    tem_wht = np.full([516,516],255)
    for i in range(x.shape[0]):
        for j in range(x.shape[1]):
            tem_blk[i+2][j+2] = x[i][j]
            tem_wht[i+2][j+2] = x[i][j]
    return tem_blk, tem_wht

kernel = []
for k in range(-2,3,1):
    for m in range(-2,3,1):
        if (k!=-2 or m!=-2) and (k!=2 or m!=2) and (k!=2 or m!=2) and (k!=2 or m!=2):
            kernel.append([k,m])

def dil(x):
    tem = np.zeros([512,512])
    for i in range(2,x.shape[0]-2,1):
        for j in range(2,x.shape[1]-2,1):
            maxi = 0
            for k in range(len(kernel)):
                if x[i+kernel[k][0]][j+kernel[k][1]]> maxi:
                    maxi = x[i+kernel[k][0]][j+kernel[k][1]]
            tem[i-2][j-2] = maxi
    print(tem.shape)
    return tem

def ero(x):
    tem = np.zeros([512,512])
    for i in range(2,x.shape[0]-2,1):
        for j in range(2,x.shape[1]-2,1):
            mini = 255
            for k in range(len(kernel)):
                if x[i+kernel[k][0]][j+kernel[k][1]]< mini:
                    mini = x[i+kernel[k][0]][j+kernel[k][1]]
            tem[i-2][j-2] = mini
    return tem
```

- **Resulting images**

1. Dilation



2. Erosion



3. Opening



4. Closing

