CTF misc常用的图像处理

原创

Hardworking666 于 2021-11-24 13:46:51 发布 3134 收藏 1

分类专栏： CTF 文章标签： unctf python 安全 网络安全 图像处理

本文链接：https://blog.csdn.net/Hardworking666/article/details/121513611

文章目录

一、Pillow

```
#基本操作
from PIL import Image  #引用Image类
im = Image.new('RGB',(110,80))  #新建图片
im = Image.open('x.jpg')  #加载图片
print(im.format,im.size,im.mode)  #format-格式，size-(宽&高)，mode-模式
im.show()  #呈现图片
im.getcolors()  #获取图片信息，一般会返回一个元组(count,(r,g,b))。该元组第一个元素count 代表该颜色像素出现的次数，第二个元素表示(rgb)。
im.convert('RGB').getpixel((0,0))  #获取像素点的RGB值
Image.save('y.png')  #保存文件

#图片剪切
#box是一个4元的坐标数组，坐标轴是左上角是(0,0)的卡迪尔坐标系。假设box是(x1,y1,x2,y2)，则所取区域是以各自坐标划线所围的区域。
im = Image.open('x.jpg')
box = (150,150,245,280)
region = im.crop(box)
region.show()

#图片粘贴
#将一张图覆盖到另一张图上面。格式为：paste(要贴的图片，要贴的图片的4元坐标组成的区域)。
im = Image.open('x.jpg')
box = (50,50,200,200)
region = im.crop(box)
# 将图片逆序旋转180后，黏贴到原来复制的位置
region = region.transpose(Image.ROTATE_180)
im.paste(region,box)
im.show()

#图像序列
#当处理GIF这种包含多个帧的图片，称之为序列文件，PIL会自动打开序列文件的第一帧。而使用seek和tell方法可以在不同帧移动。tell是帧数，而seek是取当前帧数的图片。
from PIL import Image
im = Image.open("laopo.gif")
im.seek(1)
im.show()
try:
 while 1:
 im.seek(im.tell()+1)
 im.show()
except EOFError:
 pass

#读取/修改像素
from PIL import Image
img = Image.open('x.jpg')
width , height = img.size
for i in range(0,width):
 for j in range(0,height):
  tmp = img.getpixel((i,j))
  img.putpixel((i,j),(0,0,tmp[2]))
img.show()
```

使用Pillow来进行图像处理

二、OpenCV2

```
#图像读入
import cv2
```

```
img = cv2.imread(r'x.jpg',0)
#图像显示
cv2.imshow('tupian',img)
cv2.waitKey()
#图像保存
cv2.imwrite('tupian,jpg',img)

#图像属性
print(img.shape)   #shape 返回图像行数、列数、通道数
print(img.size)    #size  返回图像像素数
print(img.dtype)   #dtype 返回图像的数据类型

#通道拆分/合并
#拆分为B、G、R三个通道
#索引拆分
B = img[:,:,0]
G = img[:,:,1]
R = img[:,:,2]
#函数拆分
B,G,R = cv2.split(img)
#合并
bgr = cv2.merge([b,g,r])

#类型转换
#将BGR模式转换为灰度图像，再将灰度图像转换为RGB模式
import cv2
import numpy as np
bgr=np.random.randint(0,256,size=[2,4,3],dtype=np.uint8)  #生成2×4×3的BGR图像
gray=cv2.cvtColor(bgr,cv2.COLOR_BGR2GRAY)   #BGR图像转换为灰度图像
rgb=cv2.cvtColor(gray,cv2.COLOR_GRAY2RGB)   #灰度图像转换为RGB图像
print('bgr=\n',bgr)
print('gray=\n',gray)
print('rgb=\n',rgb)
# 在RGB色彩空间的基础上，还可以加一个A通道，叫做Alpha通道，此时原图像类型转变为RGBA模式，例如常见的PNG类型图像就是RGBA色彩空间的。
#分析alpha通道的值
import cv2
import numpy as np
img=np.random.randint(0,256,size=[2,3,3],dtype=np.uint8)  #生成2×3×3的BGR图像
bgra=cv2.cvtColor(img,cv2.COLOR_BGR2BGRA)
print('img=\n',img)
print('bgra=\n',bgra)
b,g,r,a=cv2.split(bgra)
print('a=\n',a)
a[:,:]=125
bgra=cv2.merge([b,g,r,a])
print('bgra=\n',bgra)

#傅里叶变换
img = cv2.imread(r"x.jpg",0)
# 图像数据要转换成float32
img_float32 = np.float32(img)
#进行傅里叶变换
dft = cv2.dft(img_float32,flags = cv2.DFT_COMPLEX_OUTPUT)
# 将低频信息转换至图像中心
dft_shift = np.fft.fftshift(dft)
# 傅里叶变换后的数据是由实部虚部构成的，需要进行转换成图像格式才能显示(0,255)
magnitude = 20*np.log(cv2.magnitude(dft_shift[:,:,0],dft_shift[:,:,1]))
plt.subplot(121),plt.imshow(img,cmap = 'gray')
plt.subplot(122),plt.imshow(magnitude,cmap='gray')
```

```
plt.show()
```

三、Matplotlib

```
#显示图片
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np

picture = mpimg.imread('x.jpg') # 读取和代码处于同一目录下的图片
plt.imshow(picture) # 显示图片
plt.axis('on') # 显示坐标轴
plt.show() #因为idle是pycharm所以多一行显示命令

#显示RGB某个通道
lena1 = picture*[0,0,1]   # 显示图片的一个通道
lena2 = picture*[1,0,0]
lena3 = picture*[0,1,0]

#RGB转灰度
def rgb2gray2(rgb):
    r, g, b = rgb[:,:,0], rgb[:,:,1], rgb[:,:,2] #数组切片
    gray = 0.2989 * r + 0.5870 * g + 0.1140 * b
    return gray
gray_pro = rgb2gray2(picture)
plt.imshow(gray_pro,cmap = plt.get_cmap('gray'))
plt.axis('on')  # 显示坐标轴
plt.show()

#保存图片
plt.savefig('xx.jpg')
```