



北京航空航天大学  
BEIHANG UNIVERSITY

# 图像去噪及直方图均衡化

## 实验报告

院（系）名称	自动化科学与电气工程学院
专业名称	模式识别与智能系统
学生姓名	张家奇
学号	15031204
任课老师	郑红

2018 年 04 月 08 日



## 1. 实验目的

- (1) 学习图像处理中常用的几种噪声：高斯噪声、椒盐噪声、脉冲噪声的特点及产生原因。并熟练掌握为图像添加以上几种噪声的方法。
- (2) 掌握常用的图片去噪方法，如：中值滤波、均值滤波。深刻理解各类去噪方法的原理、特点。并编程实现。
- (3) 深入理解直方图均衡化的原理及深层思想。自主编程实现直方图均衡化的算法。

## 2. 实验内容及算法流程

### (1) 图像去噪实验

- a) 读取图片，在照片中分别添加高斯、椒盐、脉冲噪声。
- b) 改变噪声的强度，观察不同类别、不同强度噪声对图像的影响。
- c) 分别使用均值滤波与中值滤波，对椒盐噪声进行处理。并观察不同滤波方法对图像噪声处理的差别。
- d) 对不同均值、不同方差的高斯噪声分别进行中值滤波、均值滤波，观察对噪声的去除情况。（自己感兴趣，额外添加）

### (2) 直方图均衡化实验

#### a) 直方图均衡化算法流程：

①对于有  $G$  个灰度级（一般是 256）大小为  $M \times N$  的图像，创建一个长为  $G$  的数组  $H$  并初始化为 0；

②形成图像直方图，扫描每个像素，增加相应的  $H$  成员，当像素  $p$  具有亮度  $g_p$  时，做  $H[g_p] = H[g_p] + 1$ ；

③形成累计的直方图  $H_c$ ：

$$H_c[0] = H[0],$$

$$H_c[p] = H_c[p-1] + H[p], p=1,2,\dots,G-1$$

④构造查找表，置  $T[p] = \text{round}(\frac{G-1}{NM} H_c[p])$ ；

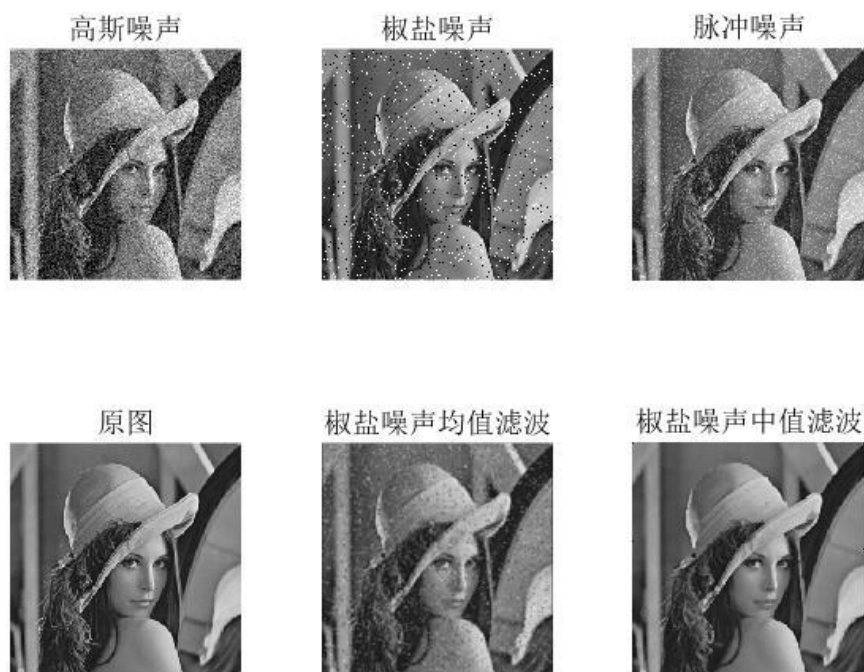
⑤重新扫描图像，写一个具有灰度级  $g_p$  的输出图像，设置  $g_p = T[g_p]$ 。

- b) 分别对彩色图片的 RGB 层进行直方图均衡化，再将均衡化后的灰度图组合成 RGB 图，观察直方图均衡化的效果。（自己感兴趣，额外添加）

### 3. 实验过程及结果分析

#### (1) 图像去噪实验

分别对图像添加高斯噪声、椒盐噪声、脉冲噪声，效果如下图第一行所示。可以看出，高斯噪声噪点分布更为平滑、密集且均匀，强度分布也更有规律，且几乎每个像素都有噪声；椒盐噪声噪点分布均匀，但较为稀疏且不是每个点都有噪声，只有少量像素点有噪声且强度基本固定；脉冲噪声则随机出现，强度固定，由参数所设定。



图一 添加不同噪声后的图像和滤波后的图像

图一第二排是图像滤波之后的结果。通过观察可以发现：

1. 均值滤波在消除噪声的同时使得图像边缘更加平滑，使图像轮廓不明显。
2. 中值滤波对于消除随机性的脉冲噪声（如：椒盐噪声、脉冲噪声等）效果明显，且能够最大限度的保证边缘不被平滑。

因此可以得出结论，在带有椒盐噪声或脉冲噪声的图像中，使用中值滤波效果更好。

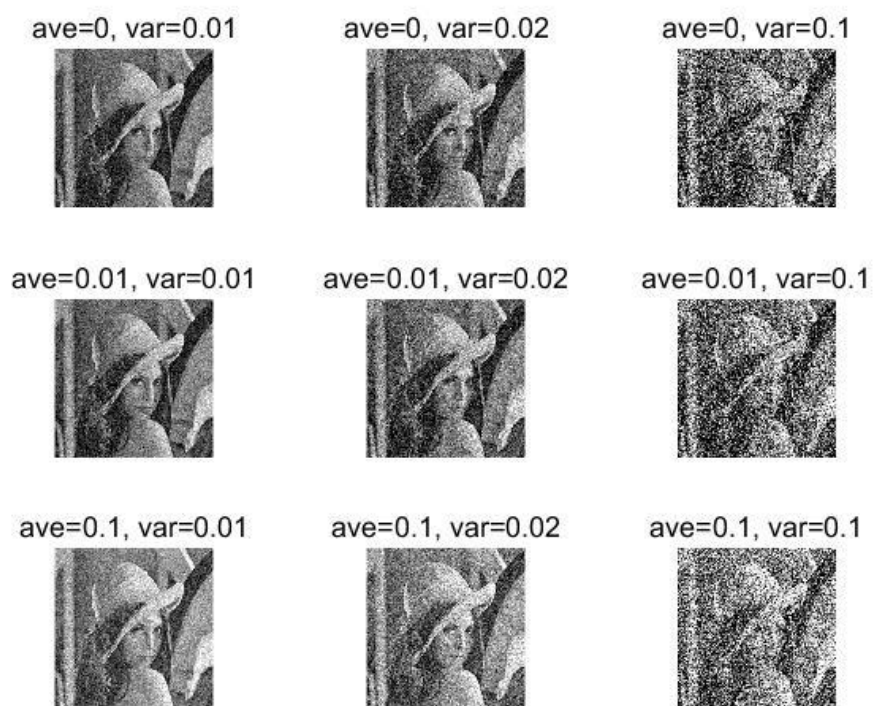
#### (2) 不同均值及方差的高斯噪声实验

对原始图片（图二）添加不同均值、不同方差的噪声。



图二 原始图片

添加不同均值、不同方差的高斯噪声后，效果如图三所示。



图三 添加不同均值、方差的高斯噪声之后

中值滤波之后效果如图四所示。



图四 中值滤波之后

均值滤波之后的效果如图五所示。



图五 均值滤波之后



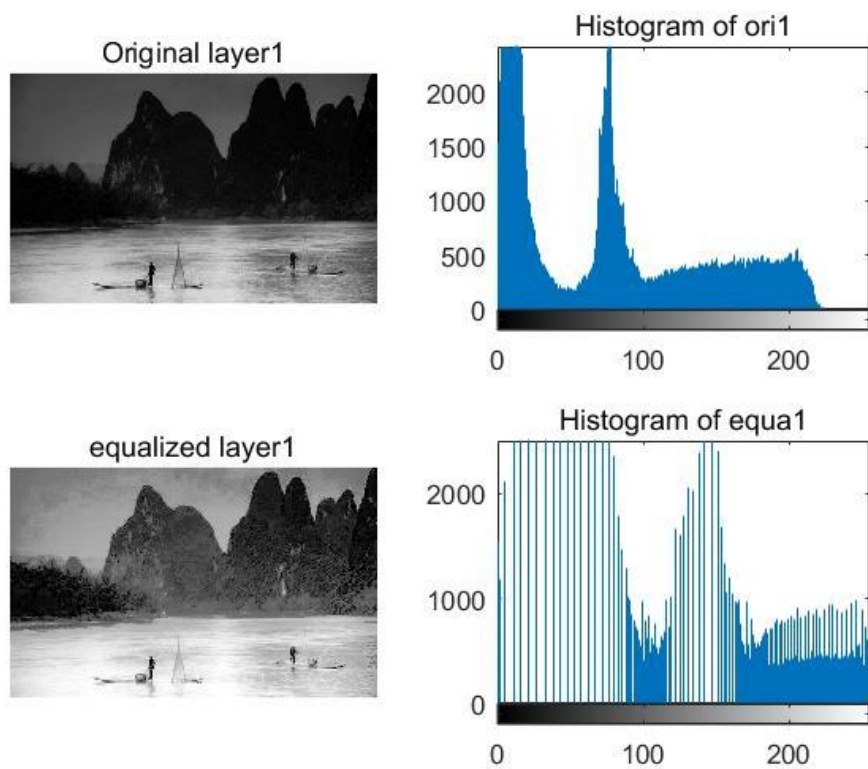
通过观察可以发现，对于高斯噪声，在噪声的均值及方差较小时，均值滤波和中值滤波的效果并没有什么明显区别。但当噪声的均值及方差加大时可以看出，均值滤波的效果要略优于中值滤波。

### (3) 直方图均衡化实验

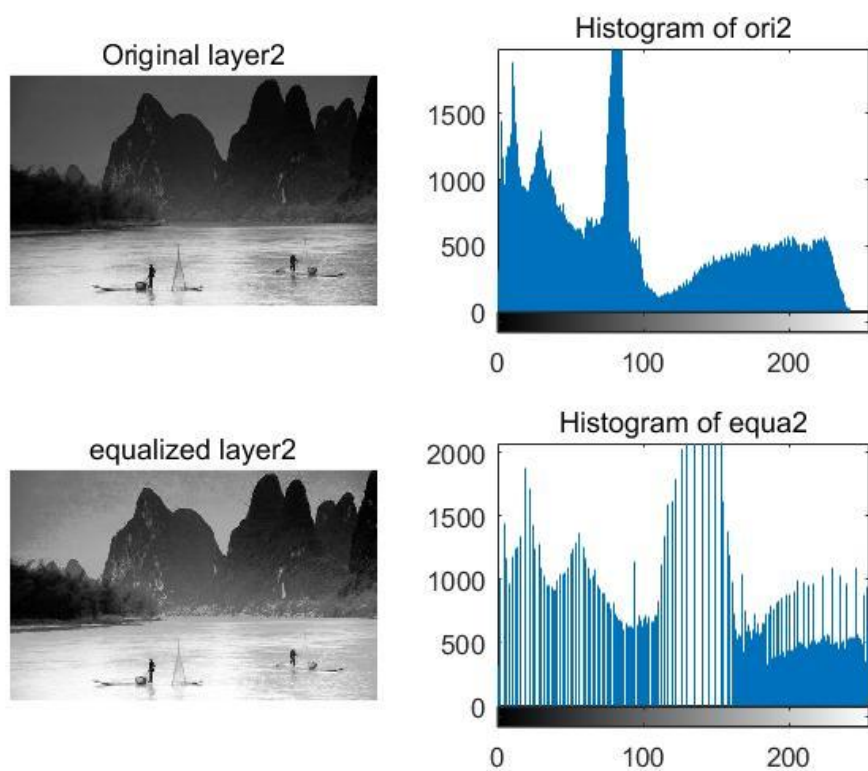
根据直方图均衡化算法的流程，写出直方图均衡化的算法实现，并封装成函数以便调用。然后对 RGB 彩色图像进行直方图均衡化。



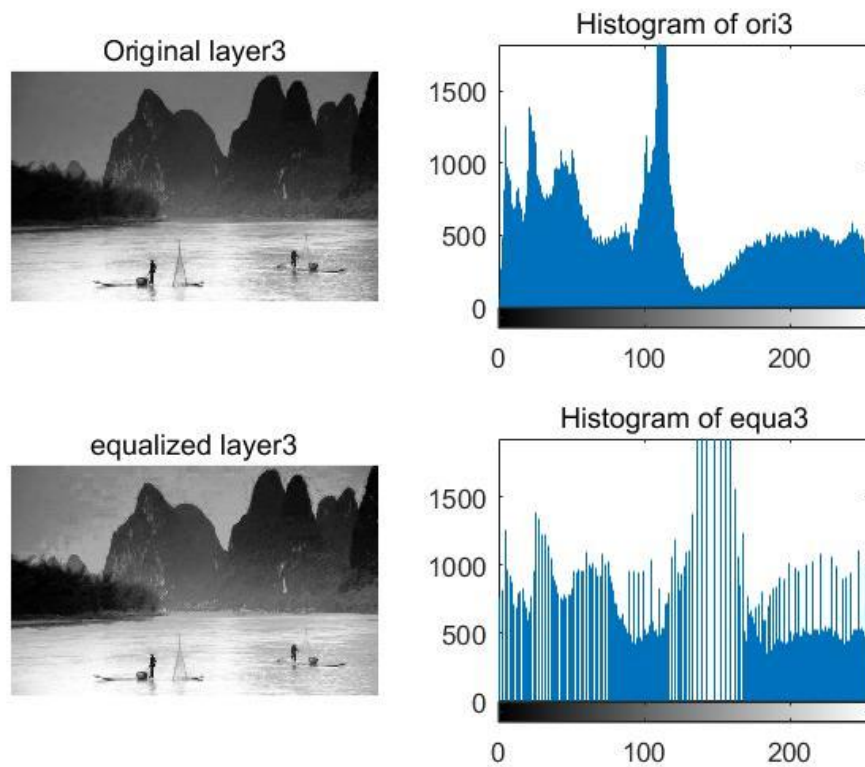
图六 原图



图七 R 层直方图均衡化



图八 G 层直方图均衡化



图九 B 层直方图均衡化



图十 RGB 直方图均衡化叠加





## 4. 总结

通过本次实验，我亲自动手实现了高斯噪声、脉冲噪声、椒盐噪声的添加；实现了中值滤波、均值滤波算法的应用；并且自己编写了直方图均衡化的算法实现。通过亲自动手实践，我对老师在课堂上讲授的各类噪声特性、滤波算法特性有了更加直观的理解。对于直方图均衡化的本质思想有了更为深刻的认识。收获颇丰。

虽然在实验的过程中遇到了许许多多的困难。比如一开始总是搞不清楚 `uint8` 和 `double` 数据类型的转换，使得程序总是出错，以及在编写直方图均衡化的算法时对矩阵的索引总是出错。但在 `debug` 的过程中，我一点点的找出并一一修正了这些错误，最终成功实现了算法。在整个实验的过程中，不仅提升了我的编程能力，更是大大提高了我积极思考解决问题的能力。在对理论知识深化理解的同时，提升了自己的综合能力。