



## 7.2 Pillow图像处理库

### ■ Python图像处理库

□ **matplotlib.image** : 仅支持导入**PNG格式**的图像, 且功能有限

□ **PIL**(Python Imaging Library)

- 功能丰富, 简单易用
- 仅支持Python2.x版本, 且已经停止更新

□ **Pillow**

- 在PIL的基础上发展而成
- 支持Python 3



### □ 安装和导入包/模块

#### ■ Pillow的**安装**

- 使用Anaconda (无需单独安装Pillow)
- 使用pip命令安装

```
pip install pillow
```

#### ■ 导入**PIL.image**模块

```
from PIL import Image
```

#### ■ 导入 **matplotlib.pyplot** 模块

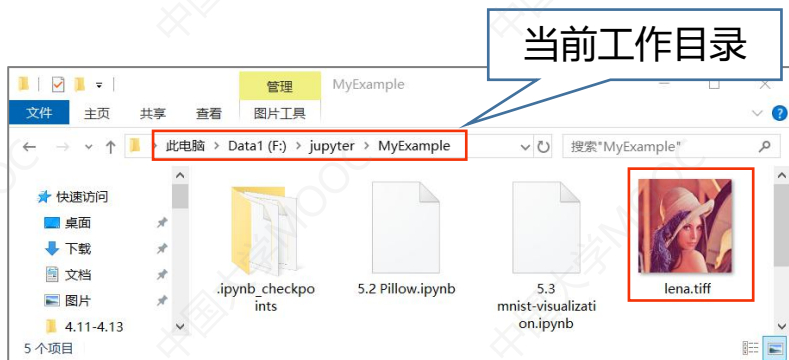
```
import matplotlib.pyplot as plt
```



### ■ 打开图像——`Image.open()`函数

`Image.open(路径)`

返回image 对象



lena.tiff



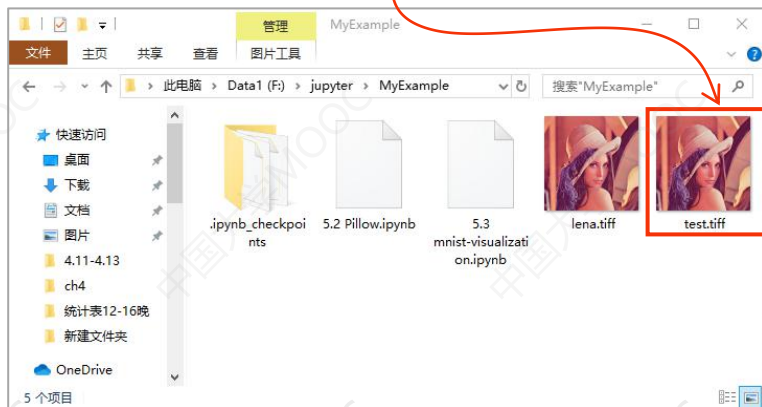
```
img=Image.open("lena.tiff")
```



### 保存图像——`save()`方法

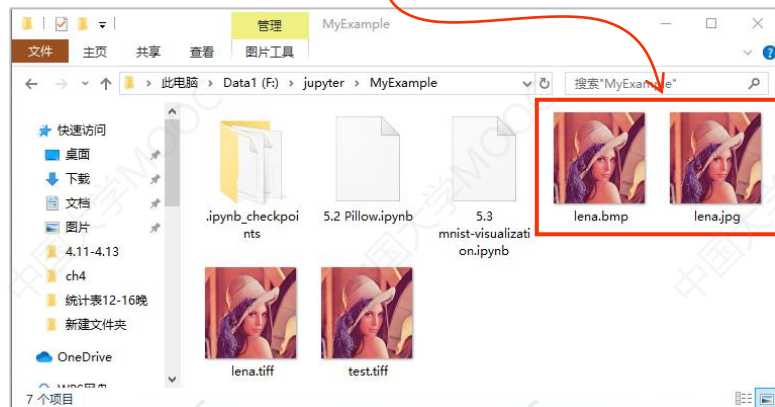
图像对象.`save()`

```
img.save("test.tiff")
```



改变文件名的**后缀**，就可以**转换图像格式**

```
img.save("lena.jpg")  
img.save("lena.bmp")
```



### ■ 图像对象的主要属性

属 性	说 明
图像对象.format	图像格式
图像对象.size	图像尺寸
图像对象.mode	色彩模式

### □ 尺寸和色彩模式

```
print ("size:",img.size)
print ("mode:",img.mode)
```

运行结果: `size:(512, 512)`  
`mode:RGB`

### □ 图像格式

```
img1=Image.open("lena.jpg")
img2=Image.open("lena.bmp")

print ("image:",img.format)
print ("image1:",img1.format)
print ("image2:",img2.format)
```

运行结果: `image: TIFF`  
`image1: JPEG`  
`image2: BMP`

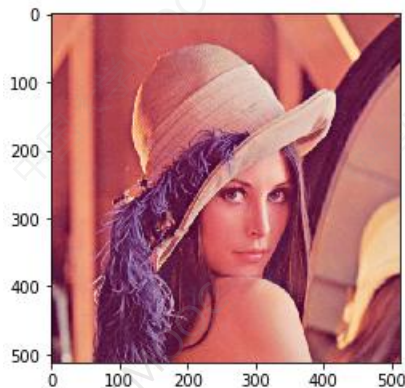


### ■ 显示图像——`imshow()`方法

`plt.imshow( image对象/Numpy数组)`

```
plt.figure(figsize=(5,5))  
plt.imshow(image)  
plt.show()
```

运行结果:





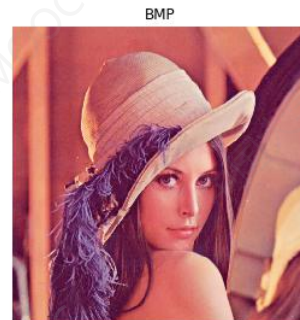
## 7.2 Pillow 图像处理库

```
1 plt.figure(figsize=(15,5))
2
3 plt.subplot(131)
4 plt.axis("off")
5 plt.imshow(img)
6 plt.title(img.format)
7
8 plt.subplot(132)
9 plt.axis("off")
10 plt.imshow(img1)
11 plt.title(img1.format)
12
13 plt.subplot(133)
14 plt.axis("off")
15 plt.imshow(img2)
16 plt.title(img2.format)
17
18 plt.show()
```

不显示坐标轴

在“子图标题”的位置，  
显示“图片格式”

运行结果：



西安科技大学

计算机科学与技术学院

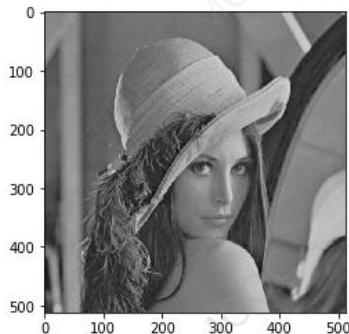


### 转换图像的色彩模式

取值	色彩模式
1	二值图像 灰度图像
P	8位彩色图像 24位彩色图像
RGBA	32位彩色图像 CMYK彩色图像
YCbCr	YCbCr彩色图像 32位整型灰度图像
F	32位浮点灰度图像

图像对象.convert(色彩模式)

```
1 img_gray=img.convert("L")
2 print("mode=",img_gray.mode)
3
4 plt.figure(figsize=(5,5))
5 plt.imshow(img_gray,cmap="gray")
6 plt.show()
7
8 img_gray.save("lena_gray.bmp")
```



出现在当前工作目录中



### 颜色通道的分离与合并

图像对象.split()

Image.merge(色彩模式, 图像列表)



彩色图像

```
img = Image.open("Lena.tiff")  
img_r, img_g, img_b = img.split()  
plt.figure(figsize=(10, 10))
```

将彩色图像分离为  
R, G, B三个颜色通道

```
plt.subplot(221)  
plt.axis("off")  
plt.imshow(img_r, cmap="gray")  
plt.title("R", fontsize=20)
```

采用灰度图表示颜色的亮度

```
plt.subplot(222)  
plt.axis("off")  
plt.imshow(img_g, cmap="gray")  
plt.title("G", fontsize=20)
```

```
plt.subplot(223)  
plt.axis("off")  
plt.imshow(img_b, cmap="gray")  
plt.title("B", fontsize=20)
```

```
img_rgb = Image.merge("RGB", [img_r, img_g, img_b])  
plt.subplot(224)  
plt.axis("off")  
plt.imshow(img_rgb)  
plt.title("RGB", fontsize=20)
```

将所有通道合并, 得到  
RGB彩色图像

```
plt.show()
```

### ■ 转化为数组

`np.array(图像对象)`

**例：**将灰度图像转化为数组

```
img_gray=Image.open("lena_gray.bmp")  
arr_img_gray= np.array(img_gray)  
print("\nshape:",arr_img_gray.shape)  
print(arr_img_gray)
```

灰度图像

运行结果：

灰度图像对应一个二维数组。  
其中每个元素对应一个像素

```
shape: (512, 512)  
[[ 162 162 162 ... 169 154 128]  
 [ 162 162 162 ... 169 154 128]  
 [ 162 162 162 ... 169 154 128]  
 ...  
 [  42  42  49 ... 104 100  98]  
 [  43  43  54 ... 103 105 107]  
 [  43  43  54 ... 103 105 107]]
```



## 7.2 Pillow图像处理库

例：将彩色图像转化为数组

```
import numpy as np
arr_img = np.array(img)
print("shape:", arr_img.shape, "\n")
print(arr_img)
```

彩色图像

shape: (512, 512, 3)

```
[[[226 137 125]
 [226 137 125]
 [223 137 133]
 ...
 [230 148 122]
 [221 130 110]
 [200 99 90]]]
```

彩色图像是一个三维数组，  
前两维对应图像的尺寸，  
第三维分别对应图像的3个通道

```
[[[ 82 22 57]
 [ 82 22 57]
 [ 96 32 62]
 ...
 [179 70 79]
 [181 71 81]
 [185 74 81]]]
```

R G B

512×512

[226 137 125]	[226 137 125]	[223 137 133]	...	[230 148 122]	[221 130 110]	[200 99 90]
[226 137 125]	[226 137 125]	[223 137 133]	...	[230 148 122]	[221 130 110]	[200 99 90]
[226 137 125]	[226 137 125]	[223 137 133]	...	[230 148 122]	[221 130 110]	[200 99 90]
...	...	...	...	...	...	...
[84 18 60]	[84 18 60]	[92 27 58]	...	[173 73 84]	[172 68 76]	[177 62 79]
[82 22 57]	[82 22 57]	[96 32 62]	...	[179 70 79]	[181 71 81]	[185 74 81]
[82 22 57]	[82 22 57]	[96 32 62]	...	[179 70 79]	[181 71 81]	[185 74 81]

### 例：图像颜色反向

对图像中**每一个像素**做**颜色反向**运算

```
1 arr_img_new=255-arr_img_gray
2
3 plt.figure(figsize=(10,5))
4
5 plt.subplot(121)
6 plt.axis("off")
7 plt.imshow(arr_img_gray,cmap="gray")
8
9 plt.subplot(122)
10 plt.axis("off")
11 plt.imshow(arr_img_new,cmap="gray")
12
13 plt.show()
```





### 对图像的缩放、旋转和镜像

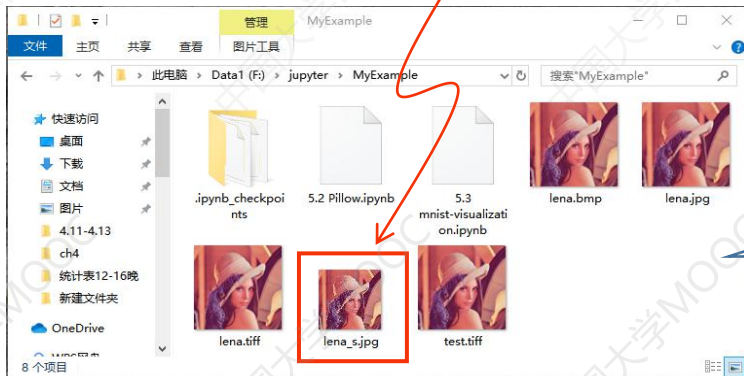
#### 缩放图像

图像对象.resize((width, height))

```
plt.figure(figsize=(5,5))  
img_small = img.resize((64, 64))  
  
plt.imshow(img_small)  
plt.show()
```

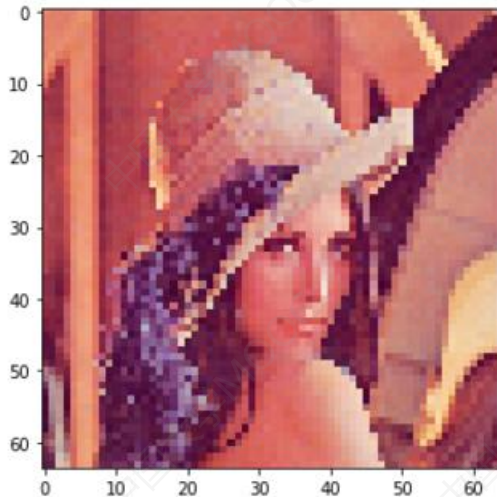
将这个缩小的图片  
保存为文件lena\_s.jpg

```
img_small.save("lena_s.jpg")
```



当前工作目录

64×64



### ■ 对图像的缩放、旋转和镜像

#### □ 缩放图像

图像对象.resize((width, height))

图像对象.thumbnail((width, height))

原地操作，返回值是None

PS:

resize()方法不对原图进行修改，

thumbnail()方法是原地操作，直接对image对象本身进行缩放。





### □ 旋转、镜像

图像对象.transpose(旋转方式)

Image.**FLIP\_LEFT\_RIGHT**: 水平翻转

Image.**FLIP\_TOP\_BOTTOM**: 上下翻转

Image.**ROTATE\_90**: 逆时针旋转90°

Image.**ROTATE\_180**: 逆时针旋转180°

Image.**ROTATE\_270**: 逆时针旋转270°

Image.**TRANSPOSE**: 将图像进行转置

Image.**TRANSVERSE**: 将图像进行转置, 再水平翻转

```
img_flr = img.transpose(Image.FLIP_LEFT_RIGHT)
img_r90 = img.transpose(Image.ROTATE_90)
img_tp = img.transpose(Image.TRANSPOSE)
```



```

1 import matplotlib.pyplot as plt
2 from PIL import Image
3
4 plt.rcParams['font.sans-serif']="SimHei"
5 img=Image.open("lena.tiff")
6
7 plt.figure(figsize=(10,10))
8
9 plt.subplot(221)
10 plt.axis("off")
11 plt.imshow(img)
12 plt.title("原图",fontsize=20)
13
14 plt.subplot(222)
15 plt.axis("off")
16 img_flr = img.transpose(Image.FLIP_LEFT_RIGHT)
17 plt.imshow(img_flr)
18 plt.title("左右翻转",fontsize=20)
19
20 plt.subplot(223)
21 plt.axis("off")
22 img_r90 = img.transpose(Image.ROTATE_90)
23 plt.imshow(img_r90)
24 plt.title("逆时针旋转90度",fontsize=20)
25
26 plt.subplot(224)
27 plt.axis("off")
28 img_tp = img.transpose(Image.TRANSPOSE)
29 plt.imshow(img_tp)
30 plt.title("转置",fontsize=20)
31
32 plt.show()

```

运行结果:



- 裁剪图像：在图像上指定的位置裁剪出一个矩形区域

图像对象.crop((x0, y0, x1, y1))

左上角

右下角

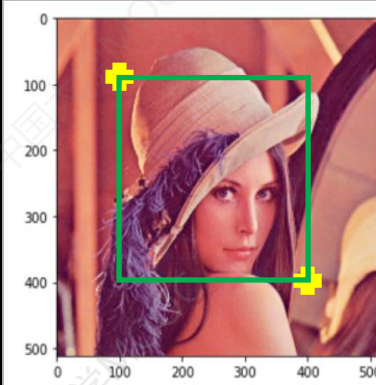
返回值：图像对象

```
import matplotlib.pyplot as plt
from PIL import Image

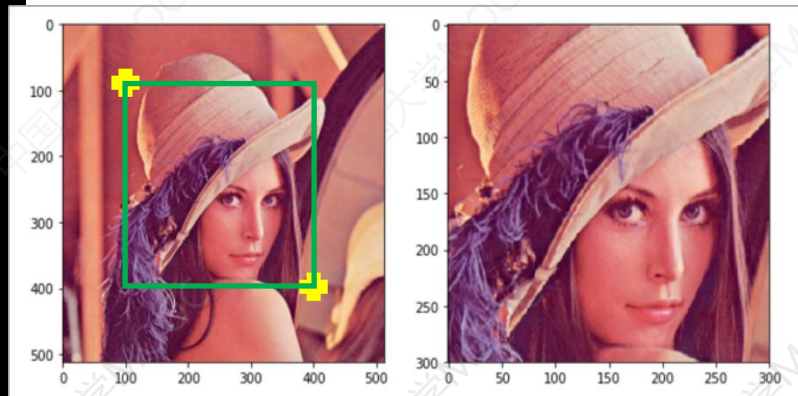
img = Image.open("Lena.tiff")

plt.figure(figsize=(10,5))

plt.subplot(121)
plt.imshow(img)

plt.subplot(122)

img_region = img.crop((100,100,400,400))
plt.imshow(img_region)

plt.show()
```



图像操作	方法/函数/属性
打开图像	Image.open(路径)
保存图像	图像对象.save()
查看图像属性	图像对象.format 图像对象.size 图像对象.mode
显示图像	plt.imshow( image对象/Numpy数组)
转换色彩模式	图像对象.convert(色彩模式)
颜色通道的分离与合并	图像对象.split() Image.merge(色彩模式, 图像列表)
将图像转换为数组	np.array(图像对象)
缩放图像	图像对象.resize((width, height))
旋转和镜像	图像对象.transpose(旋转方式)
裁剪图像	图像对象.crop((x0,y0,x1,y1))

