

lyrichu

昵称：lyrichu
园龄：2年8个月
粉丝：111
关注：6
[+加关注](#)

<2019年2月>

日	一	二	三	四	五	六
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2
3	4	5	6	7	8	9

搜索

找找看

谷歌搜索

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

我的标签

[python\(17\)](#)
[C语言\(7\)](#)
[爬虫\(4\)](#)
[生成对抗网络\(3\)](#)
[PSO\(3\)](#)
[GA\(3\)](#)
[GAN\(3\)](#)
[网易云音乐\(3\)](#)
[遗传算法\(2\)](#)
[优化算法\(2\)](#)
[更多](#)

随笔分类(48)

[Git\(1\)](#)
[haskell学习\(1\)](#)
[latex\(1\)](#)
[Linux学习\(2\)](#)
[Matlab\(1\)](#)
[python学习\(9\)](#)
[R语言学习\(1\)](#)
[scala学习\(1\)](#)
[spark\(1\)](#)
[机器学习\(2\)](#)
[计算机视觉教程\(1\)](#)
[论文解读\(1\)](#)
[爬虫\(6\)](#)
[深度学习\(5\)](#)
[数据分析\(1\)](#)
[数据结构与算法\(2\)](#)
[图像视频处理\(2\)](#)
[优化算法\(9\)](#)
[自然语言处理\(1\)](#)

随笔档案(43)

[2019年1月\(3\)](#)
[2018年6月\(5\)](#)
[2018年5月\(7\)](#)
[2018年4月\(1\)](#)
[2018年3月\(1\)](#)

[博客园](#) [首页](#) [新随笔](#) [联系](#) [订阅](#) [XML](#) [管理](#)

随笔-43 评论-90 文章-0

python PIL 图像处理库简介(一)

1. Introduction

PIL(Python Image Library)是python的第三方图像处理库，但是由于其强大的功能与众多的使用人数，几乎已经被认为是python官方图像处理库了。其官方主页为:**PIL**。 **PIL**历史悠久，原来是只支持python2.x的版本，后来出现了移植到python3的库**pillow**,pillow号称是 `friendly fork for PIL` ,其功能和**PIL**差不多，但是支持python3。本文主要介绍PIL那些最常用的特性与用法,主要参考自:<http://www.effbot.org/imagingbook>。

2. What PIL can do?

PIL可以做很多和图像处理相关的事情:

- **图像归档(Image Archives)**。PIL非常适合于图像归档以及图像的批处理任务。你可以使用PIL创建缩略图，转换图像格式，打印图像等等。
- **图像展示(Image Display)**。PIL较新的版本支持包括Tk PhotoImage，BitmapImage还有Windows DIB等接口。PIL支持众多的GUI框架接口，可以用于图像展示。
- **图像处理(Image Processing)**。PIL包括了基础的图像处理函数，包括对点的处理，使用众多的卷积核(convolution kernels)做过滤(filter),还有颜色空间的转换。PIL库同样支持图像的大小转换，图像旋转，以及任意的仿射变换。PIL还有一些直方图的方法，允许你展示图像的一些统计特性。这个可以用来实现图像的自动对比度增强，还有全局的统计分析等。

3. How to use PIL?

3.1 Image class

Image类是PIL中的核心类，你有很多方式来对它进行初始化，比如从文件中加载一张图像，处理其他形式的图像，或者是从头创造一张图像等。下面是**PIL Image**类中常用的方法：

- **open(filename,mode)**(打开一张图像)。下面的代码演示了如何从文件打开一张图像：

```
>>> from PIL import Image
>>> Image.open("dog.jpg", "r")
<PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=296x299 at 0x7F62BDB5B0F0>
>
>>> im = Image.open("dog.jpg", "r")
>>> print(im.size, im.format, im.mode)
(296, 299) JPEG RGB
```

`Image.open` 返回一个Image对象，该对象有 `size,format,mode` 等属性，其中 `size` 表示图像的宽度和高度(像素表示); `format` 表示图像的格式,常见的包括JPEG,PNG等格式; `mode` 表示图像的模式，定义了像素类型还有图像深度等，常见的有RGB,HSV等。一般来说'L'(luminance)表示灰度图像,'RGB'表示真彩图像,'CMYK'表示预先压缩的图像。一旦你得到了打开的Image对象之后，就可以使用其众多的方法对图像进行了，比如使用 `im.show()` 可以展示上面得到的图像。

- **save(filename,format)**(保存指定格式的图像)

```
>>> im.save("dog.png", 'png')
```

上面的代码将图像重新保存成png格式。

- **thumbnail(size,resample)**(创建缩略图)

```
>>> im.thumbnail((50,50),resample=Image.BICUBIC)
>>> im.show()
```

上面的代码可以创建一个指定大小(size)的缩略图,需要注意的是，thumbnail方法是原地操作，返回值是None。第一个参数是指定的缩略图的大小，第二个是采样的，有 `Image.BICUBIC` , `PIL.Image.LANCZOS` , `PIL.Image.BILINEAR` , `PIL.Image.NEAREST` 这四种采样方法。默认是 `Image.BICUBIC` 。

- **crop(box)**(裁剪矩形区域)

```
>>> im = Image.open("dog.jpg", "r")
>>> box = (100,100,200,200)
>>> region = im.crop(box)
>>> region.show()
im.crop()
```

上面的代码在im图像上裁剪了一个box矩形区域，然后显示出来。box是一个有四个数字的元组(upper_left_x,upper_left_y,lower_right_x,lower_right_y),分别表示裁剪矩形区域的左上角x,y坐标,右下角的x,y坐标,规定图

2018年2月 (1)
2017年11月 (2)
2017年9月 (1)
2017年8月 (1)
2017年7月 (3)
2017年5月 (8)
2017年4月 (3)
2017年3月 (1)
2016年12月 (6)

最新评论

1. Re:sopt:一个简单的python最优
化库
@Brues可以举一个具体的例子吗...
--lyrichu
2. Re:DCGAN 代码简单解读
博主想请教一下，不清楚 model.py
文件，discriminator函数里，为什么
要用conv_cond_concat(image, yb)
以及原理相同的代码语句，卷积层
构建了解，但不清楚为什么.....
--Mooner_lx
3. Re:Python 与 C/C++ 交互的几种
方式
如何实现 C/C++ 与 Python 的通
信？
想在 C++ 中用 Python 进行数值计
算，Python 需要访问 C++ 的变量
并计算后返回数值。有什么好办法
呢？
--chenyin~
4. Re:Python 获取 网易云音乐热门
评论
@lyrichu对于评论量大的，爬到后
面返回的都是重复内容，可能是后
台限制了。不知道老哥现在有没有
新进展。...
--菜鸟琦
5. Re:网易云音乐评论 可视化分析
楼主遇到的评论爬取不完全问题我
也遇到了，显示20w+的评论，全爬
一遍下来去重后只剩2w左右。大部
分都是重复内容，请问这个问题楼
主现在解决了吗？
--菜鸟琦

阅读排行榜

1. 怎样共享windows和linux之间的
文件(17825)
2. Python 获取 网易云音乐热门评论
(16054)
3. 介绍几个python的音频处理库
(15320)
4. python 编写简单的
setup.py(11227)
5. python PIL 图像处理库简介(一)
(10167)

评论排行榜

1. Python 获取 网易云音乐热门评论
(53)
2. 网易云音乐评论 可视化分析(12)
3. python wordcloud 对电影《我不
是潘金莲》制作词云(5)
4. EM 算法求解高斯混合模型
python实现(4)
5. sopt:一个简单的python最优化库
(3)

推荐排行榜

1. Python 获取 网易云音乐热门评论
(12)
2. 网易云音乐评论 可视化分析(6)
3. python wordcloud 对电影《我不
是潘金莲》制作词云(3)
4. 介绍几个python的音频处理库(3)
5. Python 与 C/C++ 交互的几种方
式(3)

像的最左上角的坐标为原点(0,0),宽度的方向为x轴，高度的方向为y轴，每一个像素代表一个坐标单位。crop()返回的仍然是一个Image对象。

- **transpose(method)**(图像翻转或者旋转)

```
>>> im_rotate_180 = im.transpose(Image.ROTATE_180)
>>> im_rotate_180.show()
```

上面的代码将im逆时针旋转180°，然后显示出来，`method` 是transpose的参数，表示选择什么样的翻转或者旋转方式，可以选择的值有：

- Image.FLIP_LEFT_RIGHT,表示将图像左右翻转
- Image.FLIP_TOP_BOTTOM,表示将图像上下翻转
- Image.ROTATE_90,表示将图像逆时针旋转90°
- Image.ROTATE_180,表示将图像逆时针旋转180°
- Image.ROTATE_270,表示将图像逆时针旋转270°
- Image.TRANSPOSE,表示将图像进行转置(相当于顺时针旋转90°)
- Image.TRANSVERSE,表示将图像进行转置,再水平翻转

- **paste(region,box,mask)**(将一个图像粘贴到另一个图像)

```
>>> im.paste(region, (100,100),None)
>>> im.show()
```

上面的代码将region图像粘贴到左上角为(100,100)的位置。region是要粘贴的Image对象,box是要粘贴的位置，可以是一个两个元素的元组，表示粘贴区域的左上角坐标,也可以是一个四个元素的元组，表示左上角和右下角的坐标。如果是四个元素元组的话,box的size必须要和region的size保持一致，否则将会被convert成和region一样的size。

- **split()**(颜色通道分离)

```
>>> r,g,b = im.split()
>>> r.show()
>>> g.show()
>>> b.show()
```

split()方法可以原来图像的各个通道分离,比如对于RGB图像，可以将其R,G,B三个颜色通道分离。

- **merge(mode,channels)**(颜色通道合并)

```
>>> im_merge = Image.merge("RGB", [b,r,g])
>>> im_merge.show()
```

merge方法和split方法是相对的，其将多个单一通道的序列合并起来，组成一个多通道的图像，mode是合并之后图像的模式，比如"RGB",channels是多个单一通道组成的序列。

- **resize(size,resample,box)**

```
>>> im_resize = im.resize((200,200))
>>> im_resize
<PIL.Image.Image image mode=RGB size=200x200 at 0x7F62B9E23470>
>>> im_resize.show()
>>> im_resize_box = im.resize((100,100),box = (0,0,50,50))
>>> im_resize_box.show()
```

resize方法可以将原始的图像转换大小,size是转换之后的大小,resample是重新采样使用的方法，仍然有 `Image.BICUBIC`，`PIL.Image.LANCZOS`，`PIL.Image.BILINEAR`，`PIL.Image.NEAREST` 这四种采样方法，默认是 `PIL.Image.NEAREST`，box是指定的要resize的图像区域，是一个用四个元组指定的区域(含义和上面所述box一致)。

- **convert(mode,matrix,dither,palette,colors)**(mode转换)

```
>>> im_L = im.convert("L")
>>> im_L.show()
>>> im_rgb = im_L.convert("RGB")
>>> im_rgb.show()
>>> im_L.mode
'L'
>>> im_rgb.mode
'RGB'
```

convert方法可以改变图像的mode,一般是在'RGB'(真彩图)、'L'(灰度图)、'CMYK'(压缩图)之间转换。上面的代码就是首先将图像转化为灰度图，再从灰度图转化为真彩图。值得注意的是,从灰度图转换为真彩图，虽然理论上确实转换成功了，但是实际上是很难恢复成原来的真彩模式的(不唯一)。

- **filter(filter)**(应用过滤器)

```
>>> im = Image.open("dog.jpg","r")
>>> from PIL import ImageFilter
>>> im_blur = im.filter(ImageFilter.BLUR)
>>> im_blur.show()
>>> im_find_edges = im.filter(ImageFilter.FIND_EDGES)
>>> im_find_edges.show()
>>> im_find_edges.save("find_edges.jpg")
>>> im_blur.save("blur.jpg")
```

filter方法可以将一些过滤器操作应用于原始图像，比如模糊操作，查找边、角点操作等。filter是过滤器函数，在 `PIL.ImageFilter` 函数中定义了大量内置的filter函数，比如 `BLUR` (模糊操作)，`GaussianBlur` (高斯模糊)，`MedianFilter` (中值过滤器)，`FIND_EDGES` (查找边)等。上面得到原始图像dog.jpg,find_edges.jpg以及blur.jpg从左到右如下图1所示：



图1 从左到右分别是:dog.jpg,find_edges.jpg以及blur.jpg

- **point(lut,mode)**(对图像像素操作)

```
>>> im_point = im.point(lambda x:x*1.5)
>>> im_point.show()
>>> im_point.save("im_point.jpg")
```

point方法可以对图像进行单个像素的操作，上面的代码对point方法传入了一个匿名函数,表示将图像的每个像素点大小都乘以1.5,mode是返回的图像的模式，默认是和原来图像的mode是一样的。图2是原来的dog.jpg和point操作之后的im_point.jpg之间的对比。



图2 dog.jpg和point操作之后的im_point.jpg

下面是一个结合了 `point` 函数, `split` 函数, `paste` 函数以及 `merge` 函数的小例子。

```
>>> source = im.split()
>>> R,G,B = 0,1,2
>>> mask = source[R].point(lambda x: x<100 and 255)
>>> # x<100,return 255,otherwise return 0
>>> out_G = source[G].point(lambda x:x*0.7)
>>> # 将out_G粘贴回来,但是只保留'R'通道像素值<100的部分
>>> source[G].paste(out_G,None,mask)
>>> # 合并成新的图像
>>> im_new = Image.merge(im.mode,source)
>>> im_new.show()
>>> im.show()
```

- **ImageEnhance()**(图像增强)

```
>>> from PIL import ImageEnhance
>>> brightness = ImageEnhance.Brightness(im)
>>> im_brightness = brightness.enhance(1.5)
>>> im_brightness.show()
>>> im_contrast = ImageEnhance.Contrast(im)
>>> im_contrast.enhance(1.5)
<PIL.Image.Image image mode=RGB size=296x299 at 0x7F62AE271AC8>
>>> im_contrast.enhance(1.5).show()
```

ImageEnhance是PIL下的一个子类，主要用于图像增强，比如增加亮度(Brightness),增加对比度(Contrast)等。上面的代码将原来图像的亮度增加50%,将对对比度也增加了50%。

- **ImageSequence()**(处理图像序列)

下面的代码可以遍历gif图像中的所有帧，并分别保存为图像

```
>>> from PIL import ImageSequence
>>> from PIL import Image
>>> gif = Image.open("pipixia.gif")
>>> for i,frame in enumerate(ImageSequence.Iterator(gif),1):
...     if frame.mode == 'JPEG':
```

```
...     frame.save("%d.jpg" % i)
...     else:
...         frame.save("%d.png" % i)
```

除了上面使用迭代器的方式以外，还可以一帧一帧读取gif,比如下面的代码：

```
>>> index = 0
>>> while 1:
...     try:
...         gif.seek(index)
...         gif.save("%d.%s" % (index, 'jpg' if gif.mode == 'JPEG' else 'png'))
...         index += 1
...     except EOFError:
...         print("Reach the end of gif sequence!")
...         break
```

上面的代码在读取到gif的最后一帧之后，会throw 一个 EOFError,所以我们只要捕获这个异常就可以了。

热爱编程，热爱机器学习！ github:<http://www.github.com/Lyrichu> github blog:<http://Lyrichu.github.io> 个人博客站点:<http://www.movieb2b.com>(不再维护)

分类: [python学习](#), [图像视频处理](#)

标签: [python](#), [PIL](#), [图像处理](#)



 lyrichu
关注 - 6
粉丝 - 111
[+加关注](#)

2

0

« 上一篇: [python自动制作gif并添加文字](#)

» 下一篇: [条件GAN论文简单解读](#)

posted on 2018-06-04 15:10 [lyrichu](#) 阅读(10168) 评论(1) [编辑](#) [收藏](#)

评论：

#1楼 2018-11-23 10:32 | [别动我的砖、我还可以搬太棒了！](#)

[支持\(0\)](#) [反对\(0\)](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】超50万C++/C#源码：大型实时仿真HMI组态CAD\GIS图形源码！

【推荐】专业便捷的企业级代码托管服务 - Gitee 码云

相关博文：

- [Python 之 使用 PIL 库做图像处理](#)
- [Python安装PIL库](#)
- [Python图像处理库\(PIL\)--简介和基本概念](#)
- [python PIL 库处理文件](#)
- [Python图像处理库\(PIL\)](#)

最新新闻：

- [李国庆，千万别搞区块链](#)
- [京东企业购To B：求变与进击](#)
- [搜狐21岁 张朝阳发表演讲：我们刚过半程 下半场刚刚开始](#)
- [5G折叠屏手机哪家强：三星、华为、小米、中兴旗舰机大比拼](#)
- [马斯克破局临界点：环评书称上海特斯拉今年9月完成四车间建设](#)
- » [更多新闻...](#)

