CV_HW5

R07922125羅偉倫

Language and Tool

- Python3.6
- Numpy
- PIL

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]]
```

Input and binarize lena picture

```
# origin
img = Image.open('lena.bmp')
img_array = np.array(img)
Image.fromarray(img_array).save('lena.jpg')
```

lena

Problem1 Dilation

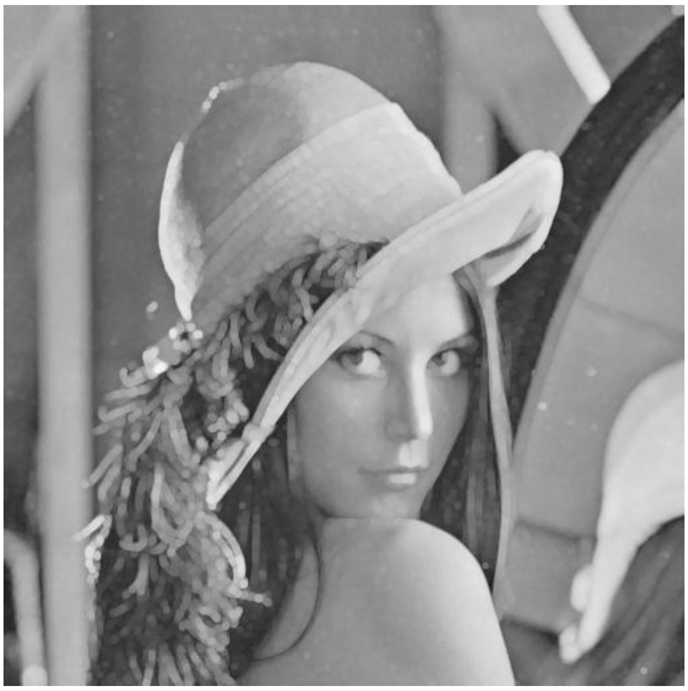
I trace all the pixels of the lena picture, and make the pixel dilated by the kernel.

img_array:numpy array, the array of a picture
kernel:list, the kernel I use

```
def dilation(img_array, kernel):
    img_d = np.copy(img_array)
    for i in range(img_array.shape[0]):
        for j in range(img_array.shape[1]):
            max_ = 0
            for k in kernel:
                new_x = i + k[0]
                new_y = j + k[1]
                if(new_x >=0 and new_x<img_array.shape[0] and new_y>=0 
and new_y<img_array.shape[1]):
                    max_ = max(max_, img_array[new_x][new_y])
            img_d[i][j] = max_

    return img_d

img_d = dilation(img_array, kernel)
Image.fromarray(img_d).save('lena_dil.jpg')
```



lena_dil

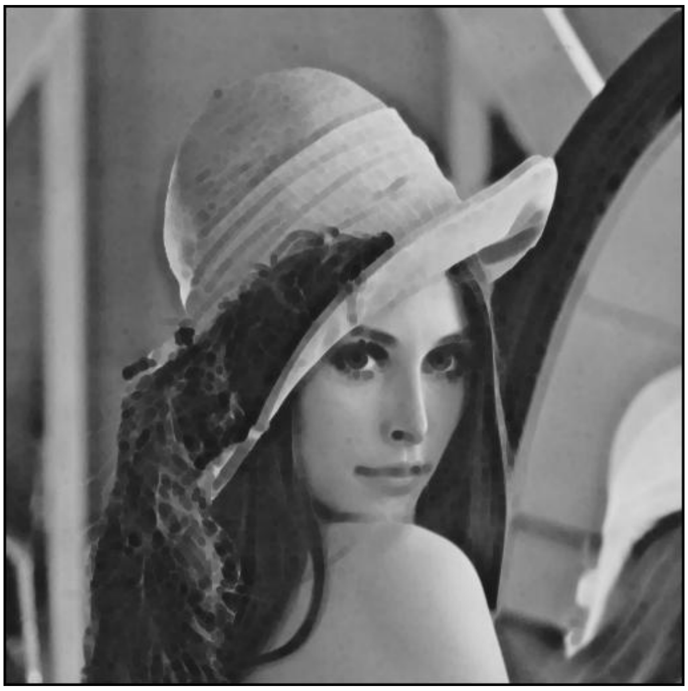
Problem2 Erosion

```
During tracing all pixels of the lena picture,I do the erosion by the
kernel.Besides, if the counting number is less than the length of
kernel, the pixel value will be 0.

img_array:numpy array, the array of the binary picture
kernel:list, the kernel I use

def erosion(img_array, kernel):
    img_e = np.copy(img_array)
    length = len(kernel)
    for i in range(img_array.shape[0]):
        for j in range(img_array.shape[1]):
            min_ = 255
            count = 0
            for k in kernel:
                new_x = i + k[0]
                new_y = j + k[1]
                if(new_x >=0 and new_x<img_array.shape[0] and new_y>=0
and new_y<img_array.shape[1]):
                    count += 1
                    min_ = min(min_,img_array[new_x][new_y])
            if (count == length):
                img_e[i][j] = min_
            else:
                img_e[i][j] = 0
    return img_e

img_e = erosion(img_array, kernel)
Image.fromarray(img_e).save('lena_ero.jpg')
```



lena_ero

Problem 3&4 Opening and Closing

To perform opening on the lena picture,I do erosion first and then do dilation.
To perform closing on the lena picture,I do dilation first and then do erosion.

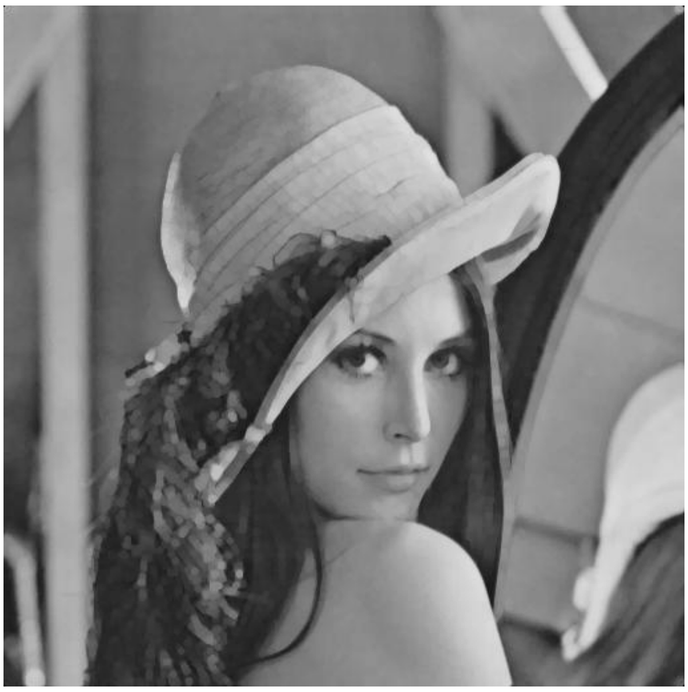
img_array:numpy array, the array of the binary picture
kernel:list, the kernel I use

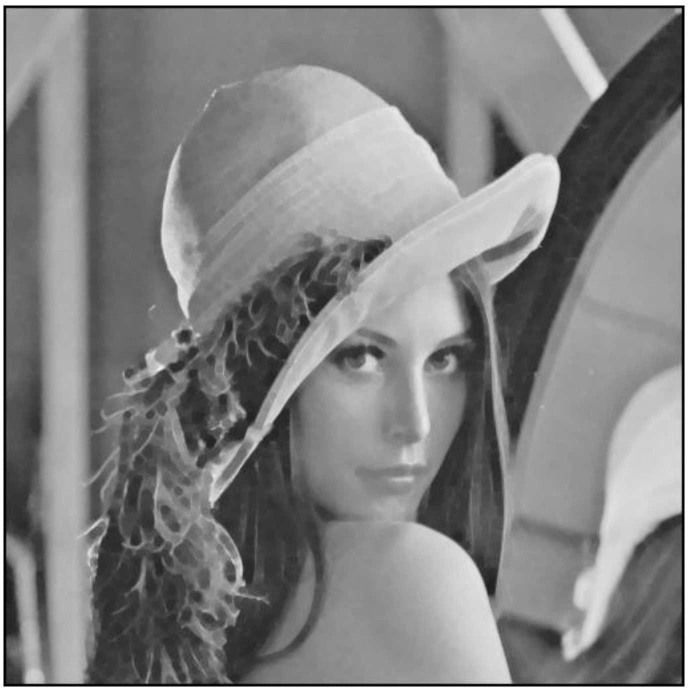
```
def opening(img_array, kernel):
    return dilation(erosion(img_array, kernel),kernel)

img_open = opening(img_array, kernel)
Image.fromarray(img_open).save(‘lena_open.jpg')

def closing(img_array, kernel, pixel):
    return erosion(dilation(img_array, kernel),kernel)

img_close = closing(img_array, kernel)
Image.fromarray(img_close).save(‘lena_close.jpg’)
```


lena_open


lena_close