[Python图像处理] 二.OpenCV+Numpy库读取与修改像素

原创 Eastmount 2018-08-28 08:33:54 20610 收藏 82 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 OpenCV 读取像素 修改像素

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

本篇文章主要讲解 OpenCV+Numpy 图像处理基础知识，包括读取像素和修改像素。知识点如下：
1.传统读取像素方法
2.传统修改像素方法
3.Numpy读取像素方法
4.Numpy修改像素方法

PS: 文章也学习了网易云高登教育的知识，推荐大家学习。

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.传统读取像素方法

1.灰度图像，返回灰度值。

返回值=图像(位置参数)，例：p = img[88,142] print§

```
# -*- coding:utf-8 -*-
import cv2

#读取图片
img = cv2.imread("picture.bmp", cv2.IMREAD_UNCHANGED)

#灰度图像
p = img[88, 142]
print(p)

#显示图像
cv2.imshow("Demo", img)

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

#写入图像
cv2.imwrite("testyxz.jpg", img)
```

输出结果如下图所示：[131 131 131]，由于该图是24位BMP，B=G=R输出三个相同的结果，有的图像仅有一个像素点则输出一个值。

2.BGR图像，返回值为B、G、R的值。

例:

b = img[78, 125, 0] print(b)

g = img[78, 125, 1] print(g)

r = img[78,125, 2] print®

```
# -*- coding:utf-8 -*-
import cv2

#读取图片
img = cv2.imread("test.jpg", cv2.IMREAD_UNCHANGED)

#BGR图像
b = img[78, 125, 0]
print(b)
g = img[78, 125, 1]
print(g)
r = img[78, 125, 2]
print(r)

#方法二
bgr = img[78, 125]
print(bgr)

#显示图像
cv2.imshow("Demo", img)

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

#写入图像
cv2.imwrite("testyxz.jpg", img)
```

输出像素和图像如下所示:

155

104

61

[155 104 61]

二.传统修改像素方法

1.修改单个像素值

BGR图像可以通过位置参数直接访问像素值并进行修改，输出结果如下所示：

```
# -*- coding:utf-8 -*-
import cv2

#读取图片
img = cv2.imread("test.jpg", cv2.IMREAD_UNCHANGED)

#BGR图像
print(img[78, 125, 0])
print(img[78, 125, 1])
print(img[78, 125, 2])

#修改像素
img[78, 125, 0] = 255
img[78, 125, 1] = 255
img[78, 125, 2] =255

print(img[78, 125])
img[78, 125] = [10, 10, 10]
print(img[78, 125, 0])
print(img[78, 125, 1])
print(img[78, 125, 2])


#方法二
print(img[78, 125])
img[78, 125] = [10, 10, 10]
print(img[78, 125])
```

输出结果如下所示，通过两种方法分别将B、G、R像素值修改为255和0。

155

104

61

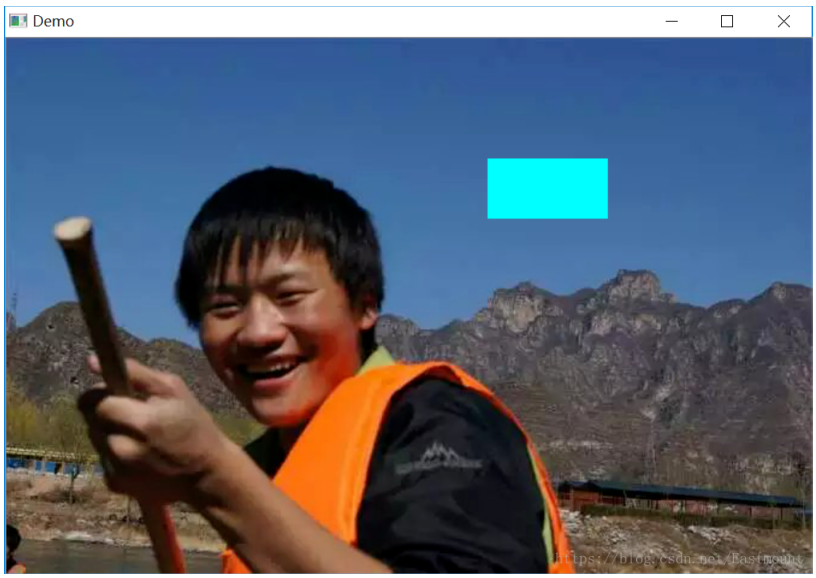
255
255
255
[255 255 255]
[10 10 10]

2.修改区域像素

通过访问图像数组的位置区域实现区域像素修改，比如 [100:150,400:500] 是访问第100到150行，400到500列的区域，再对该区域像素进行修改。代码如下所示：

```
# -*- coding:utf-8 -*-
import cv2

#读取图片
img = cv2.imread("test.jpg", cv2.IMREAD_UNCHANGED)

#BGR图像
img[100:150, 400:500] = [255, 255, 0]

#显示图像
cv2.imshow("Demo", img)

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

#写入图像
cv2.imwrite("testyxz.jpg", img)
```

输出结果如下图所示，[255, 255, 0]是浅蓝色。



三.Numpy读取像素方法

使用Numpy进行像素读取，调用方式如下：

返回值 = 图像.item(位置参数)

```
# -*- coding:utf-8 -*-
import cv2
import numpy

#读取图片
img = cv2.imread("test.jpg", cv2.IMREAD_UNCHANGED)

#Numpy读取像素
blue = img.item(78, 100, 0)
green = img.item(78, 100, 1)
red = img.item(78, 100, 2)
print(blue)
print(green)
print(red)

#显示图像
cv2.imshow("Demo", img)

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

输出结果如下，注意OpenCV读取图像通道是BGR，也可以转换成RGB在进行处理。
155
104
61

四.Numpy修改像素方法

使用Numpy的itemset函数修改像素，调用方式如下：
图像.itemset(位置, 新值)
例如：img.itemset((88,99), 255)

```
# -*- coding:utf-8 -*-
import cv2
import numpy

#读取图片
img = cv2.imread("test.jpg", cv2.IMREAD_UNCHANGED)

#Numpy读取像素
print(img.item(78, 100, 0))
print(img.item(78, 100, 1))
print(img.item(78, 100, 2))
img.itemset((78, 100, 0), 100)
img.itemset((78, 100, 1), 100)
img.itemset((78, 100, 2), 100)
print(img.item(78, 100, 0))
print(img.item(78, 100, 1))
print(img.item(78, 100, 2))
```

输出结果如下：
155
104

61
100
100
100

也可以同时输出B、G、R三个值，核心代码如下：

```
print(img[78, 78])
img.itemset((78, 78, 0), 0)
img.itemset((78, 78, 1), 0)
img.itemset((78, 78, 2), 0)
print(img[78, 78])
#[155 104  61]
#[0 0 0]
```

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。
（By：Eastmount 2018-08-28 早8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。