这里简单介绍python图像处理的类库，PIL其实只是一般的效果处理库，对像素级的支持并不好，真正科学计算，当然要熟悉numpy，基本数组操作，然后是scipy，里面封装了大量常用数学函数，也包括ndimage一个常规图像处理库，之后是scikit-image，opencv等大而全的图像处理库，以及ITK，VTK等行业内流行的类库，起丰富程度与matlab不相上下。并且这些类库大多底层由C实现，效率也不低。

**1. scikit Image**

scikit-image是Python的一个开源包，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，正好与matlab一样。它使算法和实用程序得以用于研究、教育和行业应用领域。即便是对不熟悉Python生态环境的人来说，这个库也十分简单明了。此代码由一个活跃的志愿者团队编写，质量很高且已经审阅。

**2. Numpy**

Numpy是Python编程的一个核心库，为数组提供支持。图像本质上是包含数据像素点的标准Numpy数组。因此，通过使用基本的NumPy操作，例如切片、掩膜和匹配索引，可以修改图像的像素值。也可以用skimage加载图像，用matplotlib显示图像。

3**. Scipy**

和Numpy一样，scipy是Python的另一个核心模块，可用于图像的基本处理和加工。 特别是子模块scipy.ndimage提供了可在n维NumPy数组上运行的函数。该包目前包括线性和非线性滤波法、二值图像形态学、B样条插值和对象测量等功能。

**4. PIL/ Pillow**

PIL（Python图像处理库）是一个免费的Python编程语言库，支持打开、处理和保存多种格式的图像文件。但它的发展早已停滞不前，最新一次发布还是在2009年。所幸还有Pillow这个积极开发的PIL分支，而且它安装起来更容易，还可以在所有主操作系统上运行，并支持Python 3。该库包含点操作、使用一组内置卷积内核进行图像过滤以及颜色空间转换这些基本的图像处理功能。

**5. OpenCV-Python**

OpenCV（开源计算机视觉库）是计算机视觉应用中使用最为广泛的库之一。OpenCV-Python是OpenCV的python接口。OpenCV-Python不只是因为后台由用C / C ++编写的代码组成而速度快，还因为前端的Python包装器使得编码和部署容易。因此，它成为了计算密集型计算机视觉程序的绝佳选择。

**6. SimpleCV**

SimpleCV也是用于构建计算机视觉应用程序的一个开源框架。使用SimpleCV，可以不必事先了解位深、文件格式、色彩空间就能对几个高性能的计算机视觉库如OpenCV 进行访问。它的学习曲线比OpenCV小得多，正如其宣传语所说，“SimpleCV让计算机视觉变得容易”。SimpleCV的优势在于：

**7. Mahotas**

Mahotas是Python中另一个计算机视觉和图像处理库，含有过滤和形态学操作这类的传统图像处理功能和兴趣点检测、局部描述符等用于特征计算的现代化计算机视觉功能。接口使用了Python，适合快速开发，但算法是由C ++实现的，并对速度进行了调整。Mahotas库因代码简单且依赖性最小而快速。

**8. SimpleITK**

ITK或Insight Segmentation and Registration Toolkit是一个开源的跨平台系统，为开发人员提供了一整套用于图像分析的软件工具。其中，SimpleITK是一个建立在ITK之上的简化层，旨在方便它在快速成型、教育、解释语言中的应用。SimpleITK是一个具有大量组件的图像分析工具包，支持一般过滤操作、图像分割和图像配准。SimpleITK本身是用C ++编写的，但可用于包括Python在内的多种编程语言。

**9. pgmagick**

Pgmagick是GraphicsMagick库基于Python的包装器。GraphicsMagick图像处理系统有时也被称为图像处理的瑞士军刀。它提供了强大高效的工具集和库集，这些集合支持读取、写入和操作的图像格式超过88种，其中包括DPX，GIF，JPEG，JPEG-2000，PNG，PDF，PNM和TIFF等重要格式。

**10. Pycairo**

Pycairo是一组用于图形库cairo的python包。Cairo是一个用于绘制矢量图形的2D图形库。矢量图形很有趣，在调整大小或变换时它们的清晰度不会受到影响。Pycairo可从Python中调用cairo命令用于cairo。

python库安装

可以通过pip进行安装

sudo pip install xxxx

scikit-image是基于numpy，因此需要安装numpy和scipy，同时需要安装matplotlib进行图片的实现等。

因此，需要安装如下的包：

numpy (1.13.3)

matplotlib (2.1.0)

scikit-image (0.13.1)

scipy (1.0.0)

也可以直接下载集成开发环境Anaconda,该环境已经集成了数字图像处理相关的包，因此安装起来比较方便。

可以通过如下程序简单测试下相关库是否安装成功

import numpy as np

import scipy as sp

import matplotlib.pyplot as plt

from skimage import io

img = io.imread("./cat.png")

print(img.shape)

plt.imshow(img)

plt.show()

若显示正常，则可以认为相关的库安装成功

# **skimage库子模块介绍**

skimage库的全称scikit-image Scikit， 是对scipy.ndimage进行了扩展，提供了更多的图片处理功能。skimage包含很多的子模块，各个子模块具有不同的功能，如下

子模块名称 实现功能

io 读取,保存和显示图片和视频

color 颜色空间变换

data 提供一些测试图片和样本数据

filters 图像增强，边缘检测，排序滤波器，自动阈值等

draw 操作于numpy数组上的基本图形绘制，包括线条，矩阵，圆和文本等

transform 几何变换和其他变换，如旋转，拉伸和Radon(拉东)变换等

exposure 图像强度调整，例如，直方图均衡化等

feature 特征检测和提取， 例如，纹理分析等

graph 图论操作，例如，最短路径

measure 图像属性测量，例如，相似度和轮廓

morphology 形态学操作，如开闭运算，骨架提取等

novice 简化的用于教学目的的接口

restoration 修复算法，例如去卷积算法，去噪等

segmentation 图像分割为多个区域

util 通用工具

viewer 简单图形用户界面用于可视化结果和探索参数

当要使用对应的模块中功能函数时，需要通过import导入对应的子模块即可, 若要导入多个子模块时，子模块之间用逗号隔开，如下：

**from** skimage **import** io, data, feature

## skimage操作

图像读取, 保存与显示

skimage中io子模块提供了相关的功能，同时也提供了一些data模块，其中包含一些示例图片用于练习

导入io子模块的python语句如下:

from skimage import io

1

2

从外部读取图片并显示

使用 skimage.io.imread(fname) 读取fname指定的图片，

skimage.io.imshow(arr)， 表示显示arr数组表示的图片

from skimage import io

img = io.imread('./cat.png')

io.imshow(img)

io.show()

读取单张灰度图片时，使用 skimage.io.imread(fname, as\_grey=True) 函数，第一个参数fname表示要显示的图片路径，第二个参数as\_grey，是bool类型，默认值False。

from skimage import io

img = io.imread('./cat.png', as\_grey=True)

io.imshow(img)

io.show()



## skimage自带图片

图片名称 说明

astronaut 宇航员

binary\_blobs 二元斑点

camera 相机

checkerboard 棋盘

chelsea 猫

clock 时钟

coffee 一杯咖啡

coins 硬币

horse 马

hubble\_deep\_field 星空

immunohistochemistry 结肠图片

logo 商标

moon 月球表面

page 书页内容

rocket 火箭

text 文字图片

例如

from skimage import io, data

img = data.hubble\_deep\_field()

io.imshow(img)

io.show()

1

2

3

4

5



图片名就是对应的函数名，如camera图片对应的函数名为 data.camera()。

注：这些图片存储在skimage的安装目录下，可以通过data\_dir把路径打印出来。

from skimage import data\_dir

print (data\_dir)

1

2

3

输出为：

/usr/local/lib/python3.6/dist-packages/skimage/data

## 保存图片

使用 io.imsave(fname, arr) 函数进行保存,

参数fname: 表示保存的路径和名称

参数arr：表示需要保存的数组变量

from skimage import io, data

img = data.checkerboard()

io.imshow(img)

io.imsave('checkerboard\_copy.jpg', img)

1

2

3

4

5

6

这样，在当前的工作目录下就增加了一个checkerboard\_copy.jpg文件。

注：保存图片同时也起到了转换格式的作用，若读取的是png格式图片，当保存为jpg时，则图片从png格式转换为jpg格式图片。

## 获取图片信息

from skimage import io, data

img = data.chelsea()

io.imshow(img)

io.show()

print(type(img)) # 类型

print(img.shape) # 形状

print(img.shape[0]) # 图片宽度

print(img.shape[1]) # 图片高度

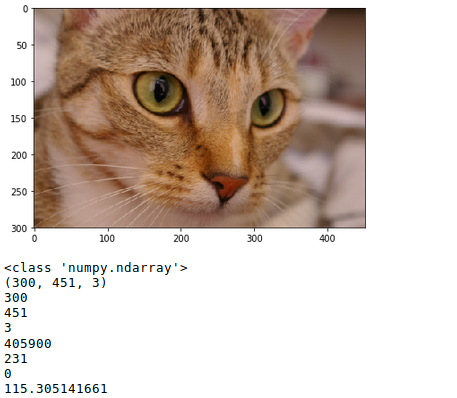
print(img.shape[2]) # 图片通道数

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



图像像素访问与裁剪

图片读入程序后，以numpy数组方式存储，因此对numpy数组的操作，都可以用于图片数组，对数组元素的访问，实际上就是对图片像素点的访问。

像素读取

对 彩色图片 的像素点访问方式如下

img[i, j, c]

其中：

i 表示图片的行数

j 表示图片的列数

c 表示图片的通道数(RGB三通道分别对应0, 1, 2)。

坐标从左上角开始

对 灰度图片 的像素点访问方式如下

gray[i, j]

例如， 对data中宇航员图片的B通道中的第20行10列的像素值

from skimage import io, data

img = data.astronaut()

pixel = img[20, 10, 2]

print(pixel)

输出

69

例如，显示红色单通道图片的程序如下

**from** skimage **import** io, data

**img** = **data**.astronaut()**R** = img[:, :, 0]**io**.imshow(**R**)**io**.show()



像素修改

例如，对宇航员图片随机添加椒盐噪声

from skimage import io, data

import numpy as np

img = data.astronaut()

# 随机生成5000个椒盐点

rows, cols, dims = img.shape

for i in range(5000):

x = np.random.randint(0, rows)

y = np.random.randint(0, cols)

img[x, y, :] = 255

io.imshow(img)

io.show()



图片裁剪

由于图片是以numpy数组进行存储，因此对于数组的裁剪，就是对图片的裁剪

例如，对宇航员图片进行裁剪

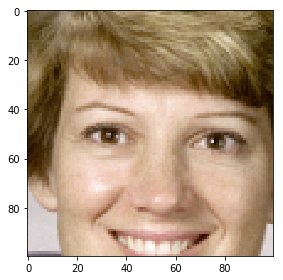
from skimage import io, data

img = data.astronaut()

partial\_img = img[50:150, 170:270, :] #50-150行，170-270列

io.imshow(partial\_img)

io.show()



对多个像素点进行操作时， 使用数组切片方式进行访问， 切片方式访问的是指定间隔内下标对应的像素点。以下是一些例子

img[i,:] = im[j,:] # 将第 j 行的数值赋值给第 i 行

img[:,i] = 100 # 将第 i 列的所有数值设为 100

img[:100,:50].sum() # 计算前 100 行、前 50 列所有数值的和

img[50:100,50:100] # 50~100 行，50~100 列（不包括第 100 行和第 100 列）

img[i].mean() # 第 i 行所有数值的平均值

img[:,-1] # 最后一列

img[-2,:] (or im[-2]) # 倒数第二行

例1： 将宇航员图片进行二值化，像素值大于128的变为1, 否在变为0

from skimage import io, data, color

img = data.astronaut()

img\_gray = color.rgb2gray(img)

rows, cols = img\_gray.shape

for i in range(rows):

for j in range(cols):

if (img\_gray[i, j] <= 0.5):

img\_gray[i, j] = 0

else:

img\_gray[i, j] = 1

io.imshow(img\_gray)

io.show()

