

1 Exercises

1.1 Color Spaces

- Given an image in RGB color format, the H component of each RGB pixel is obtained using the equation

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases}$$

with

$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^{\frac{1}{2}}} \right\}$$

The saturation component is given by

$$S = 1 - \frac{3}{R + G + B} [\min(R, G, B)]$$

Finally, the intensity component is given by

$$I = \frac{1}{3}(R + G + B)$$

It is assumed that the RGB values have been normalized to the range [0,1] and that angle θ is measured with respect to the red axis of the HIS space.

The hue channel:

<p>Yellow</p> <p>Normalized:(R,G,B)->(255,255,0)->(1,1,0)</p> $H = \theta = \cos^{-1} \frac{1}{2} = 60^\circ$ <p>Normalized hue:</p> $H = \frac{1}{6}$	<p>Magenta</p> <p>Normalized:(R,G,B)->(255,0,255)->(1,0,1)</p> $H = 360 - \theta = 360 - \cos^{-1} \frac{1}{2} = 300^\circ$ <p>Normalized hue:</p> $H = \frac{5}{6}$
<p>Cyan</p> <p>Normalized:(R,G,B)->(0,255,255)->(0,1,1)</p> $H = \theta = \cos^{-1}(-1) = 180^\circ$ <p>Normalized hue:</p> $H = \frac{1}{2}$	<p>Green</p> <p>Normalized:(R,G,B)->(0,255,0)->(0,1,0)</p> $H = \theta = \cos^{-1}\left(-\frac{1}{2}\right) = 120^\circ$ <p>Normalized hue:</p> $H = \frac{1}{3}$

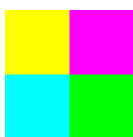
The saturation channel:

<p>Yellow</p> <p>Normalized:(R,G,B)->(255,255,0)->(1,1,0)</p> $S = 1$	<p>Magenta</p> <p>Normalized:(R,G,B)->(255,0,255)->(1,0,1)</p> $S = 1$
<p>Cyan</p> <p>Normalized:(R,G,B)->(0,255,255)->(0,1,1)</p> $S = 1$	<p>Green</p> <p>Normalized:(R,G,B)->(0,255,0)->(0,1,0)</p> $S = 1$

The intensity channel:

Yellow Normalized:(R,G,B)->(255,255,0)->(1,1,0) $I = \frac{2}{3}$	Magenta Normalized:(R,G,B)->(255,0,255)->(1,0,1) $I = \frac{2}{3}$
Cyan Normalized:(R,G,B)->(0,255,255)->(0,1,1) $I = \frac{2}{3}$	Green Normalized:(R,G,B)->(0,255,0)->(0,1,0) $I = \frac{1}{3}$

2. 由于饱和度图像的所有灰度值都一样（为 1），因此做算术均值滤波时不会改变灰度值（忽略边缘），故转化回 RGB 模型后图像表现和原来一样（忽略边缘的影响）。结果如下，和猜想一致。



3. 结果如下：



放大来观察：



可以看到边缘部分颜色发生了变化，但是没有渐变效果，这是因为在经过 RGB 到 HIS 转换、算术均值滤波和 HIS 到 RGB 的三次运算后，函数内部的取整、四舍五入等非精确运算使得精度损失，导致某些邻域的像素的颜色一样。

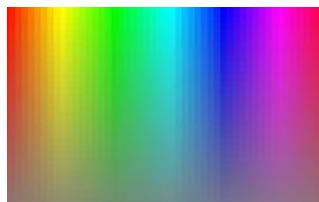
我们可以大致分析一下颜色是如何来的。对 H 通道滤波，左上、右上、左下、右下的四个色块很容易分析出颜色应当没有变化（事实上由于精度损失，颜色和原来略

有差异)，而不同颜色交接的边界例如中心点的 H 值经过滤波后为

$$H = \left(\frac{1}{6} * 4 + \frac{5}{6} * 4 + \frac{1}{2} * 4 + \frac{1}{3} * 4 \right) * \frac{1}{16} = \frac{11}{24} \approx \frac{1}{2}$$

和 Cyan 色块的 H 值相同，因此最终中心和 Cyan 对应的色块颜色一致，其他相同的色块也都可以验证是如上计算包括取近似值后得来的。

可以猜想，如果没有精度损失，滤波器经过范围内有混合颜色的区域是混合颜色的渐变，渐变效果就像下图一样：



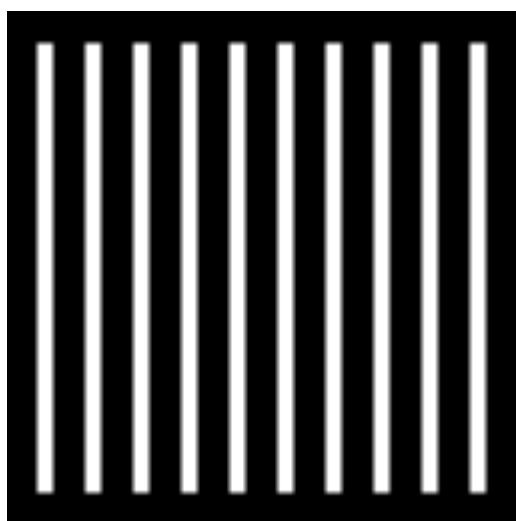
从上图也可以看到，黄蓝之间是绿色，紫红和绿色之间有两种明显不同的蓝色，黄色和紫红色之间有三种明显不同的颜色，和上图的结果契合。

2.1 Pre-requirement

2.2 Image Filtering

1. Arithmetic mean filters:

3*3:

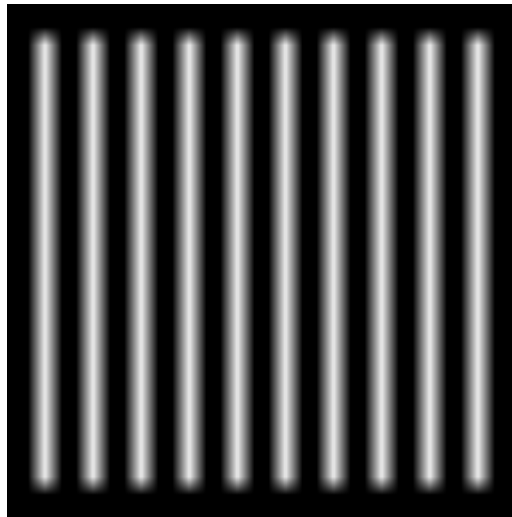


视觉上表现为黑白边界即原白条整体都变得模糊（灰色），纯白色白条缩小。

分析：由公式 $\hat{f}(x, y) = \frac{1}{mn} \sum_{(s, t) \in S_{xy}} g(s, t)$ ，滤波器覆盖范围内如果只有黑色或白色像素则得到的结果不变；如果同时有黑白像素则会根据黑白像素的比例进行加权平均。因此在 3*3 的算数均值滤波器作用下，白条宽度高度各减少 2pixels，原黑白交接线里侧的宽一个像素的白色边界的灰度值变为 $\frac{2}{3}$ ，外侧的宽一个像素的黑色边界的灰度值变为 $\frac{1}{3}$ ，四个顶角的变化见下图。下图是一个白条右上角经过滤波后的灰度值的示意图。

0	0	0	0	0
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{2}{9}$	$\frac{1}{9}$	0
$\frac{2}{3}$	$\frac{2}{3}$	$\frac{4}{9}$	$\frac{2}{9}$	0
1	1	$\frac{2}{3}$	$\frac{1}{3}$	0
1	1	$\frac{2}{3}$	$\frac{1}{3}$	0
1	1	$\frac{2}{3}$	$\frac{1}{3}$	0

9*9:

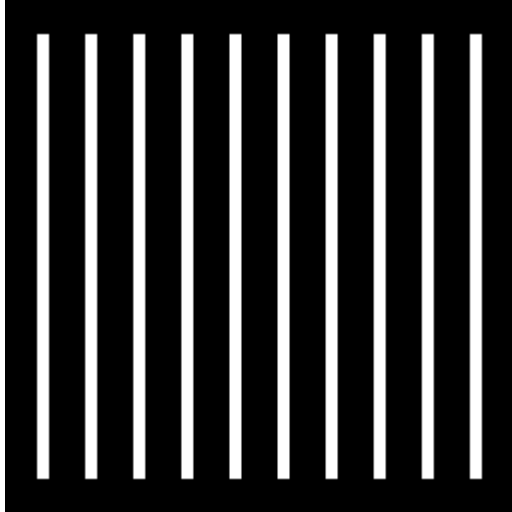


视觉上表现为黑白边界即原白条整体都变得更加模糊（灰色）了，纯白色已经没有了。

分析：由公式 $\hat{f}(x, y) = \frac{1}{mn} \sum_{(s, t) \in S_{xy}} g(s, t)$ ，滤波器覆盖范围内如果只有黑色或白色像素则得到的结果不变；如果同时有黑白像素则会根据黑白像素的比例进行加权平均。又因为 9*9 的滤波器宽度大于白条宽度（8pixels），因此白条的每一个像素都会经过黑色的加权平均，越靠近白条的边界黑色权重越大（但总体还是白色的权重大）。而宽为 8pixels 的黑色边界也变成了不同程度的灰色。具体的计算同上面 3*3 的类似。

2. Harmonic mean filters:

3*3:



原白条宽高都变小了，颜色上肉眼看不出区别。

分析：由公式 $\hat{f}(x, y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}}$ ，滤波器范围内只要有黑色点（灰度值为 0）就会使最终得到的灰度值为 0，即黑色，否则得到白色。因此对 3*3 的调和均值滤波器而言，原黑白交接线里侧的宽一个像素的白色边界变为黑色，其他不变。

9*9:

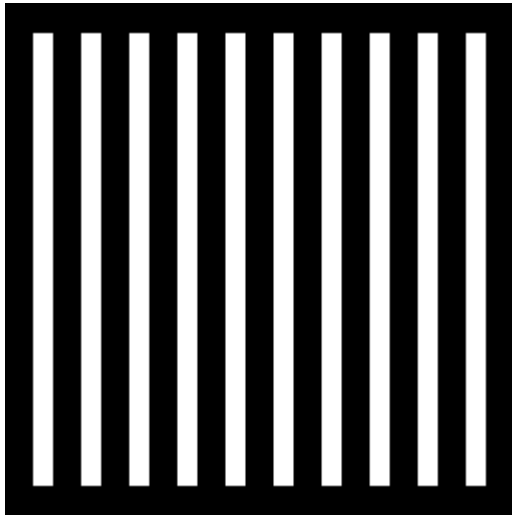


原白条已经“消失不见”。

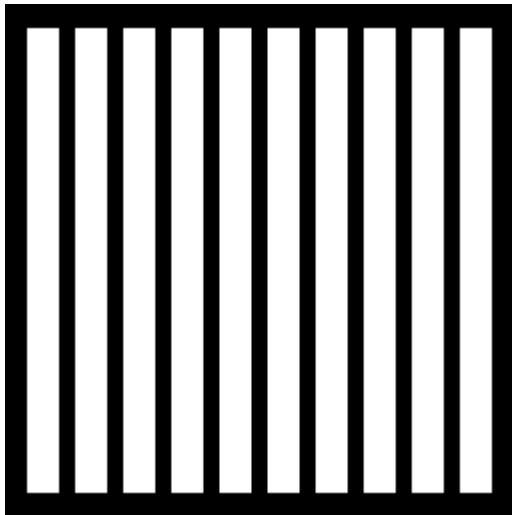
分析：由公式 $\hat{f}(x, y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}}$ ，滤波器范围内只要有黑色点（灰度值为 0）就会使最终得到的灰度值为 0，即黑色，否则得到白色。因此对 9*9 的调和均值滤波器而言，原黑白交接线里侧的宽 8 个像素的白色边界变为黑色，其他不变。由于白条的宽度只有 8 个像素，因此表现为整个白条都变成了黑色。

3. Contraharmonic mean filters:

3*3:



原白条宽高都变大了，颜色上肉眼看不出区别;
9*9:



原白条宽高都得更大了，颜色上肉眼看不出区别

2.3 Image Denoising

1. Noise generator

见 noise_generator.m 文件，用法为

```
function output_img = noise_generator(input_img, flag, x, y)
% flag = 'gaussian': Gaussian white noise
%                               with constant mean x and standard variance y
% flag = 'salt & pepper': salt-and-pepper noise
%                               with the probabilities of each of the two noise
%                               components.[Pa(x) for pepper and Pb(y) for salt]
```

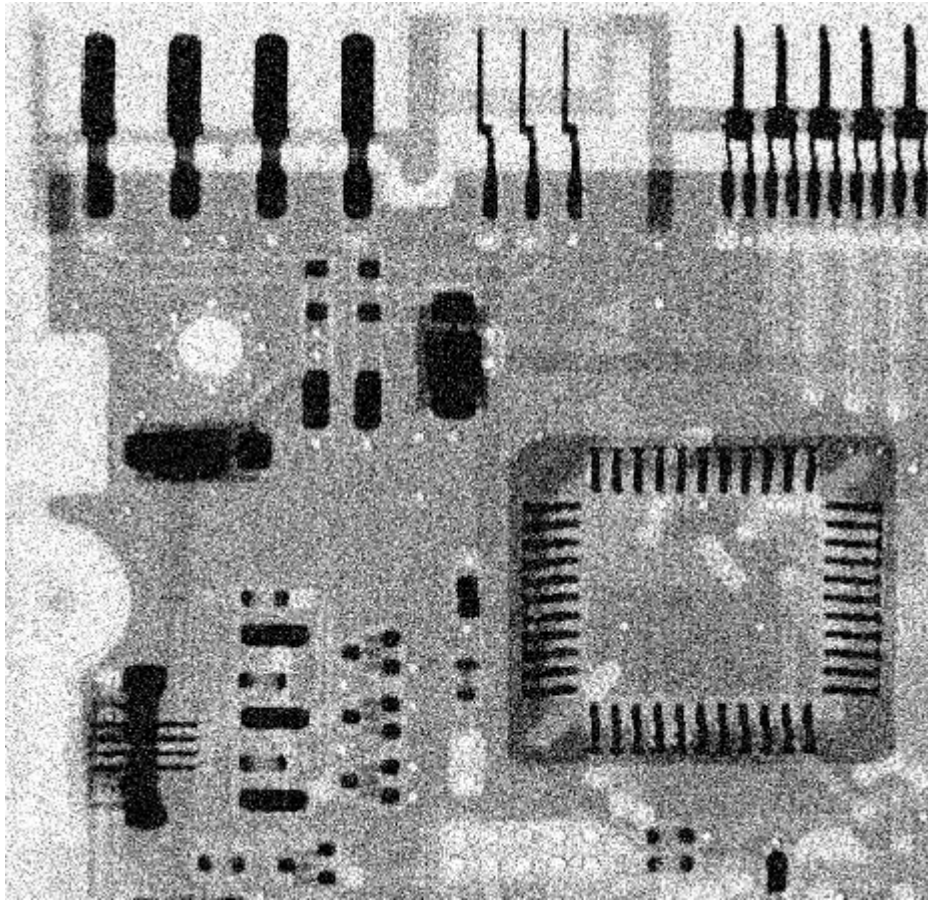
输入图像实则是彩色图像，先取一个 channel 并进行 double 转换。

高斯噪声的生成：高斯分布就是正态分布。先生成值满足标准正态分布的矩阵，标准方差和前面矩阵元素的乘积再加上均值即为所求。

椒盐噪声的生成：先生成值满足标准均匀分布的矩阵，矩阵中值小于 P_a 的位置对应的输出图像的灰度值为 0（黑色，椒噪声），值在 P_a 和 P_a+P_b 之间的位置对应的输出图像的灰度值为 1（白色，盐噪声）。

参考¹

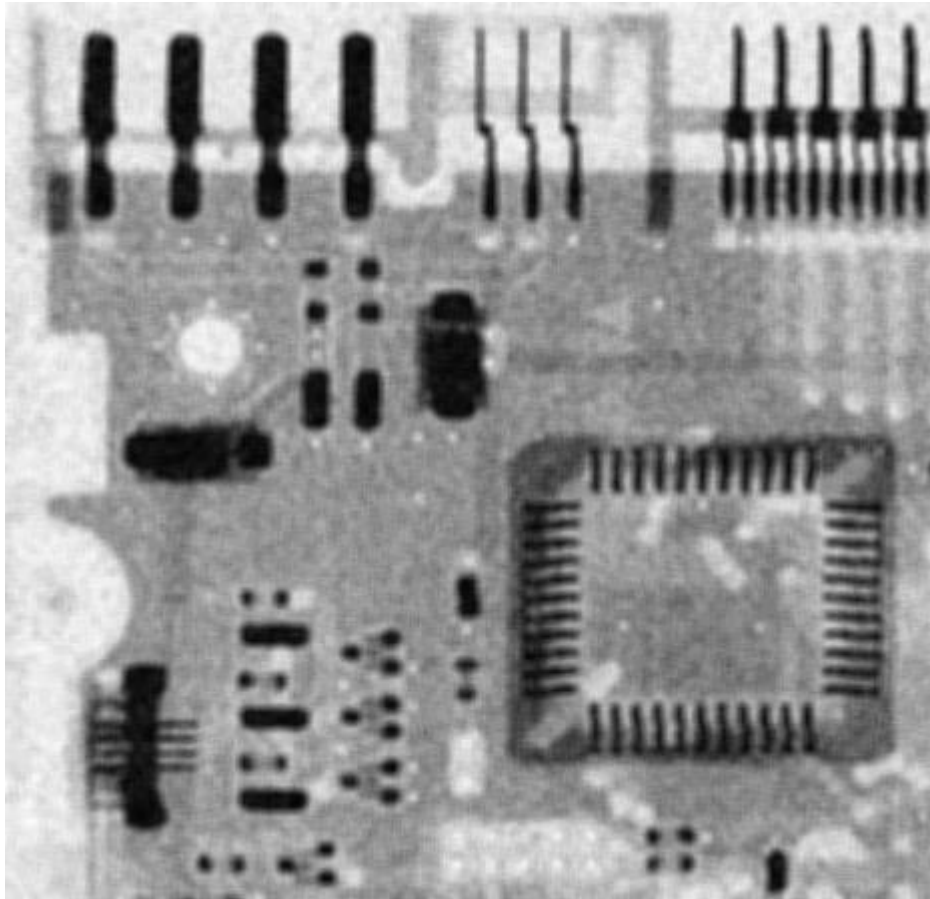
2. Gaussian noise image:



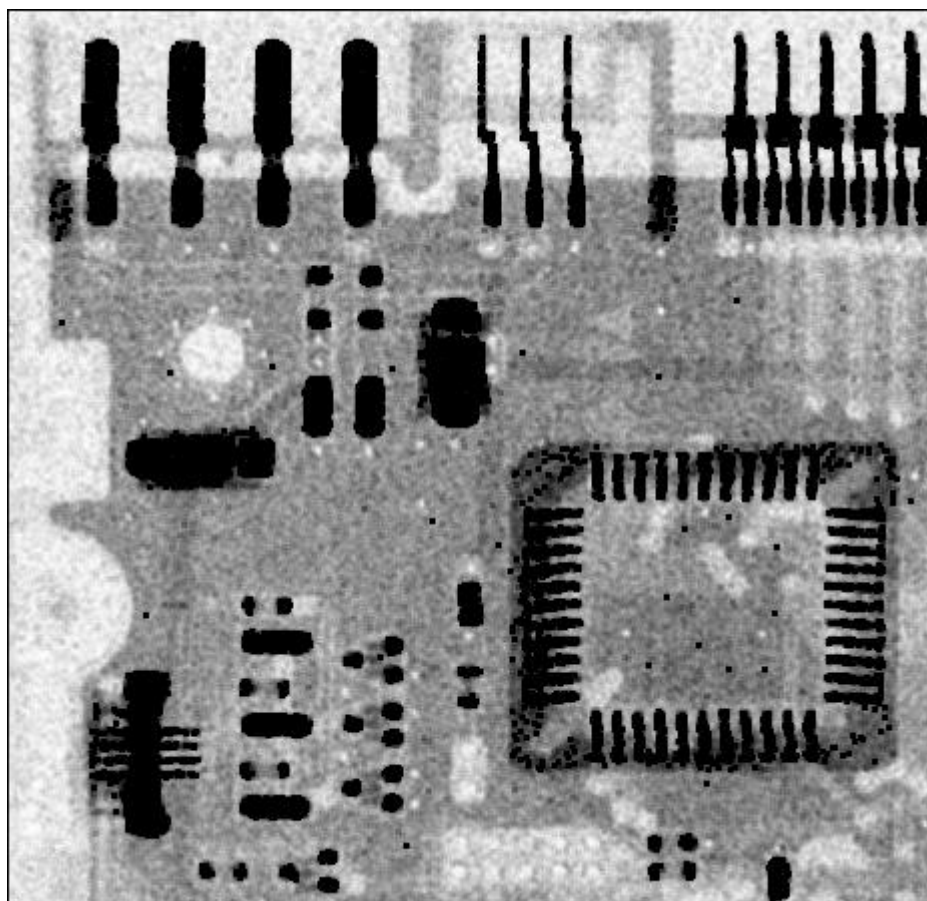
4*4 arithmetic mean filter:

1

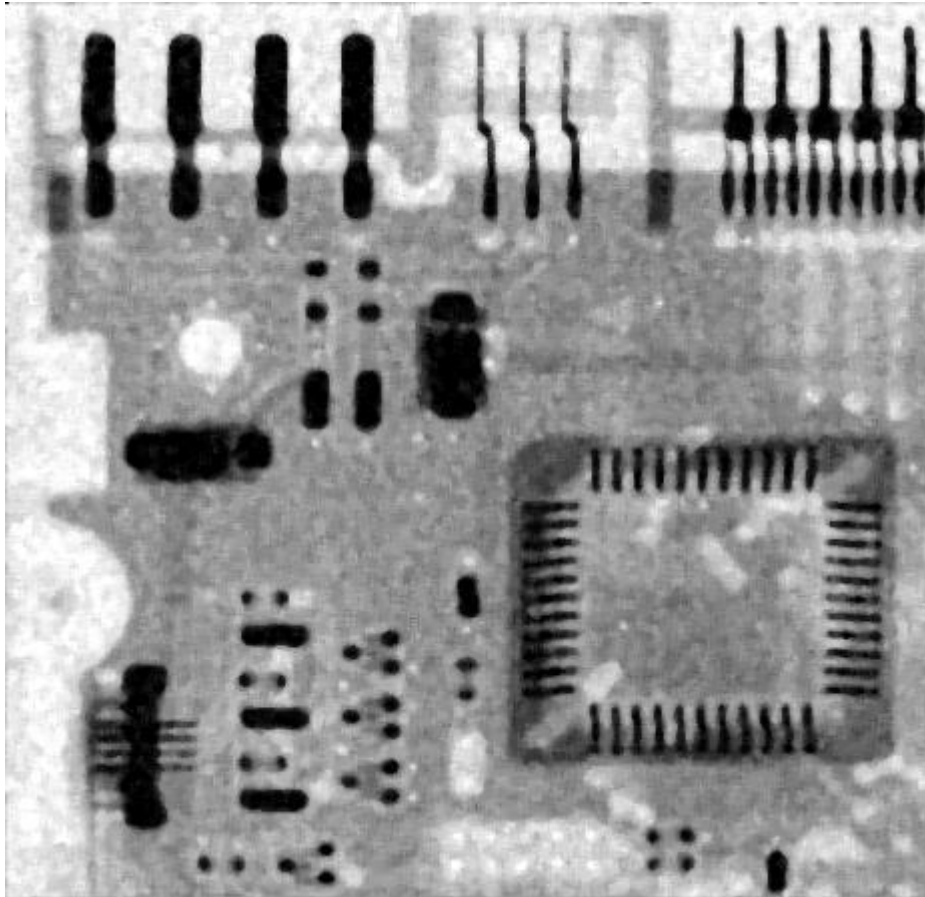
https://cn.mathworks.com/matlabcentral/fileexchange/13619-toolbox-non-local-means/content/toolbox_nlmean/toolbox/imnoise.m
<https://cn.mathworks.com/help/matlab/math/random-numbers-with-specific-mean-and-variance.html>



3*3 geometric mean filter:



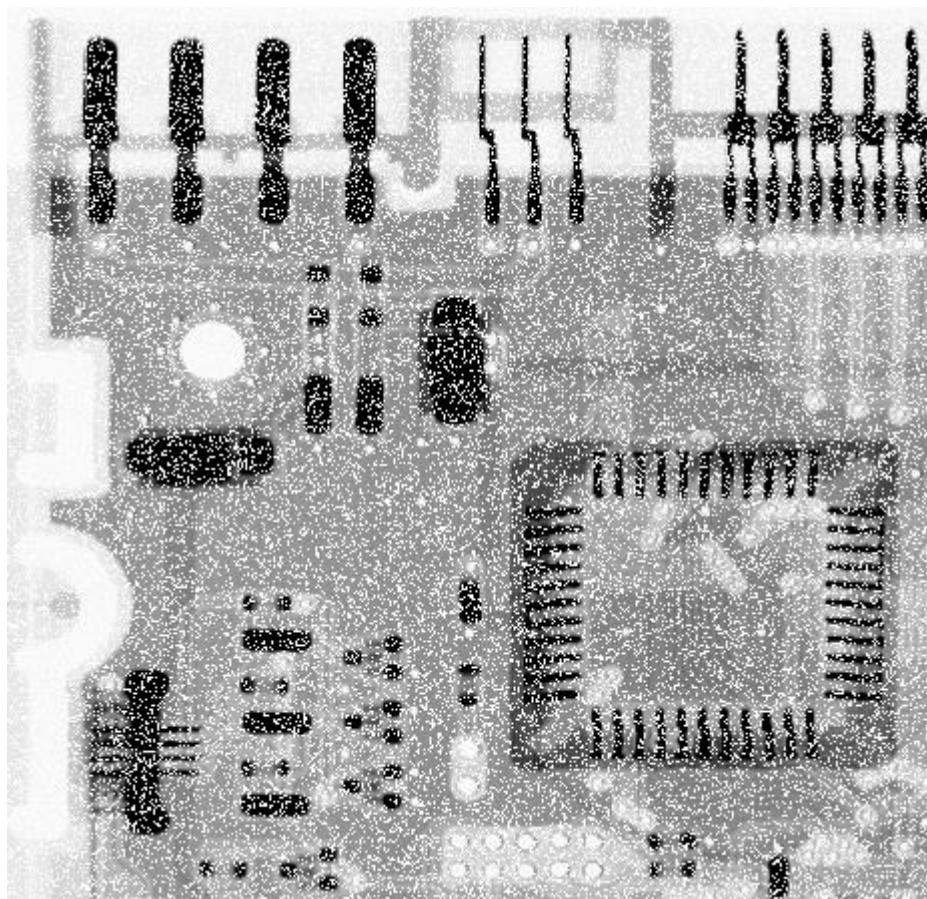
5*5 median filter:



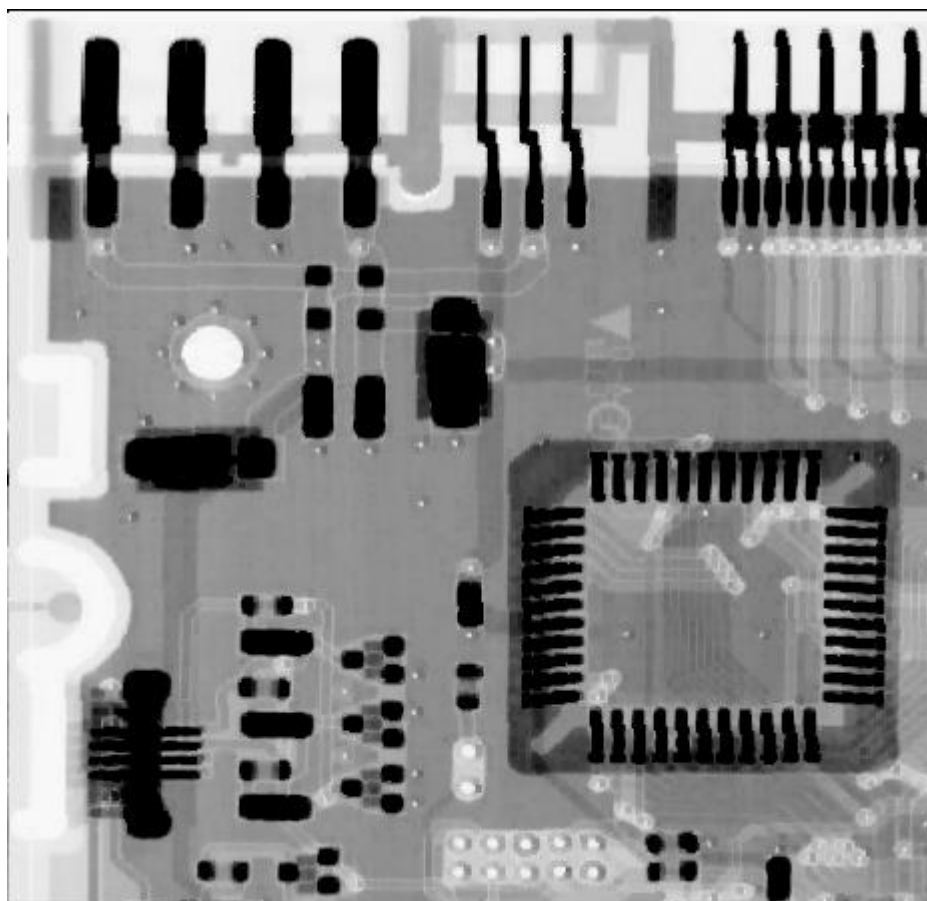
经试验和对比，去除此盐噪声效果由好到坏依次是：

应用的滤波器	效果	和原图比较	原因
5*5 中值滤波器	去除部分噪声	整体模糊，仍然有噪声没有去除	选取邻域处于中间的灰度值，有助于减少随机噪声，并且相对于算数均值滤波，减少了做平均带来的模糊。
4*4 算数均值滤波器	去除部分噪声	整体变暗、更模糊，仍然有噪声没有去除	对邻域的灰度取算数平均值，得到的结果大部分是灰色，和原图有差距，可以减弱随机噪声的影响，但不能消除，同时会造成模糊。
3*3 几何均值滤波器	不模糊，噪声减少了一些，丢失的细节最少	黑色变粗，另外出现了一些黑点，仍然有明显得噪声	将邻域的灰度相乘再做(mn)次方根运算，可以减少随机噪声，但只要领域内有一个像素为黑色(0)，相乘后该点也变为黑色，因此得到的图像黑色变粗且另外还出现了一些黑色点。

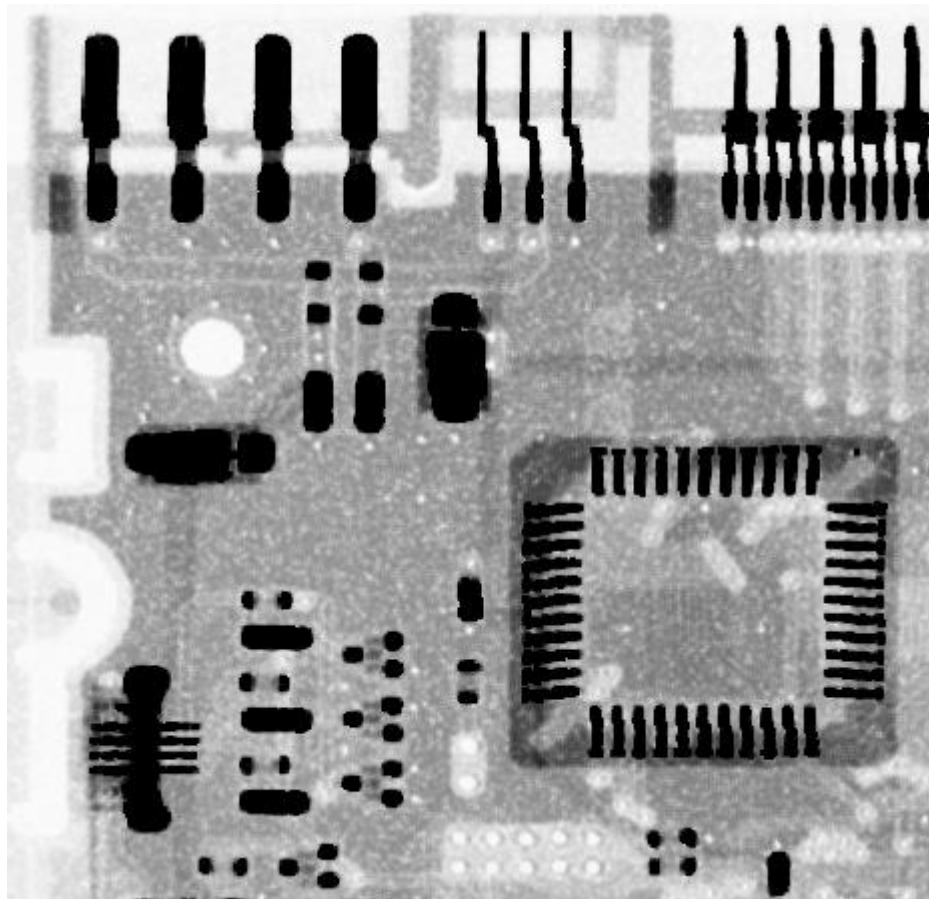
3. Salt noise image:



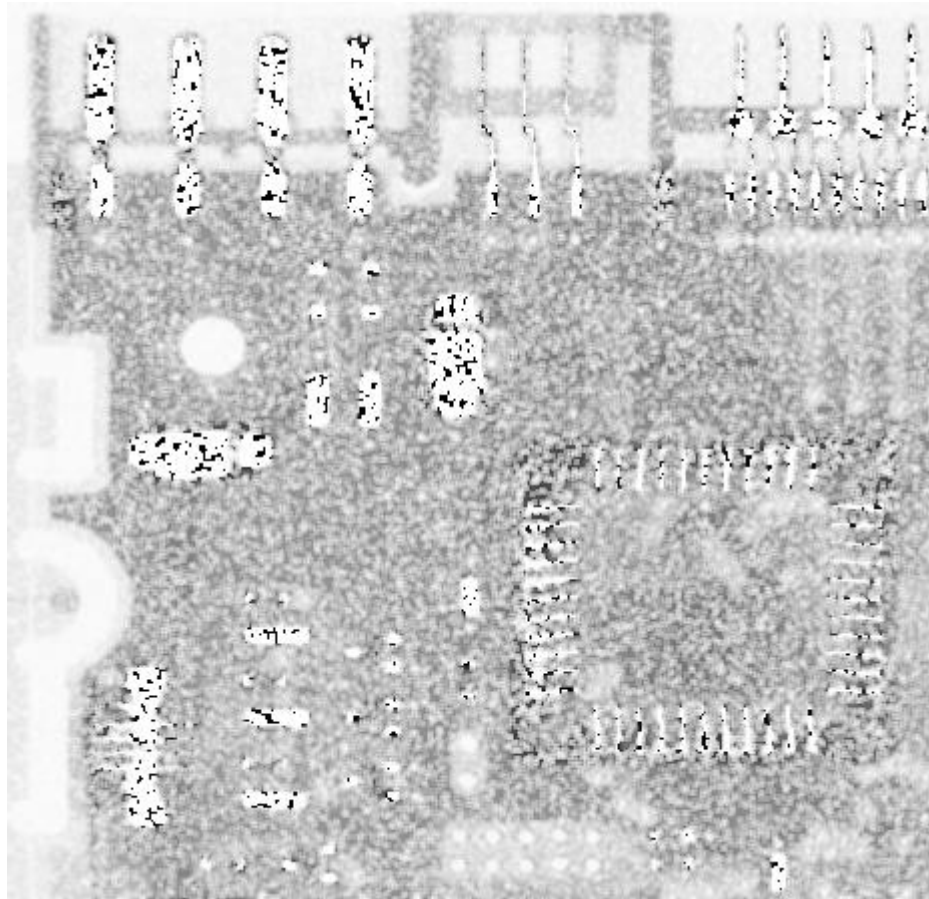
3*3 min filter:



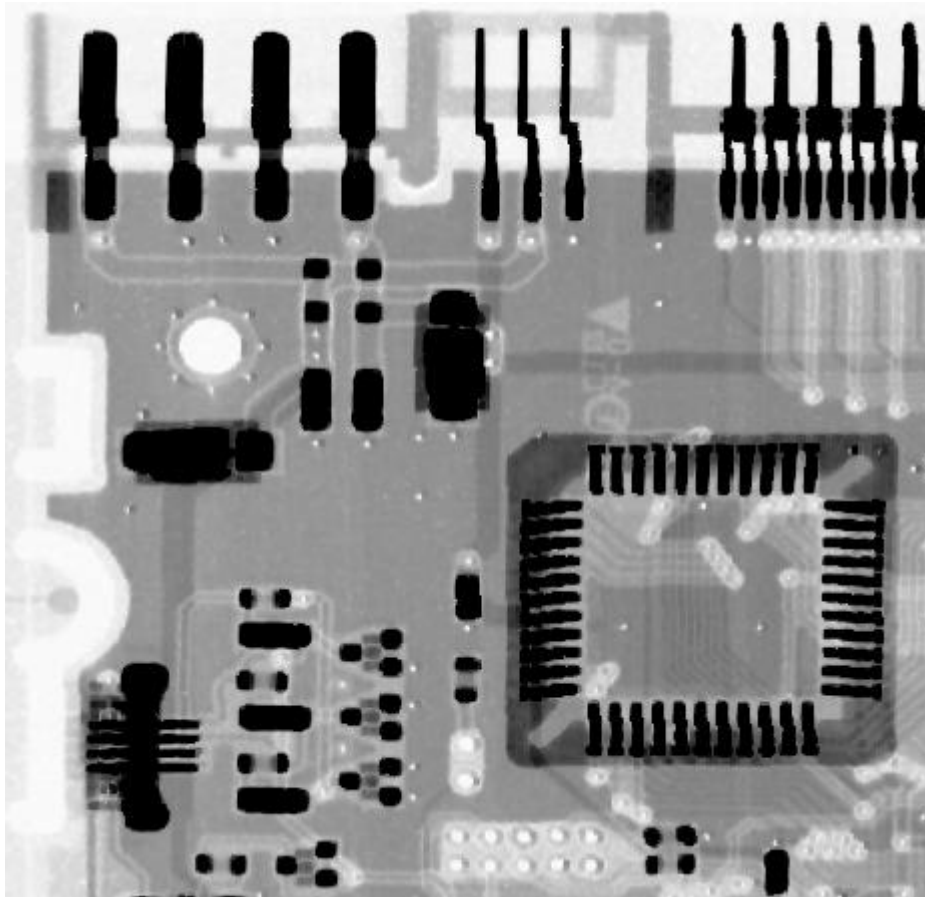
3*3 harmonic filter



3*3 contraharmonic mean filter Image, $Q=1.5$



3*3 contraharmonic mean filter Image, $Q=-5$



经试验和对比，去除此盐噪声效果由好到坏依次是：

应用的滤波器	效果	和原图比较	原因
Q=-5 的 3*3 反调和均值滤波器	盐噪声几乎全部去掉	黑色变粗，白色变细	Q<0 时，有助于去除白色的盐噪声。对于该盐噪声，Q=-5 去除效果最好。
3*3 最小滤波器	盐噪声几乎全部去掉	整体变暗，上边界变黑，黑色变粗，白色变得更细	它可以找出滤波器范围内最暗的点，而盐噪声是最亮的，因此去除效果较好；但也会因为这种选择导致整体变暗。
3*3 调和滤波器	去除大部分盐噪声	仍然有盐噪声没有去除，黑色变粗，白色变细	Q=-1 时，有助于去除白色的盐噪声，但 Q 不够小，效果没有 Q=-5 好。
Q=1.5 的 3*3 反调和均值滤波器	图像整体呈白色，几乎被毁	只剩一点轮廓	Q>0 时，会使盐噪声图像更亮而毁坏图像。

为什么错误的 Q 值会导致很差的结果：

反调和均值滤波基于下面的表达式：

$$\hat{f}(x,y) = \frac{\sum_{(s,t) \in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s,t)^Q}$$

首先，Q=0 时，

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s,t)$$

是算数均值滤波，表示将掩码范围内的灰度值以一样的权重求和。

而 $Q>0$ 时，分子比分母放大得更多，整体放大了，即变白/亮了；

$Q<0$ 时，分子比分母缩小得更多，整体缩小了，即变黑/暗了。

对每一个像素，它及它周围的像素灰度值经过这样的放大/缩小再求和等操作后，能更有效地去除个别黑/白像素的影响。

因此 $Q>0$ 时，有助于去除黑色的椒噪声，但会使盐噪声图像更亮而毁坏图像；

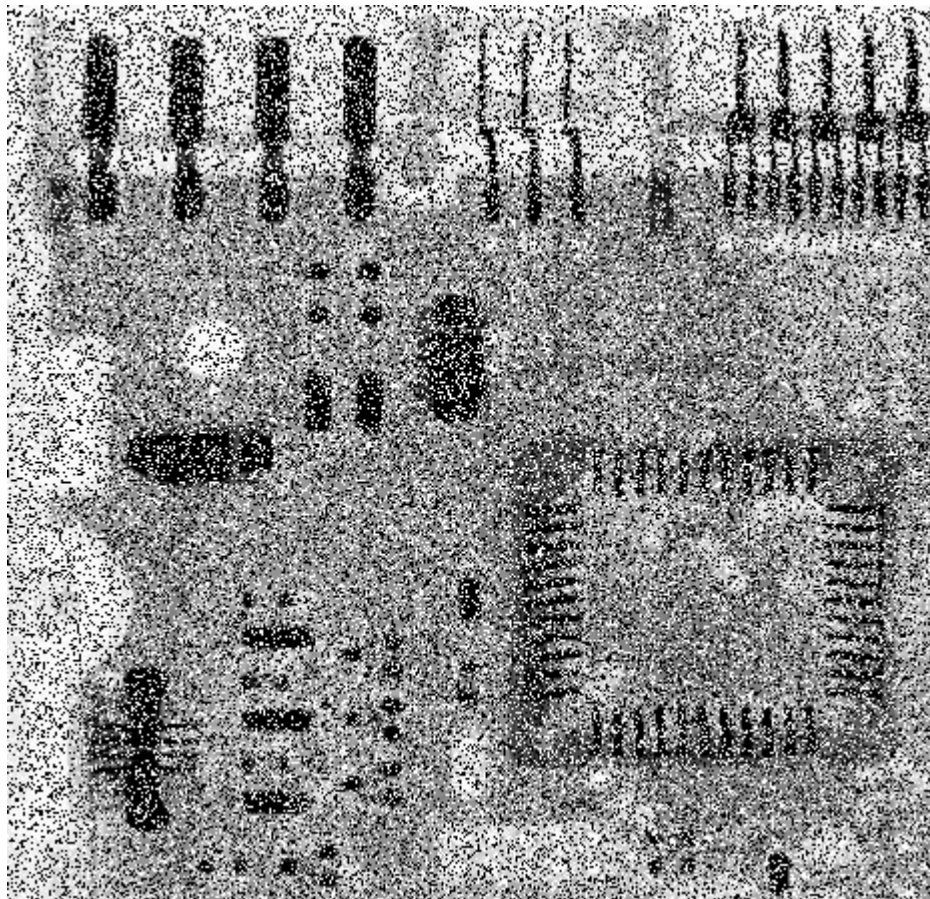
$Q<0$ 时，有助于去除白色的盐噪声，但会使椒噪声图像更暗而毁坏图像。

注意， $Q=-1$ 时，

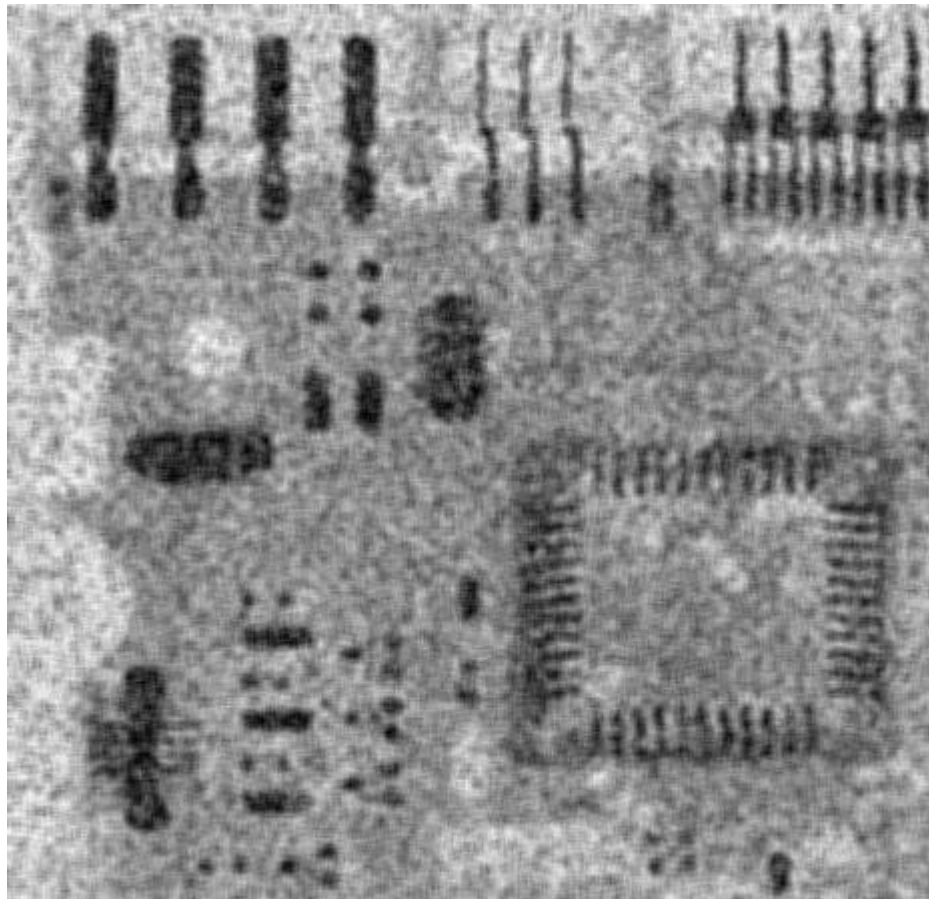
$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}}$$

即是调和均值滤波。

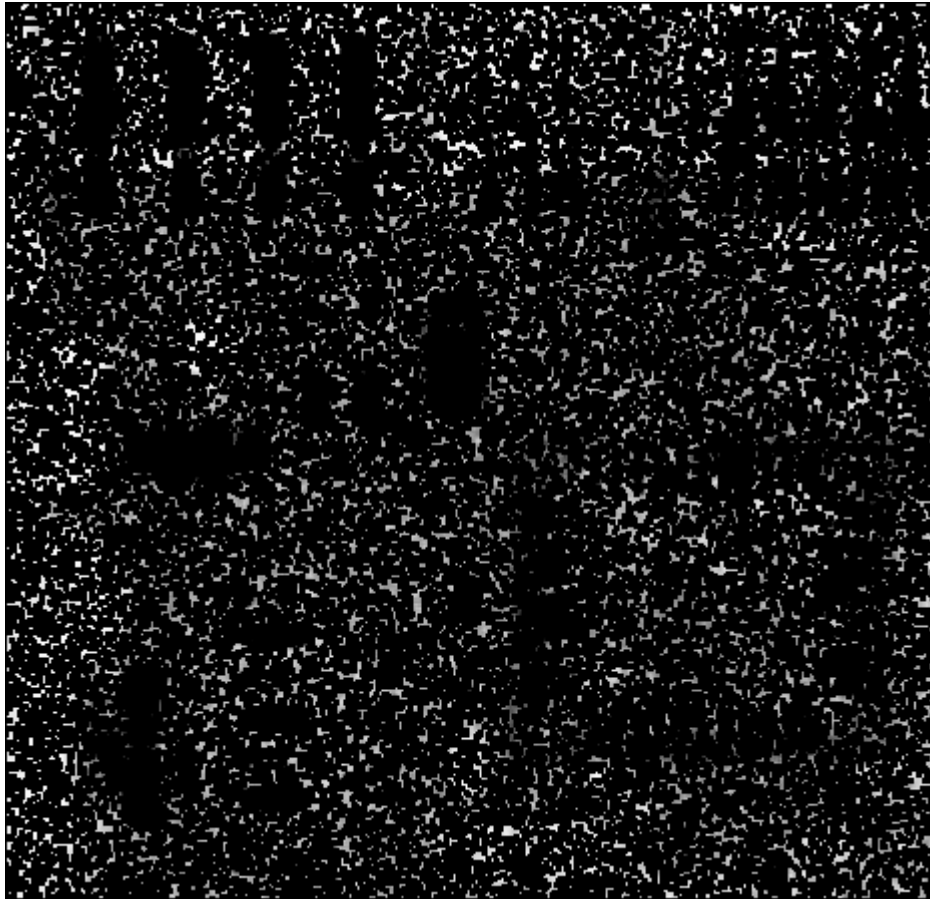
4. Salt-and-pepper noise image:



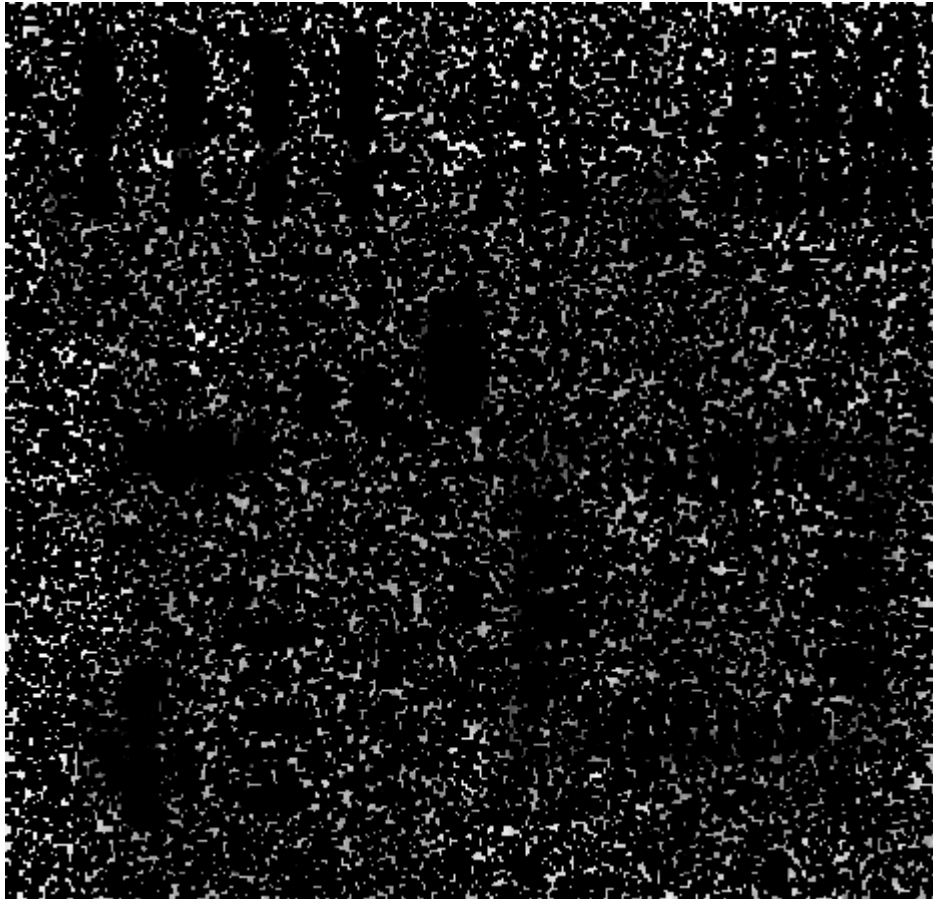
5*5 arithmetic mean filter:



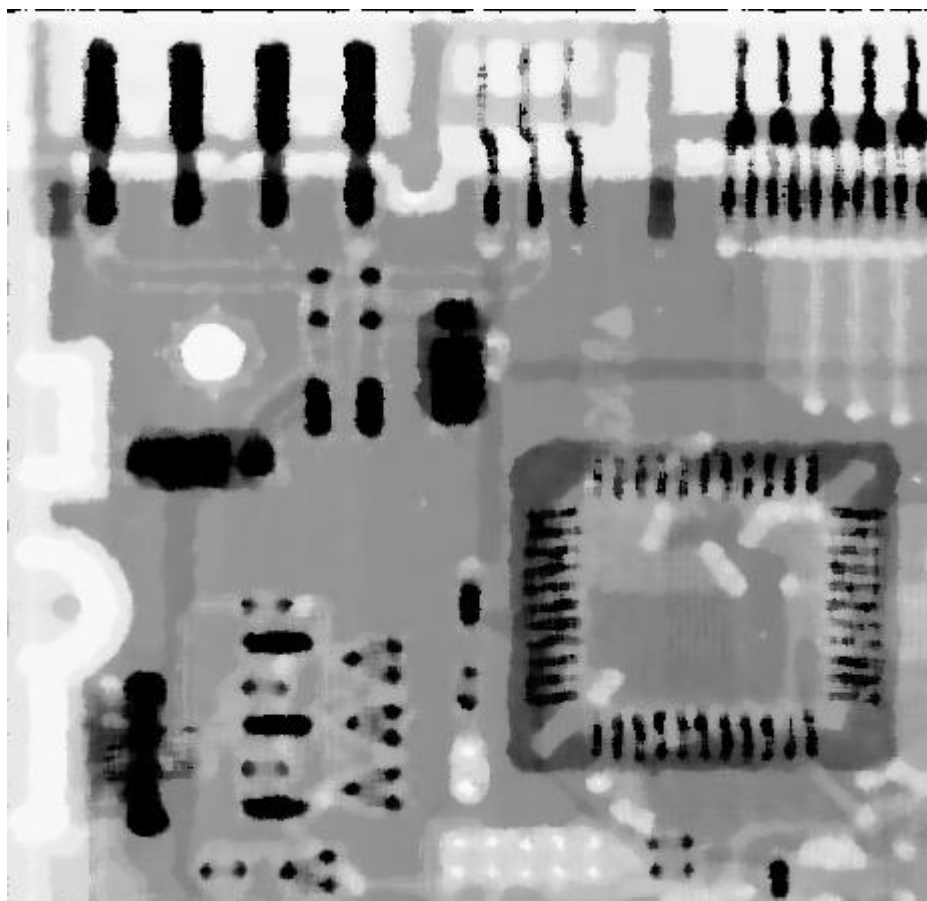
3*3 geometric mean filter:



3*3 harmonic mean filter:



7*7 median filter:



经试验和对比，去除此盐噪声效果由好到坏依次是：

应用的滤波器	效果	和原图比较	原因
7*7 中值滤波器	椒盐噪声几乎全部去掉	略模糊，损失了大量细节	由于椒盐噪声的灰度值处于两个极端，因此中值滤波器选取邻域处于中间的灰度值能很好地去除椒盐噪声，并且相对于算数均值滤波，减少了做平均带来的模糊。
5*5 算数均值滤波器	减弱了噪声，黑色轮廓保存较好	还存在较多噪声，损失了大量白色细节，整体变暗	对邻域的灰度取算数平均值，得到的结果大部分是灰色，和原图有差距，可以减弱随机噪声的影响，但不能消除，同时会造成模糊。
3*3 调和均值滤波器	效果极差，表现为黑色底色上有大量白/灰点	原来黑色块的形状还可见，白色块已经不见	将邻域的灰度求倒数再求和，最后将和求倒数。很容易验证只要领域内有一个像素为黑色（0），运算后的结果也是黑色。图像中大量的随机分布的椒噪声（黑色）附近都会变成黑色。
3*3 几何均值滤波器			将邻域的灰度相乘再做(mn)次方根运算，只要领域内有一个像素为黑色（0），相乘后该点也变为黑色。图像中大量的随机分布的椒噪声（黑色）附近都会变成黑色。很容易猜想并验证到当滤波器变大一些（9*9）时，图像全黑。

5. **算数均值滤波**（ $Q=0$ ）和**调和均值滤波**（ $Q=-1$ ）是**反调和均值滤波**的特殊情况。反调和均值滤波基于下面的表达式：

$$\hat{f}(x, y) = \frac{\sum_{(s,t) \in S_{xy}} g(s, t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s, t)^Q}$$

下面说一下上式中分母的求法（分子类似）：输出图像中的每一个像素的分母的计算都来自于输入图像相应位置的领域（滤波器范围内）像素的 Q 次方之和。

具体的如判断是否在边界内，uint8 的转换等细节已在此前的作业中谈过，不再赘述。

几何均值滤波基于下面的表达式：

$$\hat{f}(x, y) = \left[\prod_{(s,t) \in S_{xy}} g(s, t) \right]^{\frac{1}{mn}}$$

输出图像中的每一个像素都来自于对输入图像相应位置的领域（滤波器范围内）像素的乘积做 (mn) 次方根运算。可以采用一个临时变量记录每次乘积的结果，在对临时变量求根运算后赋给输出图像的方法。

最小滤波：

$$\hat{f}(x, y) = \min_{(s,t) \in S_{xy}} \{g(s, t)\}$$

输出图像中的每一个像素都等于输入图像相应位置的领域（滤波器范围内）像素的最小值。可以采用将领域像素灰度值存入一个滤波器大小矩阵，再对矩阵求最小值的方法找出最小值。

中值滤波：

$$\hat{f}(x, y) = \text{median}\{g(s, t)\}_{(s,t) \in S_{xy}}$$

输出图像中的每一个像素都等于输入图像相应位置的领域（滤波器范围内）像素的中值。可以采用将领域像素灰度值存入一个滤波器大小矩阵，再对矩阵求中值的方法找出中值。

2.4 Histogram Equalization on Color Images

1. 法一（基于颜色通道的独立均衡法）：



2. 法二（基于颜色通道平均值的均衡法）：



3. 法三（基于 HSI 颜色空间的 I 分量均衡法）:



4. 比较如下:

方法（整体视觉上由优到劣）	效果	和原图对比		运算速度	解释
基于 HSI 颜色空间的 I 分量均衡法	整体明亮、色彩饱和度更高	三幅图右上角的天空都有不均匀的色块	部分过于黑暗导致损失了细节	中等	只针对 I 分量均衡，因此不会出现色彩失真，饱和度较高
基于颜色通道平均值的均衡法	整体颜色比较均匀，色彩饱和度较低		显示了更多细节	最慢	考虑了 RGB 之间的关系，没有色彩失真；由于基于平均值做了均衡化，整体颜色较为均匀，色彩饱和度不大。
基于颜色通道的独立均衡法	整体被绿 / 蓝色笼罩		失真，变绿，车上也有不均匀色块	最快	分别对 RGB 通道进行直方图均衡化的处理是局部的，其忽略了 RGB 之间的关系，处理后的图像色彩

					可能会失真
--	--	--	--	--	-------