

Computer and Robot Vision Homework 4

資工碩一 張家源 r07922102

使用語言：Python

Dilation

變數

v_map: 以tuples的形式存此圖foreground位置

img: input圖片(已經過Binarize)

方法

根據每個foreground，讓他加上kernel的每個tuple。若沒有超出邊界(legal function)，則在result image填上此位置。

```
def dilation(img, v_map, kernel):
    result = np.zeros((img.shape[0],img.shape[1]))
    result_vmap = []
    for i in range(len(kernel)):
        for j in range(len(v_map)):
            if legal(addition(kernel[i],v_map[j])):
                e = addition(kernel[i],v_map[j])
                result[e[0]][e[1]] = 255
                result_vmap.append((e[0],e[1]))
    cv2.imwrite('dilation.jpg',result)
    return result, result_vmap
```

結果



Erosion

變數同dilation

每個foreground，檢查是否加上每個kernel後在原圖仍為foreground。若檢查後合格，則在result image填上此格。

```
def erosion(img, v_map , kernel):
    result = np.zeros((img.shape[0],img.shape[1]))
    result_vmap = []
    for i in range(len(v_map)):
        valid = True
        for j in range(len(kernel)):
            if legal(addition(v_map[i], kernel[j])):
                e = addition(v_map[i], kernel[j])
                if img[e[0]][e[1]] != 255:
                    valid = False
                    break
        if valid == True:
            e = v_map[i]
            result[e[0]][e[1]] = 255
            result_vmap.append((e[0],e[1]))
    cv2.imwrite('erosion.jpg',result)
    return result, result_vmap
```

結果



Hit and Miss

變數同上。

先檢查foreground加上J中每一個tuple在原圖中是否也是foreground。再檢查foreground加上K中每一個tuple在原圖中是否是background。若兩者皆成立，在result image填上此格。

```
def HitAndMiss(img, v_map, J, K):
    result = np.zeros((img.shape[0],img.shape[1]))
    result_vmap = []
    for i in range(len(v_map)):
        valid = True
        for j in range(len(J)):
            if legal(addition(v_map[i], J[j])):
                e = addition(v_map[i], J[j])
                if img[e[0]][e[1]] != 255:
                    valid = False
                    break
            if valid == False:
                break
        for j in range(len(K)):
            if legal(addition(v_map[i], K[j])):
                e = addition(v_map[i], K[j])
                if img[e[0]][e[1]] != 0:
                    valid = False
                    break
        if valid == True:
            e = v_map[i]
            result[e[0]][e[1]] = 255
            result_vmap.append((e[0],e[1]))
    cv2.imwrite('HitAndMiss.jpg',result)
```

結果



Opening

先做Erosion，再用Erosion的結果做Dilation。

```
def opening(img, v_map, kernel):  
    result, result_vmap = erosion(img, v_map, kernel)  
    result, result_vmap = dilation(result, result_vmap, kernel)  
  
    cv2.imwrite('opening.jpg', result)
```

結果



Closing

先做Dilation，再用Dilation的結果做Erosion。

```
def closing(img, v_map, kernel):  
    result, result_vmap = dilation(img, v_map, kernel)  
    result, result_vmap = erosion(result, result_vmap, kernel)  
  
    cv2.imwrite('closing.jpg', result)
```

結果



