Python图像处理的基本操作

导入所需要的包

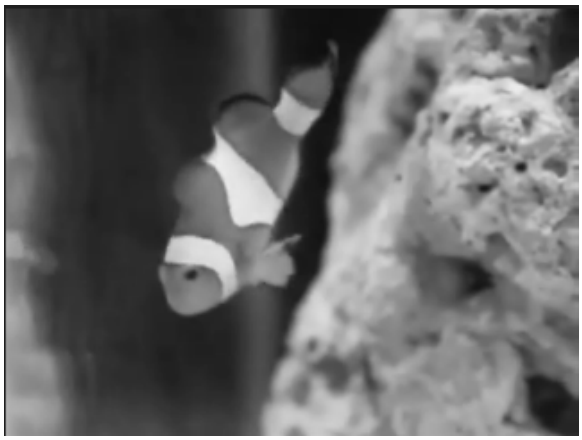
```
import numpy as np
import PIL.Image as Image
import matplotlib.pyplot as plt
import math
```

读取、显示、保存图像

```
#读取图像
image_path = "path/to/309.bmp"  #图像所在的路径
img = Image.open(image_path)

#显示图像
img.show()
#也可以使用matplotlib库来显示图像
plt.imshow(img)
plt.show()

#保存图像
save_path = "save/path"   #图像保存路径
img.save(save_path)

#打印图像大小和属性
print(img.width, img.height, img.mode, img.format, type(img))
#320 240 RGB BMP <class 'PIL.BmpImagePlugin.BmpImageFile'>
```

彩色图像转灰度图像

```
#灰度化图像
#将每个像素用8个bit表示，0表示黑，255表示白，其他数字表示不同的灰度。
#转换公式：L = R * 299/1000 + G * 587/1000+ B * 114/1000。
img_gray = img.convert('L')
img_gray.show()
```

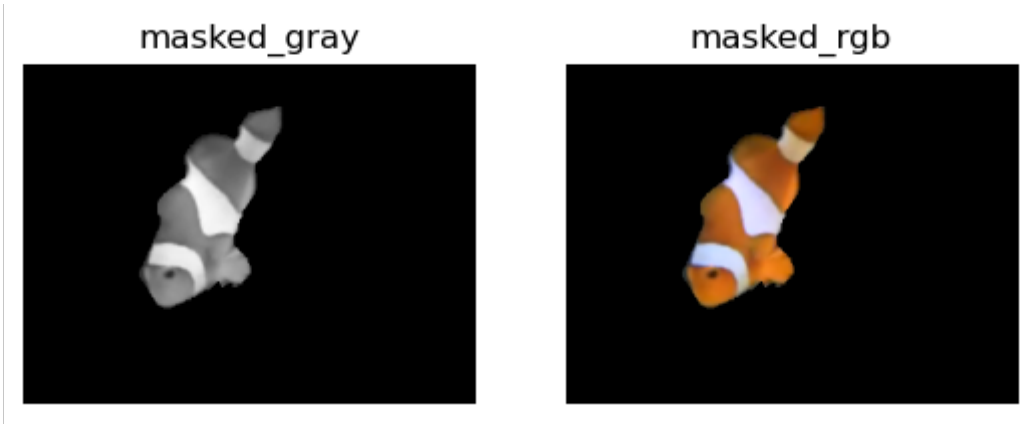


绘出mask后的图像

```
#读取掩码mask文件并转为ndarray格式
mask_path = 'path/to/mask.csv'
with open(mask_path) as f:
    mask = csv.reader(f)
    mask = list(mask)
    mask = np.array(mask, dtype=np.int8)

#展示掩码图像
mask_show = Image.fromarray(mask*255)
mask_show.show()

#对图像进行操作时需要将图像同样转为ndarray格式
img_rgb = np.array(img)
img_gray = np.array(img_gray)

#对图像进行掩码
masked_gray = img_gray * mask/255
mask_rgb = np.array([mask, mask, mask]).transpose(1, 2, 0) #transpose用于改变矩阵维度序列
masked_rgb = img * mask_rgb/255
```

matplotlib库的基本操作

```
#绘图
plt.figure()
plt.subplot(1, 2, 1)        #一个1*2的图像列表的第一幅图
plt.title('masked_gray')    #设置图像标题
plt.imshow(masked_gray, cmap='gray')
plt.axis('off')             ##关闭坐标轴
plt.subplot(1, 2, 2)
plt.title('masked_rgb')
plt.imshow(masked_rgb)
plt.axis('off')
plt.show()
#其他操作
#plt.plot()  绘制曲线
#plt.legend()  显示图例
#plt.title()  设置标题
#plt.xlabel()  设置x轴标签
#plt.ylabel()  设置y轴标签
#plt.xlim()  设置x轴范围
#plt.ylim()  设置y轴范围
#plt.xticks()  设置x轴刻度
#plt.yticks()  设置y轴刻度
#plt.savefig()  保存图片
```



绘制正态分布函数

正态分布由μ (均值)、σ (方差)两个参数确定

```
#正态分布计算函数
def pdf(x,mu,sigma):
    sqrt_2pi = math.sqrt(2*math.pi)
    index = -0.5*((x-mu)/sigma)**2
    y = (1/(sqrt_2pi*sigma))*np.exp(index)
    return y

mu = 0  # 均值
sigma = 1  # 方差
# 绘制正态分布的概率密度函数图形
x_normal = np.linspace(-4, 4, 100)  # x轴范围
y_normal = pdf(x_normal, mu, sigma)  # 计算概率密度函数值
plt.plot(x_normal, y_normal, color='red', label='N(0,1)')
plt.xticks(np.arange(-4, 5, 1))     #x轴刻度
plt.xlabel('x')
plt.ylabel('Probability Density')
plt.title('Normal Distribution')
plt.legend()
plt.show()
```