

1. Gray scale dilation : 從左上到右下每個點都用 kernel 去跑過一次附近 kernel 範圍內尋找最大的值，找到以後將原本中心點改成該最大值。



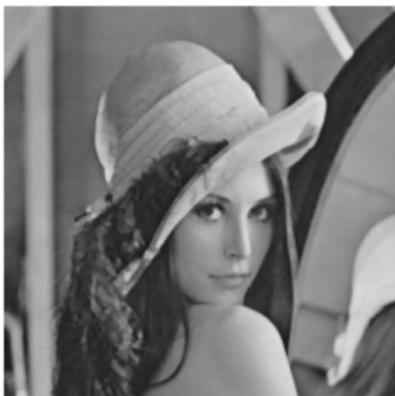
```
for i in range(h):
    for j in range(w):
        max_val = 0
        for k in range(21):
            x = i + ker[k][0]
            y = j + ker[k][1]
            if (x >= 0 and x < h and y >= 0 and y < w):
                if img[x][y] > max_val:
                    max_val = img[x][y]
        new_img[i][j] = max_val
```

2. Gray scale erosion : 從左上到右下每個點都用 kernel 去跑過一次附近 kernel 範圍內尋找最小的值，找到以後將原本中心點改成該最小值。



```
for i in range(h):
    for j in range(w):
        min_val = 255
        for k in range(21):
            x = i + ker[k][0]
            y = j + ker[k][1]
            if (x >= 0 and x < h and y >= 0 and y < w):
                if img[x][y] < min_val:
                    min_val = img[x][y]
        new_img[i][j] = min_val
```

3. Gray scale opening : 先對 img 做 erosion 再做 dilation



```
def opening(img):
    new_img = np.zeros([h, w], dtype=img.dtype)
    new_img = img
    return dilation(erosion(new_img))
```

#### 4. Gray scale closing : 先對 img 做 dilation 再做 erosion



```
def closing(img):  
    new_img = np.zeros([h, w], dtype=img.dtype)  
    new_img = img  
    return erosion(dilation(new_img))
```