

学生姓名： 丁晓琪

学生学号： 22336057

专业名称： 计科

## 一：二值膨胀

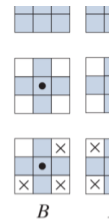
### 1. 理论：

$$A \oplus B = \left\{ z \mid \left( \hat{B} \right)_z \cap A \neq \emptyset \right\}$$

先对B反射，然后以 $z$ 对其进行平移，并确保平移后和 $A$ 至少是有一个元素是重叠的，即交集非空。

集合（结构元） $B$  相对于其原点的**反射**（表示为  $\hat{B}$ ）定义如下：

$$\hat{B} = \{ w \mid w = -b, b \in B \}$$



结构元及其点表示原点

集合  $B$  相对于点  $z = (z_1, z_2)$  的**平移**（表示为  $(B)_z$ ）定义如下：

$$(B)_z = \{ c \mid c = b + z, b \in B \}$$

### 2. 实现：（灰度值255看为1，其他灰度值看为0）

- 输入：图像(这里的灰度值仍是0-255，但是在函数处理中将灰度值255看成1，其他灰度值看成0)，结构元（3\*3的大小）

```
1 #膨胀,二值膨胀,只看像素值为0/255的位置的像素,像素不为255的默认为0,按照定义
2 def binary_dilation(image, kernel):
3     # 输入: 图像矩阵, 膨胀核
4     # 输出: 膨胀后的图像
5     result_image=np.zeros_like(image)
6     kernel_height,kernel_width=kernel.shape
7     image_height,image_width=image.shape
8     pad_height=kernel_height//2
9     pad_width=kernel_width//2
```

```

10     #保持原图像不变，周围填充0
11     padded_image=np.pad(image,((pad_height, pad_height), (pad_width,
pad_width)), mode='constant', constant_values=0)
12     for i in range(image_height):
13         for j in range(image_width):
14             # 结构元的中点在i,j时
15             for ki in range(-pad_height,pad_height+1):
16                 for kj in range(-pad_width,pad_width+1):
17                     # 只有结构元有个元素核图像位置重合
18                     if kernel[-ki+pad_height,-kj+pad_width] and
padded_image[i+ki,j+kj]==255:
19                         result_image[i,j]=255
20
21     return result_image

```

## 二：二值腐蚀

### 1. 理论：

$B$ 对 $A$ 的腐蚀是一个用 $z$ 平移的 $B$ 包含在 $A$ 中的所有的点 $z$ 的集合。我们假设 $B$ 是一个结构元，从上面描述将公式等价为：

$$A \ominus B = \left\{ z \mid (B)_z \cap A^c = \emptyset \right\}$$

### 2. 实现：灰度值为0的看成二值0，其他灰度值看成二值1

```

1  #腐蚀，像素不为0的默认为255，按照定义
2  def binary_erosion(image, kernel):
3      # 输入：图像矩阵，结构元
4      result_image=np.zeros_like(image)
5      kernel_height,kernel_width=kernel.shape
6      image_height,image_width=image.shape
7      pad_height=kernel_height//2
8      pad_width=kernel_width//2
9      # 保持图形腐蚀后大小不变
10     padded_image=np.pad(image,((pad_height, pad_height), (pad_width,
pad_width)), mode='constant', constant_values=0)
11
12     for i in range(image_height):
13         for j in range(image_width):
14             inside=True
15             # 结构元中心在i,j
16             for ki in range(-pad_height,pad_height+1):
17                 for kj in range(-pad_width,pad_width+1):
18                     # 只要结构元和图像背景有交集，则该位置为0
19                     if kernel[kj+pad_width,kj+pad_width] and
padded_image[i+ki,j+kj]==0:
20                         inside=False
21                         break
22             if inside:
23                 result_image[i,j]=255
24
25     return result_image

```

### 三：图像边界提取

1. 理论实现:

前景像素集合 $A$ 的边界 $\beta(A)$ 可按如下方式得到：首先使用合适的结构元 $B$ 腐蚀  $A$ ，然后求 $A$ 和腐蚀结果的差集。

$$\beta(A) = A - (A \ominus B)$$

2. 实现:

```
1   dilated_image=binary_dilation(gray_matrix,kernel)
2   eroded_image=binary_erosion(gray_matrix,kernel)
3   # 图像边界，按照定义
4   boundary_image=gray_matrix-eroded_image
```

### 四：实验结果:

膨胀后图像前景线条变粗，腐蚀后图像前景线条变细

1. 内核为

```
kernel = [
    [255, 255, 255],
    [255, 255, 255],
    [255, 255, 255]
]
```

Original Image



Dilated Image



Eroded Image



Boundary Image



2. 内核为

```
kernel = [
    [255, 255, 255],
    [0, 255, 0],
    [0, 255, 0]
]
```

Original Image



Dilated Image



Eroded Image



Boundary Image

