電腦視覺hw4

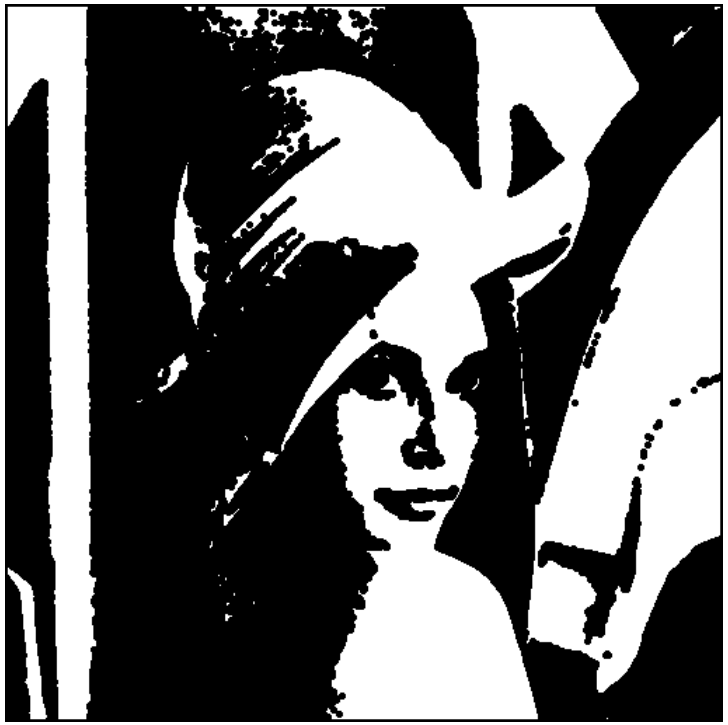
R11922131 周柏諺

```python
def dilation():
    img = cv2.imread("Lena.bmp")
    height=img.shape[0]
    width=img.shape[1]
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    cv2.imwrite("gray.bmp", gray)
    #二值化
    binary=cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    for i in range(height):
        for j in range(width):
            if binary[i,j]<128:
                binary[i,j]=0
            elif binary[i,j]>=128:
                binary[i,j]=255
    cv2.imwrite("binary.bmp", binary)
    #dilation
    binary.flags.writeable = True
    kernel=np.array([[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]])
    dilation=np.zeros((height,width),dtype=int)
    for i in range(height):
        for j in range(width):
            if binary[i,j]<128:
                binary[i,j]=0
            elif binary[i,j]>=128:
                binary[i,j]=1
    for i in range(height):
        for j in range(width):
            if(binary[i][j]==1):
                for x in range(-2,3,1):
                    for y in range(-2,3,1):
                        if (0<=x+i and x+i<height and 0<=y+j and y+j<width):
                            if kernel[x+2][y+2]==1:
                                dilation[i+x][y+j]=255
    dilation_img=Image.fromarray(dilation)
    dilation_img=dilation_img.convert("L")
    #dilation_img.show()
    dilation_img.save("dilation.bmp")
```

dilation

```python
#ersion
img = cv2.imread("Lena.bmp")
height=img.shape[0]
width=img.shape[1]
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imwrite("gray.bmp", gray)
#二値化
binary=cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
for i in range(height):
    for j in range(width):
        if binary[i,j]<128:
            binary[i,j]=0
        elif binary[i,j]>=128:
            binary[i,j]=255
cv2.imwrite("binary.bmp", binary)
binary.flags.writeable = True

for i in range(height):
    for j in range(width):
        if binary[i,j]<128:
            binary[i,j]=0
        elif binary[i,j]>=128:
            binary[i,j]=1
#print(binary[1][1])
kernel=np.array([[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]])
kernel_sum=0
for i in range(kernel.shape[0]):
    for j in range(kernel.shape[1]):
        kernel_sum=kernel_sum+kernel[i][j]
erosion=np.zeros((height,width),dtype=int)
for i in range(height):
    for j in range(width):
        count=0
        for x in range(-2,3,1):
            for y in range(-2,3,1):
                if (0<=x+i and x+i<height and 0<=y+j and y+j<width):
                    count=count+(binary[i+x][j+y]*kernel[x+2][y+2])
        if count==kernel_sum:
            erosion[i][j]=255
erosion_img=Image.fromarray(erosion)
erosion_img=erosion_img.convert("L")
erosion_img.save("erosion.bmp")
```



erosion

Opening

closing

```python
#hit and miss
img = cv2.imread("Lena.bmp")
height=img.shape[0]
width=img.shape[1]
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imwrite("gray.bmp", gray)
#二值化
binary=cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
for i in range(height):
    for j in range(width):
        if binary[i,j]<128:
            binary[i,j]=0
        elif binary[i,j]>=128:
            binary[i,j]=255
cv2.imwrite("binary.bmp", binary)
binary.flags.writeable = True
for i in range(height):
    for j in range(width):
        if binary[i,j]<128:
            binary[i,j]=0
        elif binary[i,j]>=128:
            binary[i,j]=1
#對binary的inverse
binary_c=cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
for i in range(height):
    for j in range(width):
        if binary_c[i,j]<128:
            binary_c[i,j]=0
        elif binary_c[i,j]>=128:
            binary_c[i,j]=255
binary_c.flags.writeable = True
for i in range(height):
    for j in range(width):
        if binary_c[i,j]<128:
            binary_c[i,j]=0
        elif binary_c[i,j]>=128:
            binary_c[i,j]=1
for i in range(height):
    for j in range(width):
        if binary_c[i,j]==1:
            binary_c[i,j]=0
        elif binary_c[i,j]==0:
            binary_c[i,j]=1
cv2.imwrite("binary_c.bmp", binary_c)

J_kernel=np.array([[0,0,0],
                   [1,1,0],
                   [0,1,0]])
J_kernel_sum=0
for i in range(J_kernel.shape[0]):
    for j in range(J_kernel.shape[1]):
        J_kernel_sum=J_kernel_sum+J_kernel[i][j]
K_kernel=np.array([[0,1,1],
                   [0,0,1],
                   [0,0,0]])
K_kernel_sum=0
for i in range(K_kernel.shape[0]):
    for j in range(K_kernel.shape[1]):
        K_kernel_sum=K_kernel_sum+K_kernel[i][j]

A_erosion_J=np.zeros((height,width),dtype=int)
Ac_erosion_K=np.zeros((height,width),dtype=int)
hit_and_miss=np.zeros((height,width),dtype=int)

for i in range(height):
    for j in range(width):
        count=0
        for x in range(-1,2,1):
            for y in range(-1,2,1):
                if (0<=x+i and x+i<height and 0<=y+j and y+j<width):
                    count=count+(binary[i+x][j+y]*J_kernel[x+1][y+1])
        if count==J_kernel_sum:
            A_erosion_J[i][j]=255

for i in range(height):
    for j in range(width):
        count=0
        for x in range(-1,2,1):
            for y in range(-1,2,1):
                if (0<=x+i and x+i<height and 0<=y+j and y+j<width):
                    count=count+(binary_c[i+x][j+y]*K_kernel[x+1][y+1])
        if count==K_kernel_sum:
            Ac_erosion_K[i][j]=255

for i in range(height):
    for j in range(width):
        if A_erosion_J[i][j]==Ac_erosion_K[i][j]==255:
            hit_and_miss[i][j]=255
        else:
            hit_and_miss[i][j]=0
hit_and_miss_img=Image.fromarray(hit_and_miss)
hit_and_miss_img=hit_and_miss_img.convert("L")
hit_and_miss_img.save("hit_and_miss.bmp")
```

Hit and miss