**torchvision.transforms.Compose()**

此函数功能是串联多个图片变换的操作

**可选操作**

transforms.Resize(224)：将图象短边缩放至224，保持长宽比例不变；transforms.Resize([224,448])：将加载图像变为224×448大小

transforms.RandomRotation(degrees=(-30, 40))：保持图像大小不变随机旋转-30到40度，正数表示顺时针，负数表示逆时针，旋转图像后的空缺区域用黑框补齐

transforms.CenterCrop(224)：在中心点处按224×224大小裁剪，防止旋转后出现黑框部分

transforms.RandomHorizontalFlip(p=0.3)：以0.3的概率随机水平翻转

transforms.RandomVerticalFlip(p=0.3)：以0.3的概率随机垂直翻转

transforms.ToTensor()：把shape=(H×W×C)的像素值范围为[0, 255]的PIL.Image或者numpy.ndarray转换成shape=(C×H×W)的像素值范围为[0.0, 1.0]的torch.FloatTensor

transforms.Normalize(mean=[0.485, 0.456, 0.406]，std = [0.229, 0.224, 0.225])：用公式进行标准化，其中mean和std一般从数据集中随机抽样计算得到的

设loader是transforms.Compose()的实例，则loader(image)可以对图片进行处理

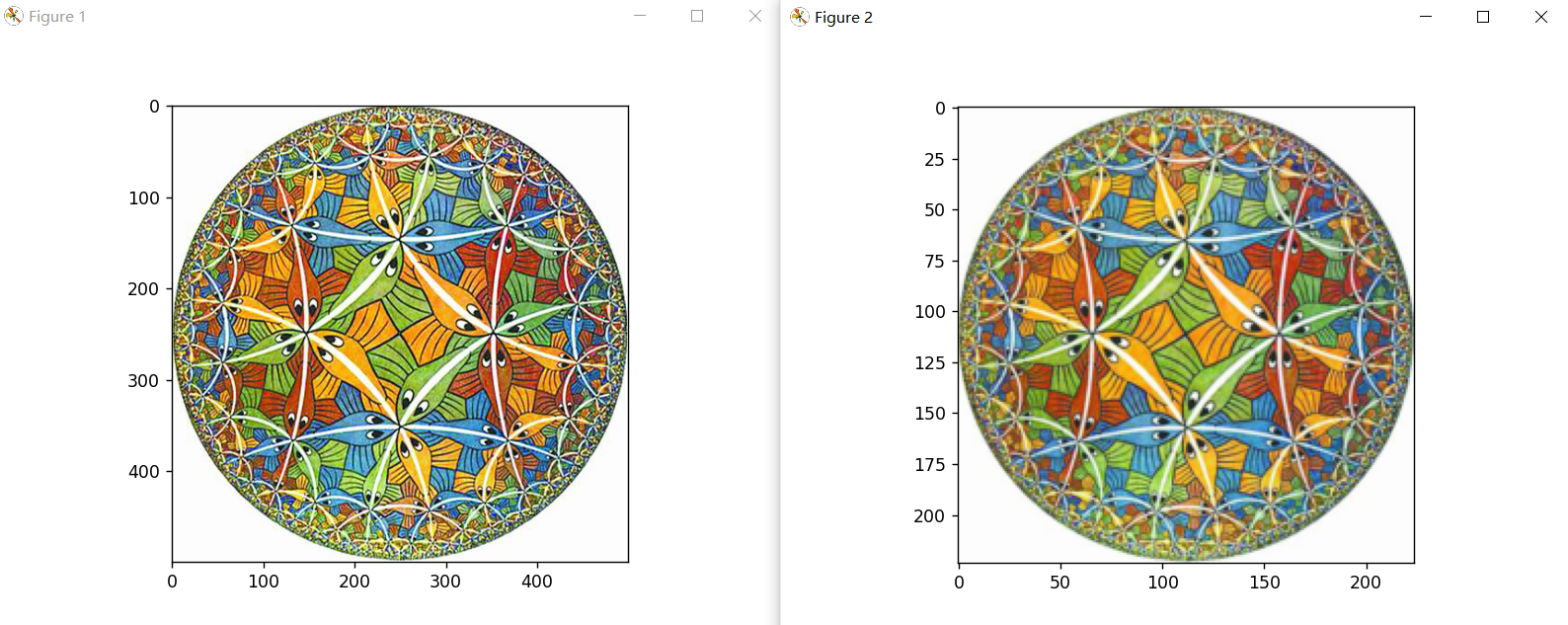
**输入图片格式**

image可以是PIL，numpy.ndarray

import torchvision.transforms as transforms  
from PIL import Image  
import matplotlib.pyplot as plt  
loader = transforms.Compose([  
 transforms.Resize(224), *#将加载的图像(500×500×3)转变(224×224×3)* transforms.ToTensor()]) *#将图像转化为(3×224×224)的tensor并归一化*image = Image.open('escher.jpg')  
unloader = transforms.ToPILImage()  
plt.figure()  
plt.imshow(image)  
plt.figure()  
image = loader(image)  
print(image.size())  
image = unloader(image)  
plt.imshow(image)  
plt.show()

>>>

torch.Size([3, 224, 224])



import torchvision.transforms as transforms  
from PIL import Image  
import matplotlib.pyplot as plt  
loader = transforms.Compose([  
 transforms.RandomRotation(degrees=60),  
 #transforms.CenterCrop(224),  
 transforms.ToTensor()])

image = Image.open('portrait1.jpg')  
unloader = transforms.ToPILImage()  
plt.figure()  
plt.imshow(image)  
plt.figure()  
image = loader(image)  
image = unloader(image)  
plt.imshow(image)  
plt.show()

>>>

