Computer Vision hw4

R06922075 翁瑋

- Write programs which do binary morphological dilation, erosion, opening, closing, and hit-and-miss transform on a binary image

```python
def binarize(im , threshold) :
    pixels = im.load()
    print (pixels[0,0])
    im_thre = im.copy()

    for i in range(int(im.size[0])) :
        for j in range(int(im.size[1])) :
            im_thre.putpixel((i,j), (pixels[i,j] > threshold)*255 )

    return im_thre
```

先將圖片用 128 當 threshold 做 binarize

```python
def dilation(im , kernel):
    pixels = im.load()
    im_dil = Image.new('L', im.size, 'black')

    for i in range(1,int(im.size[0])-1) :
        for j in range(1,int(im.size[1])-1) :
            for k in range(len(kernel)):
                if(pixels[i,j] == 255):
                    im_dil.putpixel((i+kernel[k][0],j+kernel[k][1]), 255 )
    return im_dil
```

實作 dilation 部分

```python
def erosion(im , kernel) :
    pixels = im.load()
    im_ero = Image.new('L', im.size, 'white')
    for i in range(int(im.size[0])) :
        im_ero.putpixel((i,0), 0)
        im_ero.putpixel((i,511), 0)
        im_ero.putpixel((0,i), 0)
        im_ero.putpixel((511,i), 0)


    for i in range(1,int(im.size[0])-1) :
        for j in range(1,int(im.size[1])-1) :
            for k in range(len(kernel)):
                if(pixels[i,j] == 0):
                    im_ero.putpixel((i+kernel[k][0],j+kernel[k][1]), 0 )
    return im_ero
```

實作 erosion 部分，採用 $(A \ominus B)^c = A^c \oplus \check{B}$

```python
def opening(im , kernel):
    return dilation(erosion(im , kernel) , kernel)

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

Opening 跟 Closing 實作部分，用 dilation 跟 erosion 結合來實現

```python
def hit_miss(im , J , K):
    pixels = im.load()
    im_c = im.copy()

    for i in range(int(im.size[0])) :
        for j in range(int(im.size[1])) :
            im_c.putpixel((i,j), (pixels[i,j] == 0)*255 )

    im1 = erosion(im , J)
    im2 = erosion(im_c , K)
    pixels1 = im1.load()
    pixels2 = im2.load()

    im_result = im.copy()
    for i in range(int(im.size[0])) :
        for j in range(int(im.size[1])) :
            im_result.putpixel((i,j), ((pixels1[i,j] == 255)&(pixels2[i,j] == 255))*255 )

    return im_result
```

Hit_and_miss 實作，先分開做成兩張圖再求聯集

Result :

Origin	Dilation
	
Erosion	Opening
	

Closing	Hit-and-miss
	

Environment :

Anaconda3

Python 3.6.1

Using Library :

PIL

Benchmark :

Lena.bmp