

Python+OpenCV图像处理

---

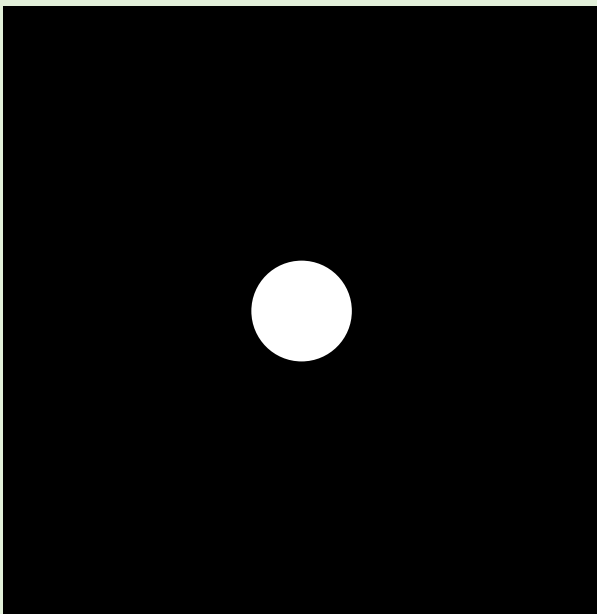
# 形态学转换

图像膨胀

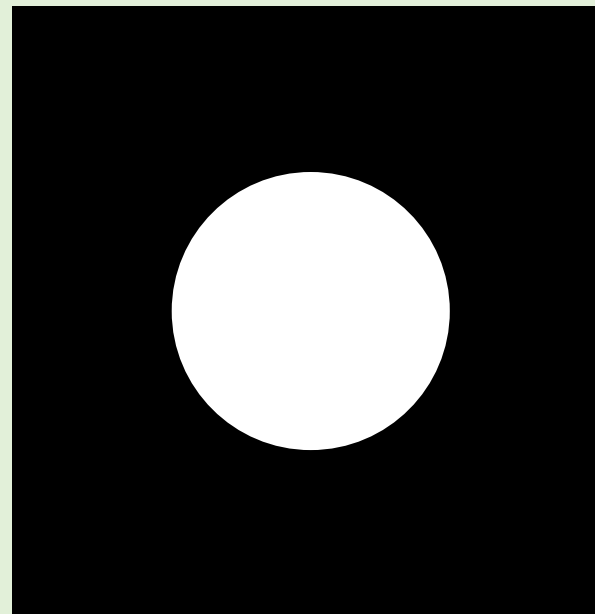
---

李大羊

## 膨胀示例



原始图像



膨胀图像

是腐蚀操作的逆操作。

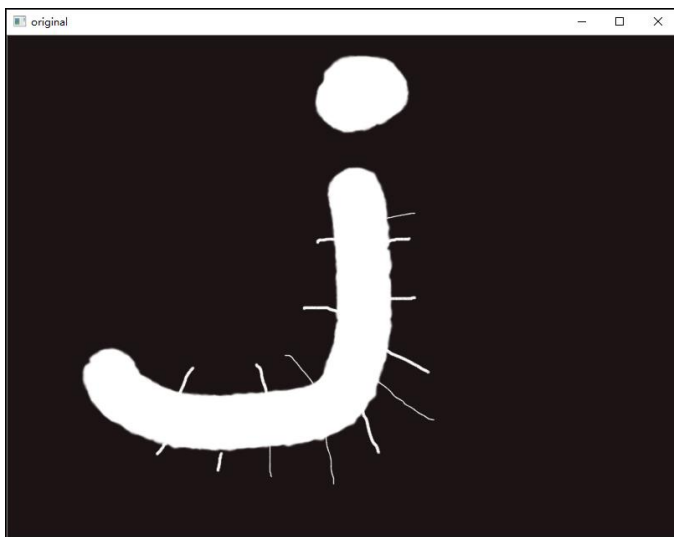


原始图像



膨胀图像

## 膨胀操作



原始图像

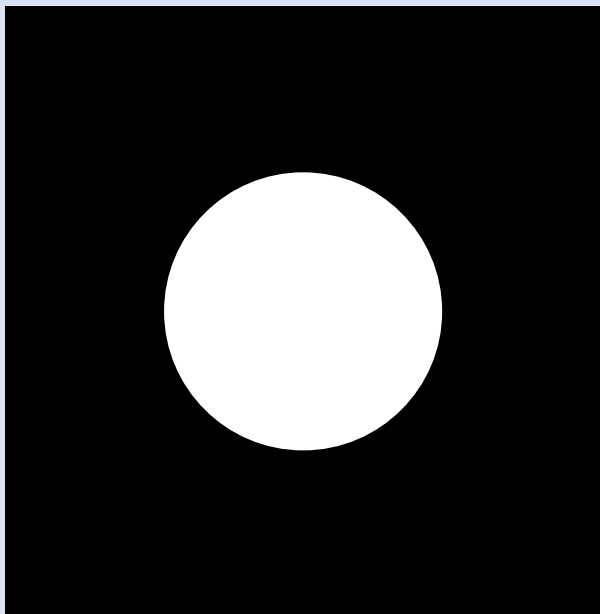


腐蚀图像



膨胀图像

- 图像被腐蚀后，去除了噪声，但是会压缩图像。
- 对腐蚀过的图像，进行膨胀处理，可以去除噪声，并保持原有形状。



原始图像

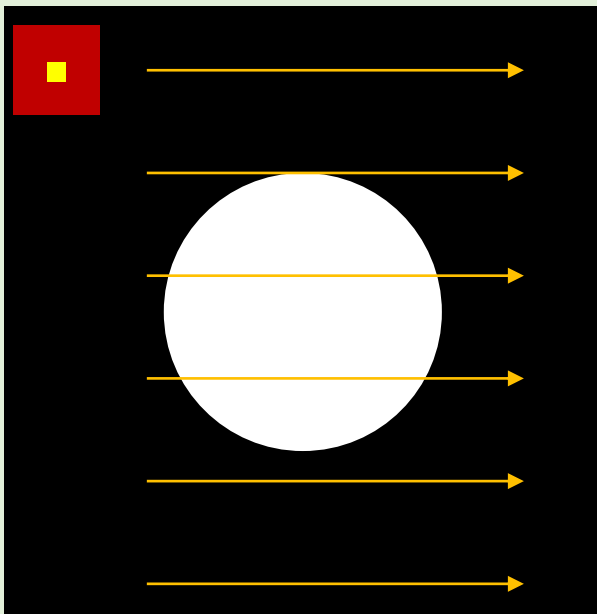


卷积核

## 基础

1. 形态学转换主要针对的是二值图像。
2. 两个输入对象。  
对象1：二值图像  
对象2：卷积核

## 膨胀



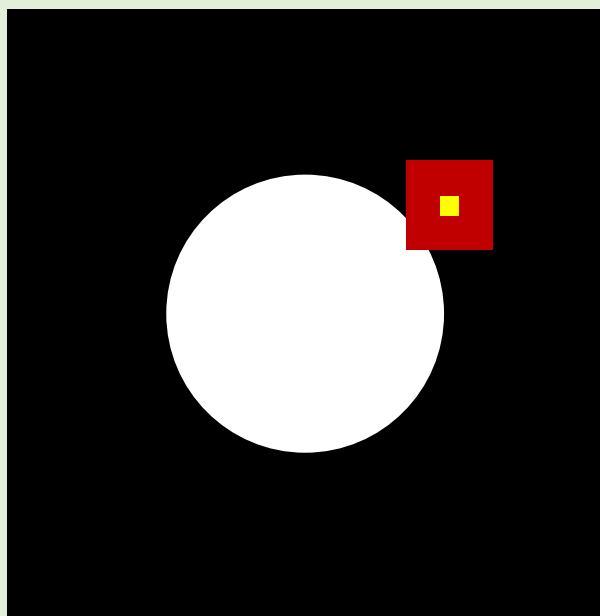
原始图像



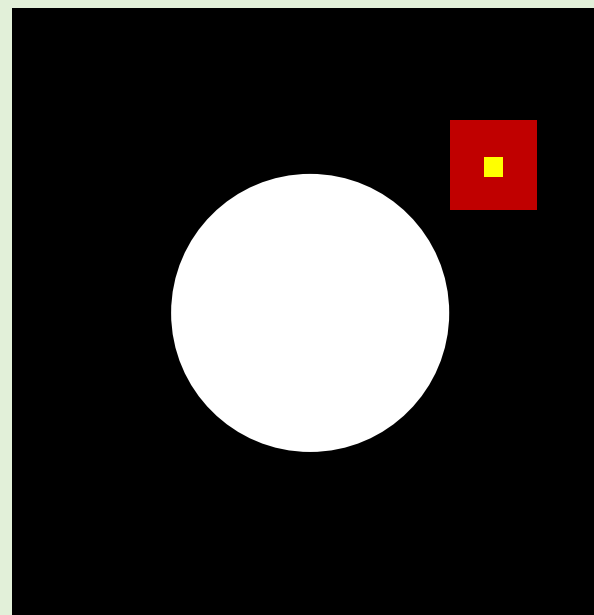
卷积核

卷积核的中心点逐个像素扫描原始图像。

## 膨胀



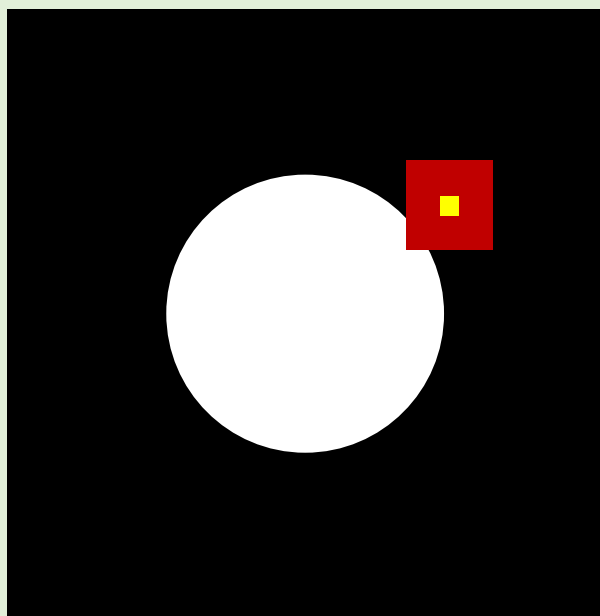
像素值置为1



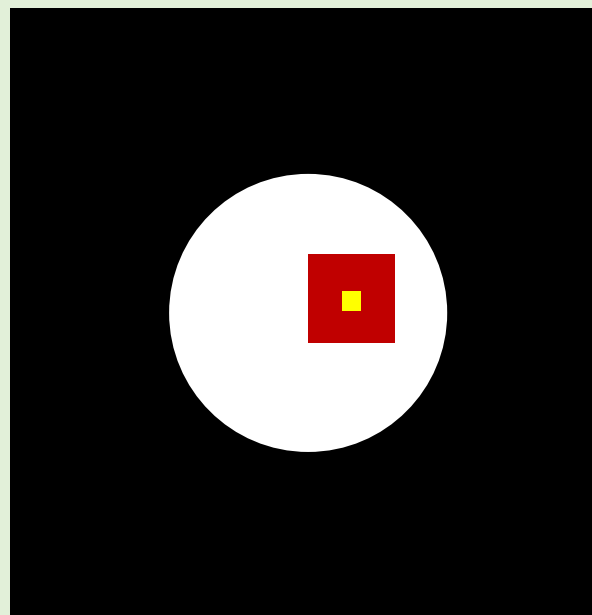
像素值保留为0

被扫描到的原始图像中的像素点，当卷积核对应的元素值只要有一个为1时，其值就为1，否则值为0.

## 膨胀



像素值置为1



像素值保留为1

被扫描到的原始图像中的像素点，当卷积核对应的元素值只要有一个为1时，其值就为1，否则值为0.



## 函数dilate

```
dst = cv2.dilate ( src , kernel , iterations )
```

dst , 处理结果

src , 源图像

kernel , 卷积核

iterations , 迭代次数

## 函数dilate

```
dst = cv2.dilate ( src , kernel , iterations )
```

dst , 处理结果

src , 源图像

kernel , 卷积核

iterations , 迭代次数

## 函数dilate

```
dst = cv2.dilate ( src , kernel , iterations )
```

dst , 处理结果

src , 源图像

kernel , 卷积核

iterations , 迭代次数

## 函数dilate

```
dst = cv2.dilate ( src , kernel , iterations )
```

dst , 处理结果

src , 源图像

kernel , 卷积核

iterations , 迭代次数

## 函数dilate

`dst = cv2.dilate ( src , kernel , iterations )`

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

## 函数dilate

```
dst = cv2.dilate ( src , kernel , iterations )
```

```
kernel = np.ones((5,5),np.uint8)
```

## 函数dilate

```
dst = cv2.dilate ( src , kernel , iterations )
```

```
kernel = np.ones((5,5),np.uint8)
```

## 函数dilate

```
dst = cv2.dilate ( src , kernel , iterations )
```

```
kernel = np.ones((5,5),np.uint8)
```



## 函数dilate

```
dst = cv2.dilate ( src , kernel , iterations )
```

```
kernel = np.ones((5,5),np.uint8)
```

## 使用的卷积核

```
kernel = np.ones((5,5),np.uint8)
```

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

## 函数dilate

```
dst = cv2.dilate ( src , kernel , iterations )
```

dst , 处理结果

src , 源图像

kernel , 卷积核

iterations , 迭代次数

## 函数dilate

```
dst = cv2.dilate ( src , kernel , iterations )
```

iterations , 迭代次数

默认情况下，迭代次数是1，根据需要可以进行多次膨胀操作。

## 函数dilate

```
dst = cv2.dilate ( src , kernel , iterations )
```

dst , 处理结果

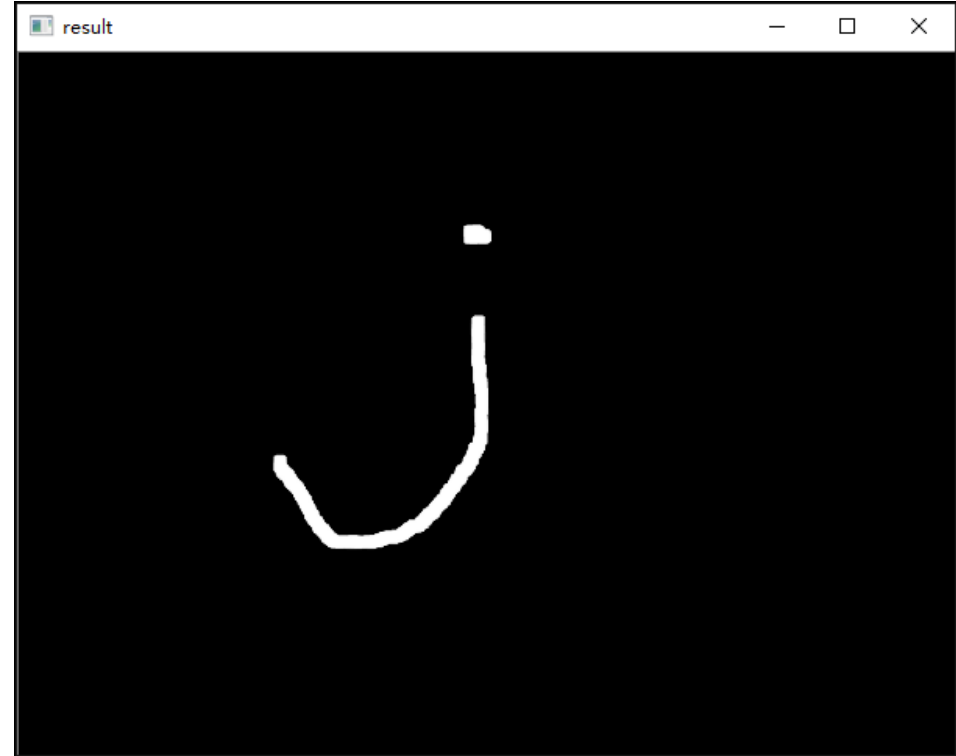
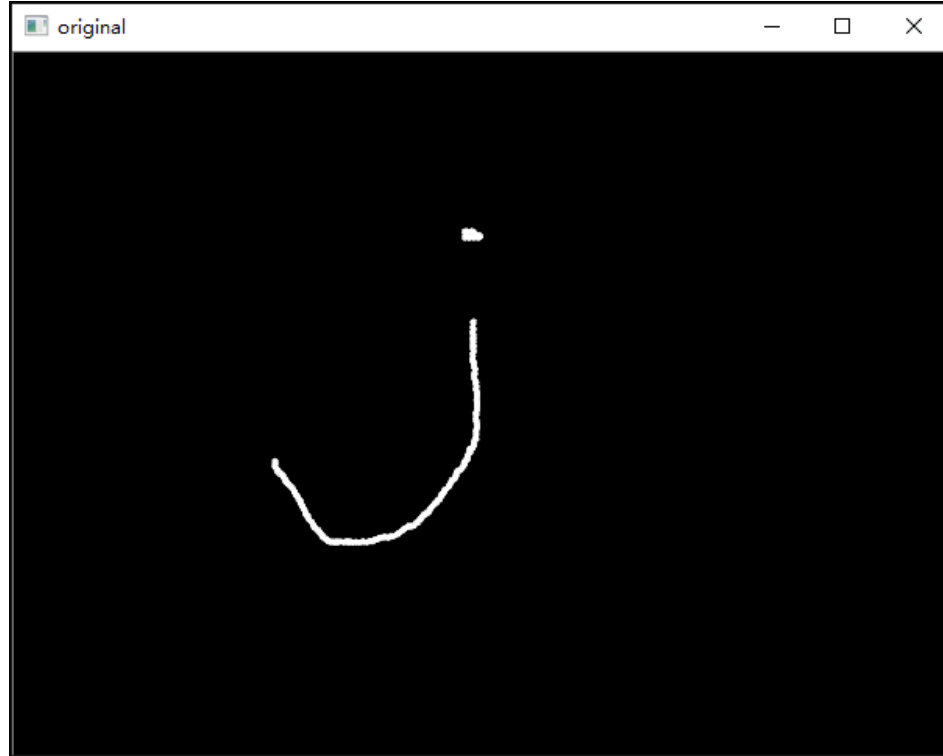
src , 源图像

kernel , 卷积核

iterations , 迭代次数

```
import cv2
import numpy as np
o=cv2.imread("image\\dilate.bmp",cv2.IMREAD_UNCHANGED)
kernel = np.ones((5,5),np.uint8)
dilation = cv2.dilate(o,kernel)
cv2.imshow("original",o)
cv2.imshow("dilation",dilation)
cv2.waitKey()
cv2.destroyAllWindows()
```

## 膨胀操作

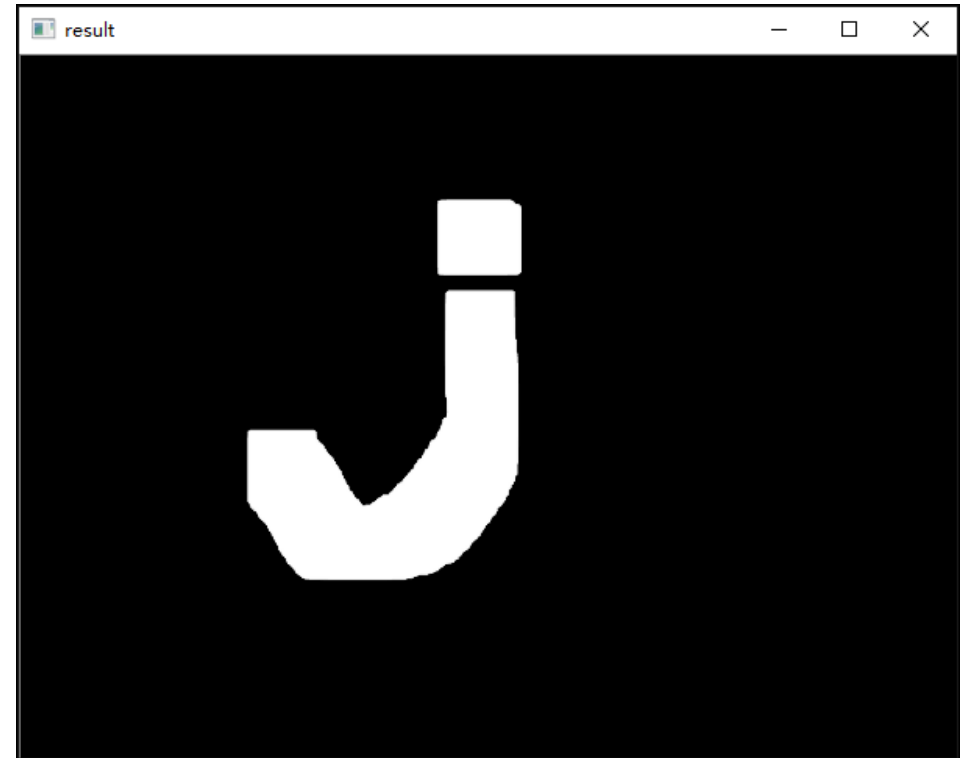
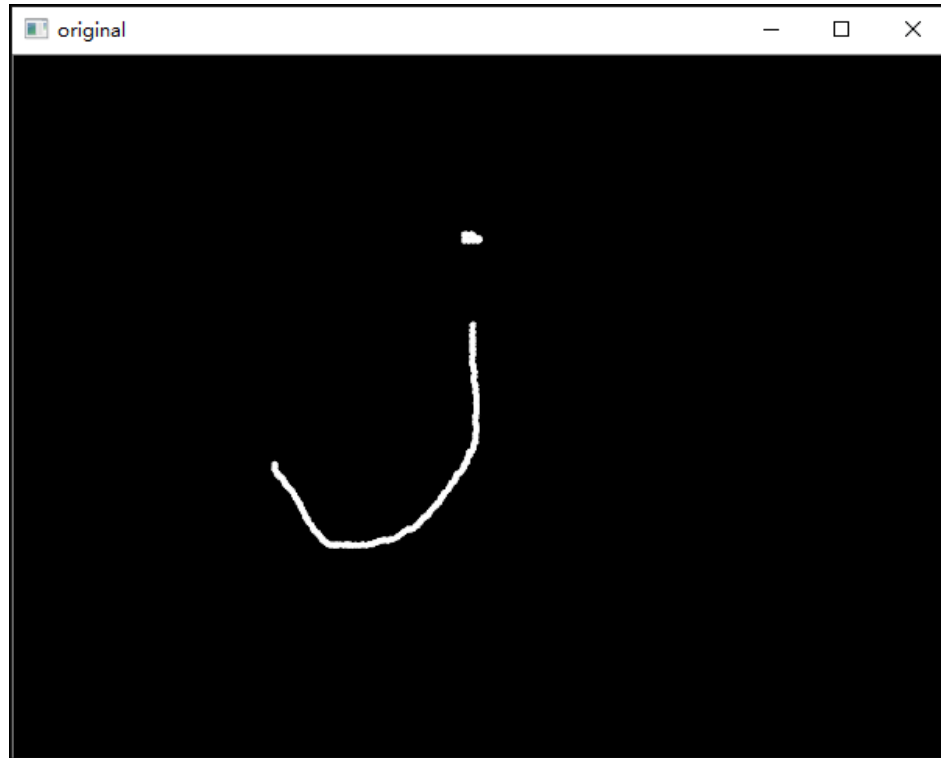


## 膨胀操作

```
import cv2
import numpy as np
o=cv2.imread("image\\dilate.bmp",cv2.IMREAD_UNCHANGED)
kernel = np.ones((5,5),np.uint8)
dilation = cv2.dilate(o,kernel,iterations = 9)
cv2.imshow("original",o)
cv2.imshow("dilation", dilation)
cv2.waitKey()
cv2.destroyAllWindows()
```



## 膨胀操作



Python+OpenCV图像处理

---

# 形态学转换

图像膨胀

---

李大羊

[lilizong@gmail.com](mailto:lilizong@gmail.com)