

6 图像的基础操作

1. 获取并修改图像中的像素点

我们可以通过行和列的坐标值获取该像素点的像素值。对于 BGR 图像，它返回一个蓝，绿，红值的数组；对于灰度图像，仅返回相应的强度值。使用相同的方法对像素值进行修改。

- 1) **img[x,y]**: 获取某个像素点的值，返回该像素点的 BGR 值。
- 2) **img[x,y,index]**: 获取像素点某个颜色通道的值，颜色通道顺序是 BGR。
- 3) **img[x,y]=[B,G,R]**: 修改该像素点的颜色通道数值。

```
img=cv2.imread("test.jpg")
px=img[100,100]
blue=img[100,100,0]
img[100,100]=[255,255,255]
```

2. 获取图像的属性

1) **shape**: 如果是彩色图像，获取图像的形状，返回包含行数、列数、通道数的数组；如果是二值图像或者灰度图像，则仅返回行数和列数。通过该属性的返回值是否包含通道数，可以判断一幅图像是灰度图像（或二值图像）还是彩色图像。

2) **size**: 返回图像的像素数目。其值为“行×列×通道数”，灰度图像或者二值图像的通道数为 1。

- 3) **dtype**: 返回图像的数据类型。

```
print("shape=",img.shape)#"(1600,1200,3)"
print("size=",img.size)#"57600000"
print("dtype=",img.dtype)#"uint8"
```

3.图像通道的拆分与合并

3.1 图像通道的拆分

split: 输入要拆分的图像，返回三个单独颜色通道的图片对象。

```
img=cv2.imread("test.jpg")
B,G,R=cv2.split(img)
cv2.imshow("blue",B)
cv2.imshow("green",G)
cv2.imshow("red",R)
```

3.2 图像通道的合并

merge: 把 B,G,R 单通道合并为三通道 BGR 图像。

```
img=cv2.imread("test.jpg")
B,G,R=cv2.split(img)
img=cv2.merge((B,G,R))
```

4.颜色空间的转换

OpenCV 中有 150 多种颜色转换方法，常用的有 BGR 转 GRAY 和 BGR 转 HSV。函数格式：**cvtColor(img,flag)**。

- 1) **img:** 进行颜色空间转换的图像。
- 2) **flag:** 转换类型，例如：**cv2.COLOR_BGR2HSV**，就是 BGR 转 HSV。

```
img=cv2.imread("test.jpg")
img=cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
```