OpenCV-python 学习笔记 Opencv图像基本操作

1. openCV-python 环境搭建：

使用pip工具:

```
pip install opencv-python
```

2. OpenCV 的图像读取显示及保存

1. 导入图像：

   cv2.imread(文件名， 标记)

   ```
   import numpy as np
   import cv2

   img = cv2.imread('1.jpg', 0)  # 读取图像
   ```

   此时img属性：`<class 'numpy.ndarray'>`

   注：

   ```
   cv2.IMREAD_COLOR() # 读取彩色图像
   cv2.IMREAD_GRAYSCALE()  # 以灰度模式读入图像
   ```

2. 显示图像：

   cv2.imshow(窗口名， 图片)

   ```
   import numpu as np
   import cv2

   img = cv2.imread('1.jpg')  # 读取图像
   cv2.imshow('image', img)  # 显示图像
   cv2.waitKey(0)  # 等待键盘输入
   ```

   注：

   ```
   cv2.waitKey()  # 等待键盘输入，毫秒级
   cv2.destoryAllWindows()  # 删除任何我们建立的窗口
   ```

通常可以先创建一个窗口，再加载图像，这样可以决定窗口是否需要调整大小。

使用到函数`cv2.namedWindow()`，默认设定函数标签为`cv2.WINDOW_AUTOSIZE`，可以将标签改为`cv2.WINDOW_NORMAL`,这样就可以调整窗口大小了。

```
cv2.namedWindow('image',cv2.WINDOW_NORMAL)
cv2.imshow('image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

3. 保存图像：

cv2.imwrite(文件名， 图片)

```
cv2.imwrite('1.jpg', img)
```

3. OpenCV图像基本操作

1. 获取并修改像素值

读取一幅图像，根据像素的行和列的坐标获取它的像素值，对于RGB图像而言，返回RGB的值，对于灰度图则返回灰度值。

```
import numpy as np
import cv2

img = cv2.imread('1.jpg')
print("图片的规格：{}".format(np.shape(img)))
px = img[100,100]  # 提取单个像素
print(px)
blue = img[100, 100, 0]  # 提取单个像素blue通道的值
print(blue)
img[101, 101] = [255, 255, 255]  # 修改单点像素值
print(img[101])
```

输出：

```
图片的规格：(4000, 6360, 3)
[0 1 2]
0
[255 255 255]
```

2. 获取图像属性

图像属性包括：行，列，通道，图像数据类型，像素数目等

(1)img.shape

可以获得图像的形状，返回值是一个包含行数，列数，通道数的元组

例如：

```
img = cv2.imread('1.jpg')
print('图像的形状：{}'.format(img.shape))
```

输出：

```
图像的形状：(4000, 6360, 3)
```

如果图像是灰度图，返回值仅有行数和列数，所以通过检查返回值可以判断是灰度图还是彩色图

(2)img.size

可以返回图像的像素数目

```
img = cv2.imread('1.jpg')
print('图像的像素数目：{}'.format(img.size))
```

输出：

```
图像的像素数目：76320000
```

(3)img.dtype

返回图像的数据类型

在debug时很重要，因为OpenCV-Python代码中经常出现数据类型的不一致

例如：

```
img = cv2.imread('1.jpg')
print('图像的数据类型：{}'.format(img.dtype))
```

输出：

```
图像的数据类型：uint8
```

3. 图像ROI

对图像的特定区域操作。ROI是使用numpy索引来获得的。

例如：选择球的部分并拷贝到其他区域

```
import cv2
import numpy
img = cv2.imread('roi.jpg')
ball =img[20:30,30:30]
img[40:40,50:50]=ball
```

并且要先知道图像尺寸，以及你要移动的图像的像素坐标(可结合`matplotlib`进行使用)

4. 拆分及合并图像通道

使用`cv2.split()`函数

```
r,g,b=cv2.split(img)  #拆分
img=cv2.merge(r,g,b)  #合并
```

或者(推荐):

```
b=img[:,:,0]  #拆分b通道
```

5. 为图像扩边（填充）

想为图像周围建一个边可以使用`cv2.copyMakeBorder()`函数。这经常在卷积运算或0填充时被用到。具体参数如下:

(1) `src`输入图像

(2) `top,bottom,left,right`对应边界的像素数目

(3) `borderType`要添加哪种类型的边界:

a. `cv2.BORDER_CONSTANT`添加有颜色的常数值边界，还需要下一个参数(value)

b. `cv2.BORDER_REFLIECT`边界元素的镜像。例如：fedcba | abcdefgh | hgfedcb

c. `cv2.BORDER_101`或者`cv2.BORDER_DEFAULT`跟上面一样，但稍作改动，例如：gfedcb | abcdefgh | gfedcba

d. `cv2.BORDER_REPLICATE`复后一个元素。例如: aaaaaa|

abcdefgh|hhhhhhh

e. `cv2.BORDER_WRAP` 不知怎么了, 就像样: cdefgh| abcdefgh|abcdefg

f. `value`边界颜色