

哈爾濱工業大學

視聽覺信號處理實驗報告

實驗一

題	目	<u>視覺信號處理實驗一</u>
學	院	<u>計算學部</u>
專	業	<u></u>
學	號	<u></u>
學	生	<u></u>
任	課	教
師		<u>姚鴻勛</u>

哈爾濱工業大學計算機科學與技術學院

2022 年秋季

实验一 报告

一、 实验内容 (contents)

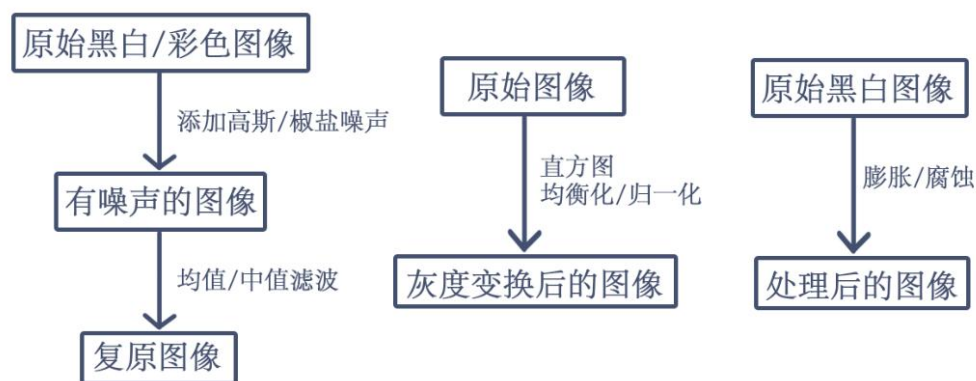
1. 实现高斯噪声和椒盐噪声的添加。
2. 实现直方图均衡化和归一化操作。
3. 对实验内容 1 得到的加噪声后的图片实现中值滤波和均值滤波处理。
4. 实现对图像的腐蚀与膨胀。

二、 实验目的 (purposes)

1. 掌握高斯噪声和椒盐噪声的添加。
2. 掌握直方图均衡化和归一化操作。
3. 掌握中值滤波和均值滤波。
4. 掌握图像的腐蚀与膨胀运算。

三、 实验设计、算法和流程(Design, algorithm and procedure)

实验设计：



实验算法：

1. 添加高斯噪声

高斯噪声 Gaussian noise，是指它的概率密度函数服从高斯分布（即正态分布）的一类噪声，通常是因为不良照明和高温引起的传感器噪声。在生成噪声过程中，对灰度值进行归一化处理后，添加一个 $X \sim N(0, \sigma^2)$ 的噪声 noise. 对每个像素叠加一个随机的噪声干扰。

```
# 添加高斯噪声

def gauss_noise(img: ndarray, **args) -> ndarray:

    noise = np.random.normal(0, 0.05, img.shape).astype(dtype=np.float32)

    img = img + noise

    img = np.clip(img, 0, 1)

    return np.uint8(img * 255)
```

2. 添加椒盐噪声

椒盐噪声 salt-and-pepper noise, 又称为脉冲噪声, 它是一种随机出现的白点 (盐噪声) 或者黑点 (椒噪声), 通常是由图像传感器, 传输信道, 解压处理等产生的黑白相间的亮暗点噪声 (椒-黑, 盐-白)。在生成噪声过程中, 控制总体噪声出现概率, 且黑色噪声和白色噪声的出现概率各为 50%。

```
# 添加椒盐噪声

def sp_noise(img: ndarray, **args) -> ndarray:

    for i in range(img.shape[0]):

        for j in range(img.shape[1]):

            if random.random() < 0.02:

                img[i][j] = 0 if random.random() < 0.5 else 1

    return np.uint8(img * 255)
```

3. 实现中值滤波

中值滤波法是一种非线性平滑技术, 它将每一像素点的灰度值设置为该点某邻域窗口内的所有像素点灰度值的中值。中值滤波对脉冲噪声有良好的滤除作用, 特别是在滤除噪声的同时, 能够保护信号的边缘, 使之不被模糊。这些优良特性是线性滤波方法所不具有的。此外, 中值滤波的算法比较简单, 也易于用硬件实现, 但是因为涉及排序, 处理速度较慢。

4. 实现均值滤波

均值滤波是典型的线性滤波算法, 在图像中应用比较多, 原理是以该像素点周围的八个像素点取平均操作, 然后替代该像素点, 也就是卷积操作。对于处理简单的线性数据 $y=ax+b$, 原理也是类似的, 取该点周围的 n 个点取平均即可, n

可以看作是一个滑窗。因此，可以取该点的前后 n 个数据的平均值，也可以取前 n 个数据的平均值，根据不同场景数据设计即可。

5. 直方图均衡化

直方图均衡化 (Histogram Equalization) 是一种增强图像对比度 (Image Contrast) 的方法，其主要思想是将一副图像的直方图分布通过累积分布函数变成近似均匀分布，从而增强图像的对比度。为了将原图像的亮度范围进行扩展，需要一个映射函数，将原图像的像素值均衡映射到新直方图中，这个映射函数有两个条件：

- ①不能打乱原有的像素值大小顺序，映射后亮、暗的大小关系不能改变；
- ② 映射后必须在原有的范围内，即像素映射函数的值域应在 0 和 255 之间；

综合以上两个条件，累积分布函数是个好的选择，因为累积分布函数是单调增函数（控制大小关系），并且值域是 0 到 1（控制越界问题），所以直方图均衡化中使用的是累积分布函数。

因为图像由一个个像素点组成，所以图像直方图均衡化是通过离散形式的累积分布函数求解的，直方图均衡化过程中，映射方法是：

$$s_k = \sum_{j=0}^k \frac{n_j}{n} \quad k=0,1,2,\dots,L-1$$

其中， s_k 指当前灰度级经过累积分布函数映射后的值， n 是图像中像素的总和， n_j 是当前灰度级的像素个数， L 是图像中的灰度级总数。

直方图均衡化的步骤：

- ①依次扫描原始灰度图像的每一个像素，计算出图像的灰度直方图；
- ②算灰度直方图的累积分布函数；
- ③根据累积分布函数和直方图均衡化原理得到输入与输出之间的映射关系。
- ④最后根据映射关系得到结果进行图像变换

6. 直方图归一化

有时直方图会偏向一边。比如说，数据集中在 0 处（左侧）的图像全体会偏暗，数据集中在 255 处（右侧）的图像会偏亮。如果直方图有所偏向，那么其动态范围（dynamic range）就会较低。为了使人能更清楚地看见图片，让直方

图归一化、平坦化是十分必要的。

这种归一化直方图的操作被称作灰度变换（Grayscale Transformation）。像素点取值范围从[c,d]转换到[a,b]的过程由下式定义。这回我们将灰度扩展到[0,255]范围：

$$x_{\text{out}} = \begin{cases} a & (\text{if } x_{\text{in}} < c) \\ \frac{b-a}{d-c} (x_{\text{in}} - c) + a & (\text{else if } c \leq x_{\text{in}} < d) \\ b & (\text{else}) \end{cases}$$

7. 实现膨胀运算

膨胀就是求局部最大值的操作。按数学方面来说，膨胀或者腐蚀操作就是将图像（或图像的一部分区域，我们称之为 A）与核（我们称之为 B）进行卷积。核可以是任何的形状和大小，它拥有一个单独定义出来的参考点，我们称其为锚点。多数情况下，核是一个小的中间带有参考点和实心正方形或者圆盘，其实，我们可以把核视为模板或者掩码。而膨胀就是求局部最大值的操作，核 B 与图形卷积，即计算核 B 覆盖的区域的像素点的最大值，并把这个最大值赋值给参考点指定的像素。这样就会使图像中的高亮区域逐渐增长。

8. 实现腐蚀运算

与膨胀相反，腐蚀就是求局部最小值的操作。腐蚀可以理解为 B 的中心（锚点）沿着 A 的内边界走了一圈。腐蚀也是对高亮部分而言，A 区域之外的部分 < A 的高亮像素，所里里面被外面取代。A 中能完全包含 B 的像素被留下来了。腐蚀可以简单理解为消除物体 A 所有边界点的过程。

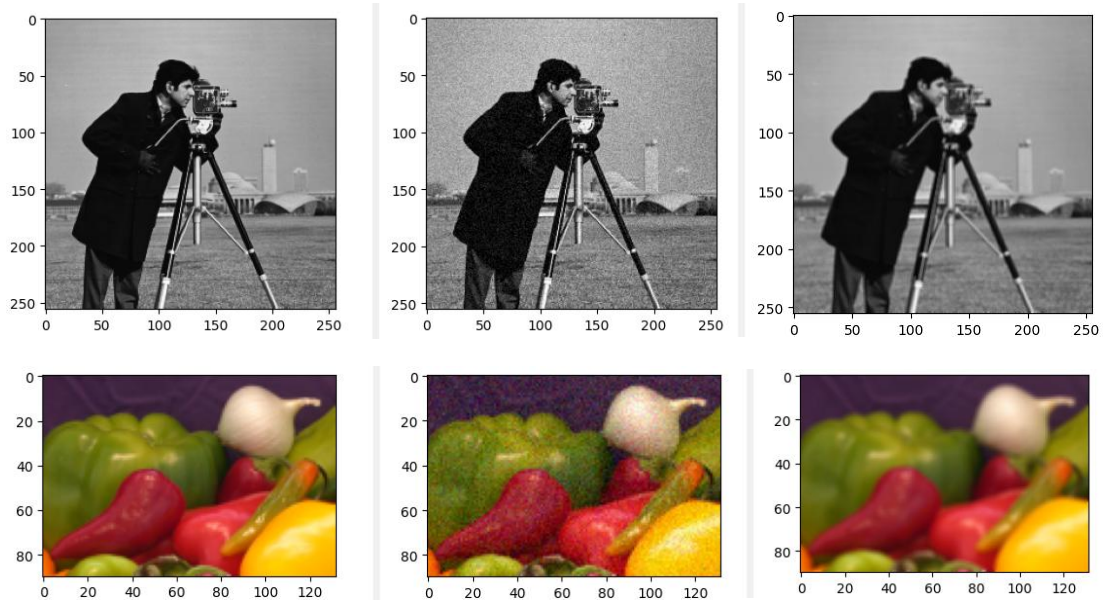
此外，先膨胀后腐蚀称为开(open)，先腐蚀后膨胀称为闭(close).开运算能够去除孤立的小点,毛刺和小桥(即连通两块区域的小点)，而总的位置和形状不变。一般来说，闭运算能够填平小湖(即小孔),弥合小裂缝，而总的位置和形状不变。

四、 实验结果(results)

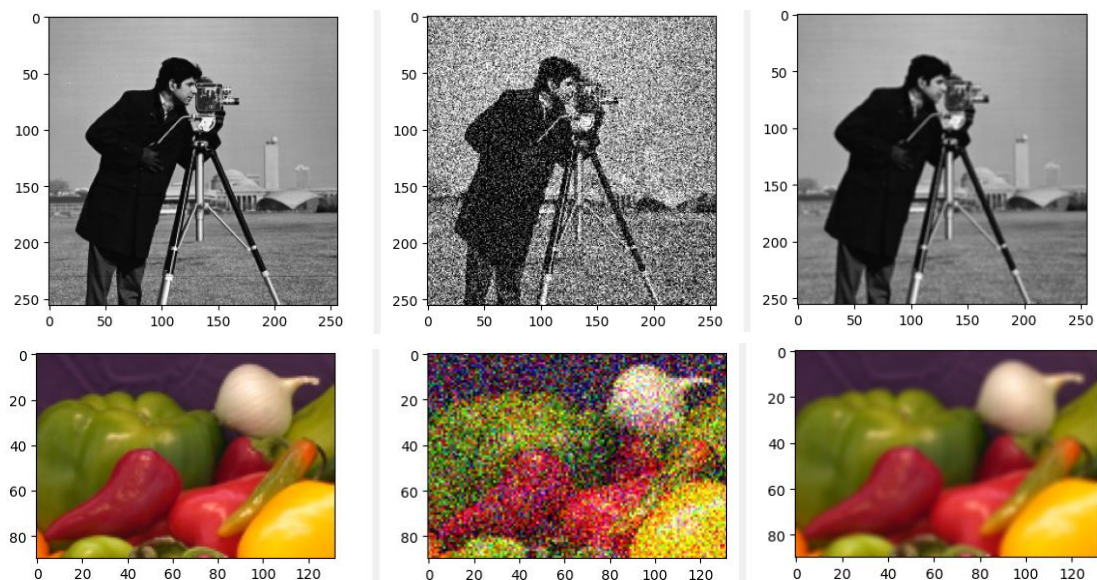
1. 添加噪声与降噪

①添加高斯噪声 + 均值滤波（黑白图像）

当添加低强度高斯噪声， $\sigma^2 = 0.05$ 时（从左到右依次为原图、添加噪声后的图像、降噪后的图像，下同）：



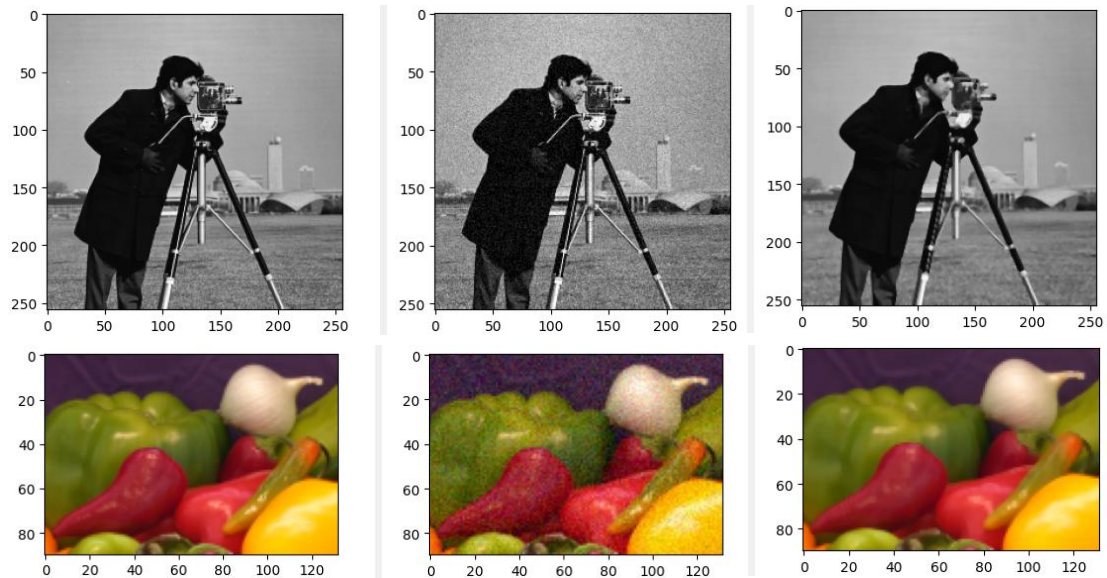
当添加高强度高斯噪声， $\sigma^2 = 0.25$ 时：



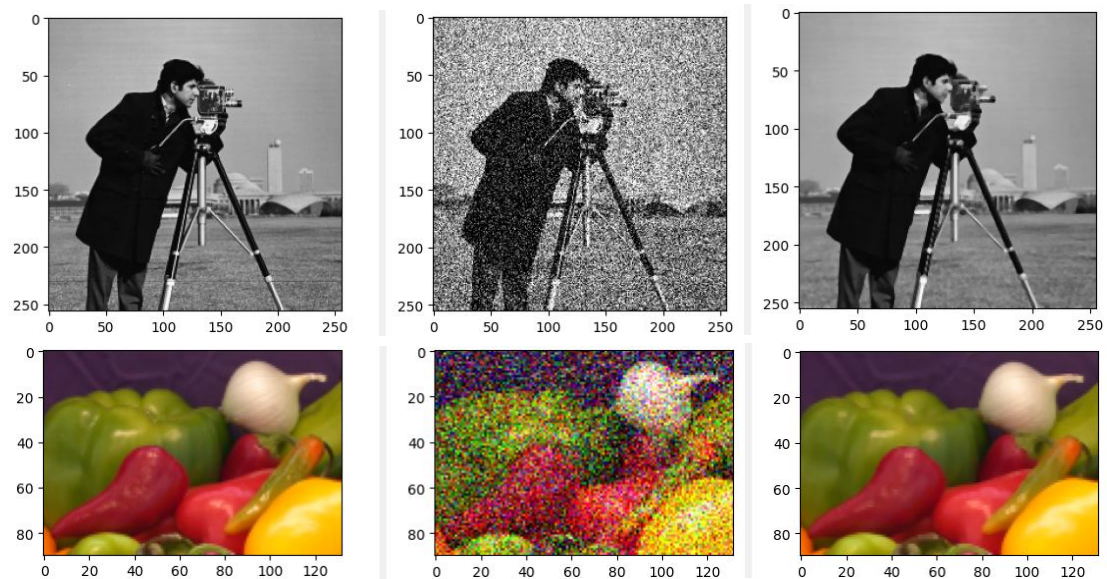
从中可以看出，不论时低强度高斯噪声还是高强度高斯噪声，均值滤波的效果都很好，能够复原图像信息，只是一些边缘轮廓处略有些模糊。

②添加高斯噪声 + 中值滤波（黑白图像）

当添加低强度高斯噪声， $\sigma^2 = 0.05$ 时：



当添加高强度高斯噪声， $\sigma^2 = 0.25$ 时：



从中可以看出，不论是低强度高斯噪声还是高强度高斯噪声，中值滤波的效果也很好，能够复原图像信息，边缘轮廓处模糊的现象比起均值滤波略有好转。但是由于中值滤波需要对滑窗内的像素进行排序，时间复杂度相比均值滤波增大，运行时间较慢，对于尺寸较大的图像尤为明显。

③添加椒盐噪声 + 均值滤波（黑白图像）

当添加低强度椒盐噪声，噪声出现率为 0.02 时：



当添加高强度椒盐噪声，噪声出现率为 0.2 时：



从中可以看出，不论是低强度椒盐噪声还是高强度椒盐噪声，均值滤波的效果都很差，不仅没有改善图像质量、去除噪点，还使得原本的噪声图像边缘轮廓更加模糊。

④添加椒盐噪声 + 中值滤波（黑白图像）

当添加低强度椒盐噪声，噪声出现率为 0.02 时：



当添加高强度椒盐噪声，噪声出现率为 0.2 时：

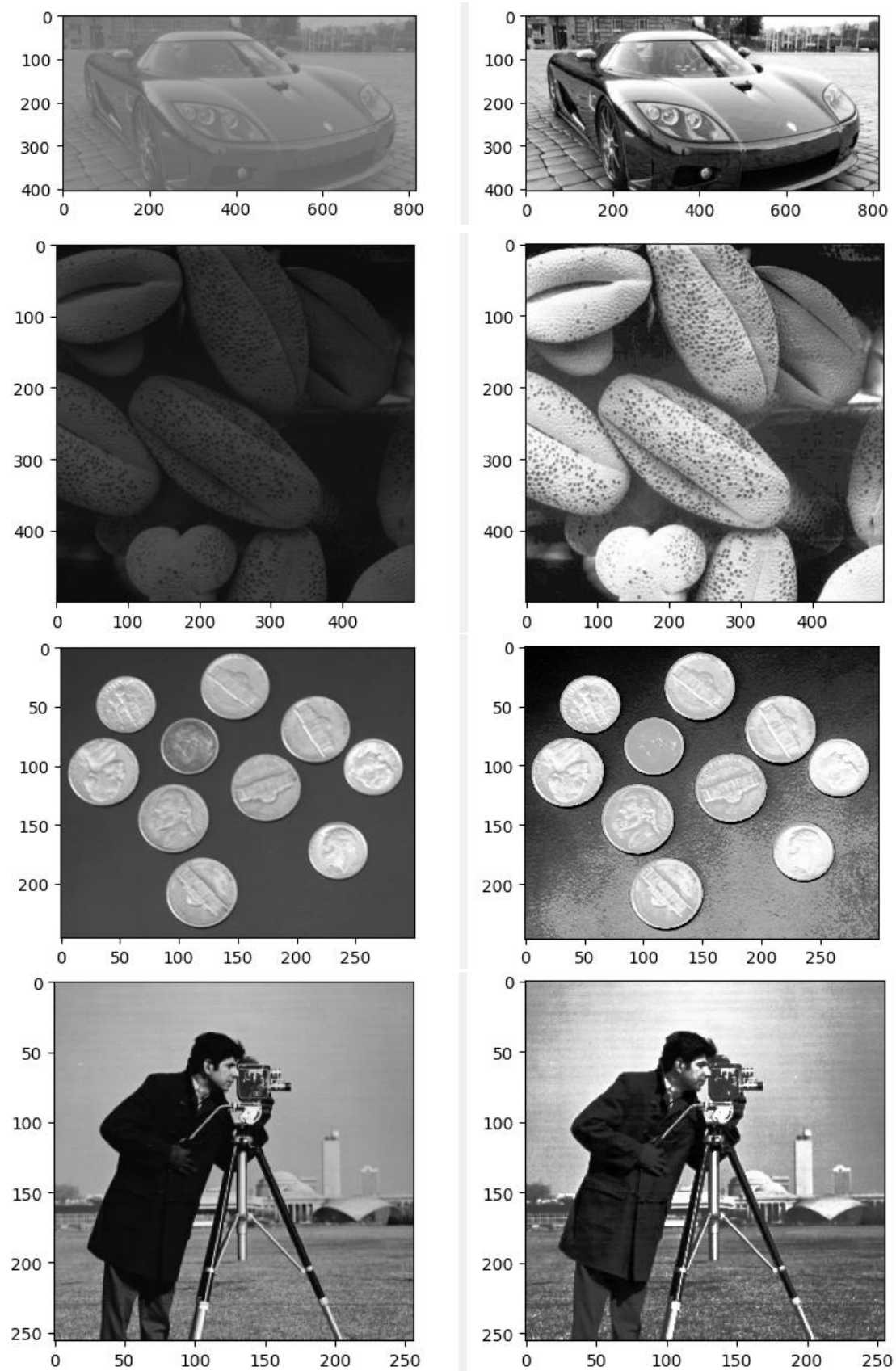


从中可以看出，当添加低强度椒盐噪声时，中值滤波的效果很好，能够有效地复原图像信息，去除噪点。当添加低强度椒盐噪声时，大部分噪点也被去除，仅保留了小部分连成一片的大噪点，很大程度上改善了图像质量。

2. 图像的灰度变换

①直方图均衡化

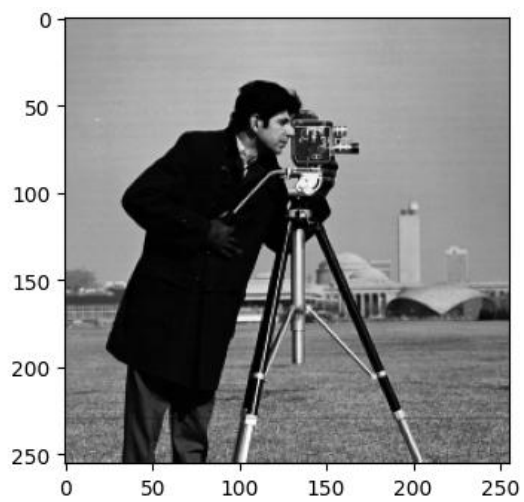
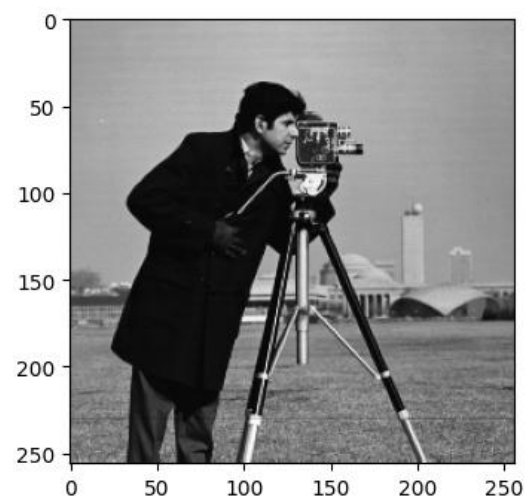
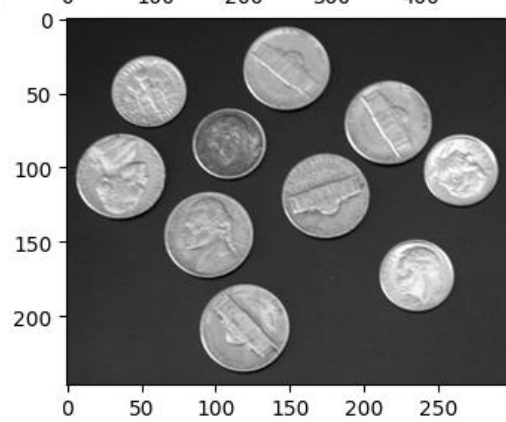
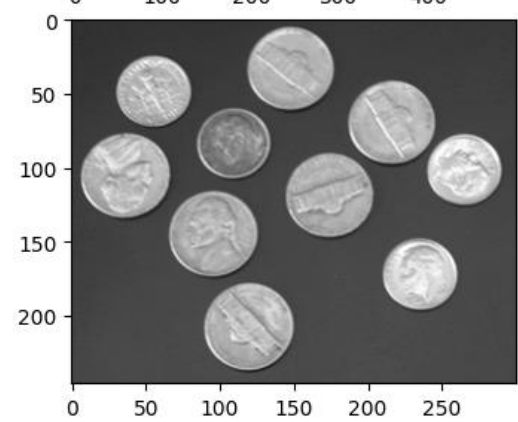
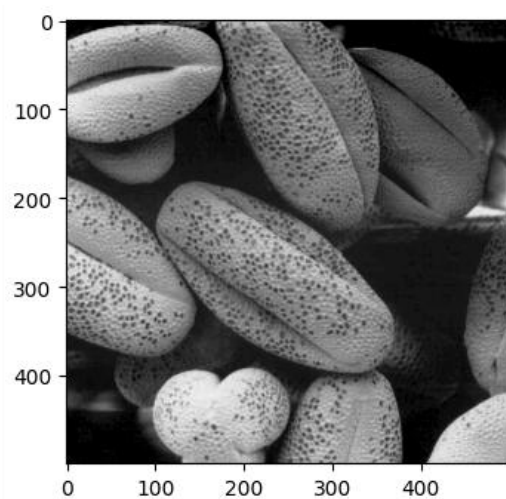
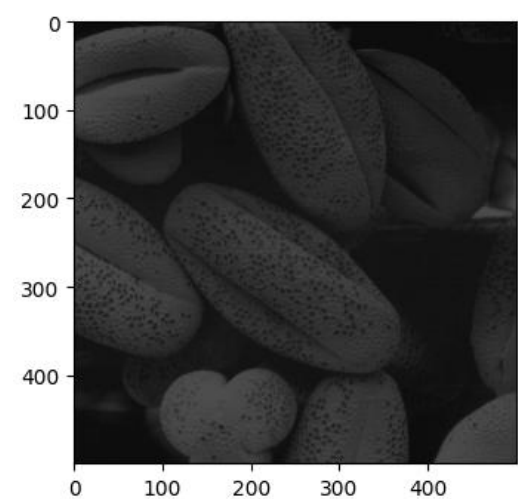
对以下几幅图像进行直方图均衡化，其中左图为原始图像，右图为直方图均衡化后的图像。



从中可以看出，通过直方图均衡化，原图的直方图趋于均匀，效果更好。表现在具体图像上，能够使得图像的明暗关系、前后景关系、材质、凹凸等等更加

清晰，使得人眼接收到更多有关画面的信息。

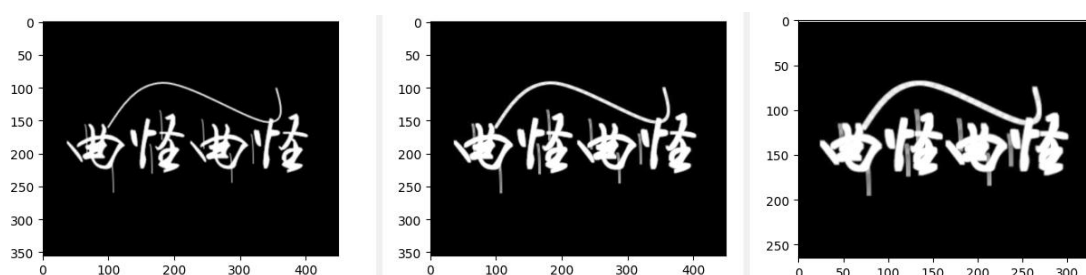
②直方图归一化



从中可以看出，通过直方图归一化，原图的直方图从整个灰度值的一部分映射到了[全黑，全白]的区间上，调节了画面整体过亮或者过暗的问题，明暗对比度增强，更易识别。相比直方图均衡化，保留了原有的灰度的变化趋势，局部灰度变化本身较少的区域，经过归一化后，对比度不如直方图均衡化后的效果强。

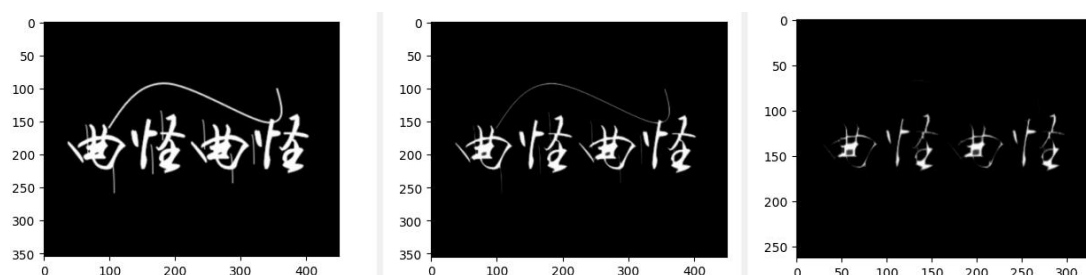
3. 图像的膨胀、腐蚀、开启、闭合

①膨胀（从左到右依次为：原图、一次膨胀、两次膨胀的图像）



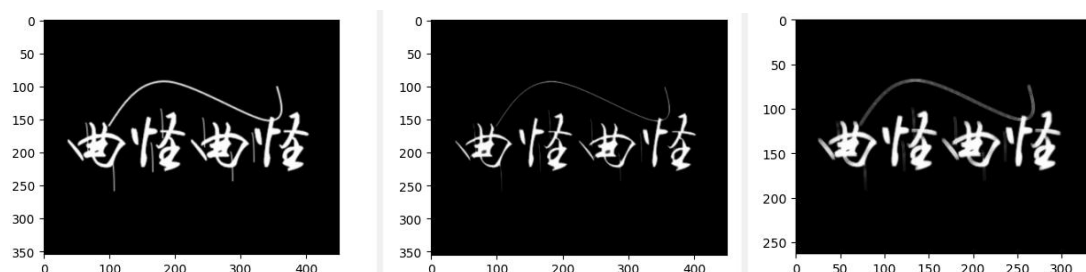
可以看出，膨胀能使得高亮（白色）区域沿着边缘逐次增长。

②腐蚀（从左到右依次为：原图、一次腐蚀、两次腐蚀的图像）



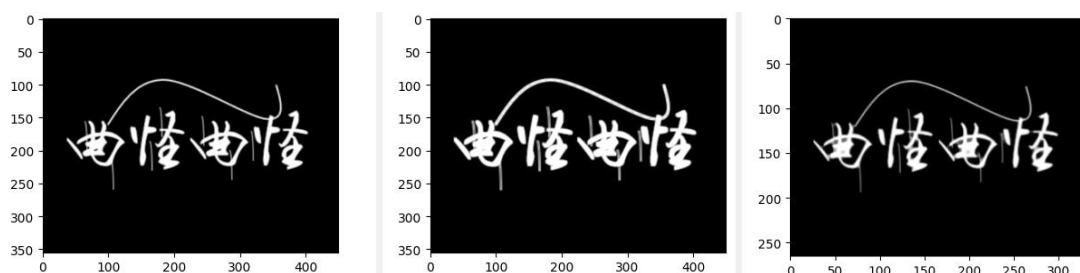
可以看出，膨胀能使得高亮（白色）区域沿着边缘逐次缩小，黑色增长。

③开启（从左到右依次为：原图、先腐蚀、后膨胀的图像）



可以看出，开启能运算能够去除孤立的毛刺和细线，而总的位置和形状不变。一般来说，闭运算能够填平小湖(即小孔),弥合小裂缝，而总的位置和形状不变

④闭合（从左到右依次为：原图、先膨胀、后腐蚀的图像）



可以看出，闭合运算能够填平小孔，弥合小裂缝，而总的位置和形状不变。

五、 结论(conclusion)

1. 中值滤波和均值滤波对高斯噪声的去噪效果都较好，但均值滤波会模糊边缘，而中值滤波处理速度速度较慢。

均值滤波不适合对椒盐噪点进行去噪，而中值滤波对椒盐噪声的去噪效果较好。

2. 直方图均衡化能将图像的灰度处理地接近均匀分布，使得显像效果更好，灰度对比强烈。

3. 直方图归一化能使得灰度分布不均匀或者分布在小区间的图像，经过处理后分布在更大的灰度区域上，使得更易辨识，但对于本就灰度范围较大的图像处理区别不大。

4. 通过膨胀、腐蚀的组合，能够达到对图像中的毛刺、细线、孤立点进行处理的目的，实现去除或连接操作，进而实现所需的降噪处理。

六、 参考文献(reference)

[1]Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing (Third Edition)

[2]左飞. 图像处理中的数学修炼