

**随机过程实验**

**实验题目 椒盐噪声及去噪实践**

**学 号 1180300922**

**姓 名 王志泓**

**指导教师 范晓鹏**

**日 期 2020.12.5**

1. 实验目的

理解椒盐噪声的产生原理及其滤除方式。

1. 实验内容

**问题一：**

编程读取一幅灰度bitmap图像，并读取图像中任意位置的像素值，理解基本的位图结构，并显示图像。

思考：如果是RGB数据会怎么样？如果是JPEG，PNG格式，最后在显示的时候如何显示？

**问题二：**

给定一幅8-比特灰度图像，调用噪声函数添加椒盐噪声。

**问题三：**

给定带椒盐噪声的图像，输出恢复的图像。

基本方法：利用非椒盐位置的真实像素信息（图像信号的先验信息，或者马尔可夫场的局部势能函数），来推导椒盐位置的真实像素值

思考:如何原来图像又含有高斯噪声，这时候怎么办 ?如果只含有高斯噪声，如何去噪效果会好？

1. 实验过程

**对于问题一**，python2中的库函数提供了一系列函数用于对于图像的处理，首先使用读取图像，可以将其灰度化，最后将其转为数组即可。

**对于问题二**，使用随机函数将数组中的值转为0或255即可模拟椒、盐两种情况的噪声。

**对于问题三**，对每个像素点，若其值为0或255，就假设其为噪声，并将其赋值为周围n\*n矩阵的加权平均值即可，其中n根据信噪比确定，噪声越多，所需要参考的周边矩阵就需要越大，即邻域越大。权重则根据邻域内某点到该点的距离确定，距离越近，权重越大。

1. 实验结果

**问题一代码如下：**

**﻿**#! python2

from PIL import Image

from numpy import \*

img = array(Image.open('original.jpg').convert('L'))

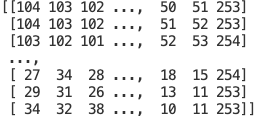
im = Image.fromarray(img)

im.save('gray.jpg')

**问题一结果如下：**

** **

**原始图像 灰度图像**

****

**Img数组内容**

**思考：**对于RGB数据，可以用三元数组形式存储颜色数据。

**问题二代码如下：**

# -\*- coding: utf-8 -\*-

from PIL import Image

from pylab import \*

from numpy import \*

#读取图片,灰度化，并转为数组

img = array(Image.open('original.jpg').convert('L'))

#信噪比

SNR = 0.95

#计算总像素数目 SP， 得到要加噪的像素数目 NP = SP \* (1-SNR)

noiseNum=int((1- SNR)\*img.shape[0]\*img.shape[1])

#于随机位置将像素值随机指定为0或者255

for i in range(noiseNum):

randX=random.random\_integers(0,img.shape[0]-1)

randY=random.random\_integers(0,img.shape[1]-1)

if random.random\_integers(0,1)==0:

img[randX,randY]=0

else:

img[randX,randY]=255

#显示图像

gray()

im = Image.fromarray(img)

im.save('output.jpg')

**问题二结果如下：**

** **

**噪声量为0.05 噪声量为0.5**

****

**噪声量为0.95**

**问题三代码如下：**

**﻿** # -\*- coding: utf-8 -\*-

from PIL import Image

def depoint(img):

pixdata = img.load()

w, h = img.size

index = 2

for y in range(index, h-index):

for x in range(index, w-index):

if pixdata[x,y] == 0 or pixdata[x,y] == 255:

count = 0

tmp = 0

for i in range(x-index,x+1+index):

for j in range(y-index,y+1+index):

counttmp = 0

if pixdata[i,j] == 0 or pixdata[i,j] == 255:

continue

else:

if i != x:

counttmp += abs(1/(i-x))

if j != y:

counttmp += abs(1/(j-y))

tmp += counttmp\*pixdata[i,j]

count += counttmp

if count == 0:

continue

pixdata[x,y] = int(tmp/count)

return img

if \_\_name\_\_ == '\_\_main\_\_':

img = Image.open('output.jpg')

w, h = img.size

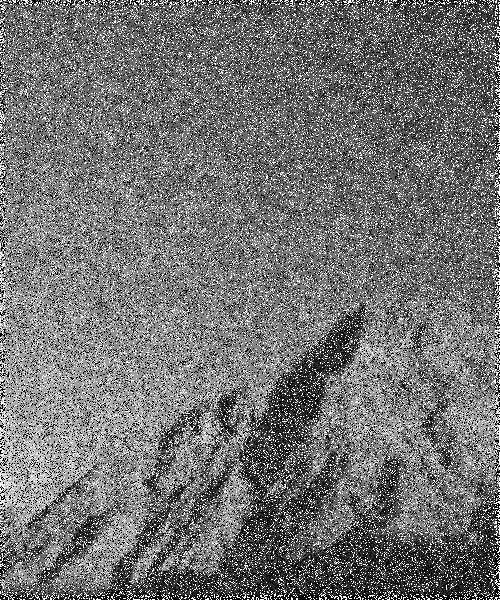
img = depoint(img)

img.save('recovery.jpg')

**问题三结果如下：**

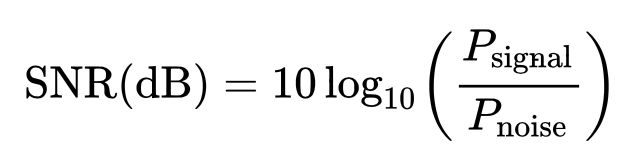
 

**噪声量为0.05，邻域为1 噪声量为0.5，邻域为2**



**噪声量为0.95，邻域为5**

**由下式可计算SNR值：**



噪声量为0.05时，SNR = 12.78dB

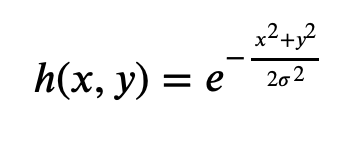
噪声量为0.5时，SNR = 0dB

噪声量为0.95时，SNR = -12.78dB

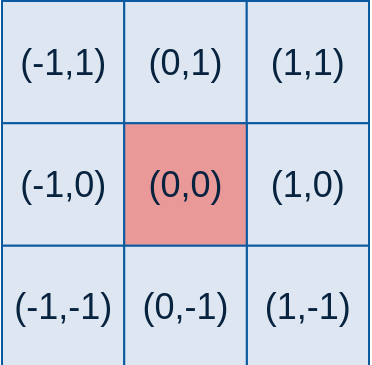
**思考：**

1. 含有高斯噪声时，可以扩大判断某像素点是否为椒盐噪声的范围，如高斯噪声的均值为5时，则像素值在[0,5]时，均可认定为椒盐噪声，然后进行椒盐噪声的滤除即可。
2. 如果只含有高斯噪声，使用高斯滤波器即可。

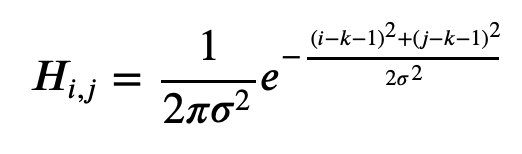
一个二维的高斯函数如下：



其中(𝑥,𝑦)为点坐标，在图像处理中可认为是整数；𝜎是标准差。要想得到一个高斯滤波器的模板，可以对高斯函数进行离散化，得到的高斯函数值作为模板的系数。例如：要产生一个3×3的高斯滤波器模板，以模板的中心位置为坐标原点进行取样。模板在各个位置的坐标，如下所示（x轴水平向右，y轴竖直向下）



这样，将各个位置的坐标带入到高斯函数中，得到的值就是模板的系数。  
对于窗口模板的大小为 (2𝑘+1)×(2𝑘+1)，模板中各个元素值的计算公式如下：



1. 心得体会

椒盐噪声是在实际场合中由于像素传感器或别的机制会导致像素值出现这种0，255的异常，学习椒盐噪声的滤除具有其实际意义。