

图像处理库Scikit-image

实验目标

通过本案例的学习和课后作业的练习：

1. 了解 skimage 对图像的简单操作；
2. 了解 skimage 颜色空间的转换、特征提取和增加滤波等操作。

你也可以将本案例相关的 ipynb 学习笔记分享到 [AI Gallery Notebook \(https://marketplace.huaweicloud.com/markets/aihub/notebook/list/\)](https://marketplace.huaweicloud.com/markets/aihub/notebook/list/) 版块获得成长值 (https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=9b8d7e7a-a150-449e-ac17-2dcf76d8b492)，分享方法请查看[此文档 \(https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=8afec58a-b797-4bf9-acca-76ed512a3acb\)](https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=8afec58a-b797-4bf9-acca-76ed512a3acb)。

Scikit-image介绍

基于python脚本语言开发的数字图片处理包有 PIL，Pillow，opencv，scikit-image等，其中：

- PIL和Pillow只提供最基础的数字图像处理，功能有限。
- PIL只支持 Python2, Pillow是PIL的一个派生分支，支持 Python3.opencv 实际上是一个c++库，只是提供了python接口，更新速度非常慢。
- scikit-image 是基于scipy的一款图像处理包，它将图片作为 numpy 数组进行处理，与matlab一样。

skimage包的全称是scikit-image SciKit (toolkit for SciPy)，它对scipy.ndimage进行了扩展，提供了更多的图片处理功能。它是由python语言编写的，由scipy 社区开发和维护。skimage包由许多的子模块组成，各个子模块提供不同的功能。

本案例推荐的理论学习视频：

- 《AI基础课程--常用框架工具》图像处理库Scikit-image (<https://education.huaweicloud.com/courses/course-v1:HuaweiX+CBUCNxE081+Self-paced/courseware/260c88440da34f13879977b6db7bdd8a/1b7a0dedb2224369aeafd18953a4a728/>)

注意事项

1. 本案例推荐使用AI引擎：**XGBoost-Sklearn**。
2. 如果你是第一次使用 JupyterLab，请查看 [《ModelArts JupyterLab使用指导》](https://bbs.huaweicloud.com/forum/thread-97603-1-1.html) (<https://bbs.huaweicloud.com/forum/thread-97603-1-1.html>) 了解使用方法；
3. 如果你在使用 JupyterLab 过程中碰到报错，请参考 [《ModelArts JupyterLab常见问题解决办法》](https://bbs.huaweicloud.com/forum/thread-98681-1-1.html) (<https://bbs.huaweicloud.com/forum/thread-98681-1-1.html>) 尝试解决问题。

实验步骤

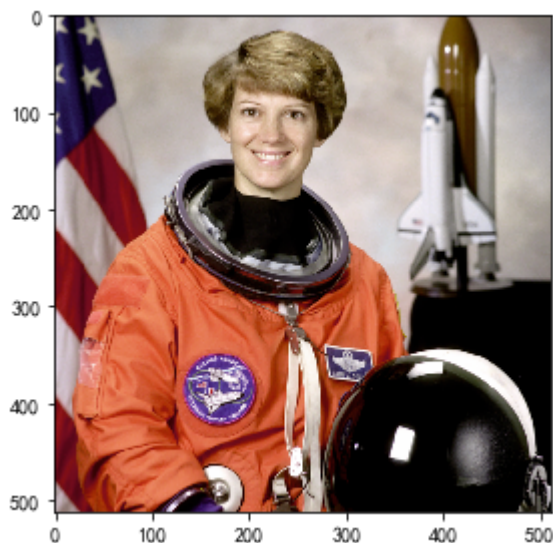
1. 图像的简单操作

```
In [1]: from skimage import novice
        from skimage import data # skimage中自带的图片
        %matplotlib inline
```

```
/home/ma-user/anaconda3/envs/XGBoost-Sklearn/lib/python3.6/site-packa
ges/skimage/novice/__init__.py:103: UserWarning: The `skimage.novice`
module was deprecated in version 0.14. It will be removed in 0.16.
```

```
warnings.warn("The `skimage.novice` module was deprecated in versio
n 0.14. "
```

```
In [2]: # 打开图片并展示
        picture = novice.open(data.data_dir + '/astronaut.png')
        picture.show()
```



```
In [3]: #查看图片格式
        picture.format
```

```
Out[3]: 'png'
```

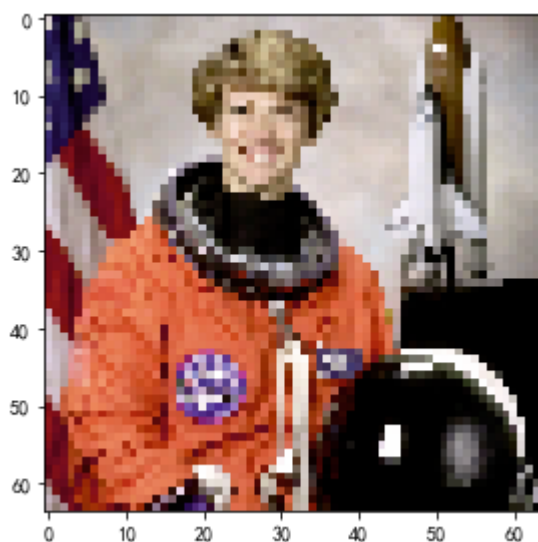
```
In [4]: #查看图片名称字符。  
picture.path.endswith('astronaut.png')
```

```
Out[4]: True
```

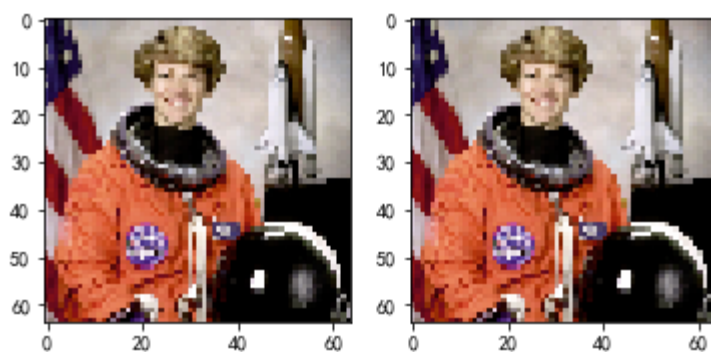
```
In [5]: #查看图片尺寸。  
picture.size
```

```
Out[5]: (512, 512)
```

```
In [6]: #定义图片大小。  
picture.size = (64, 64)  
picture.show()
```



```
In [7]: #预览图片。  
picture.compare() #compare命令预览对图片所做的更改:
```



```
In [8]: #查看原始文件是否被修改。  
picture.modified
```

```
Out[8]: True
```

```
In [9]: #保存图片。  
picture.save('save-demo.png')
```

2. io模块读取图片

```
In [10]: from skimage.io import *
```

```
In [11]: #拷贝测试图片
import os
import moxing as mox

if not os.path.isdir("./sk_image_data"):
    mox.file.copy_parallel("obs://modelarts-labs-bj4/course/hwc_edu/pyt
hon_module_framework/datasets/sk_image_data/", "sk_image_data")
```

```
INFO:root:Using MoXing-v1.17.3-
```

```
INFO:root:Using OBS-Python-SDK-3.20.7
```

```
In [12]: #查看图片数据。
img = imread("sk_image_data/lena.jpeg")
img
```

```
Out[12]: array([[232, 136, 120],
                [245, 149, 133],
                [225, 129, 113],
                ...,
                [184, 106, 122],
                [156, 93, 124],
                [180, 133, 177]],

               [[229, 134, 116],
                [240, 145, 127],
                [208, 113, 95],
                ...,
                [206, 129, 147],
                [195, 134, 165],
                [247, 202, 241]],

               [[234, 139, 121],
                [204, 109, 91],
                [244, 149, 131],
                ...,
                [132, 57, 77],
                [131, 72, 100],
                [232, 192, 226]],

               ...,

               [[206, 194, 194],
                [215, 203, 205],
                [207, 195, 199],
                ...,
                [109, 37, 38],
                [103, 35, 34],
                [88, 25, 20]],

               [[196, 201, 195],
                [199, 204, 200],
                [206, 210, 209],
                ...,
                [126, 54, 55],
                [135, 67, 64],
                [143, 80, 75]],

               [[190, 206, 196],
                [189, 202, 193],
                [207, 220, 213],
                ...,
                [136, 65, 63],
                [150, 82, 79],
                [162, 99, 94]]], dtype=uint8)
```

```
In [13]: #使用imshow方法展示图片  
imshow(img)
```

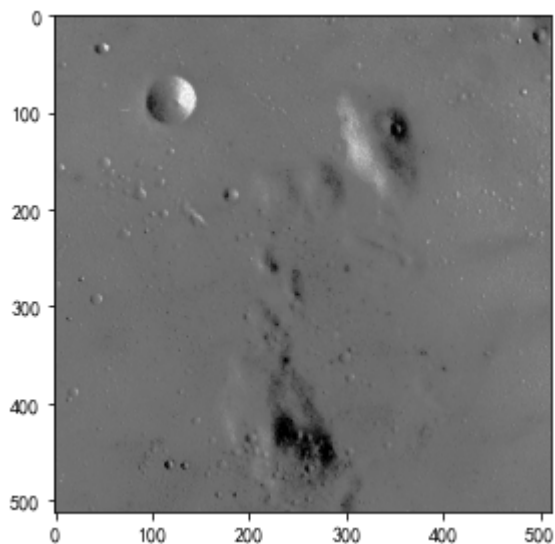
```
Out[13]: <matplotlib.image.AxesImage at 0x7f2e67cf5710>
```



3. Data模块中内置的图片

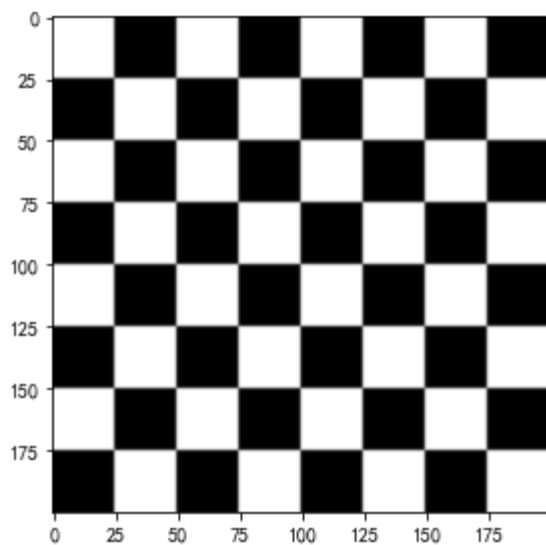
```
In [14]: import skimage  
  
moon = skimage.data.moon()  
imshow(moon)
```

```
Out[14]: <matplotlib.image.AxesImage at 0x7f2e66455160>
```



```
In [15]: che = skimage.data.checkerboard()  
imshow(che)
```

```
Out[15]: <matplotlib.image.AxesImage at 0x7f2e6642fb70>
```



```
In [16]: image = skimage.data.coffee()  
imshow(image)
```

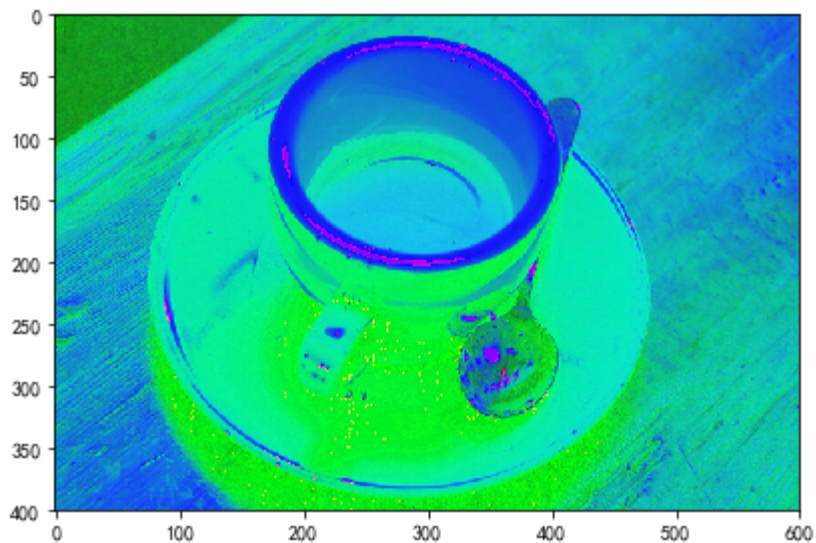
```
Out[16]: <matplotlib.image.AxesImage at 0x7f2e663969e8>
```



4. 色彩空间转换

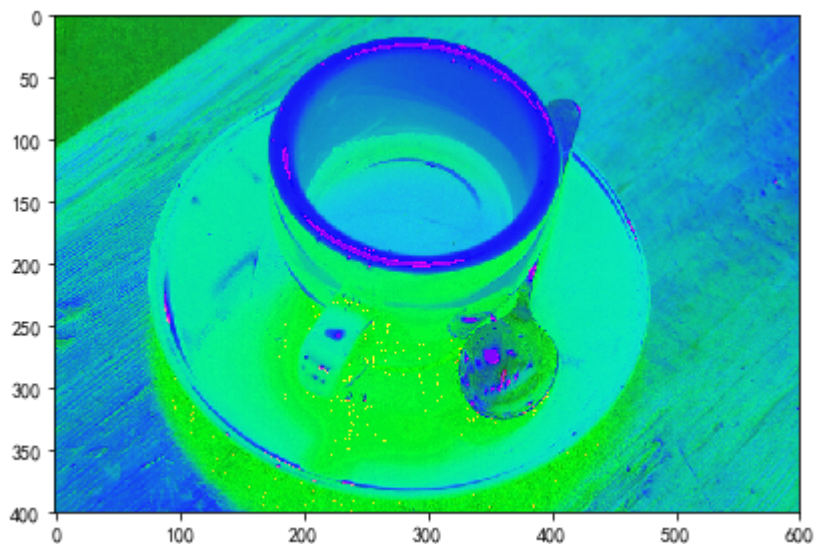

```
In [17]: from skimage.color import *  
  
img = data.coffee()  
img_hsv = convert_colorspace(img, 'RGB', 'HSV') # RGB-> HSV  
imshow(img_hsv)
```

Out[17]: <matplotlib.image.AxesImage at 0x7f2e663812e8>



```
In [18]: # convert_colorspace(img, 'RGB', 'HSV') 等价于方法 rgb2hsv。  
img_hsv = rgb2hsv(img)  
imshow(img_hsv)
```

Out[18]: <matplotlib.image.AxesImage at 0x7f2e662e2780>



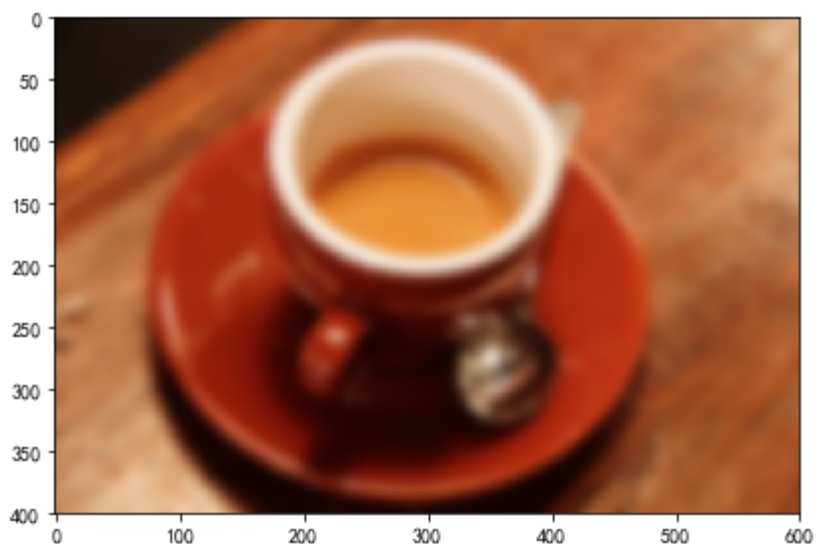
5. 滤镜

为图片增加高斯滤波。


```
In [19]: from skimage.filters import gaussian

image = data.coffee()
filtered_img = gaussian(image, sigma=5, multichannel=True)
imshow(filtered_img)
```

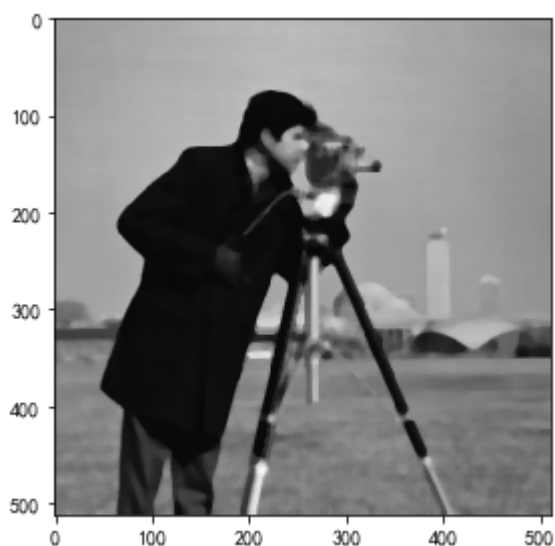
Out[19]: <matplotlib.image.AxesImage at 0x7f2e02b4e6d8>



```
In [20]: # 局部中值
from skimage.filters import median
from skimage.morphology import disk

img = data.camera()
med = median(img, disk(5))
imshow(med)
```

Out[20]: <matplotlib.image.AxesImage at 0x7f2e662b8a58>

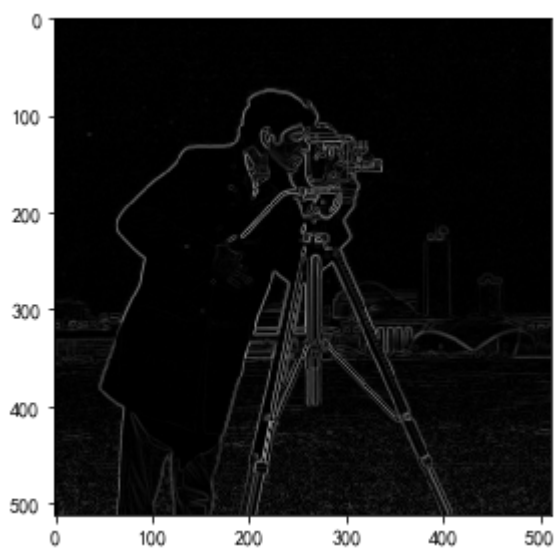


6. 边缘检测

```
In [21]: # 使用Sobel变换查找边缘幅度
from skimage.filters import sobel
from skimage import data

camera = data.camera()
edges = sobel(camera)
imshow(edges)
```

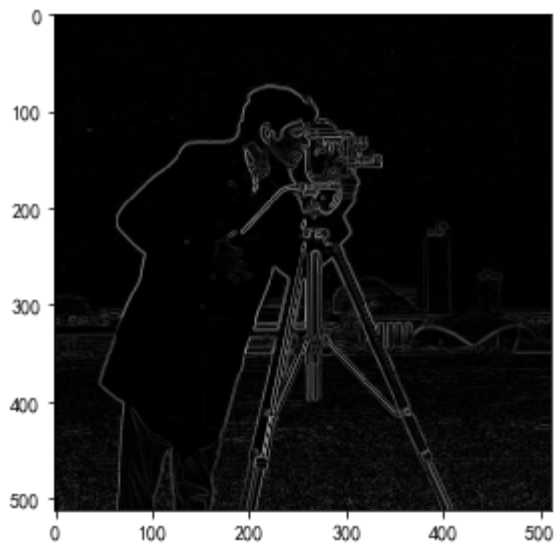
Out[21]: <matplotlib.image.AxesImage at 0x7f2e65db3128>



```
In [22]: # 使用Scharr变换查找边缘幅度
from skimage.filters import scharr

edges = scharr(camera)
imshow(edges)
```

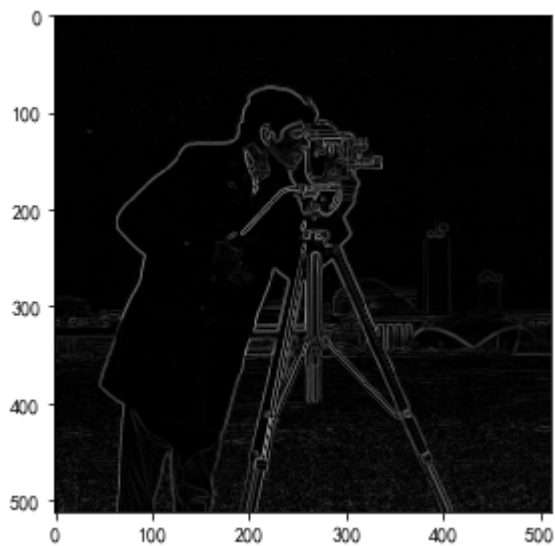
Out[22]: <matplotlib.image.AxesImage at 0x7f2e65d89be0>



```
In [23]: # 使用Prewitt变换查找边缘幅度
from skimage.filters import prewitt

edges = prewitt(camera)
imshow(edges)
```

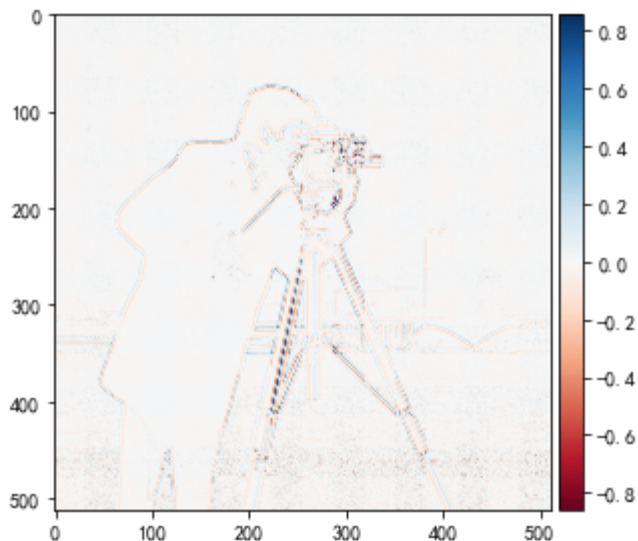
Out[23]: <matplotlib.image.AxesImage at 0x7f2e3dee76d8>



```
In [24]: # 使用拉普拉斯算子查找图像的边缘
from skimage.filters import laplace

edges = laplace(camera)
imshow(edges)
```

Out[24]: <matplotlib.image.AxesImage at 0x7f2e3dec8208>



以上是 Scikit_image 的基本使用，受限于篇幅原因，本案例未完全覆盖 Scikit_image 中的全部操作，欢迎你将更全面的 Scikit_image 学习笔记分享到 [AI Gallery Notebook \(https://marketplace.huaweicloud.com/markets/aihub/notebook/list/\)](https://marketplace.huaweicloud.com/markets/aihub/notebook/list/) 版块获得成长值 (https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=9b8d7e7a-a150-449e-ac17-2dcf76d8b492)，分享方法请查看 [此文档 \(https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=8afec58a-b797-4bf9-acca-76ed512a3acb\)](https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=8afec58a-b797-4bf9-acca-76ed512a3acb)。

作业

请你利用本课程中学到的知识和已掌握的知识，完成以下编程题：

1. [给灰度图像着色 \(https://marketplace.huaweicloud.com/markets/aihub/notebook/detail/?id=4dd04369-9a75-4995-bb37-2b3edd188b26\)](https://marketplace.huaweicloud.com/markets/aihub/notebook/detail/?id=4dd04369-9a75-4995-bb37-2b3edd188b26)

使用scikit-image载入一张灰度图像，如案例中的 data.camera()，将其着色为红色和青色（绿色和蓝色的结合），并显示。