

[Python图像处理] 十.形态学之图像顶帽运算和黑帽运算

原创

Eastmount

2018-11-04 00:03:23

11178

收藏 27

编辑 版权

分类专栏:

Python图像处理及图像识别

文章标签:

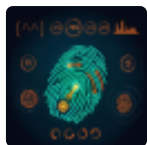
python

图像处理

黑帽运算

顶帽运算

OpenCV



Python图像处理及图像识别

¥9.90

本专栏主要结合Python语言讲述图像处理相关的知识,从二值图像、灰度图像到RGB图像基础知识,再到常见的图像处理算法,包括:灰度...



Eastmount

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

[Python图像处理] 九.形态学之图像开运算、闭运算、梯度运算

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,包括图像顶帽运算和图像黑帽运算,基础性知识希望对您有所帮助。

1.图像顶帽运算

2. 图像黑帽运算

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



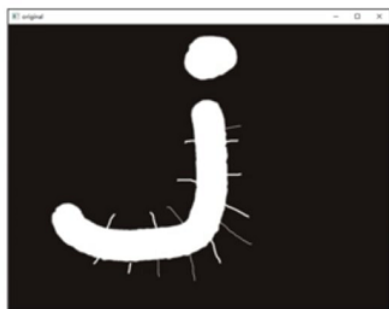
五年来写了314篇博客，12个专栏，是真的热爱分享，热爱CSDN这个平台，也想帮助更多的人，专栏包括Python、数据挖掘、网络爬虫、图像处理、C#、Android等。现在也当了两年老师，更是觉得有义务教好每一个学生，让贵州学子好好写点代码，学点技术，"师者，传到授业解惑也"，提前祝大家新年快乐。2019我们携手共进，为爱而生。

一. 图像顶帽运算

1. 基本原理

图像顶帽（或图像礼帽）运算是原始图像减去图像开运算的结果，得到图像的噪声。如下图所示：

$$\text{顶帽运算}(\text{img}) = \text{原始图像}(\text{img}) - \text{开运算}(\text{img})$$



原始图像



开运算图像



礼帽图像

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

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_TOPHAT对应开运算。其原型如下：

dst = cv2.morphologyEx(src, cv2.MORPH_TOPHAT, kernel)

参数dst表示处理的结果，src表示原图像，cv2.MORPH_TOPHAT表示顶帽运算，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

result = cv2.morphologyEx(img, cv2.MORPH_TOPHAT, kernel)

result , 礼帽结果

img

, 源图像

cv2.MORPH_TOPHAT

, 礼帽

kernel

, 卷积核

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

卷积如下图所示：

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

<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)

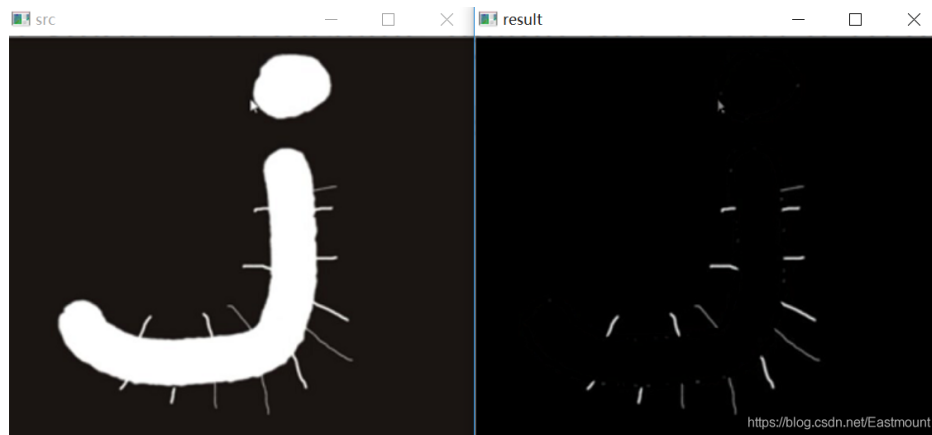
#设置卷积核
kernel = np.ones((5,5), np.uint8)

#图像顶帽运算
result = cv2.morphologyEx(src, cv2.MORPH_TOPHAT, kernel)

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

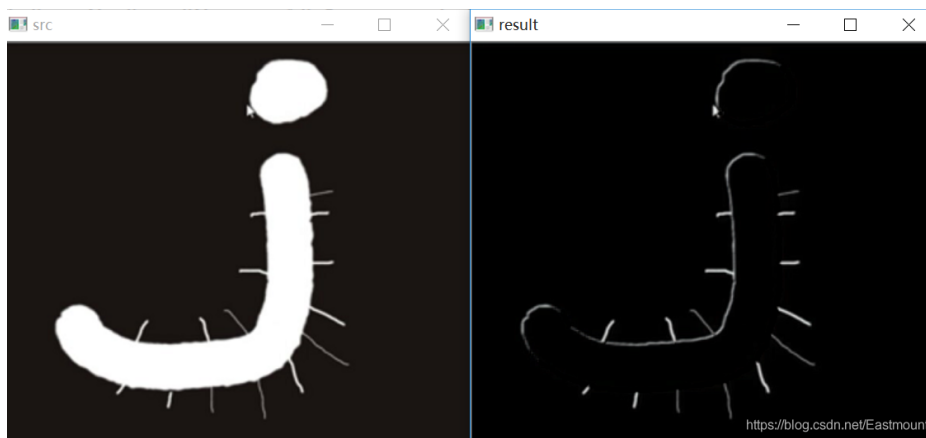
#等待显示
cv2.waitKey(0)
cv2.destroyAllWindows()
```

输出结果如下图所示，可以看到外部噪声被提取出来。



如果想获取更多的细节，可以将卷积设置为10*10，如下图所示：

```
kernel = np.ones((10,10), np.uint8)
result = cv2.morphologyEx(src, cv2.MORPH_TOPHAT, kernel)
```



二. 图像黑帽运算

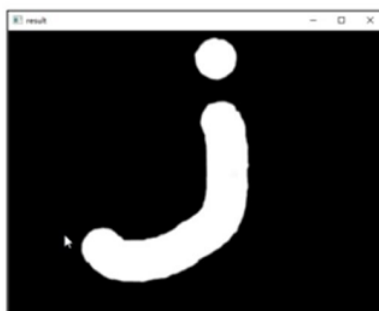
1. 基本原理

图像黑帽运算是图像闭运算操作减去原始图像的结果，得到图像内部的小孔，或者前景色中的小黑点。如下图所示：

$$\text{黑帽运算}(\text{img}) = \text{闭运算图像}(\text{img}) - \text{原始图像}(\text{img})$$



原始图像



闭运算图像



黑帽图像

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_BLACKHAT对应开运算。其原型如下：

dst = cv2.morphologyEx(src, cv2.MORPH_BLACKHAT, kernel)

参数dst表示处理的结果，src表示原图像，cv2.MORPH_BLACKHAT表示黑帽运算，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

```
result = cv2.morphologyEx(img, cv2.MORPH_BLACKHAT, kernel)
```

result , 黑帽结果

img , 源图像

cv2.MORPH_BLACKHAT , 黑帽

kernel , 卷积核

<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('test02.png', cv2.IMREAD_UNCHANGED)

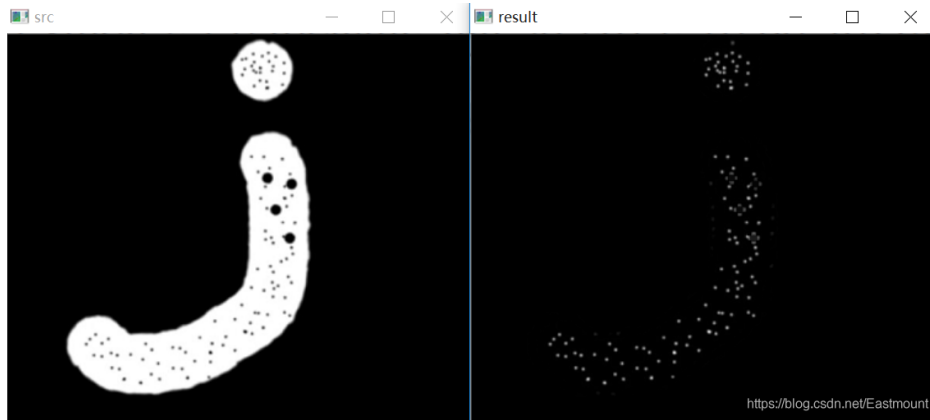
#设置卷积核
kernel = np.ones((5,5), np.uint8)

#图像黑帽运算
result = cv2.morphologyEx(src, cv2.MORPH_BLACKHAT, 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, 可以看到图像内部黑点被提取出来。



但内部比较大的四个黑点没有被提取，如果想获取更多的细节，可以将卷积设置为10*10，如下图所示：

```
kernel = np.ones((10,10), np.uint8)
```

```
result = cv2.morphologyEx(src, cv2.MORPH_BLACKHAT, kernel)
```



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 2018-11-04 深夜12点 <https://blog.csdn.net/Eastmount/>)

2020年8月18新开的“娜璋AI安全之家”，主要围绕Python大数据分析、网络空间安全、人工智能、Web渗透及攻防技术进行讲解，同时分享CCF、SCI、南核北核论文的算法实现。娜璋之家会更加系统，并重构作者的所有文章，从零讲解Python和安全，写了近十年文章，真心想把自己所学所感所做分享出来，还请各位多多指教，真诚邀请您的关注！谢谢。

