

Computer Vision HW4

R12922054 資工所 邱信璋

Description

- 根據作業要求，將原圖進行128進行二值化。
- kernel的大小也設成3-5-5-5-3的大小。
- kernel-L, kernel-k也按照題目的敘述

1. Dilation Image

How to implement :

把原圖所有的點遍歷一遍然後當遇到的不是空白時，根據kernel的形狀拓展(注意這裡要小心不要讓拓展之後的圖片大小超過原圖)

```
dilation_img = np.zeros((row_size, col_size), dtype=int)
for i in range(row_size):
    for j in range(col_size):
        if np_img[i][j]:
            for k in kernel:
                new_i = i + k[0]
                new_j = j + k[1]
                if new_i >= 0 and new_i < row_size and new_j >= 0 and new_j < col_size:
                    dilation_img[new_i][new_j] = 255
```



Figure 1: image for Dilation.

2. Erosion Image

How to implement :

把原圖所有的點遍歷一遍，當該點根據kernel所作的擴展，全部都包含在原圖，則將該點保留。(同樣如果做擴展的當下擴展後的圖片大小已經超過原圖大小則可以直接忽略該點)

```
erosion_img = np.zeros((row_size, col_size), dtype=int)
for i in range(row_size):
    for j in range(col_size):
        flag = True
        for k in kernel:
            new_i = i + k[0]
            new_j = j + k[1]
            if new_i < 0 or new_i >= row_size or new_j < 0 or new_j >= col_size or not np_img[new_i][new_j]:
                flag = False
                break
        if flag == True:
            erosion_img[i][j] = 255
```



Figure 2: image for Erosion.

3. Opening Image

How to implement :

先做erosion再做dilation(先把東西變背景再把背景變東西)

```
np_opening_img = dilation(erosion(np_img, kernel, row_size, col_size), kernel, row_size, col_size)
```



Figure 3: image for Opening.

4 Closing Image

How to implement :

先做dilation再做erosion(先把背景變東西再把東西變背景)

```
np_closing_img = erosion(dilation(np_img, kernel, row_size, col_size), kernel, row_size, col_size)
```



Figure 4: image for Closing.

5. Hit and miss Image

How to implement :

1. 將原圖對kernel L做erosion
2. 將原圖做complement後再對kernel K做erosion
3. 最後把包含1. 2.結果的點保留，即可。

```
origin_img = copy.deepcopy(np_img)
origin_img_complement = get_complement(origin_img, row_size, col_size)
j_kernel_img = erosion(origin_img, kernel_j, row_size, col_size)
k_kernel_img = erosion(origin_img_complement, kernel_k, row_size, col_size)

hit_and_miss_img = copy.deepcopy(np_img)
for i in range(row_size):
    for j in range(col_size):
        if j_kernel_img[i][j] and k_kernel_img[i][j]:
            hit_and_miss_img[i][j] = 255
        else:
            hit_and_miss_img[i][j] = 0
```

```
complement_img = copy.deepcopy(np_img)
for i in range(row_size):
    for j in range(col_size):
        complement_img[i][j] = 255 - np_img[i][j]
```

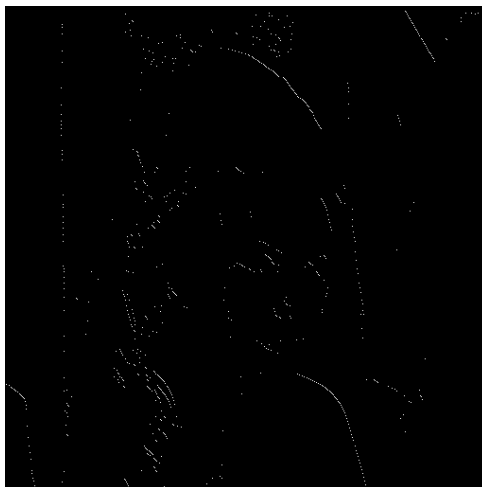


Figure 5: image for Hit and miss