Python 图像处理库 Pillow 入门（含代码）

Pillow 是 Python 里的图像处理库（PIL：Python Image Library），提供了了广泛的文件格式支持，强大的图像处理能力，主要包括图像储存、图像显示、格式转换以及基本的图像处理操作等。

1）使用 Image 类

PIL 最重要的类是 Image class，你可以通过多种方法创建这个类的实例；你可以从文件加载图像，或者处理其他图像，或者从 scratch 创建。

要从文件加载图像，可以使用 open( )函数，在 Image 模块中：

```
>>> from PIL import Image
>>> im = Image.open("E:/photoshop/1.jpg")
```

加载成功后，将返回一个 Image 对象，可以通过使用示例属性查看文件内容：

```
>>> print(im.format, im.size, im.mode)
('JPEG', (600, 351), 'RGB')

>>>
```

format 这个属性标识了图像来源。如果图像不是从文件读取它的值就是 None。size 属性是一个二元 tuple，包含 width 和 height（宽度和高度，单位都是 px）。mode 属性定义了图像 bands 的数量和名称，以及像素类型和深度。常见的 modes 有 “L” (luminance) 表示灰度图像，“RGB” 表示真彩色图像, and “CMYK” 表示出版图像。

如果文件打开错误，返回 IOError 错误。

只要你有了 Image 类的实例，你就可以通过类的方法处理图像。比如，下列方法可以显示图像：

```
im.show()
```

2）读写图像

PIL 模块支持大量图片格式。使用在 Image 模块的 open() 函数从磁盘读取文件。你不需要知道文件格式就能打开它，这个库能够根据文件内容自动确定文件格式。要保存文件，使用 Image 类的 save() 方法。保存文件的时候文件名变得重要了。除非你指定格式，否则这个库将会以文件名的扩展名作为格式保存。加载文件，并转化为 png 格式：

```
"Python Image Library Test"

from PIL import Image

import os

import sys


for infile in sys.argv[1:]:
    f,e = os.path.splitext(infile)
    outfile = f +".png"
    if infile != outfile:
        try:
            Image.open(infile).save(outfile)
        except IOError:
            print("Cannot convert", infile)
```

save() 方法的第二个参数可以指定文件格式。

3）创建缩略图

缩略图是网络开发或图像软件预览常用的一种基本技术，使用 Python 的 Pillow 图像库可以很方便的建立缩略图，如下：

```
# create thumbnail
size = (128,128)
for infile in glob.glob("E:/photoshop/*.jpg"):
    f, ext = os.path.splitext(infile)
    img = Image.open(infile)
    img.thumbnail(size,Image.ANTIALIAS)
```

```
        img.save(f+".thumbnail","JPEG")
```

上段代码对 photoshop 下的 jpg 图像文件全部创建缩略图，并保存，glob 模块是一种智能化的文件名匹配技术，在批图像处理中经常会用到。

注意：Pillow 库不会直接解码或者加载图像栅格数据。当你打开一个文件，只会读取文件头信息用来确定格式，颜色模式，大小等等，文件的剩余部分不会主动处理。这意味着打开一个图像文件的操作十分快速，跟图片大小和压缩方式无关。

4）图像的剪切、粘贴与合并操作

Image 类包含的方法允许你操作图像部分选区，PIL.Image.Image.crop 方法获取图像的一个子矩形选区，如：

```
# crop, paste and merge
im = Image.open("E:/photoshop/lena.jpg")
box = (100,100,300,300)
region = im.crop(box)
```

矩形选区有一个 4 元元组定义，分别表示左、上、右、下的坐标。这个库以左上角为坐标原点，单位是 px，所以上诉代码复制了一个 200×200 pixels 的矩形选区。这个选区现在可以被处理并且粘贴到原图。

```
region = region.transpose(Image.ROTATE_180)
im.paste(region, box)
```

当你粘贴矩形选区的时候必须保证尺寸一致。此外，矩形选区不能在图像外。然而你不必保证矩形选区和原图的颜色模式一致，因为矩形选区会被自动转换颜色。

5）分离和合并颜色通道

对于多通道图像，有时候在处理时希望能够分别对每个通道处理，处理完成后重新合成多通道，在 Pillow 中，很简单，如下：

```
r,g,b = im.split()
im = Image.merge("RGB", (r,g,b))
```

对于 split（ ）函数，如果是单通道的，则返回其本身，否则，返回各个通道。

6）几何变换

对图像进行几何变换是一种基本处理，在 Pillow 中包括 resize( )和 rotate( )，如用法如下：

```
out = im.resize((128,128))
out = im.rotate(45)  # degree conter-clockwise
```

其中，resize( )函数的参数是一个新图像大小的元祖，而 rotate( )则需要输入顺时针的旋转角度。在 Pillow 中，对于一些常见的旋转作了专门的定义：

```
out = im.transpose(Image.FLIP_LEFT_RIGHT)
out = im.transpose(Image.FLIP_TOP_BOTTOM)
out = im.transpose(Image.ROTATE_90)
out = im.transpose(Image.ROTATE_180)
out = im.transpose(Image.ROTATE_270)
```

7）颜色空间变换

在处理图像时，根据需要进行颜色空间的转换，如将彩色转换为灰度：

```
cmyk = im.convert("CMYK")
gray = im.convert("L")
```

8）图像滤波

图像滤波在 ImageFilter 模块中，在该模块中，预先定义了很多增强滤波器，可以通过 filter( )函数使用，预定义滤波器包括：

BLUR、CONTOUR、DETAIL、EDGE_ENHANCE、EDGE_ENHANCE_MORE、EMBOSS、FIND_EDGES、SMOOTH、SMOOTH_MORE、SHARPEN。其中 BLUR 就是均值滤波，CONTOUR 找轮廓，FIND_EDGES 边缘检测，使用该模块时，需先导入，使用方法如下：

```
from PIL import ImageFilter
```

```
imgF = Image.open("E:/photoshop/lena.jpg")
outF = imgF.filter(ImageFilter.DETAIL)
conF = imgF.filter(ImageFilter.CONTOUR)
edgeF = imgF.filter(ImageFilter.FIND_EDGES)
imgF.show()
outF.show()
conF.show()
edgeF.show()
```

除此以外，ImageFilter 模块还包括一些扩展性强的滤波器：

class PIL.ImageFilter.GaussianBlur(radius=2)

Gaussian blur filter.

参数

radius – Blur radius.

class PIL.ImageFilter.UnsharpMask(radius=2, percent=150, threshold=3)

Unsharp mask filter.

See Wikipedia’s entry on digital unsharp masking for an explanation of the parameters.

class PIL.ImageFilter.Kernel(size, kernel, scale=None, offset=0)

Create a convolution kernel. The current version only supports 3×3 and 5×5 integer and floating point kernels.

In the current version, kernels can only be applied to “L” and “RGB” images.

参数：

size – Kernel size, given as (width, height). In the current version, this must be (3,3) or (5,5).

kernel – A sequence containing kernel weights.

scale – Scale factor. If given, the result for each pixel is divided by this value. the default is the sum of the kernel weights.

offset – Offset. If given, this value is added to the result, after it has been divided by the scale factor.

class PIL.ImageFilter.RankFilter(size, rank)

Create a rank filter. The rank filter sorts all pixels in a window of the given size, and returns therank 'th value.

参数：

size – The kernel size, in pixels.

rank – What pixel value to pick. Use 0 for a min filter, size * size / 2 for a median filter, size * size - 1 for a max filter, etc.

class PIL.ImageFilter.MedianFilter(size=3)

Create a median filter. Picks the median pixel value in a window with the given size.

参数：

size – The kernel size, in pixels.

class PIL.ImageFilter.MinFilter(size=3)

Create a min filter. Picks the lowest pixel value in a window with the given size.

参数:

size – The kernel size, in pixels.

class PIL.ImageFilter.MaxFilter(size=3)

Create a max filter. Picks the largest pixel value in a window with the given size.

参数:

size – The kernel size, in pixels.

class PIL.ImageFilter.ModeFilter(size=3)

Create a mode filter. Picks the most frequent pixel value in a box with the given size. Pixel values that occur only once or twice are ignored; if no pixel value occurs more than twice, the original pixel value is preserved.

参数:

size – The kernel size, in pixels.更多详细内容可以参考：PIL/ImageFilter

9）图像增强

图像增强也是图像预处理中的一个基本技术，Pillow 中的图像增强函数主要在 ImageEnhance 模块下，通过该模块可以调节图像的颜色、对比度和饱和度和锐化等：

```
from PIL import ImageEnhance

imgE = Image.open("E:/photoshop/lena.jpg")
imgEH = ImageEnhance.Contrast(imgE)
imgEH.enhance(1.3).show("30% more contrast")
```

图像增强：

class PIL.ImageEnhance.Color(image)

Adjust image color balance.

This class can be used to adjust the colour balance of an image, in a manner similar to the controls on a colour TV set. An enhancement factor of 0.0 gives a black and white image. A factor of 1.0 gives the original image.

class PIL.ImageEnhance.Contrast(image)

Adjust image contrast.

This class can be used to control the contrast of an image, similar to the contrast control on a TV set. An enhancement factor of 0.0 gives a solid grey image. A factor of 1.0 gives the original image.

class PIL.ImageEnhance.Brightness(image)

Adjust image brightness.

This class can be used to control the brighntess of an image. An enhancement factor of 0.0 gives a black image. A factor of 1.0 gives the original image.

class PIL.ImageEnhance.Sharpness(image)

Adjust image sharpness.

This class can be used to adjust the sharpness of an image. An enhancement factor of 0.0 gives a blurred image, a factor of 1.0 gives the original image, and a factor of 2.0 gives a sharpened image.

图像增强的详细内容可以参考：PIL/ImageEnhance

除了以上介绍的内容外，Pillow 还有很多强大的功能：

PIL.Image.alpha_composite(im1, im2)

PIL.Image.blend(im1, im2, alpha)

PIL.Image.composite(image1, image2, mask)

PIL.Image.eval(image, *args)

PIL.Image.fromarray(obj, mode=None)

PIL.Image.frombuffer(mode, size, data, decoder_name=’raw’, *args)