

该系列文章是讲解Python OpenCV图像处理知识，前期主要讲解图像入门、OpenCV基础用法，中期讲解图像处理的各种算法，包括图像锐化算子、图像增强技术、图像分割等，后期结合深度学习研究图像识别、图像分类应用。希望文章对您有所帮助，如果有不足之处，还请海涵~

该系列在github所有源代码：<https://github.com/eastmountyxz/ImageProcessing-Python>
PS：请求帮忙点个Star，哈哈，第一次使用Github，以后会分享更多代码，一起加油。

同时推荐作者的C++图像系列知识：

[数字图像处理] 一.MFC详解显示BMP格式图片

[数字图像处理] 二.MFC单文档分割窗口显示图片

[数字图像处理] 三.MFC实现图像灰度、采样和量化功能详解

[数字图像处理] 四.MFC对话框绘制灰度直方图

[数字图像处理] 五.MFC图像点运算之灰度线性变化、灰度非线性变化、阈值化和均衡化处理详解

[数字图像处理] 六.MFC空间几何变换之图像平移、镜像、旋转、缩放详解

[数字图像处理] 七.MFC图像增强之图像普通平滑、高斯平滑、Laplacian、Sobel、Prewitt锐化详解

前文参考：

[Python图像处理] 一.图像处理基础知识及OpenCV入门函数

[Python图像处理] 二.OpenCV+Numpy库读取与修改像素

[Python图像处理] 三.获取图像属性、兴趣ROI区域及通道处理

[Python图像处理] 四.图像平滑之均值滤波、方框滤波、高斯滤波及中值滤波

[Python图像处理] 五.图像融合、加法运算及图像类型转换

[Python图像处理] 六.图像缩放、图像旋转、图像翻转与图像平移

[Python图像处理] 七.图像阈值化处理及算法对比

[Python图像处理] 八.图像腐蚀与图像膨胀

数学形态学（Mathematical morphology）是一门建立在格论和拓扑学基础之上的图像分析学科，是数学形态学图像处理的基本理论。其基本的运算包括：腐蚀和膨胀、开运算和闭运算、骨架抽取、极限腐蚀、击中击不中变换、形态学梯度、Top-hat变换、颗粒分析、流域变换等。

本篇文章主要讲解Python调用OpenCV实现图像形态学转化，包括图像开运算、图像闭运算和梯度运算，基础性知识希望对您有所帮助。

1.图像开运算

2.图像闭运算

3.图像梯度运算

PS：文章参考自己以前系列图像处理文章及OpenCV库函数，同时部分参考网易云lilizong老师的视频，推荐大家去学习。同时，本篇文章涉及到《计算机图形学》基础知

识，请大家下来补充。

PSS：2019年1~2月作者参加了CSDN2018年博客评选，希望您能投出宝贵的一票。我是59号，Eastmount，杨秀璋。投票地址：

https://bss.csdn.net/m/topic/blog_star2018/index



五年来写了314篇博客，12个专栏，是真的热爱分享，热爱CSDN这个平台，也想帮助更多的人，专栏包括Python、数据挖掘、网络爬虫、图像处理、C#、Android等。现在也当了两年老师，更是觉得有义务教好每一个学生，让贵州学子好好写点代码，学点技术，"师者，传到授业解惑也"，提前祝大家新年快乐。2019我们携手共进，为爱而生。

一. 图像开运算

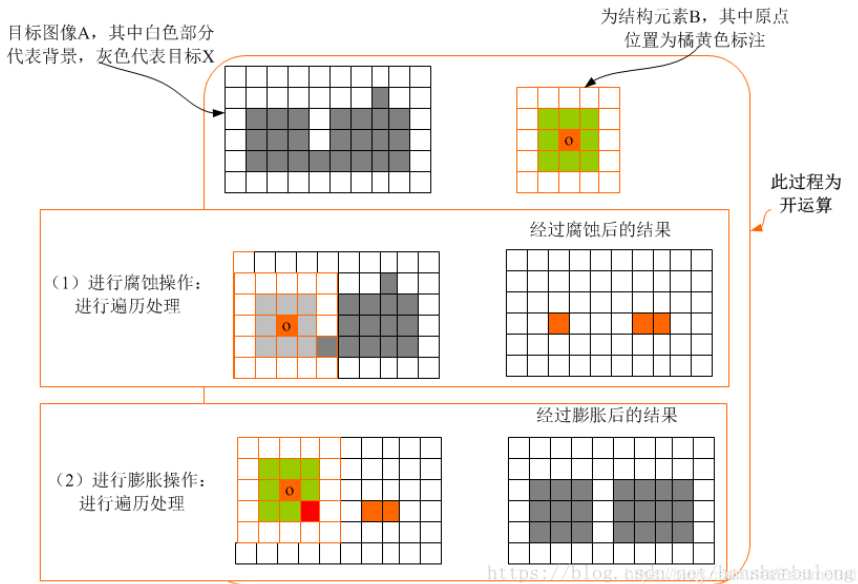
1.基本原理

图像开运算是图像依次经过腐蚀、膨胀处理后的过程。图像被腐蚀后，去除了噪声，但是也压缩了图像；接着对腐蚀过的图像进行膨胀处理，可以去除噪声，并保留原有图像。如下图所示：

开运算 = 膨胀(腐蚀)



下图是hanshanbuleng博主提供的开运算效果图，推荐大家学习他的文章。
<https://blog.csdn.net/hanshanbuleng/article/details/80657148>



2.函数原型

图像开运算主要使用的函数morphologyEx，它是形态学扩展的一组函数，其参数cv2.MORPH_OPEN对应开运算。其原型如下：

```
dst = cv2.morphologyEx(src, cv2.MORPH_OPEN, kernel)
```

参数dst表示处理的结果，src表示原图像，cv2.MORPH_OPEN表示开运算，kernel表示卷积核。下图表示5*5的卷积核，可以采用函数 np.ones((5,5), np.uint8) 构建。



函数morphologyEx

```
opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
```

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

<https://blog.csdn.net/Eastmount>

运行结果如下图所示：



原始图像



开运算图像

<https://blog.csdn.net/Eastmount>

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.png', cv2.IMREAD_UNCHANGED)

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

# 图像开运算
result = cv2.morphologyEx(src, cv2.MORPH_OPEN, kernel)

# 显示图像
```

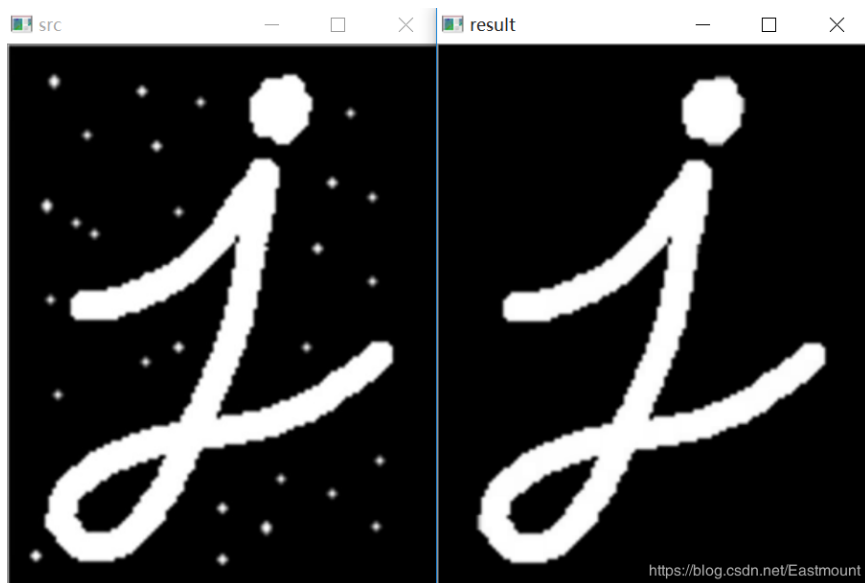
```
cv2.imshow("src", src)
cv2.imshow("result", result)

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

输出结果如下图所示，可以看到噪声已经被去除了。



但是结果result中仍然有部分噪声，如果想去除更彻底将卷积设置为10*10的。 kernel = np.ones((10,10), np.uint8) result = cv2.morphologyEx(src, cv2.MORPH_OPEN, kernel)



二. 图像闭运算

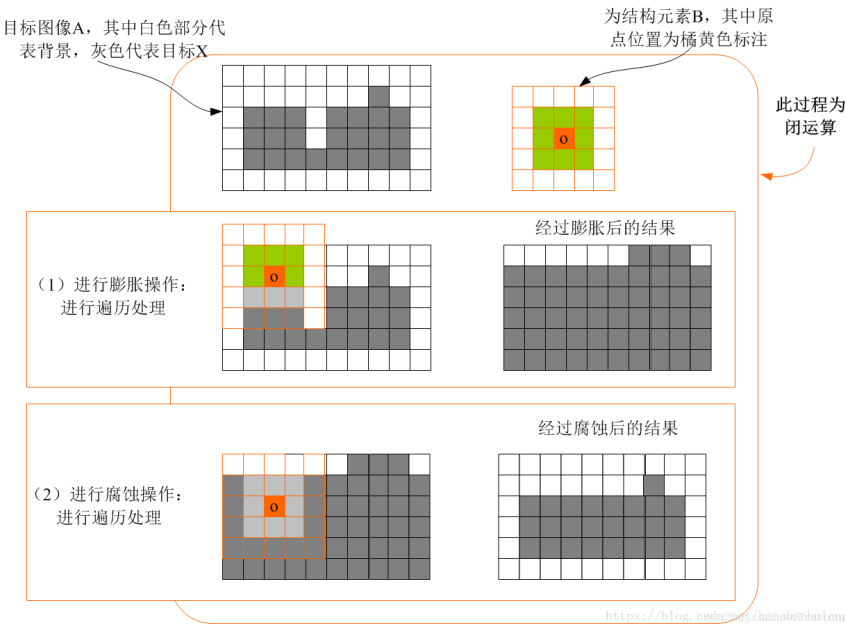
1.基本原理

图像闭运算是图像依次经过膨胀、腐蚀处理后的过程。图像先膨胀，后腐蚀，它有助于关闭前景物体内部的小孔，或物体上的小黑点。如下图所示：

$$\text{闭运算}(\text{img}) = \text{腐蚀}(\text{膨胀}(\text{img}))$$



下图是hanshanbuleng博主提供的开运算效果图，推荐大家学习他的文章。



<https://blog.csdn.net/hanshanbuleng/article/details/80657148>

2.函数原型

图像闭运算主要使用的函数morphologyEx，其原型如下：

```
dst = cv2.morphologyEx(src, cv2.MORPH_CLOSE, kernel)
```

参数dst表示处理的结果，src表示原图像，cv2.MORPH_CLOSE表示闭运算，kernel表示卷积核。下图表示5*5的卷积核，可以采用函数 np.ones((5,5), np.uint8) 构建。

函数morphologyEx

```
closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
```

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

<https://blog.csdn.net/Eastmount>

运行结果如下图所示：



原始图像



闭运算图像

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('test03.png', cv2.IMREAD_UNCHANGED)

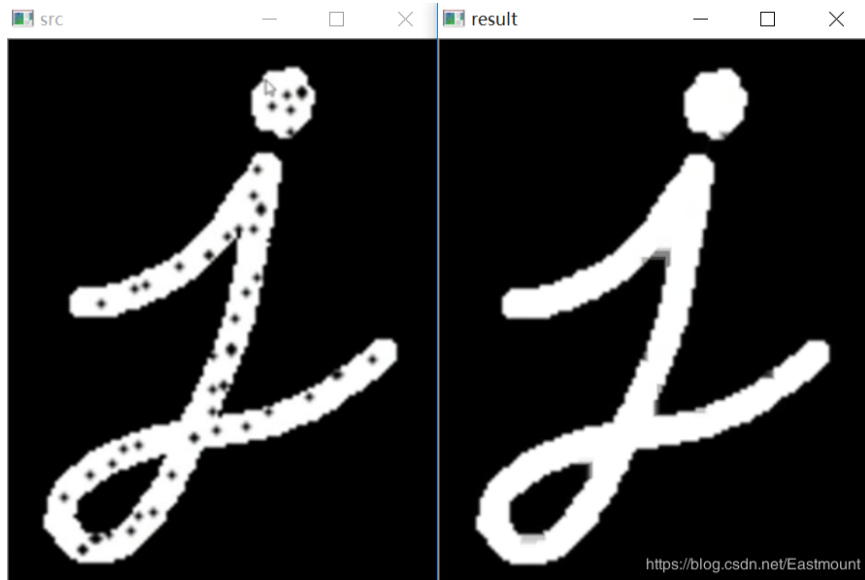
#设置卷积核
kernel = np.ones((10,10), np.uint8)

#图像闭运算
result = cv2.morphologyEx(src, cv2.MORPH_CLOSE, kernel)

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

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

输出结果如下图所示，可以看到中间的噪声去掉。

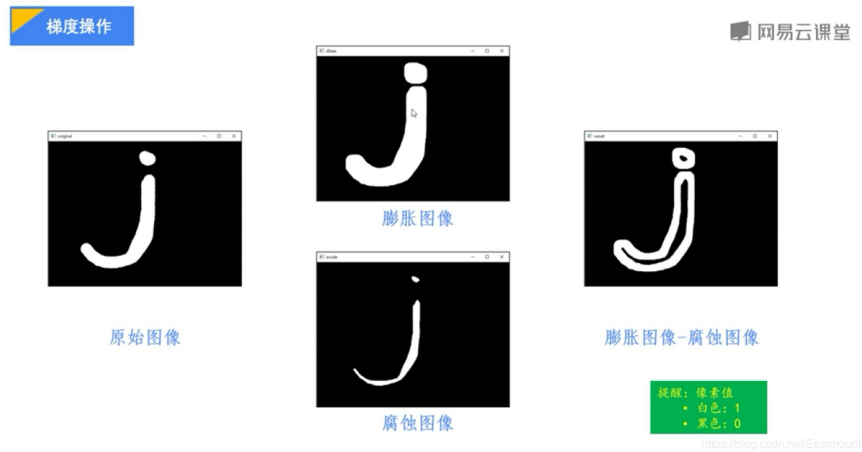


三. 图像梯度运算

1. 基本原理

图像梯度运算是膨胀图像减去腐蚀图像的结果，得到图像的轮廓，其中二值图像1表示白色点，0表示黑色点。如下图所示：

$$\text{梯度运算}(\text{img}) = \text{膨胀}(\text{img}) - \text{腐蚀}(\text{img})$$



2.函数原型

图像梯度运算主要使用的函数`morphologyEx`，参数为`cv2.MORPH_GRADIENT`。其原型如下：

```
dst = cv2.morphologyEx(src, cv2.MORPH_GRADIENT, kernel)
```

参数`dst`表示处理的结果，`src`表示原图像，`cv2.MORPH_GRADIENT`表示梯度运算，`kernel`表示卷积核。5*5的卷积核可以采用函数 `np.ones((5,5), np.uint8)` 构建。运行结果如下图所示：



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('test04.png', cv2.IMREAD_UNCHANGED)

# 设置卷积核
kernel = np.ones((10,10), np.uint8)

# 图像闭运算
result = cv2.morphologyEx(src, cv2.MORPH_GRADIENT, kernel)

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

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

输出结果如下图所示，可以看到中间的噪声去掉。



希望文章对大家有所帮助，如果有错误或不足之处，还请海涵。最近经历的事情太多，有喜有悲，关闭了朋友圈，希望通过不断学习和写文章来忘记烦劳，将忧郁转换为动力，每周学习都记录下来，加油!!!

(By: Eastmount 2018-11-02 中午12点 <https://blog.csdn.net/Eastmount/>)