

# openCV中图像的存储方式

原创

bestpig@1314

已于 2022-10-10 10:53:13 修改

阅读量1.7k

收藏 4

点赞数 2

版权

文章标签：

opencv

计算机视觉

python

opencv<sup>Q</sup>中灰度图实际在理论内存中的布局，如下图

	col0	col1	col2	col3	col4	col5	col6	col7	col8	col9	col10	col11	col12	col13	col14	col15	col16	col17	col18
row0																			
row1																			
row2																			
row3																			
row4																			
row5																			
row6																			
row7																			
row8																			
row9																			
row10																			
row11																			
row12																			
row13																			
row14																			
row15																			
row16																			
row17																			
row18																			
row19																			

其中的每个小格子就是一个图像中的像素值。0-255

opencv中彩色图实际在理论内存中的布局，如下图

	col0			col1			col2			col3			col4			col5		
row0	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row1	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row2	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row3	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row4	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row5	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row6	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row7	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row8	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row9	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row10	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row11	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row12	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row13	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row14	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row15	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row16	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row17	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row18	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R
row19	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R	B	G	R

```
1 | img = cv2.imread('xxx.jpg')
2 | print(img.shape)
```

输出的图像 shape 为 H \* W \* 3，即 高度 \* 宽度 \* 3通道 或者 行数 \* 列数 \* 3通道

图像的 shape 为 (1080, 1920, 3)，第一维 1080 表示有 1080 行，所以有 1080 个小的二维矩阵，**每个二维矩阵是三通道图像中的一行**

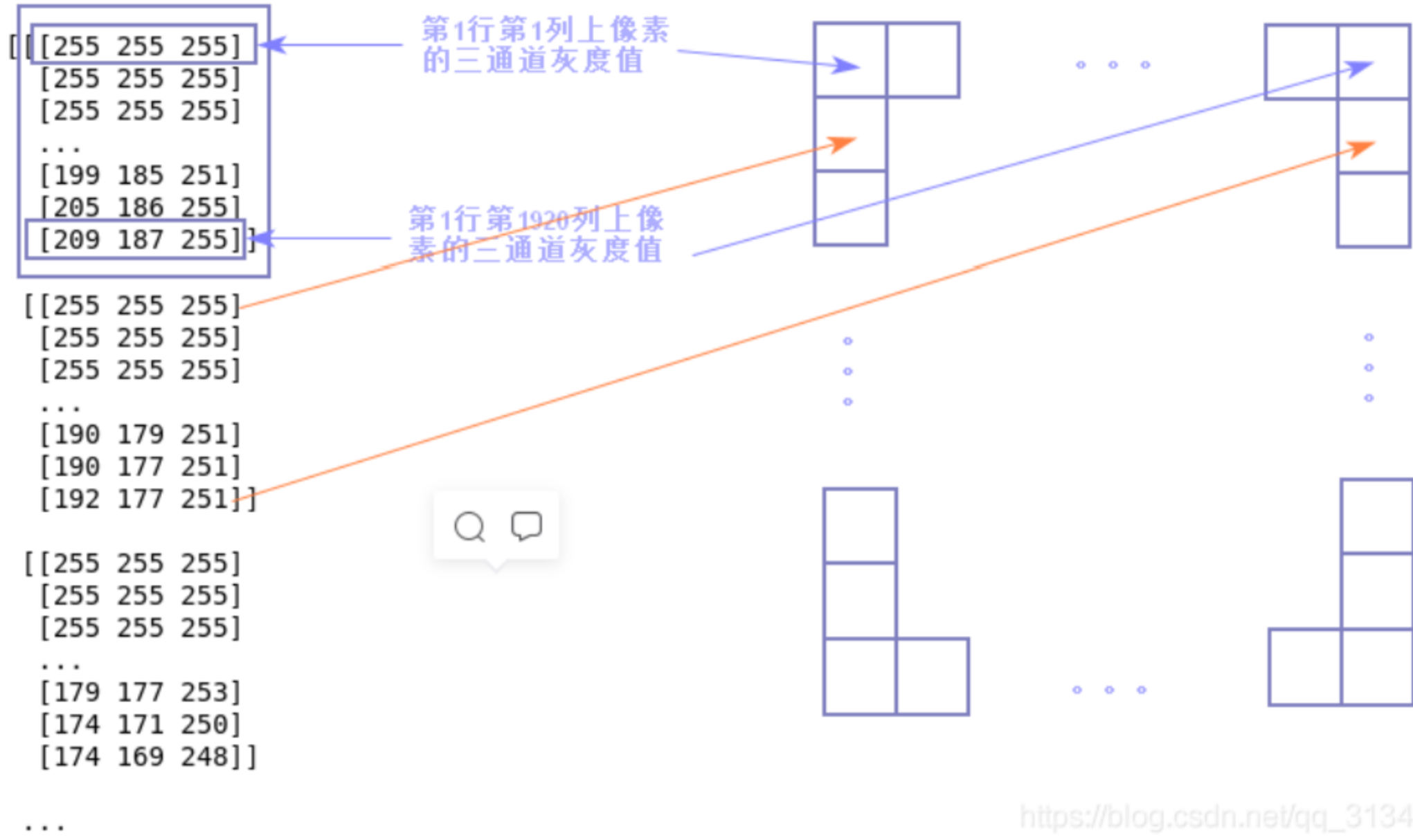
[[	[255	255	255]
	[255	255	255]
	[255	255	255]
	...		
	[199	185	251]
	[205	186	255]
	[209	187	255]]

```
[ [255 255 255]
  [255 255 255]
  [255 255 255]
  ...
  [190 179 251]
  [190 177 251]
  [192 177 251]]

[ [255 255 255]
  [255 255 255]
  [255 255 255]
  ...
  [179 177 253]
  [174 171 250]
  [174 169 248]]

...
```

在的每一个小矩阵中，有 1920 行，每一行有 3 列，每一行的三个像素值就对应当前位置的 B G R 三个通道.



如何读取彩色图像的每个通道的值呢？  
可以使用表达式访问数组内的值。例如，可以使用 `image[0,0,0]` 访问图像 `image` 的 B 通道内的第 0 行第 0 列上的像素点，式中：  
第 1 个索引表示第 0 行  
第 2 个索引表示第 0 列  
第 3 个索引表示第 0 个颜色通道

根据上述分析可知，假设有一个红色（其 R 通道值为 255，G 通道值为 0，B 通道值为 0）图像，不同的访问方式得到的值如下。  
`img[0,0]`: 访问图像 `img` 第 0 行第 0 列像素点的 BGR 值。图像是 BGR 格式的，得到的数值为 `[0,0,255]`。

img[0,0,0]: 访问图像 img 第 0 行第 0 列第 0 个通道的像素值。图像是 BGR 格式的，所以第 0 个通道是 B 通道，会得到 B 通道内第 0 行第 0 列的位置所对应的值 0。

img[0,0,1]: 访问图像 img 第 0 行第 0 列第 1 个通道的像素值。图像是 BGR 格式的，所以第 1 个通道是 G 通道，会得到 G 通道内第 0 行第 0 列的位置所对应的值 0。

img[0,0,2]: 访问图像 img 第 0 行第 0 列第 2 个通道的像素值。图像是 BGR 格式的，所以第 2 个通道是 R 通道，会得到 R 通道内第 0 行第 0 列的位置所对应的值 255。



显示推荐内容