Computer Vision Hw #4

B06902058 吳崇維

Homework Description

- Write programs which do binary morphology on a binary image:
  - (a) Dilation
  - (b) Erosion
  - (c) Opening
  - (d) Closing
  - (e) Hit-and-miss transform

- **Dilation**

```
def Dilation(arr, kernel):
    new_graph = np.zeros((length, length))
    for i in range(length):
        for j in range(length):
            if (arr[i][j] == 255):
                for dx, dy in kernel:
                    nx = i + dx
                    ny = j + dy
                    if (In(nx, ny)):
                        new_graph[nx][ny] = 255
    return new_graph
```

- **Erosion**

```
def Erosion(arr, kernel):
    new_graph = np.zeros((length, length))
    for i in range(length):
        for j in range(length):
            flag = 1
            for dx, dy in kernel:
                nx = i + dx
                ny = j + dy
                if (In(nx, ny) and arr[nx][ny] == 0):
                    flag = 0
                    break
            if (flag):
                new_graph[i][j] = 255
    return new_graph
```

- **Opening**

Just call the function : Erosion & Dilation

```
new_graph = Dilation(Erosion(arr, kernel), kernel)
```

- **Closing**

Just call the function : Erosion & Dilation

```
new_graph = Erosion(Dilation(arr, kernel), kernel)
```

- **Hit and Miss**

```
arr1 = Erosion(arr, J)       # J, K are the 'L' shape kernel
arr2 = Erosion(inv(arr), K) # inv function inverse the pixels between black
and white
res = intersec(arr1, arr2)  # do the intersection calculation
```