Computer Vision HW5

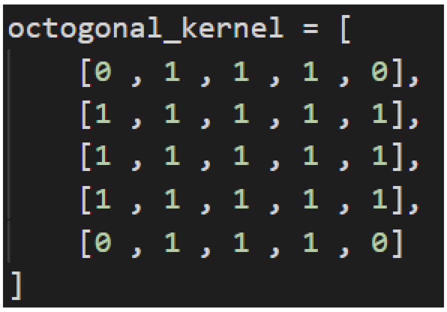
R04921040 李中原

Language: Python3

Description: I use OpenCV to do image I/O, and load raw pixel data (height , length , channels) from lena.bmp.

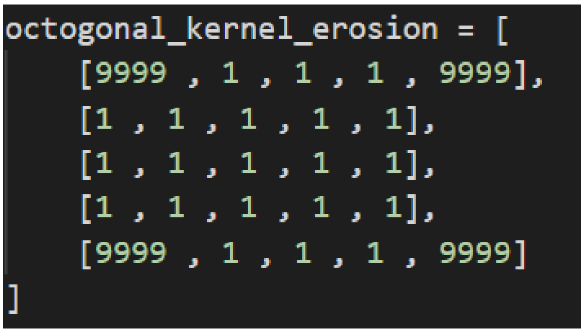
Run: python hw5.py

Kernel:



```
octogonal_kernel = [
    [0 , 1 , 1 , 1 , 0],
    [1 , 1 , 1 , 1 , 1],
    [1 , 1 , 1 , 1 , 1],
    [1 , 1 , 1 , 1 , 1],
    [0 , 1 , 1 , 1 , 0]
]
```

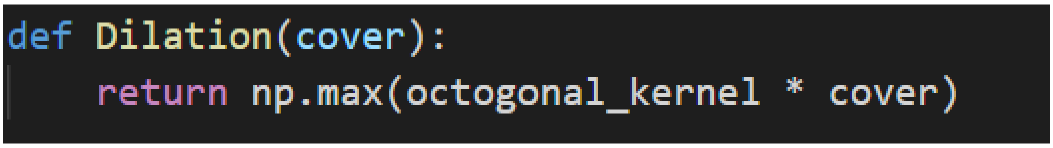
Erosion 所用的 kernel:



```
octogonal_kernel_erosion = [
    [9999 , 1 , 1 , 1 , 9999],
    [1 , 1 , 1 , 1 , 1],
    [1 , 1 , 1 , 1 , 1],
    [1 , 1 , 1 , 1 , 1],
    [9999 , 1 , 1 , 1 , 9999]
]
```

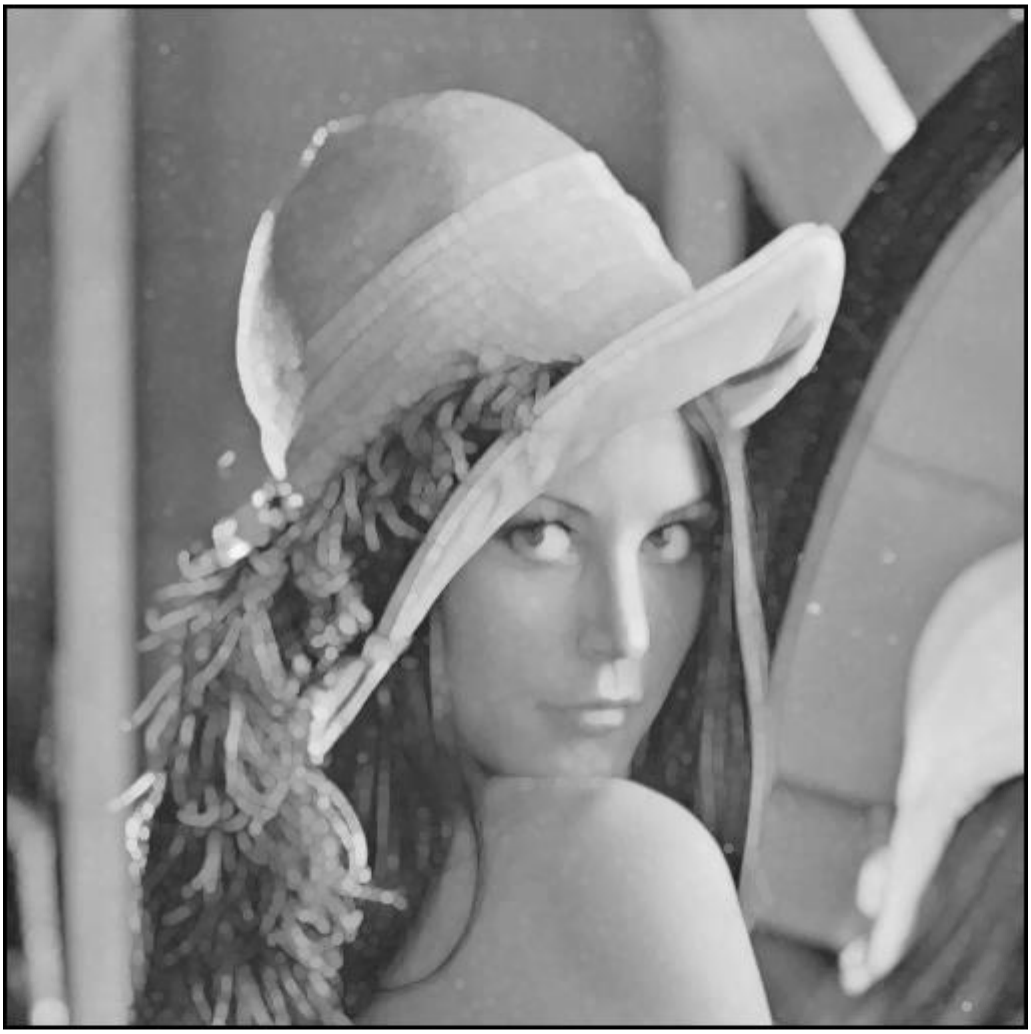
四個角設為 9999 是為了方便計算 local min 所使用

Dilation:

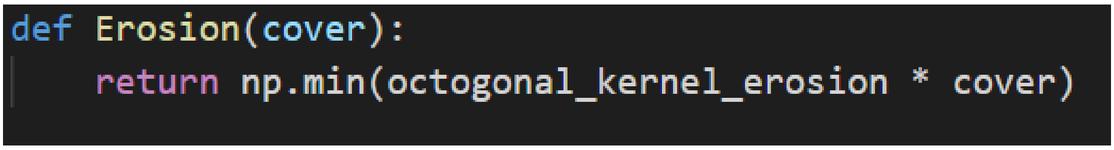


```
def Dilation(cover):
    return np.max(octogonal_kernel * cover)
```

每個點的值是將周遭5*5的範圍的點傳入Dilation做判斷，回傳對應kernel形狀中的max值

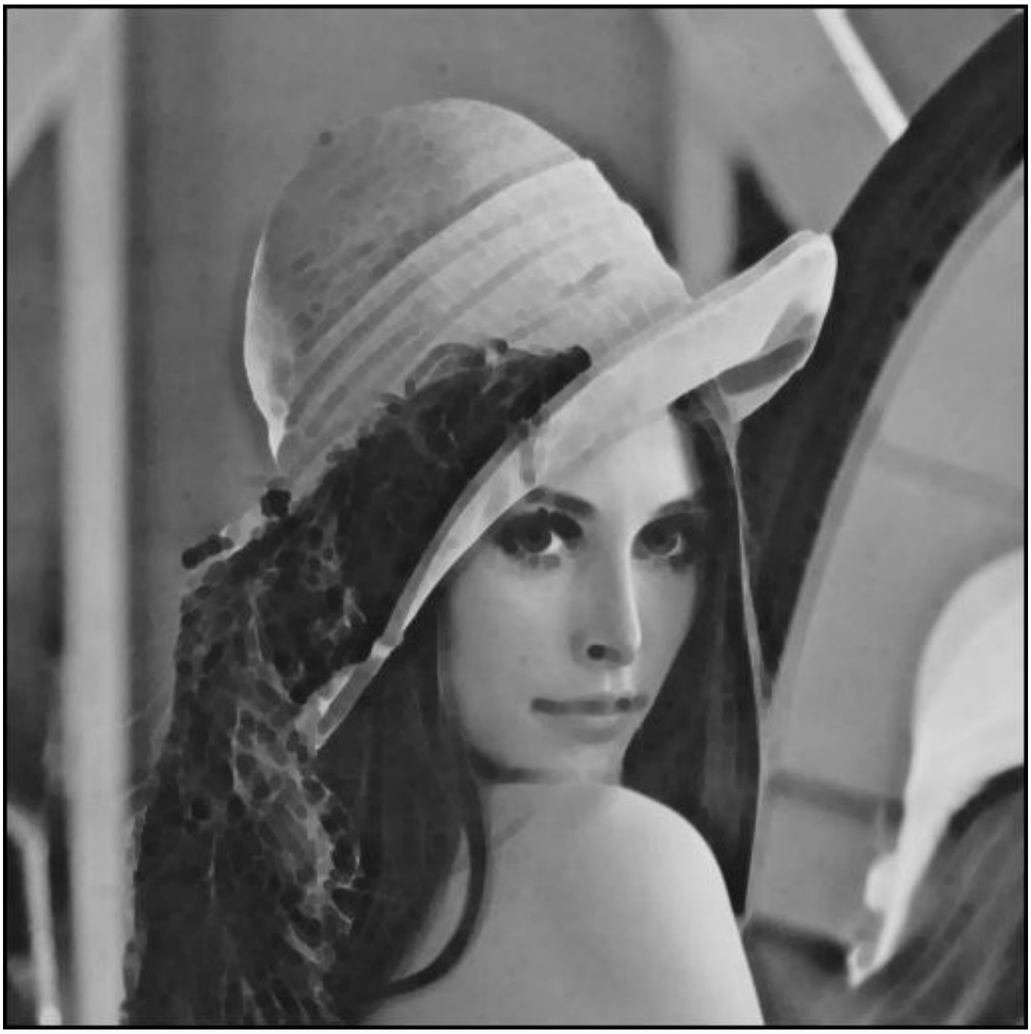


Erosion:

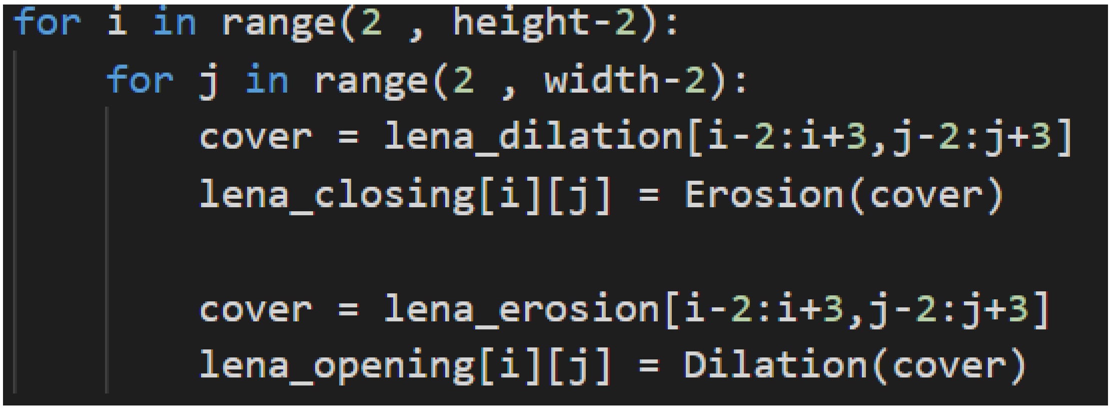


```python
def Erosion(cover):
    return np.min(octogonal_kernel_erosion * cover)
```

跟 Dilation 概念類似，不過改為回傳在 kernel 覆蓋範圍中的 min 值



Opening、Closing:

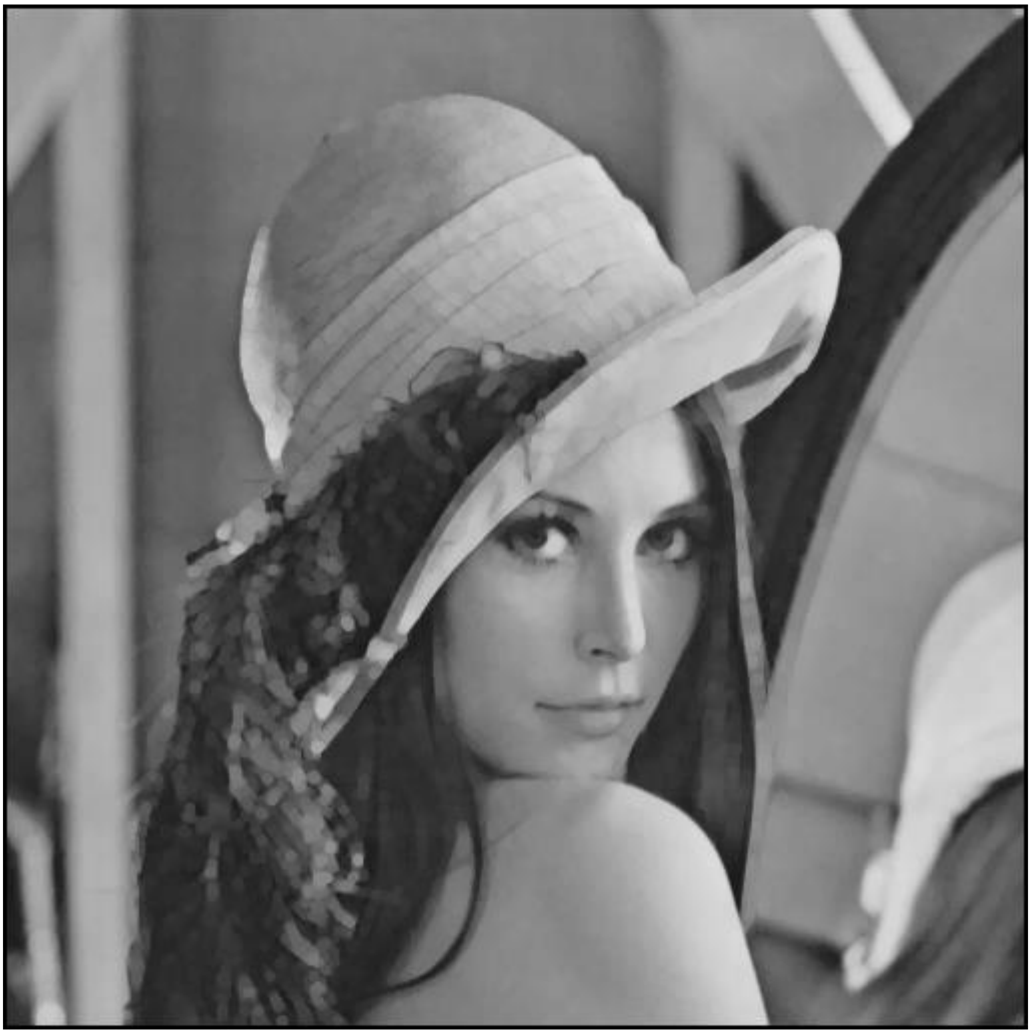


```
for i in range(2 , height-2):
    for j in range(2 , width-2):
        cover = lena_dilation[i-2:i+3,j-2:j+3]
        lena_closing[i][j] = Erosion(cover)

        cover = lena_erosion[i-2:i+3,j-2:j+3]
        lena_opening[i][j] = Dilation(cover)
```

Opening 是先做 Erosion 再做 Dilation，Closing 則是先 Dilation 再做 Erosion

Opening:



Closing: