



周围的人都比你厉害，你才会慢慢变强

公告

昵称： 山上有风景
园龄： 1年11个月
粉丝： 170
关注： 19
+加关注

< 2019年12月 >

日	一	二	三	四	五	六
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

搜索

找找看

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

STL(18)
SDN(15)
ThinkPHP3.2(1)

积分与排名

积分 - 211835
排名 - 1998

随笔分类

C/C++(74)
Html(2)
Java(33)
Javascript(19)
OpenCV(29)
PHP(2)
Python(155)

OpenCV---ROI（region of interest）和泛洪填充

目录

一：ROI

（一）获取感兴趣区域

还原操作：

二：泛洪填充floodFill

一：ROI

感兴趣区(Region of Interest,ROIs) 是图像的一部分，它通过在图像上选择或使用诸如设定阈值(thresholding) 或者从其他文件(如矢量> 转换获得等方法生成。感兴趣区可以是点、线、面不规则的形状，通常用来作为图像分类的样本、掩膜、裁剪区及其他操作。

（一）获取感兴趣区域

```
src = cv.imread("./1.png") #读取图片
cv.namedWindow("input image",cv.WINDOW_AUTOSIZE) #创建GUI窗口,形式为自适应
cv.imshow("input image",src) #通过名字将图像和窗口联系
print(src.shape)
flower = src[200:400,100:250]
cv.imshow("flower",flower)
```

原图：

0

0

关注 | 顶部 | 评论

- STL泛型编程(18)
- 单片机笔记（复习用）(3)
- 计算机网络(32)
- 其他知识(14)
- 设计模式(27)
- 数据结构(57)
- 数据库(8)
- 算法习题(43)
- 算法训练营
- 随笔所想(4)
- 图形界面编程
- 正则表达式(2)
- 转载推文(4)

随笔档案

2019年11月(5)

2019年10月(32)

2019年9月(21)

2019年7月(10)

2019年5月(8)

2019年4月(25)

2019年3月(8)

2019年2月(1)

2019年1月(12)

2018年12月(19)

2018年9月(5)

2018年8月(95)

2018年7月(78)

2018年6月(26)

2018年5月(17)

2018年4月(22)

2018年3月(111)

最新评论

1. Re:python---websocket的使用

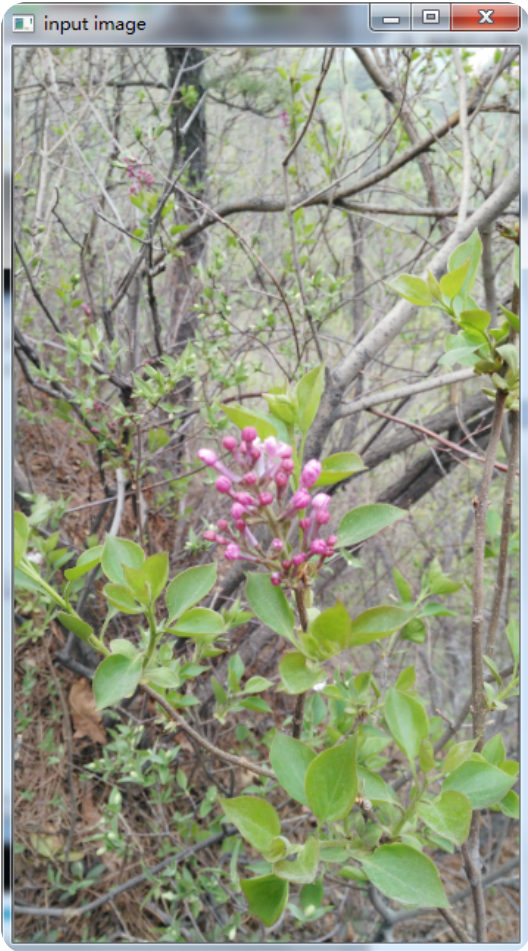
网上的方法都不行，换成gbk会报如下的错：
IndexError: index out of range
--缘分天空0320

2. Re:python---websocket的使用

@ 缘分天空0320太久没用，忘了。应该是字符编码问题吧。这类问题网上应该可以很容易找到方法解决。你看看Python的默认编码和代码是不是一致。一般就是gbk和utf8之间出错...
--山上有风景

3. Re:python---websocket的使用


您好，用了您的代码，报如下错误，麻烦问下如何解决呢？
Traceback (most recent call last): File "server3.py", line 101, in <module...
--缘分天空0320



感兴趣区域:




还原操作:



```
src = cv.imread("./1.png") #读取图片
cv.namedWindow("input image",cv.WINDOW_AUTOSIZE) #创建GUI窗口,形式为自适应
cv.imshow("input image",src) #通过名字将图像和窗口联系
print(src.shape)
flower = src[200:400,100:250]
cv.imshow("flower",flower)

gray = cv.cvtColor(flower,cv.COLOR_BGR2GRAY) #获取一张灰度图像,单一通道
backface = cv.cvtColor(gray,cv.COLOR_GRAY2BGR) #单一通道转3通道
src[200:400,100:250] = backface
cv.imshow("new image",src)

cv.waitKey(0) #等待用户操作,里面等待参数是毫秒,我们填写0,代表是永远,等待用户操作
cv.destroyAllWindows() #销毁所有窗口
```



00

关注 | 顶部 | 评论

4. Re:数据结构（六）查找---线性索引查找
- 请问最后倒序排序的那个代码怎么实现的？可以发一下吗？
- Viki-
5. Re:SDN实验---Ryu的源码分析
- @ 山上有风景谢谢！ ...
- iRoy_33

阅读排行榜

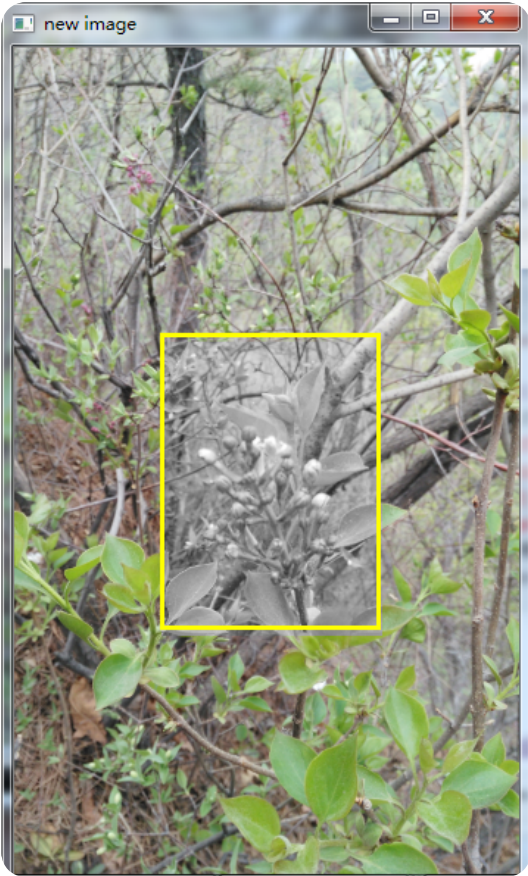
1. python---websocket的使用(17249)
2. OpenCV---图像二值化(12510)
3. OpenCV---模板匹配matchTemplate(11531)
4. OpenCV---直线检测(8206)
5. python---基础知识回顾（九）图形用户界面-----WxPython(7986)

评论排行榜

1. python---基础知识回顾（九）图形用户界面-----Tkinter(4)
2. python---websocket的使用(3)
3. SDN实验---Ryu的源码分析(3)
4. 数据结构（三）串---KMP模式匹配算法之获取next数组(2)
5. 数据结构（四）树---树的存储结构(1)

推荐排行榜

1. 数据结构（七）排序---堆排序(11)
2. python---aiohttp的使用(6)
3. python---websocket的使用(4)
4. python---基础知识回顾（九）图形用户界面-----WxPython（3）
5. OpenCV---图像金字塔原理(3)



二：泛洪填充floodFill

推文：图像处理-----泛洪填充算法(Flood Fill Algorithm) 油漆桶功能

(一)

所谓的floodFill 漫水填充就是在一张图片中，和种子点像素相差在 $[-loDiff, +upDiff]$ 的时候就用newVal来填充这个点。

```
def floodFill(image, mask, seedPoint, newVal, loDiff=None, upDiff=None, flags=None): # real signature
unknown; restored from __doc__
```

- 1.操作的图像，
- 2.掩码，
- 3.起始像素值， #我们指定的第一个点的三个通道值
- 4.填充的颜色，
- 5.填充颜色的低值， #低值三通道
- 6.填充颜色的高值， #高值三通道
- 7.填充的方法

参数5.填充颜色的低值就是：参数3的三通道值 减去 参数5
参数6.填充颜色的高值就是：参数3的三通道值 加上 参数6

7.填充的方法
彩色图像一般是FLOODFILL_FIXED_RANGE 指定颜色填充

还有一种是FLOODFILL_MASK_ONLY, mask的指定的位置为零时才填充，不为零不填充

```
def fill_color_demo(image):
    copyImg = image.copy() #从源图像上克隆出一张新图像
```

00

关注 | 顶部 | 评论

```
h,w = image.shape[:2]
mask = np.zeros([h+2,w+2],np.uint8) #+2是为了使边缘像素也被修改
print(image[30,30]) #[187 192 191] 用它进行操作
cv.floodFill(copyImg,mask,(30,30),(0,255,255),(50,50,50),(50,50,50),cv.FLOODFILL_FIXED_RANGE)
cv.imshow("fill_color_demo",copyImg)
```



```
src = cv.imread("./1.png") #读取图片
fill_color_demo(src)
```



(二) FLOODFILL_MASK_ONLY 我们可以自定义填充图像，掩码为1不填充，为0进行填充



```
def fill_binary():
    image = np.zeros([400,400,3],np.uint8)
    image[100:300,100:300,:]=255
    cv.imshow("fill_banary",image)

    mask = np.ones([402,402,1],np.uint8)
    mask[101:301,101:301] = 0 #这里设置需要加1--->101:301 这是我们填充的图像掩码为0
    cv.floodFill(image,mask,(200,200),(0,255,0),cv.FLOODFILL_MASK_ONLY) #最好写到我们填充的图像中间
    cv.imshow("fill_banary2",image)
```



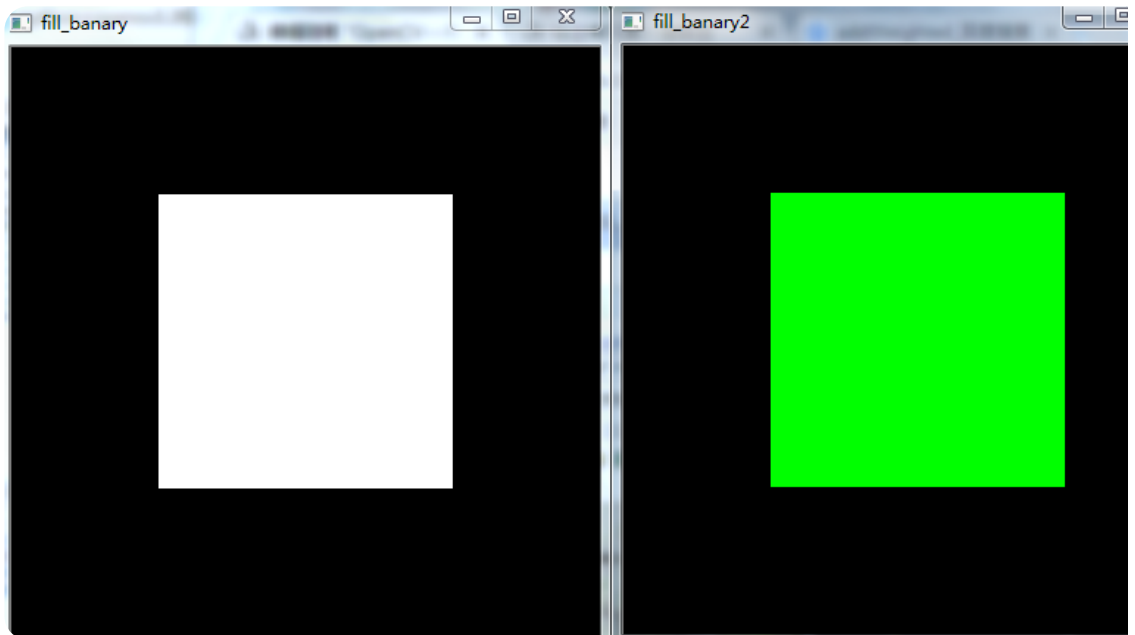
```
fill_binary()
```

- 1.操作的图像,
- 2.掩码, 为1不填充, 为0才进行填充
- 3.填充图像的中央部分
- 4.填充的颜色, 5.填充的方法

0

0

关注 | 顶部 | 评论



- ▶ 泛洪填充，如何填充一个对象内部区域
 - FLOODFILL_FIXED_RANGE - 改变图像，泛洪填充
 - FLOODFILL_MASK_ONLY - 不改变图像、只填充遮罩层本身、忽略新的颜色值参数
- ▶ `floodFill(Mat image, Mat mask, Point seedPoint, Scalar newVal)`
- ▶ `floodFill(image, mask, seedPoint, newVal, rect, loDiff, upDiff, flags)`
 $src(seed.x, seed.y) - loDiff \leq src(x, y) \leq src(seed.x, seed.y) + upDiff$

作者：山上有风景

欢迎任何形式的转载，但请务必注明出处。

限于本人水平，如果文章和代码有表述不当之处，还请不吝赐教。

分类：OpenCV

好文要顶

关注我

收藏该文



山上有风景

关注 - 19

粉丝 - 170

+加关注

« 上一篇：OpenCV---像素运算

» 下一篇：OpenCV---模糊操作

posted @ 2018-07-03 23:09 山上有风景 阅读(290) 评论(0) 编辑 收藏

0

0

返回顶部

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

关注 | 顶部 | 评论

Copyright © 2019 山上有风景

Powered by .NET Core 3.1.0 on Linux

0

0

[关注](#) | [顶部](#) | [评论](#)