安装： pip install scikit-image

官网： http://scikit-image.org/

skimage提供了io模块，顾名思义，这个模块是用来图片输入输出操作的。为了方便练习，也提供一个data模块，里面嵌套了一些示例图片，我们可以直接使用。

引入skimage模块可用：

|  |  |
| --- | --- |
| 1 | from skimage import io |

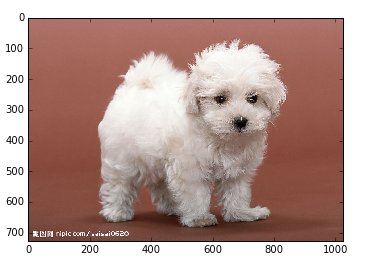
****一、从外部读取图片并显示****

读取单张彩色rgb图片，使用skimage.io.imread（fname）函数,带一个参数，表示需要读取的文件路径。显示图片使用skimage.io.imshow（arr）函数，带一个参数，表示需要显示的arr数组（读取的图片以numpy数组形式计算）。

from skimage import io

img=io.imread('d:/dog.jpg')

io.imshow(img)

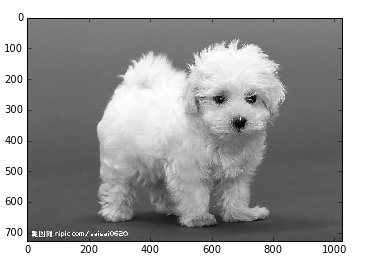


读取单张灰度图片，使用skimage.io.imread（fname，as\_grey=True）函数，第一个参数为图片路径，第二个参数为as\_grey, bool型值，默认为False

from skimage import io

img=io.imread('d:/dog.jpg',as\_grey=True)

io.imshow(img)



****二、程序自带图片****

skimage程序自带了一些示例图片，如果我们不想从外部读取图片，就可以直接使用这些示例图片：

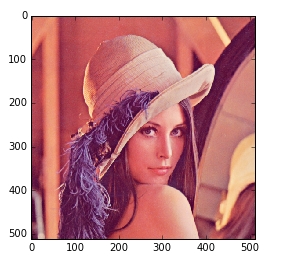
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ****astronaut**** | 宇航员图片 | ****coffee**** | 一杯咖啡图片 | ****lena**** | lena美女图片 |
| ****camera**** | 拿相机的人图片 | ****coins**** | 硬币图片 | ****moon**** | 月亮图片 |
| ****checkerboard**** | 棋盘图片 | ****horse**** | 马图片 | ****page**** | 书页图片 |
| ****chelsea**** | 小猫图片 | ****hubble\_deep\_field**** | 星空图片 | ****text**** | 文字图片 |
| ****clock**** | 时钟图片 | ****immunohistochemistry**** | 结肠图片 |  |  |

显示这些图片可用如下代码，不带任何参数

from skimage import io,data

img=data.lena()

io.imshow(img)



图片名对应的就是函数名，如camera图片对应的函数名为camera(). 这些示例图片存放在skimage的安装目录下面，路径名称为data\_dir,我们可以将这个路径打印出来看看：

from skimage import data\_dirprint(data\_dir)

显示为： D:\Anaconda3\lib\site-packages\skimage\data

也就是说，下面两行读取图片的代码效果是一样的：

from skimage import data\_dir,data,io

img1=data.lena() #读取lean图片

img2=io.imread(data\_dir+'/lena.png') #读取lena图片

****三、保存图片****

使用io模块的imsave（fname,arr）函数来实现。第一个参数表示保存的路径和名称，第二个参数表示需要保存的数组变量。

from skimage import io,data

img=data.chelsea()

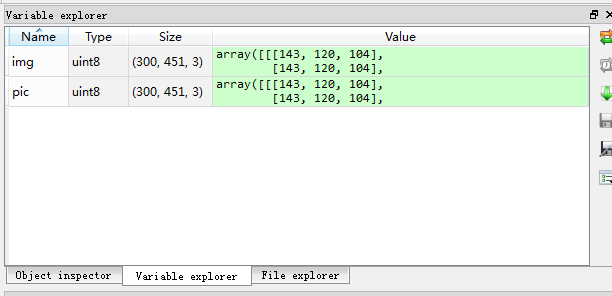
io.imshow(img)

io.imsave('d:/cat.jpg',img)

保存图片的同时也起到了转换格式的作用。如果读取时图片格式为jpg图片，保存为png格式，则将图片从jpg图片转换为png图片并保存。

****四、图片信息****

如果我们想知道一些图片信息，可以在spyder编辑器的右上角显示：



也可以直接以程序方式打印输出



from skimage import io,data

img=data.chelsea()

io.imshow(img)print(type(img)) #显示类型print(img.shape) #显示尺寸print(img.shape[0]) #图片宽度print(img.shape[1]) #图片高度print(img.shape[2]) #图片通道数print(img.size) #显示总像素个数print(img.max()) #最大像素值print(img.min()) #最小像素值print(img.mean()) #像素平均值



结果输出：

<class 'numpy.ndarray'>  
(300, 451, 3)  
300  
451  
3  
405900  
231  
0  
115.305141661