Homework 5　　資工所碩一 R05922068 彭宇劭

1. Mathematical Morphology - Gray Scale Morphology

source code: hw5.py

執行方式：python hw5.py

版本：Python 2.7.10

Kernel：3-5-5-5-3 kernel for dilation, erosion, opening and closing

簡述：

1. 在main function定義 3-5-5-5-3 kernel及呼叫不同transform function。

```
def main():
    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]
             ]

    img = cv2.imread('lena.bmp', 0)

    lena_dilation = dilation(img, kernel)
    lena_erosion = erosion(img, kernel)
    lena_opening = opening(img, kernel)
    lena_closing = closing(img, kernel)

    cv2.imwrite('lena_dilation.bmp',lena_dilation)
    cv2.imwrite('lena_erosion.bmp',lena_erosion)
    cv2.imwrite('lena_opening.bmp',lena_opening)
    cv2.imwrite('lena_closing.bmp',lena_closing)
```

2. dilation transform

由圖片上而下左而右跑過每個點，若中心大於0，則以該點為中心擴展且在圖片範圍內的pixels皆須改值為Kernel所涵蓋pixels中，對大的pixels值

```
def dilation(img, kernel):
    lena_dilation = np.zeros(img.shape, dtype=np.int)
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            if(img[i][j]>0):
                max_pixel = 0
                for k, l in kernel:
                    if ((i+k)>=0 and (j+l)>=0 and (i+k)<img.shape[0] and (j+l)<img.shape[1]):
                        max_pixel = img[i+k][j+l] if img[i+k][j+l] > max_pixel else max_pixel

                for k, l in kernel:
                    if ((i+k)>=0 and (j+l)>=0 and (i+k)<img.shape[0] and (j+l)<img.shape[1]):
                        lena_dilation[i+k][j+l] = max_pixel
    return lena_dilation
```

3. erosion transform

由圖片上而下左而右跑過每個pixel，若可以滿足每個pixels在以某點為中心伸展開的Kernel中皆大於0，則該點必須設為Kernel所涵蓋pixels中，最小的pixels值，不滿足的點維持0

```
def erosion(img, kernel):
    lena_erosion = np.zeros(img.shape, dtype=np.int)
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            if(img[i][j]>0):
                fit = True
                min_pixel = 255
                for k, l in kernel:
                    if ((i+k)<0 or (j+l)<0 or (i+k)>=img.shape[0] or (j+l)>=img.shape[1] or img[i+k][j+l]==0):
                        fit = False
                        break
                if fit:
                    for k, l in kernel:
                        min_pixel = img[i+k][j+l] if img[i+k][j+l] < min_pixel else min_pixel
                    lena_erosion[i][j] = min_pixel;
    return lena_erosion
```

4. opening and closing

同hw4

opening: 先做erosion再做dilation

closing: 先做dilation再做erosion

```
def opening(img, kernel):
    return dilation(erosion(img, kernel), kernel)

def closing(img, kernel):
    return erosion(dilation(img, kernel), kernel)
```

結果：

Dilation

Erosion

Opening

Closing