Python图像处理丨图像腐蚀与图像膨胀

华为云开发者联盟
已认证帐号

摘要：本篇文章主要讲解Python调用OpenCV实现图像腐蚀和图像膨胀的算法。

本文分享自华为云社区《[Python图像处理] 八.图像腐蚀与图像膨胀》，作者： eastmount 。

本篇文章主要讲解Python调用OpenCV实现图像腐蚀和图像膨胀的算法，基础性知识希望对您有所帮助。

- **1.基础理论**
- **2.图像腐蚀代码实现**
- **3.图像膨胀代码实现**

一. 基础知识

（注：该部分参考作者论文《一种改进的Sobel算子及区域择优的身份证智能识别方法》）

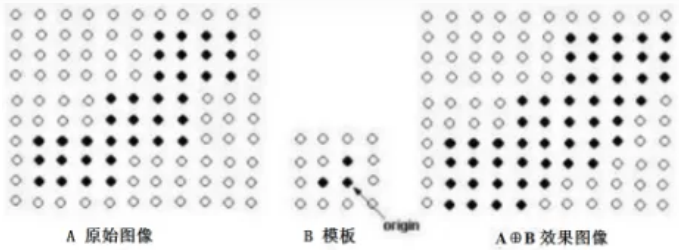
图像的膨胀（Dilation）和腐蚀（Erosion）是两种基本的形态学运算，主要用来寻找图像中的极大区域和极小区域。其中膨胀类似于“领域扩张”，将图像中的高亮区域或白色部分进行扩张，其运行结果图比原图的高亮区域更大；腐蚀类似于“领域被蚕食”，将图像中的高亮区域或白色部分进行缩减细化，其运行结果图比原图的高亮区域更小。

1.图像膨胀

膨胀的运算符是“⊕”，其定义如下：

$$A \oplus B = \{x \mid (B)_x \cap A \neq \Theta\}$$

该公式表示用B来对图像A进行膨胀处理，其中B是一个卷积模板或卷积核，其形状可以为正方形或



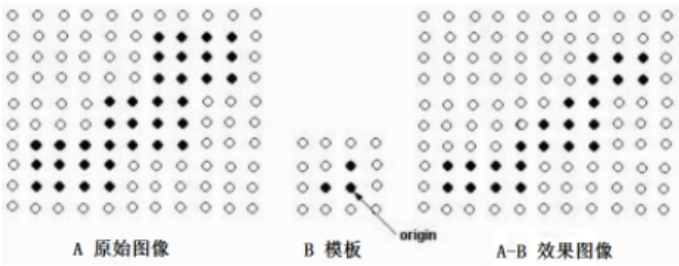
处理结果如下图所示：

2.图像腐蚀

腐蚀的运算符是“ - ”，其定义如下：

$$A - B = \{x \mid B_x \subseteq A\}$$

该公式表示图像A用卷积模板B来进行腐蚀处理，通过模板B与图像A进行卷积计算，得出B覆盖区域的像素点最小值，并用这个最小值来替代参考点的像素值。如图所示，将左边的原始图像A腐蚀处理为右边的效果图A-B。



处理结果如下图所示：

原始图像

腐蚀图像

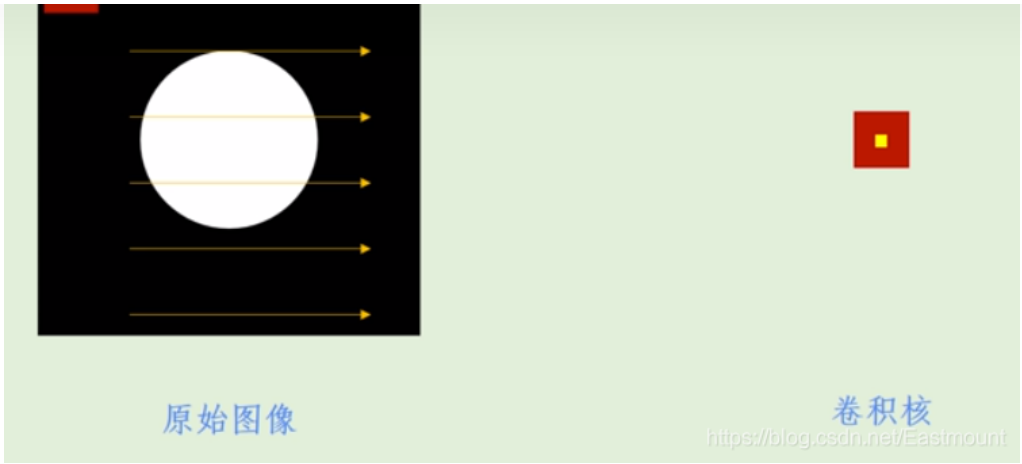
二. 图像腐蚀代码实现

1.基础理论

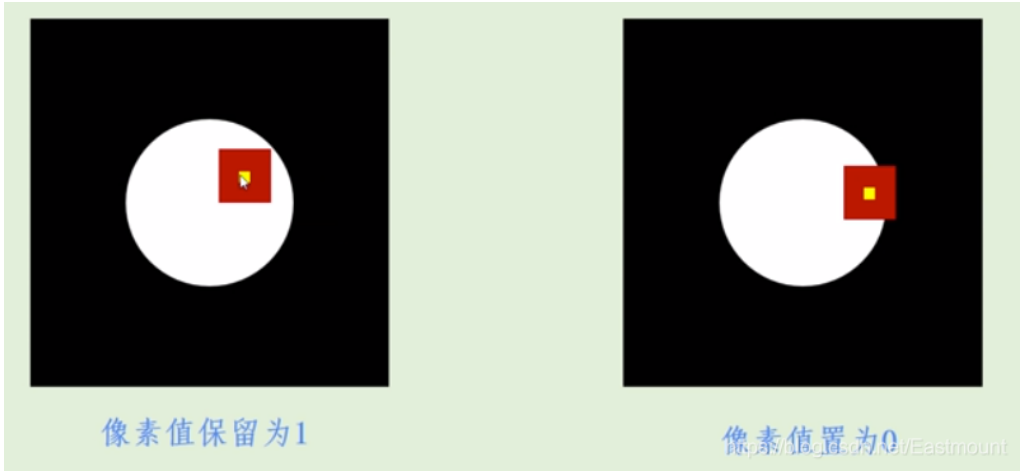
形态学转换主要针对的是二值图像（0或1）。图像腐蚀类似于“领域被蚕食”，将图像中的高亮区域或白色部分进行缩减细化，其运行结果图比原图的高亮区域更小。其主要包括两个输入对象：

(1)二值图像
(2)卷积核

卷积核是腐蚀中的关键数组，采用numpy库可以生成。卷积核的中心点逐个像素扫描原始图像，如下图所示：

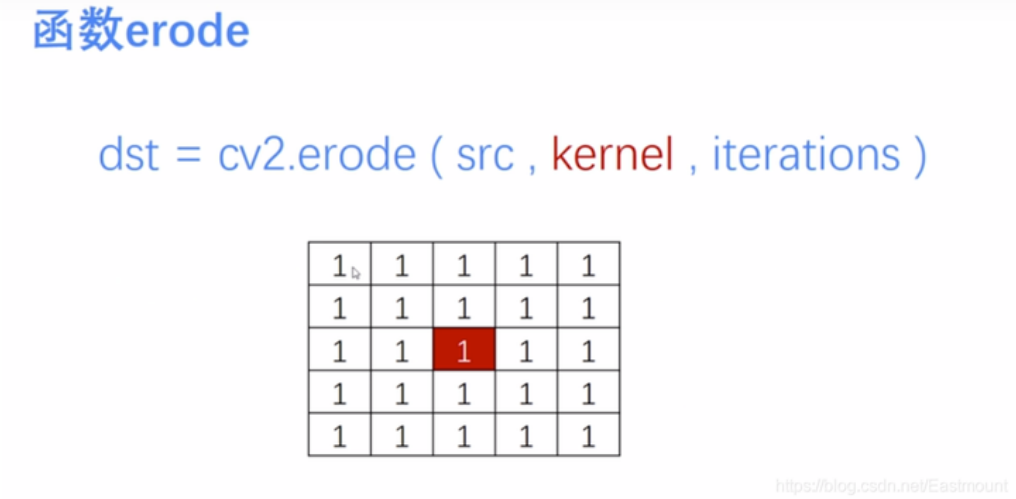


被扫描到的原始图像中的像素点，只有当卷积核对应的元素值均为1时，其值才为1，否则其值修改为0。换句话说，遍历到的黄色点位置，其周围全部是白色，保留白色，否则变为黑色，图像腐蚀变小。



2.函数原型

图像腐蚀主要使用的函数为erode，其原型如下：



注意：迭代次数默认是1，表示进行一次腐蚀，也可以根据需要进行多次迭代，进行多次腐蚀。

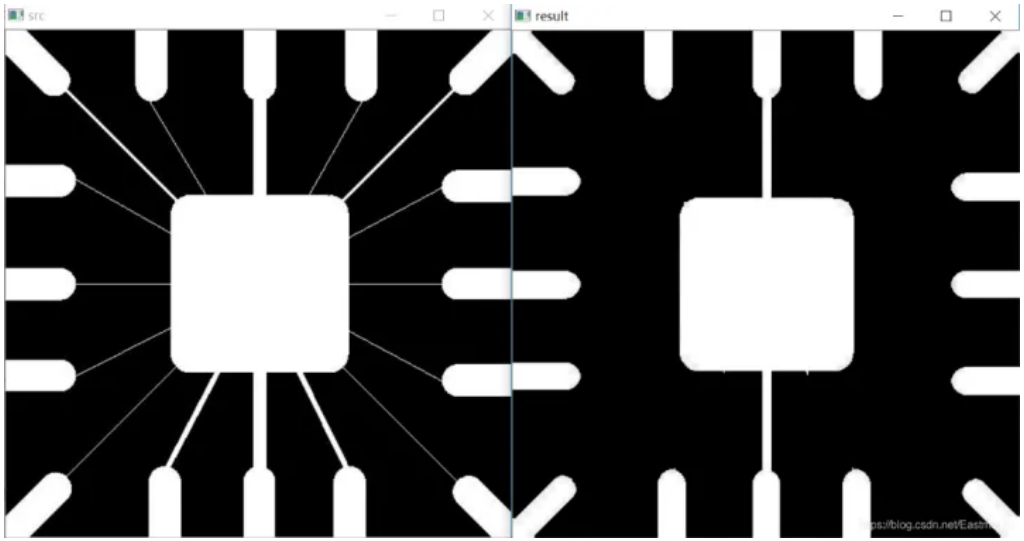
3.代码实现

完整代码如下所示：

```
#encoding:utf-8
import cv2
import numpy as np

#读取图片
src = cv2.imread('test01.jpg', cv2.IMREAD_UNCHANGED)

#设置卷积核
kernel = np.ones((5,5), np.uint8)

#图像腐蚀处理
erosion = cv2.erode(src, kernel)

#显示图像
```

```
cv2.waitKey(0)
cv2.destroyAllWindows()
```

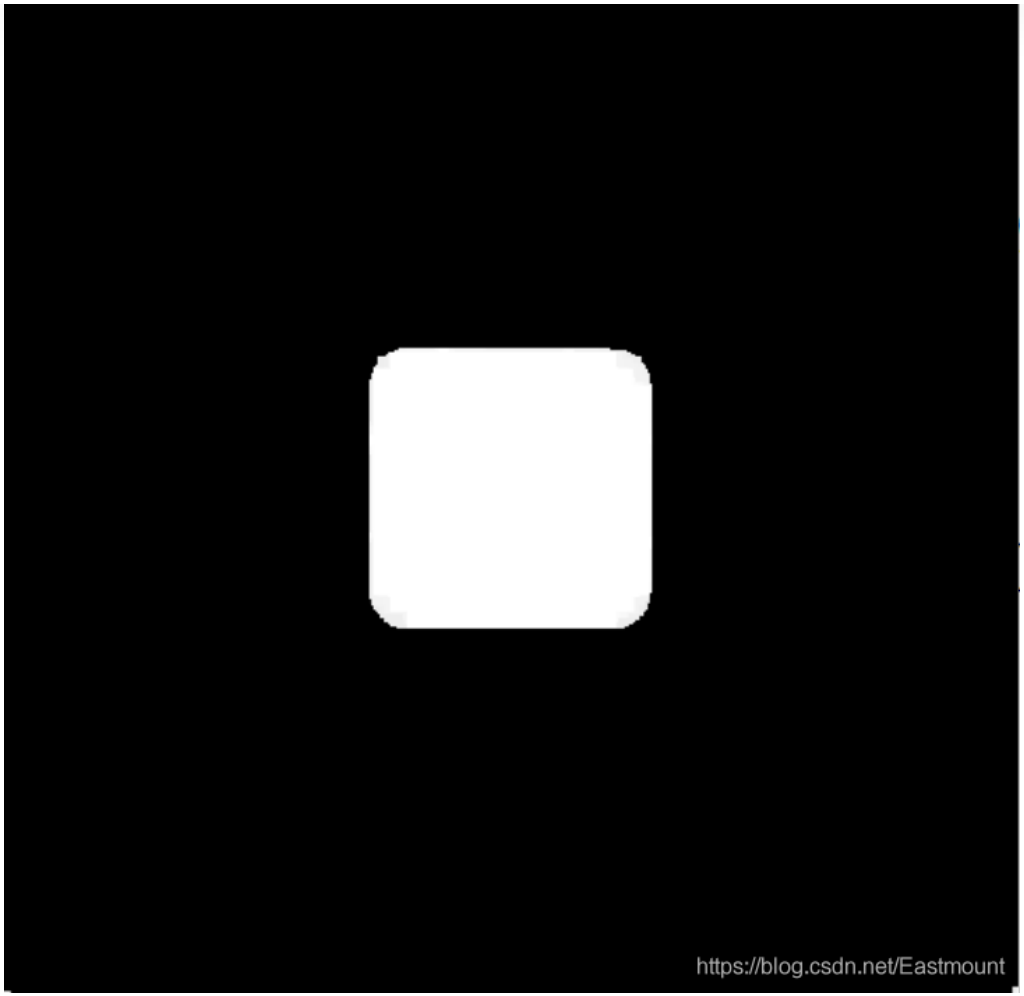
输出结果如下图所示：



由图可见，干扰的细线被进行了清洗，但仍然有些轮廓，此时可设置迭代次数进行腐蚀。

erosion = cv2.erode(src, kernel,iterations=9)

输出结果如下图所示：



三. 图像膨胀代码实现

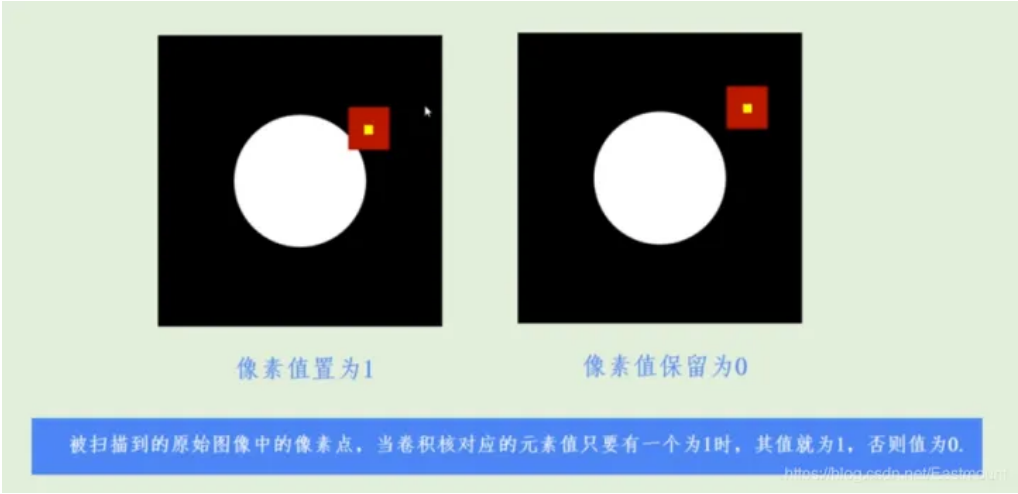
1.基础理论

图像膨胀是腐蚀操作的逆操作，类似于“领域扩张”，将图像中的高亮区域或白色部分进行扩张，其运行结果图比原图的高亮区域更大，线条变粗了，主要用于去噪

它也包括两个输入对象：

(1)二值图像或原始图像
(2)卷积核

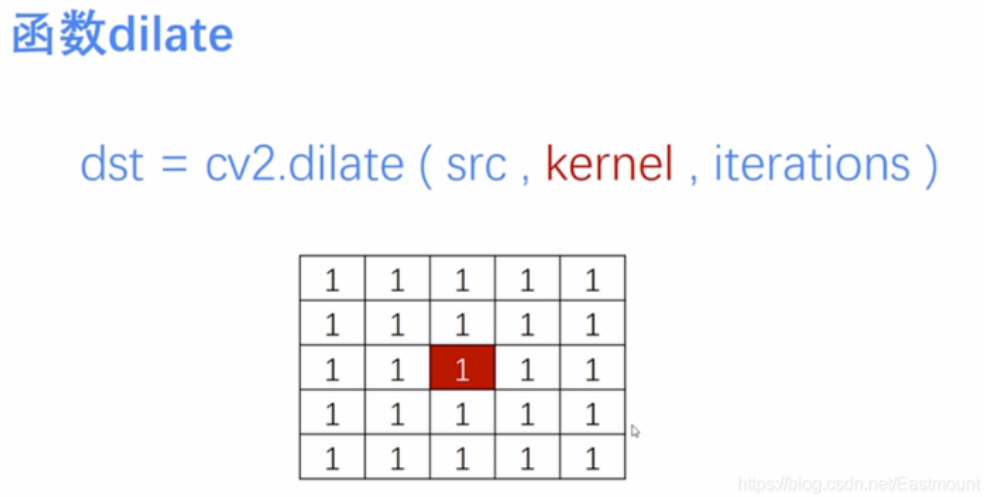
卷积核是腐蚀中的关键数组，采用numpy库可以生成。卷积核的中心点逐个像素扫描原始图像，如下图所示：



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，否则为0。**

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

参数dst表示处理的结果，src表示原图像，kernel表示卷积核，iterations表示迭代次数。下图表示5*5的卷积核，可以采用函数 np.ones((5,5), np.uint8) 构建。



注意：迭代次数默认是1，表示进行一次膨胀，也可以根据需要进行多次迭代，进行多次膨胀。通常进行1次膨胀即可。

3.代码实现

完整代码如下所示：

```
#encoding:utf-8
import cv2
import numpy as np

#读取图片
src = cv2.imread('test02.png', cv2.IMREAD_UNCHANGED)

#设置卷积核
kernel = np.ones((5,5), np.uint8)
```

```
#显示图像
cv2.imshow("src", src)
cv2.imshow("result", erosion)

#等待显示
cv2.waitKey(0)
cv2.destroyAllWindows()
```

输出结果如下所示：

图像去噪通常需要先腐蚀后膨胀，这又称为开运算，下篇文章将详细介绍。如下图所示：

erosion = cv2.erode(src, kernel)
result = cv2.dilate(erosion, kernel)