




数字图像处理实验--实验项目一 图像的基本操作和基本运算

原创

冰山、一角  已于 2022-01-30 11:10:09 修改  3383  收藏 4

文章标签：[图像处理](#) [计算机视觉](#) [python](#)

于 2022-01-28 12:01:52 首次发布

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/qq_57171795/article/details/122727860

版权

目录

前言

实验项目一 图像的基本操作和基本运算

1. 【图像的读取操作】
2. 【图像的基本运算】
3. 【图像的几何变换】
4. 【图像的灰度变换】

前言

数字图像处理（Digital Image Processing）又称为计算机图像处理，它是指将图像信号转换成数字信号并利用计算机对其进行处理的过程。

数字图像处理的主要目的一般来讲可以划分为：提高图像的视感质量；提取图像中所包含的某些特征或特殊信息，这些被提取的特征或信息往往为计算机分析图像提供便利，提取特征或信息的过程是模式识别或计算机视觉的预处理；图像数据的变换、编码和压缩，以便于图像的存储和传输。

数字图像处理常用方法有以下几个方面：图像变换；图像编码压缩；图像增强和复原；图像分割；图像描述；图像分类（识别）。

为了熟悉和锻炼数字图像处理能力，通过实验的方法进行一些简单的数字图像处理操作。使用的编程语言是python。使用的图片在文章末尾（侵权联系删除）

实验项目一 图像的基本操作和基本运算

一、实验目的：了解图像处理软件的使用，掌握图像的读取，显示，保存；以及图像的基本运算和基本操作。

二、实验要求：会编程实现图像的读取、显示、保存；会编程实现图像的加、减、乘、除运算；会进行图像的缩放、旋转、平移、镜像等几何变换；会通过编程实现彩色图像的灰度化和二值化，以及通过点运算实现图像的增强变换，同时会通过直方图均衡化和规定化实现图的像增强。

三、实验内容

1. 【图像的读取操作】

读入一幅RGB图像，变换为灰度图像和二值图像，并在同一个窗口内分成三个子窗口来分别显示RGB图像和灰度图像，注上文字标题

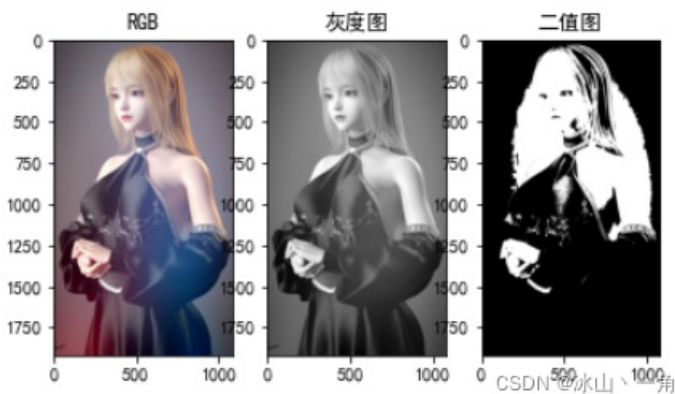
```

from skimage import io,color
from matplotlib import pyplot as plt
import numpy as np
#中文显示
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus'] = False
#读取
img=io.imread('./picture/001.jpg')
#灰度变换
img2=color.rgb2gray(img)
#二值化
img3=np.zeros((img2.shape[0],img2.shape[1]))
mid=(np.max(img2)-np.min(img2))/2+np.min(img2)
two=lambda x: 1 if x > mid else 0#lambda
for i in range(img2.shape[0]):
    for j in range(img2.shape[1]):

        # if img2[i][j]>mid:
        #     img3[i][j]=1
        # else:
        #     img3[i][j]=0
        #使用lambda函数进行判断
        img3[i][j]=two(img2[i][j])
plt.subplot(131),plt.imshow(img,'gray'),plt.title('RGB')
plt.subplot(132),plt.imshow(img2,cmap=plt.cm.gray),plt.title("灰度图")
plt.subplot(133),plt.imshow(img3,cmap=plt.cm.gray),plt.title("二值图")
plt.show()

```

结果:



2 【图像的基本运算】

对两幅不同图像执行加、减、乘、除操作，在同一个窗口内分成六个子窗口来分别显示，注上文字标题。

```

from skimage import io,color
from matplotlib import pyplot as plt
import numpy as np
img1=io.imread(r'./01 (1).jpg')
img2=io.imread(r'./01 (2).jpg')
row=np.min((img1.shape[0],img2.shape[0]))
col=np.min((img1.shape[1],img2.shape[1]))
for i in range(3):
    img1=img1[:row,:col,:]
    img2=img2[:row,:col,:]

imgjia=(img1+img2)/256#加
imgjian=(img1-img2)/255#减
imgcheng=(img1*img2)/255#乘
imgchu=(img1/img2)/255#除
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus']=False
plt.subplot(231),plt.imshow(img1),plt.title("第一张图")
plt.subplot(232),plt.imshow(img2),plt.title("第二张图")
plt.subplot(233),plt.imshow(imgjia),plt.title("加")
plt.subplot(234),plt.imshow(imgjian),plt.title("减")
plt.subplot(235),plt.imshow(imgcheng),plt.title("乘")
plt.subplot(236),plt.imshow(imgchu),plt.title("除")
plt.show()

```

结果:



3【图像的几何变换】

对一幅图像进行缩放、旋转、平移、裁剪、镜像变换，在同一个窗口内分成六个子窗口来分别显示原图和变换后的图像，注上文字标题

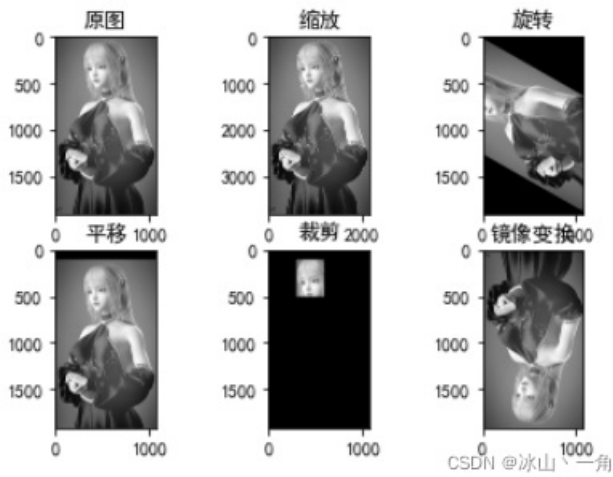
注意：这里的图像是灰度图，只有一层通道，相当与是对二维的矩阵进行操作。对彩色图像处理时，需要考虑三层通道。

```

from skimage import io,color
from matplotlib import pyplot as plt
import numpy as np
from skimage import io,color
from matplotlib import pyplot as plt
import numpy as np
from skimage import transform
img=io.imread(r'./01 (1).jpg')
img=color.rgb2gray(img)
img1=transform.rescale(img,2)#缩小为原来0.5倍数
img2=transform.rotate(img,60)#旋转60度
def Move(pic,delta_x,delta_y):      #平移
    #delta_x>0下移, delta_x<0上移
    #delta_y>0右移, delta_y<0左移
    newpic=np.zeros((pic.shape[0],pic.shape[1]))
    for i in range(pic.shape[0]):
        for j in range(pic.shape[1]):
            if i+delta_x>=pic.shape[0] or j+delta_y>=pic.shape[1]:
                pass
            else:
                newpic[i+delta_x][j+delta_y]=pic[i][j]
    return newpic
def Cut(pic,x_start,x_end,y_start,y_end):#裁剪 x纵坐标
    newpic=np.zeros((pic.shape[0],pic.shape[1]))
    for i in range(x_start,x_end):
        for j in range(y_start,y_end):
            newpic[i][j]=pic[i][j]
    return newpic
def Horizontal(pic):                #垂直镜像
    newpic=np.zeros((pic.shape[0],pic.shape[1]))
    mid=(pic.shape[0]-1)/2
    for i in range(pic.shape[0]):
        x=int(2*mid-i)
        for j in range(pic.shape[1]):
            newpic[x][j]=pic[i][j]
    return newpic
img3=Move(img,100,0)
img4=Cut(img,100,500,300,600)
img5=Horizontal(img)
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus']=False
plt.subplot(231),plt.imshow(img,plt.cm.gray),plt.title('原图')
plt.subplot(232),plt.imshow(img1,plt.cm.gray),plt.title('缩放')
plt.subplot(233),plt.imshow(img2,plt.cm.gray),plt.title('旋转')
plt.subplot(234),plt.imshow(img3,plt.cm.gray),plt.title('平移')
plt.subplot(235),plt.imshow(img4,plt.cm.gray),plt.title('裁剪')
plt.subplot(236),plt.imshow(img5,plt.cm.gray),plt.title('镜像变换')

```

结果:



4【图像的灰度变换】

对一幅图像进行灰度变化（自行设计线性变换或者非线性变换），实现图像变亮、变暗和负片效果，在同一个窗口内分成四个子窗口来分别显示，注上文字标题。

```

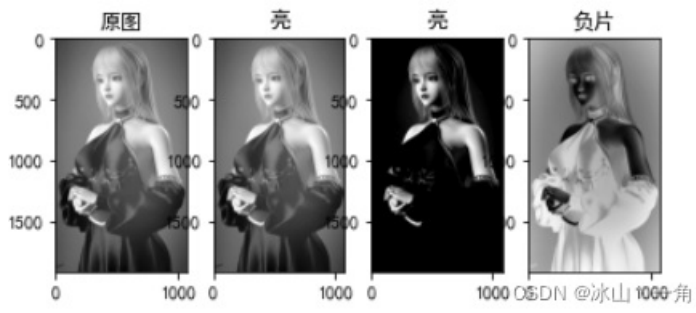
from skimage import io,color
from matplotlib import pyplot as plt
import numpy as np
#中文显示
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus'] = False

def linear(a,x,b):
    return a*x+b
# def notLinear(a,x,b):#使用一元二次函数
#     return a*x*x+b
def negative(x):
    return 255-x
img=io.imread(r'./01 (1).jpg')
img=color.rgb2gray(img)
img1=np.zeros((img.shape[0],img.shape[1]))
img2=np.zeros((img.shape[0],img.shape[1]))
img3=np.zeros((img.shape[0],img.shape[1]))
for i in range(img.shape[0]):
    for j in range(img.shape[1]):
        #亮
        a=2
        b=100
        img1[i][j]=linear(a,img[i][j],b) if linear(a,img[i][j],b)<255 else 255
        #亮
        a=1
        b=-0.5
        img2[i][j]=linear(a,img[i][j],b) if linear(a,img[i][j],b)>0 else 0
        #负片
        img3[i][j]=negative(img[i][j])/255

plt.subplot(141),plt.imshow(img,plt.cm.gray),plt.title('原图')
plt.subplot(142),plt.imshow(img1,plt.cm.gray),plt.title('亮')
plt.subplot(143),plt.imshow(img2,plt.cm.gray),plt.title('亮')
plt.subplot(144),plt.imshow(img3,plt.cm.gray),plt.title('负片')

```

结果:



01 (1).jpg



01 (2).jpg

还有其他实验将在后面陆续整理出来，一起学习一起进步吧。有问题请私信我
觉得有用的可以ctrl+D收藏哦