ISE128 Artificial Intelligence Programming Language Introduction Assignment

人工智能编程语言期末大作业

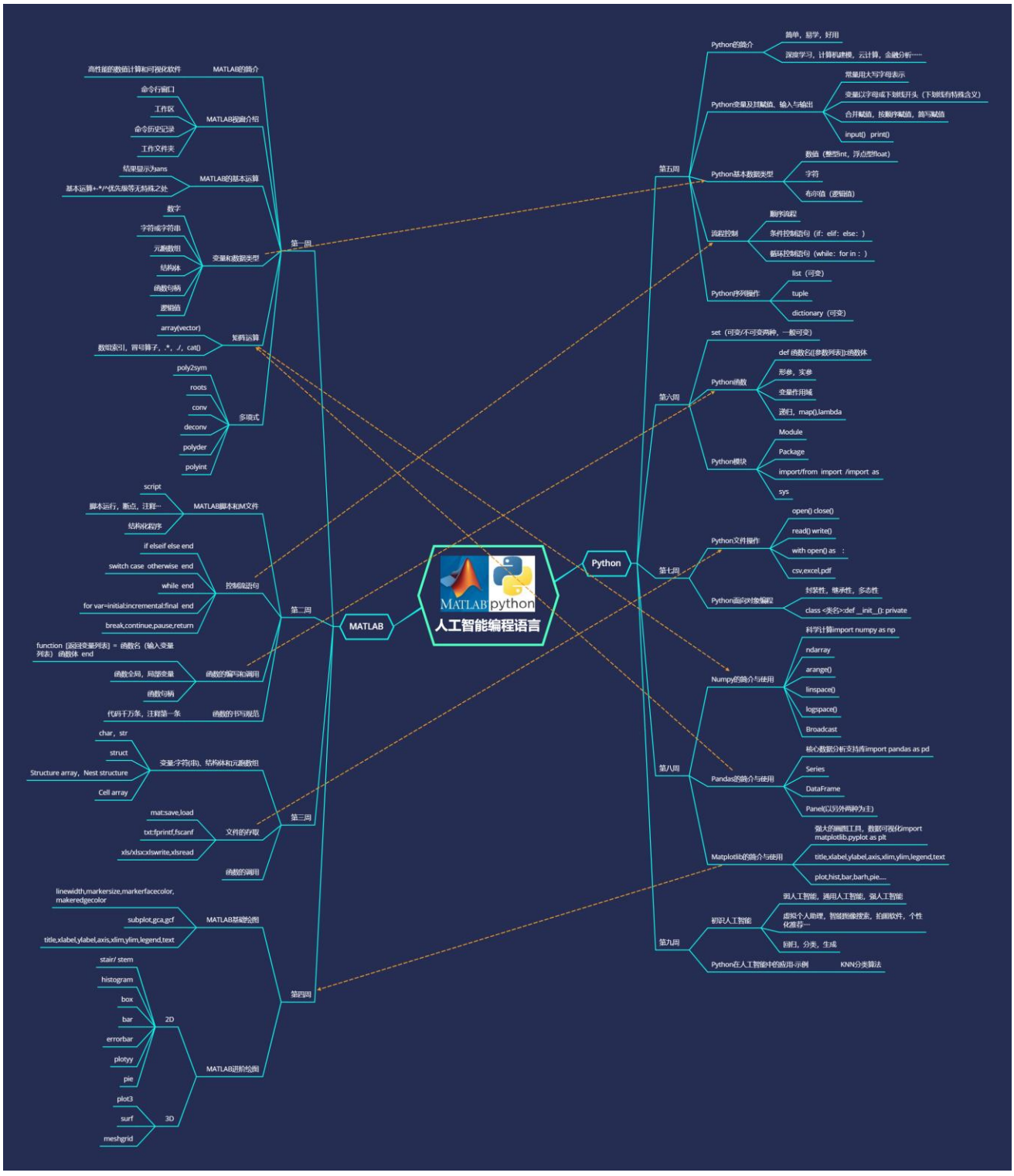
姓名：方桂安

学号：20354027

Problem 1 (10 分)

题目：请将本课程中教授的内容整理归纳，并用思维导图画出主要内容。

解答内容：



Problem 2 (15 分)

题目：请利用“对比法”，对比 C 语言、MATLAB 和 Python 语言就本课程所教授知识点上的异同。可以使用列表、举例、画图等，你认为能清晰表达异同的方式写出。

解答内容：

C语言，matlab，python的异同

	C语言	matlab	python
注释	//	%	#
注释快捷键	Ctrl + Shift + /(vs2019)	Ctrl + R/T	Ctrl + /(Pycharm) Ctrl+1(Spyder)
清除工作区内变量	free()	clear	reset
清除历史命令	无	clc	clear
索引序号		以1为起点	以0为起点
平方	math.h中的pow()函数	^	**
不等号	!=	~=	!=
与或非	&&，||，！	& | ~	and or not

条件语句	if (condition)
	;
	else
	;
		if condition
		elseif condition
		else
		end
			if condition :
			elif condition:
			else:

循环语句	for (int i=0; i<n; i++)
	while (condition)
	do { }
	while (condition)
	break 跳出整个循环
	continue 跳出该次循环
		for number = 1:n
		end
		while condition
		end
		break 跳出整个循环
		continue 跳出该次循环
			for number in range(1,n):
			while :
			break 跳出整个循环
			continue 跳出该次循环

函数定义	dataType functionName(){
	//body
	}
		function [output]=name(parameters)
		end
			def name(parameters):
			return output

运算符	+, -, *, /	+, -, *, /	+, -, *, /
整除问题	/两边是整数时候为整除 5/3 结果是1	/直接取得准确结果 如，5/3 结果是1.6667	/两边是整数时候为整除 5/3 结果是1
数据长度	string.h中的strlen()函数	length()	len()
数组大小	只有单目运算符sizeof，在编译时计算缓冲区的长度	size()	np.size()
正态分布数据	无直接方法，需编写函数	randn()	np.random.randn()
元素访问	中括号[]	小括号 ()	中括号[]
打印输出	printf	disp	print
数据结构	线性表，栈，队列，串，树，图····	矩阵，结构体，元胞数组····	列表，元组，字典，集合····
匿名函数	通过宏来定义 #define lambda(return_type, function_body) \ ({ \ return_type $ this function_body \ $ this ; \ }) #define $ lambda	f=@(x,y) x+y f(2,3)	f = lambda x, y: x+y f(2,3) filter(l ambda x: x+3，[1,2,3，4])
查看数据类型	无，c++才有相关头文件能够实现	a[1，2,3,4,5,6]; class(a) double	a=[1,2,3] type(a) <class'int'>
引号	字符用单引号，字符串用双引号	字符字符串都用单引号	单双引号没有区别
进制转换	无直接方法，需编写函数	x1 = bin2dec('1010') x2 = hex2dec('a') x3 = dec2hex (10) x4 = dec2bin(10)	x1 = int(0b1010) x2 = int (0x10) x3 = hex (10) x4 = bin (20)
.....	...	...	...

这三种编程语言的异同还有很多，但值得注意的是，matlab和python底层都是C语言写的。因此C语言虽然没有前两者已经封装好的许多丰富且强大的功能，但也能将其实现。虽说数据结构不同，但例如递归分治贪心动态规划等算法都有各自的写法。

Problem 3 (75 分)

使用 MATLAB、Python 两种语言实现简单图像去噪算法

要求：

1 提供的报告里应有文字分析（对题目的分析和对结果的分析、讨论）、对应的图片结果展示和代码截图；

2 提供 MATLAB 和 Python 文件的源代码作为附件（同一种语言，三个题的代码写在一个文件里）。

(1) 在生活中由于拍摄设备、拍摄环境、传输网络等的影响，我们获得的图片通常包含一定噪声影响。如果用公式表达可以表示为

$$I_n = I + n$$

其中，I_n 表示我们观测到的有噪声图像，I 表示原始干净图像，n 表示噪声。在实验中科研人员常采用高斯白噪声和椒盐噪声模拟图像中包含的噪声。因此，请同学们分别用 MATLAB 和 Python 实现读取照片“peppers.png”， 并为图片分别加入高斯白噪声（White Gaussian noise）和椒盐噪声(Salt & Pepper noise)，将有噪声图片保存为“peppers_Gaussion.png”和“peppers_sp.png”。对比两种噪声的特点。（15 分）

peppers.png

解答内容：

题目分析：

椒盐噪声又称为脉冲噪声，它是一种随机出现的白点（盐噪声）或者黑点（椒噪声）。

椒盐噪声=椒噪声+盐噪声， 椒盐噪声的值为 0(黑色)或者 255(白色)，

假设等概率的出现 0 或者 255。

添加椒盐噪声步骤：

① 依 SNR 制作 mask，用于判断像素点是原始信号，还是噪声

② 依 mask 给原图像赋噪声值

Python：

```python
def addsalt_pepper(img, SNR):
    '''
        添加椒盐噪声
        SNR : 信噪比
    '''
    img_ = img.copy()
    c, h, w = img_.shape
    mask = np.random.choice((0, 1, 2), size=(1, h, w), p=[SNR, (1 - SNR) / 2., (1 - SNR) / 2.])
    mask = np.repeat(mask, c, axis=0)
    img_[mask == 1] = 255    # 盐噪声
    img_[mask == 2] = 0      # 椒噪声

    return img_
```

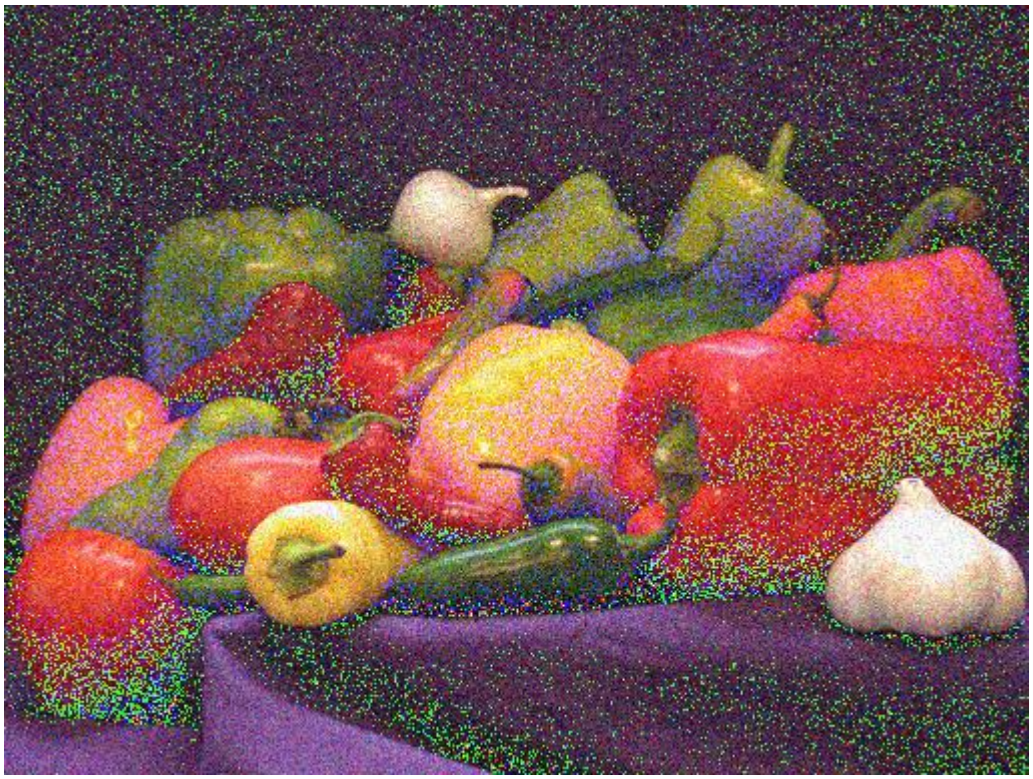
Matlab：

```
image = imread('peppers.png');
[width,height,z]=size(image);
result1 = double(image)./255;
%k1、k2作为判断临界点
k=0.1;%SNR=0.9
%rand(m,n)是随机生成m行n列的矩阵，每个矩阵元素都在0-1之间
%这里k是0.1，所以小于k的元素在矩阵中为1，反之为0
a1=rand(width,height)<k;
a2=rand(width,height)<k;
%合成彩色图像
t1=result1(:,:,1);
t2=result1(:,:,2);
t3=result1(:,:,3);
%分成黑点 白点 随机
t1(a1&a2)=0;
t2(a1&a2)=0;
t3(a1&a2)=0;
t1(a1& ~a2)=255;
t2(a1& ~a2)=255;
t3(a1& ~a2)=255;
result1(:,:,1)=t1;
result1(:,:,2)=t2;
result1(:,:,3)=t3;
result1=uint8(result1);
imwrite(result1, 'peppers_sp.png');
```

高斯噪声是指概率密度函数服从高斯分布的一类噪声。特别的，如果一个噪声，它的幅度分布

服从高斯分布，而它的功率谱密度又是均匀分布的，则称这个噪声为高斯白噪声。
添加高斯白噪声步骤：
① 读取图片像素值并进行类型转换，除以 255 便于后续计算
② 生成均值为 0，方差为 0.01 的噪声并添加至图片中，限制范围后再乘 255 恢复
Python：

```python
def gasuss_noise(image, mean=0, var=0.001):
    '''
        添加高斯噪声
        mean : 均值
        var : 方差
    '''
    image = np.array(image/255, dtype=float)
    noise = np.random.normal(mean, var ** 0.5, image.shape)
    out = image + noise
    if out.min() < 0:
        low_clip = -1.
    else:
        low_clip = 0.
    out = np.clip(out, low_clip, 1.0)
    out = np.uint8(out*255)
    return out
```



Matlab：

```
av=0;
std=0.1;
u1=rand(width,height);
u2=rand(width,height);
x=std*sqrt(-2*log(u1)).*cos(2*pi*u2)+av;
result2=double(image)./255+x*255;
result2=uint8(result2);
imwrite(result2, 'peppers_Gaussion.png');
```

结果分析：

椒盐噪声的特点：出现位置是随机的，但噪声的幅值是基本相同的。

高斯噪声的特点：出现位置是一定的（每一点上），但噪声的幅值是随机的。

1. 在噪声的概念中，通常采用信噪比（Signal-Noise Rate, SNR）衡量图像噪声。通俗的讲就是信号占多少，噪声占多少，SNR 越小，噪声占比越大。在这里，采用信号像素点的占比充当 SNR，以衡量所添加噪声的多少。

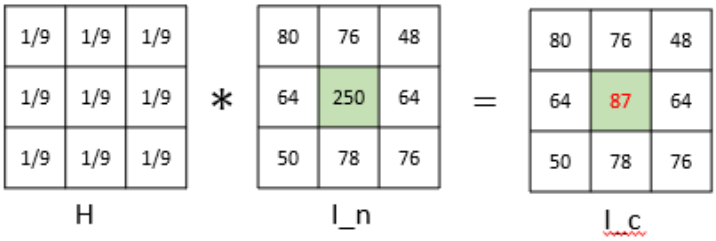
2.均值决定图像的整体灰度范围，随着均值的增大，那么图像整体的灰度同样也会增长，当然，超过 255 依然为 255，但更多的像素靠近 255，从视觉上看起来更加偏白，偏亮。

3.方差决定着图像噪声的密集程度及概率分布程度，方差越大，图像中的噪声对图片的影响也就越大，从视觉上看就是更加密集。

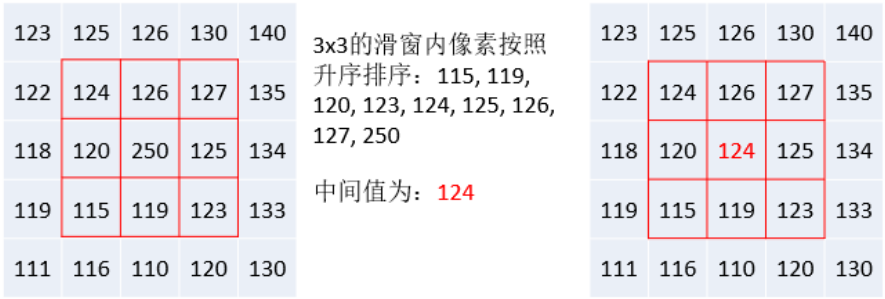
另外生成图片时注意 RGB 的顺序，保持图片原有的颜色。

（2）常用的去噪方法可分为空间域和变换域两大类。其中空间域的去噪算法又可分为线性的均值滤波和非线性的中值滤波。均值滤波主要思想为邻域平均法，即用几个像素灰度的平均值来代替每个像素的灰度。

$$I\_c(x,y) = \sum_{i=-(k-1)/2}^{(k-1)/2} \sum_{j=-(k-1)/2}^{(k-1)/2} H(i,j) I\_n(x+i, y+j)$$



中值滤波是跟均值滤波唯一不同是，不是用均值来替换中心每个像素，而是将周围像素和中心像素排序以后，取中值。中值滤波的特点即是首先确定一个以某个像素为中心点的邻域，然后将邻域中各像素的灰度值排序，取其中间值作为中心像素灰度的新值，这里领域被称为窗口，当窗口移动时，利用中值滤波可以对图像进行平滑处理。



请同学们分别用 MATLAB 和 Python 实现对上题中两幅有噪声图像使用均值和中值两种方式进行去噪处理，同时滑动窗口（滤波核）大小分别设置为 3、 5、 7，对比两种去噪方式分别对两种噪声的去噪效果和不同滤波核大小的影响。（25 分）

解答内容：

题目分析：

图像去噪的实质是对数据本身恢复和重建，以起到排除污染的作用。

均值滤波是一种低通滤波器，高频信号将会去掉，因此可以帮助消除图像尖锐噪声，实现图像平滑，模糊等功能。

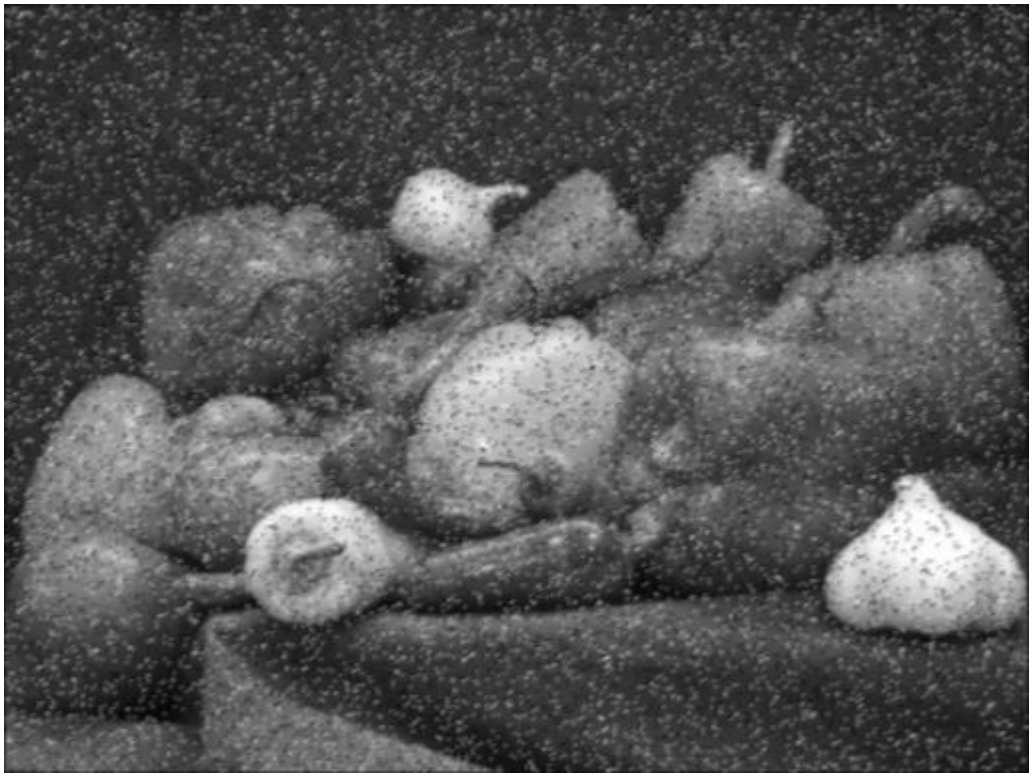
Python：

```python
from tkinter import *
from skimage import io
import numpy as np


im=io.imread('png', as_grey=True)
im_copy_med = io.imread('png', as_grey=True)
im_copy_mea = io.imread('png', as_grey=True)
for i in range(0,im.shape[0]):
    for j in range(0,im.shape[1]):
        im_copy_med[i][j]=im[i][j]
        im_copy_mea[i][j]=im[i][j]
#ui
root = Tk()
root.title("peppers")
root.geometry('300x200')

medL = Label(root, text="中值滤波：")
medL.pack()
med_text = StringVar()
med = Entry(root, textvariable = med_text)
med_text.set("")
med.pack()

meaL = Label(root, text="均值滤波：")
meaL.pack()
mea_text = StringVar()
mea = Entry(root, textvariable = mea_text)
mea_text.set("")
mea.pack()
```

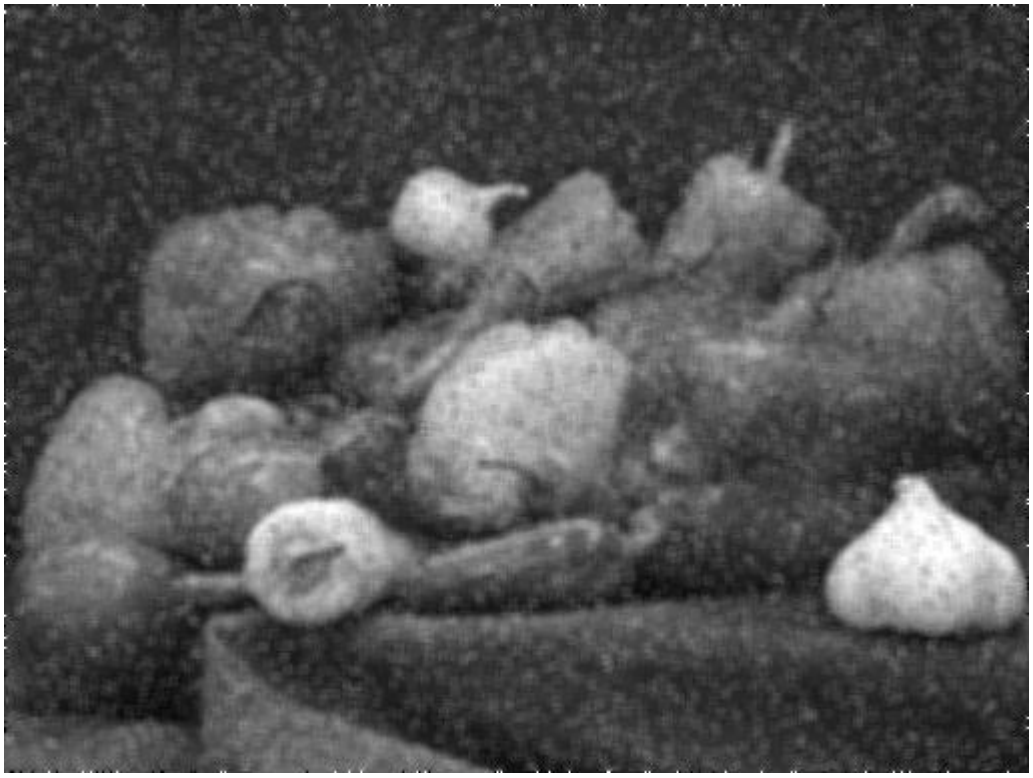
```
def m_filter(x, y, step):
    sum_s=[]
    for k in range(-int(step/2),int(step/2)+1):
        for m in range(-int(step/2),int(step/2)+1):
            sum_s.append(im[x+k][y+m])
    sum_s.sort()
    return sum_s[(int(step*step/2)+1)]

def mean_filter(x, y, step):
    sum_s = 0
    for k in range(-int(step/2),int(step/2)+1):
        for m in range(-int(step/2),int(step/2)+1):
            sum_s += im[x+k][y+m] / (step*step)
    return sum_s

def on_click():
    if(med_text):
        medStep = int(med_text.get())
        for i in range(int(medStep/2),im.shape[0]-int(medStep/2)):
            for j in range(int(medStep/2),im.shape[1]-int(medStep/2)):
                im_copy_med[i][j] = m_filter(i, j, medStep)
    if(mea_text):
        meaStep = int(mea_text.get())
        for i in range(int(meaStep/2),im.shape[0]-int(meaStep/2)):
            for j in range(int(meaStep/2),im.shape[1]-int(meaStep/2)):
                im_copy_mea[i][j] = mean_filter(i, j, meaStep)
    io.imshow(im_copy_med)
    io.imsave(str(medStep) + 'med.jpg', im_copy_med)
    io.imshow(im_copy_mea)
    io.imsave(str(meaStep) + 'mea.jpg', im_copy_mea)

Button(root, text="filterGo", command = on_click).pack()

root.mainloop()
```

对椒盐噪声处理：

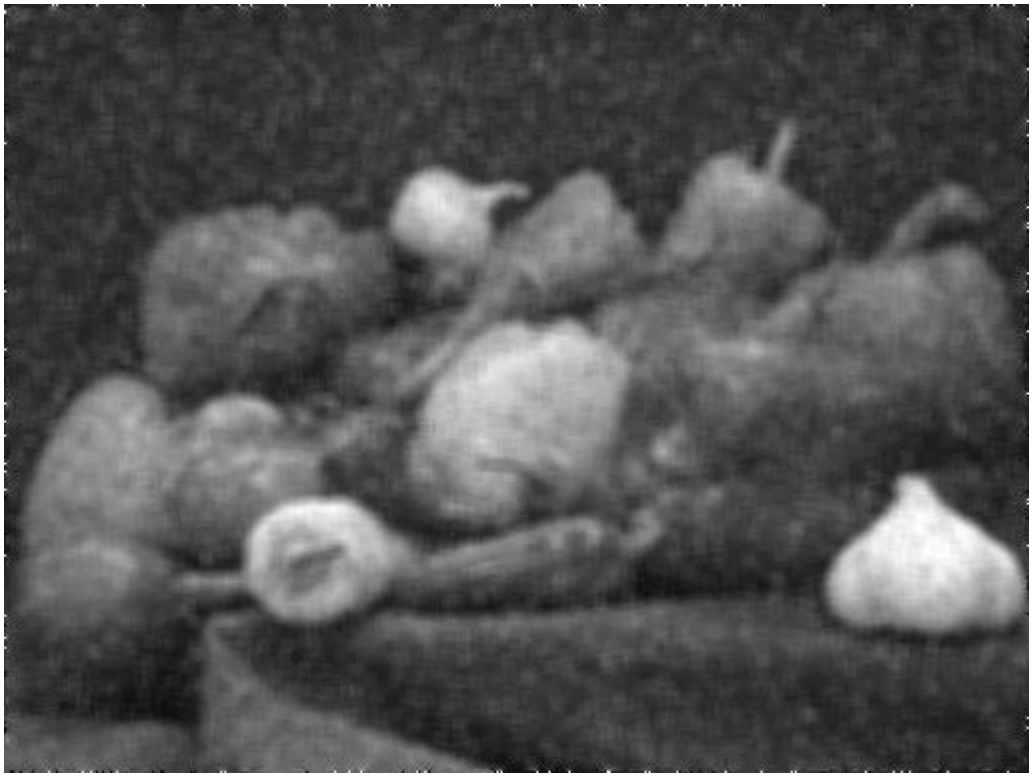
3x3：



5x5：

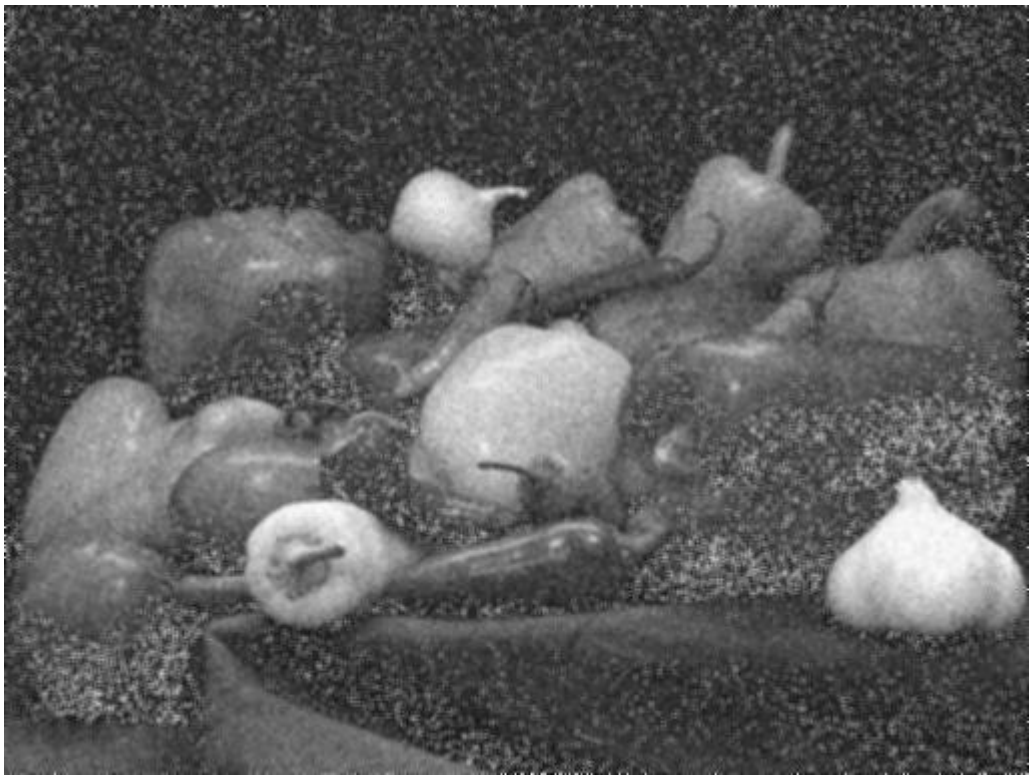


7x7：

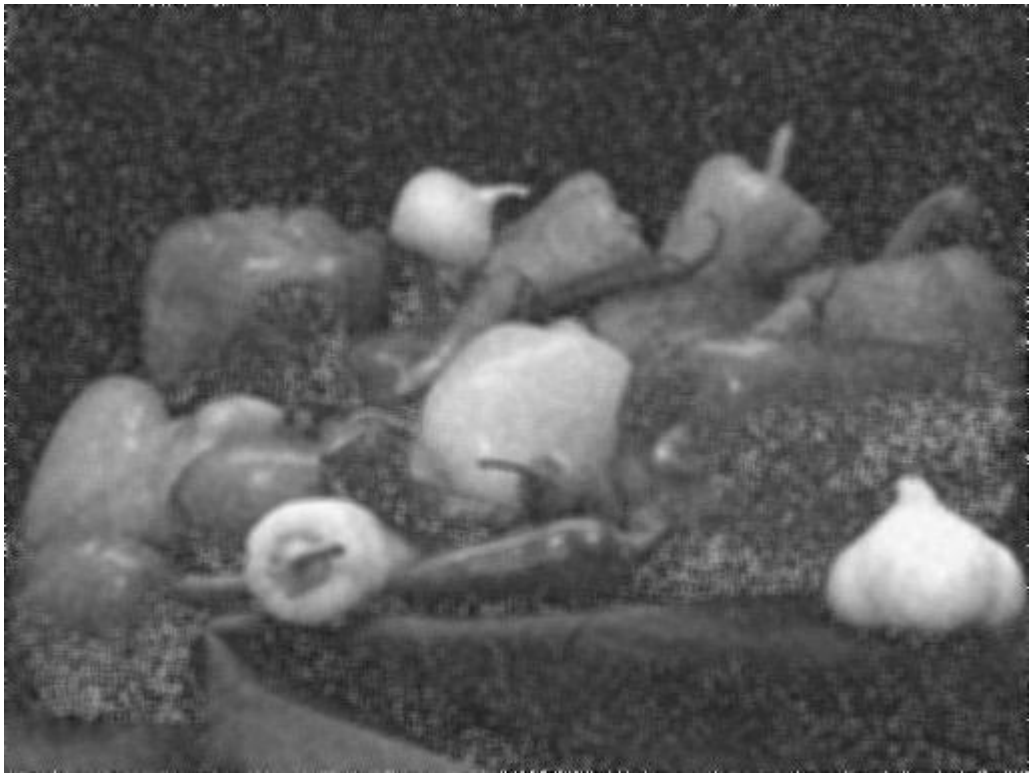


对高斯白噪声处理：

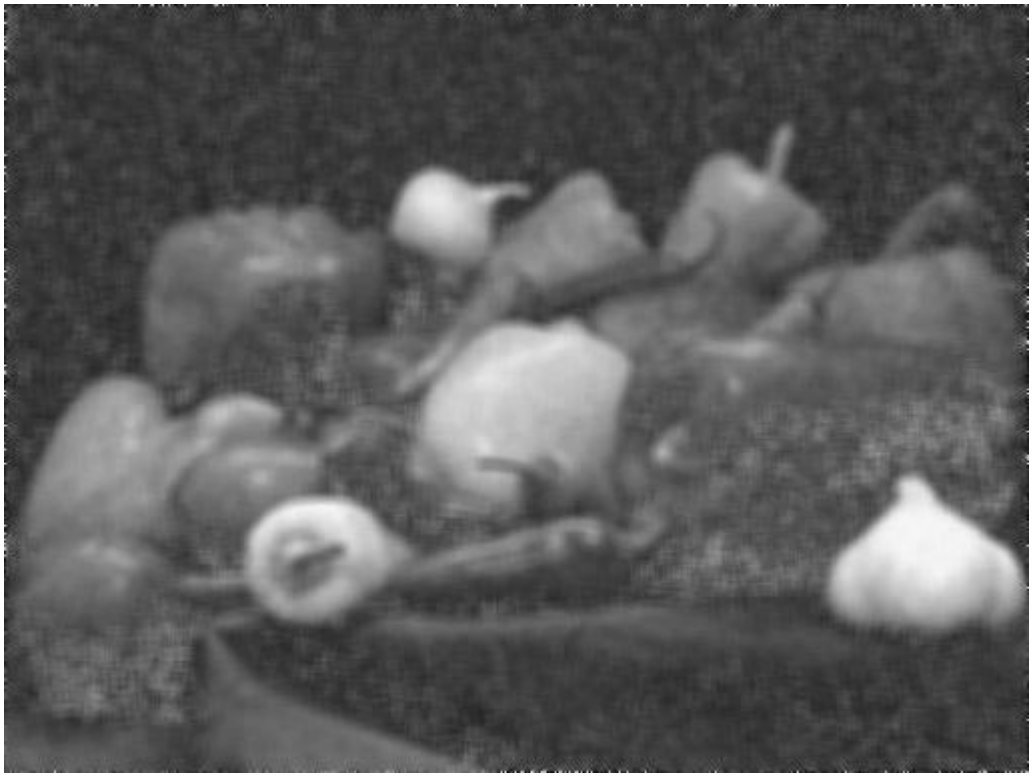
3x3：



5x5：



7x7：

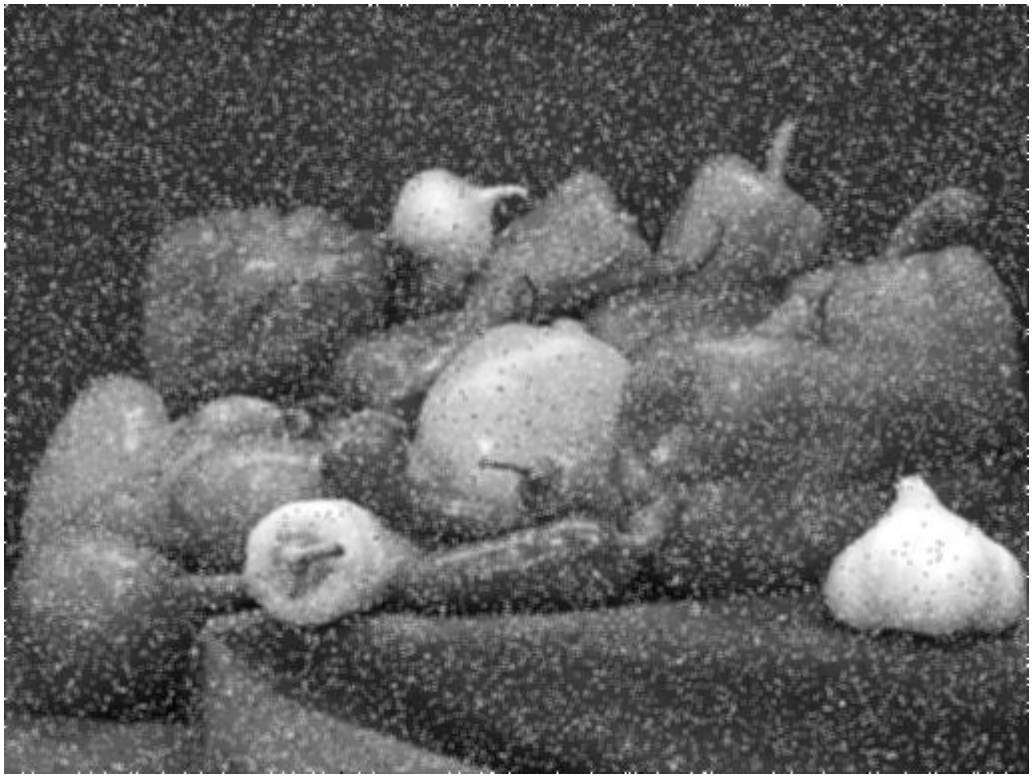


Matlab：

```
%x是需要滤波的图像,n是模板大小(即n×n)
function d=avg_filter(x,n)
a(1:n,1:n)=1;    %a即n×n模板,元素全是1
[hight, width]=size(x);    %输入图像是hightxwidth的,且hight>n,width>n
x1=double(x);
x2=x1;
for i=1:hight-n+1
    for j=1:width-n+1
        c=x1(i:i+(n-1),j:j+(n-1)).*a; %取出x1中从(i,j)开始的n行n列元素与模板相乘
        s=sum(sum(c));                %求c矩阵中各元素之和
        x2(i+(n-1)/2,j+(n-1)/2)=s/(n*n); %将与模板运算后的各元素的均值赋给模板中心位置的元素
    end
end
%未被赋值的元素取原值
d=uint8(x2);
```

对椒盐噪声处理：

3x3：



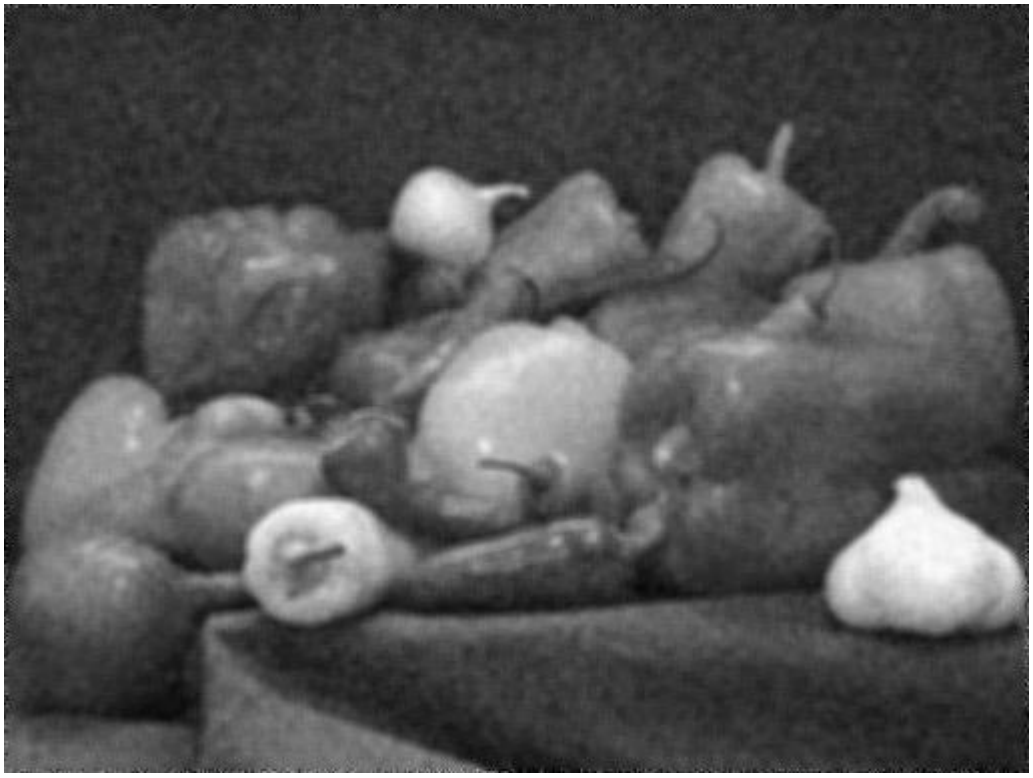
5x5：

7x7：

对高斯白噪声处理：

3x3：

5x5：



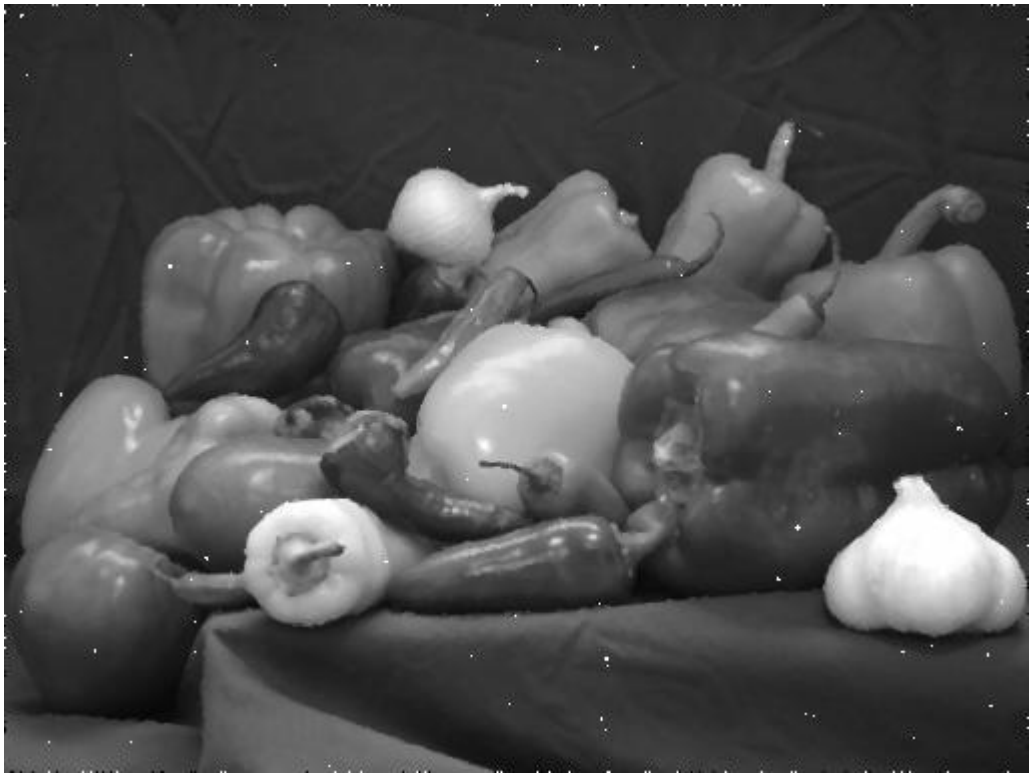
7x7：

中值滤波在一定的条件下可以克服常见线性滤波器如方框滤波器、均值滤波等带来的图像细节模糊，而且对滤除脉冲干扰及图像扫描噪声非常有效，但耗时较长。

Python：

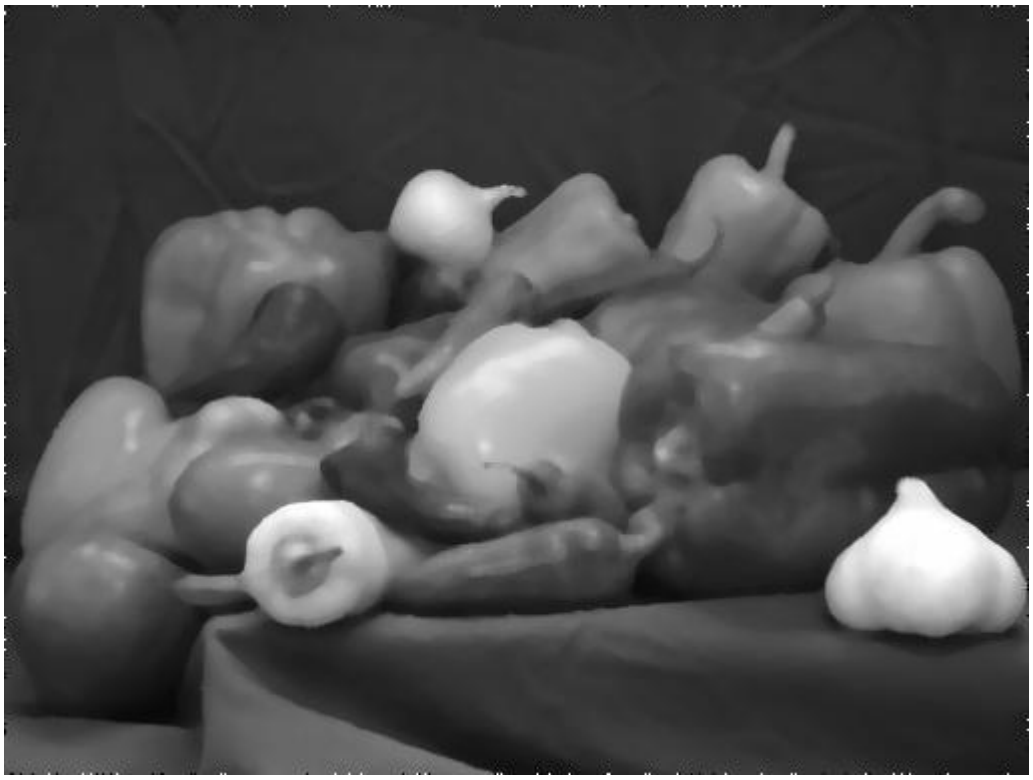
对椒盐噪声处理：

3x3：



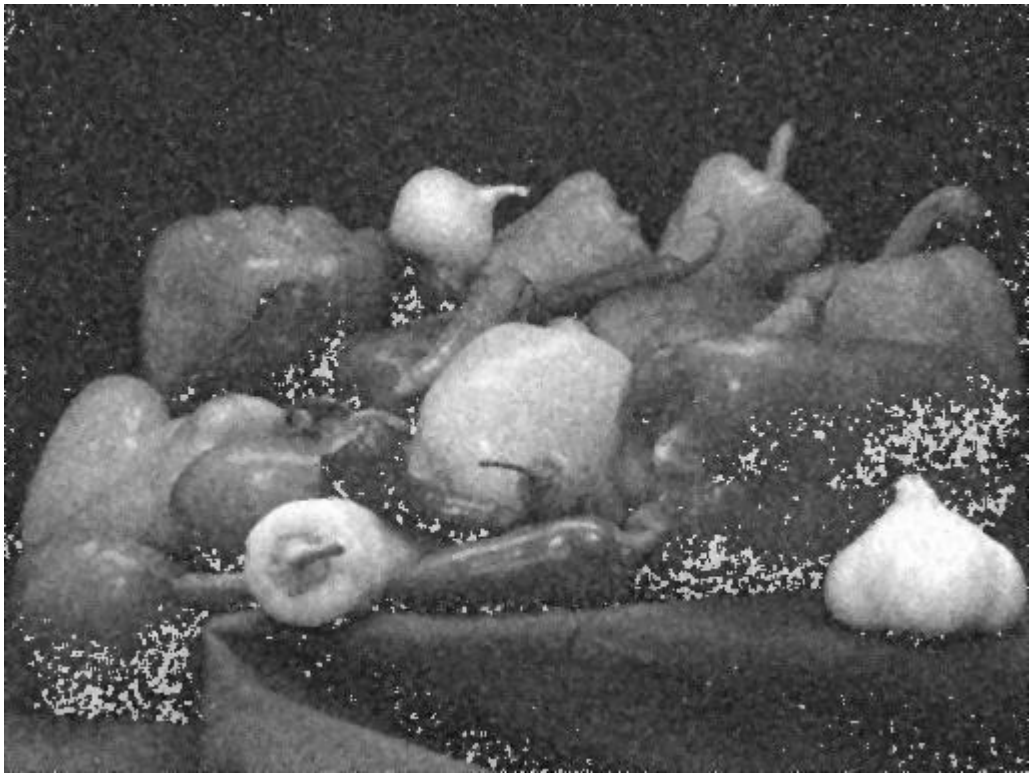
5x5：

7x7：



对高斯白噪声处理：

3x3：



5x5：

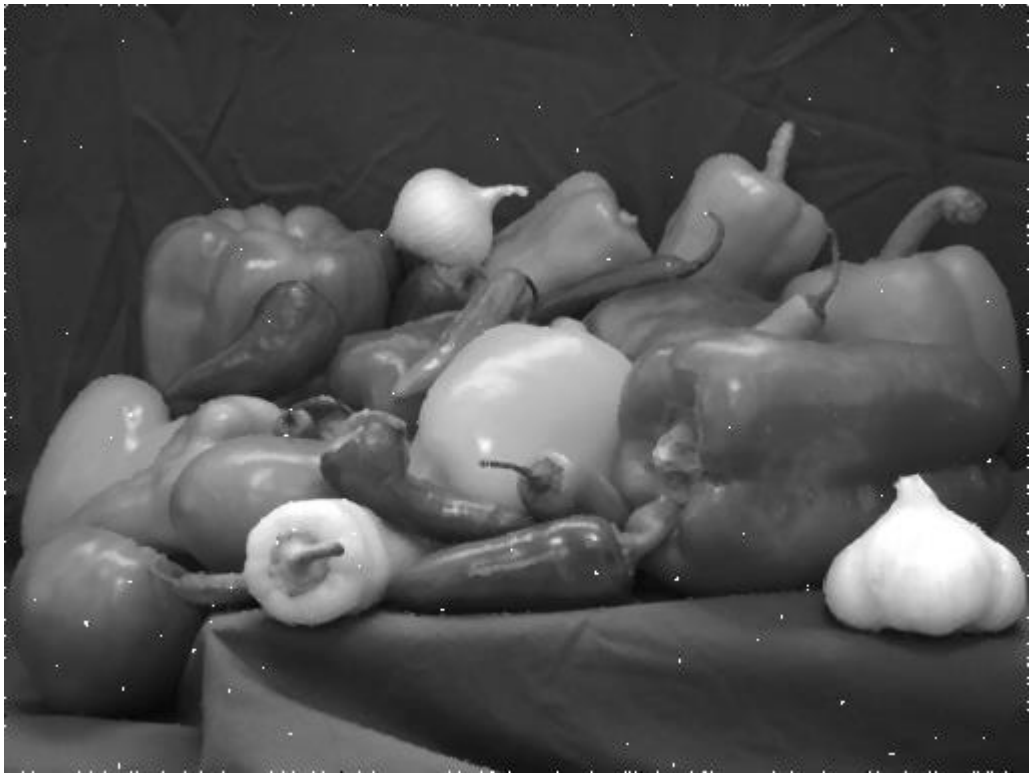
7x7:

Matlab:

```
function [ img ] = median_filter( image, m )
%-------------------------------------------
%中值滤波
%输入：
%image：原图
%m：模板的大小3*3的模板，m=3
%输出：
%img：中值滤波处理后的图像
%-------------------------------------------
    n = m;
    [ height, width ] = size(image);
    x1 = double(image);
    x2 = x1;
    for i = 1: height-n+1
        for j = 1:width-n+1
            mb = x1( i:(i+n-1),  j:(j+n-1) );
            mb = mb(:);
            mm = median(mb);
            x2( i+(n-1)/2,  j+(n-1)/2 ) = mm;

        end
    end

    img = uint8(x2);


end
```

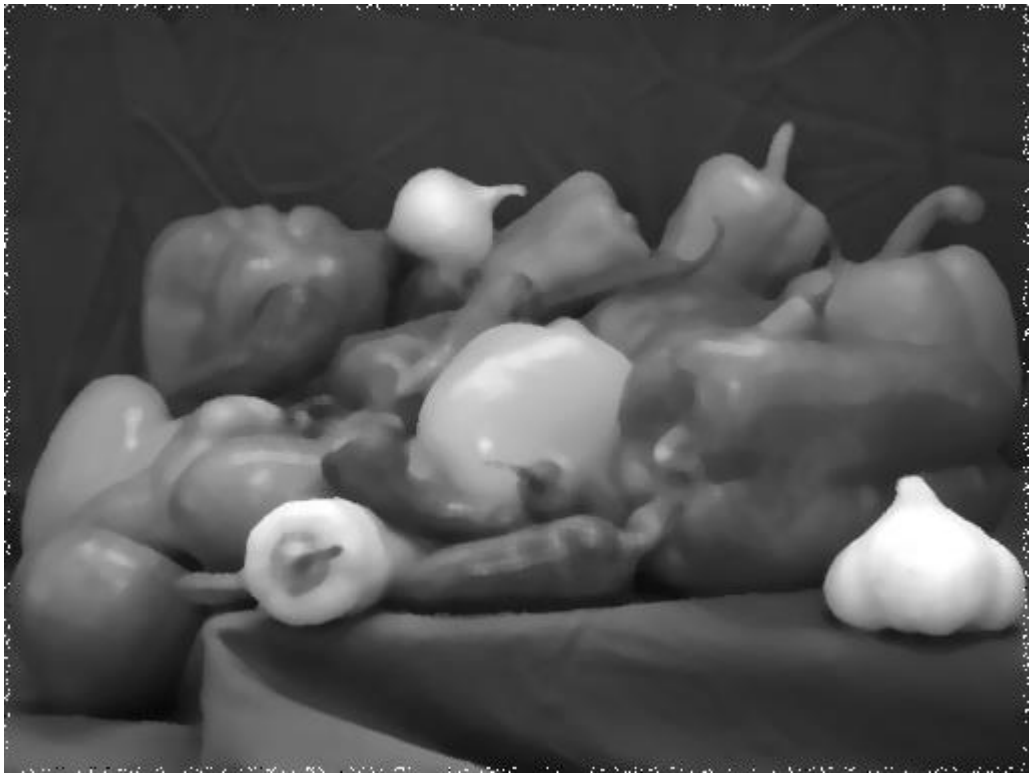
对椒盐噪声处理：

3x3：



5x5：

7x7：



对高斯白噪声处理：

3x3：

5x5：

7x7：

结果分析：

均值滤波和和中值滤波都可以起到平滑图像，虑去噪声的功能。

① 均值滤波采用线性的方法，平均整个窗口范围内的像素值，均值滤波本身存在着固有的缺陷，即它不能很好地保护图像细节，在图像去噪的同时也破坏了图像的细节部分，从而使

图像变得模糊，不能很好地去除噪声点。均值滤波对高斯噪声表现较好，对椒盐噪声表现较差。

② 中值滤波采用非线性的方法，它在平滑脉冲噪声方面非常有效,同时它可以保护图像尖锐的边缘，选择适当的点来替代污染点的值，所以处理效果好，对椒盐噪声表现较好，对高斯噪声表现较差。

③ 两种滤波处理后的图像都会随着滤波核的增大而模糊，且中值滤波核必须为奇数，均值滤波核则没有这一限制。

（3）均值和中值滤波方式也可以被用来进行图片美化、人脸美化。请在均值和中值滤波中选择一种你认为合适的滤波方式和滤波核大小分别用 MATLAB 和 Python 对图像“original_face.jpeg”进行美化处理，给出你对处理后图片的观察、你选择该滤波方式和滤波核的理由。（20 分）

original_face.jpeg

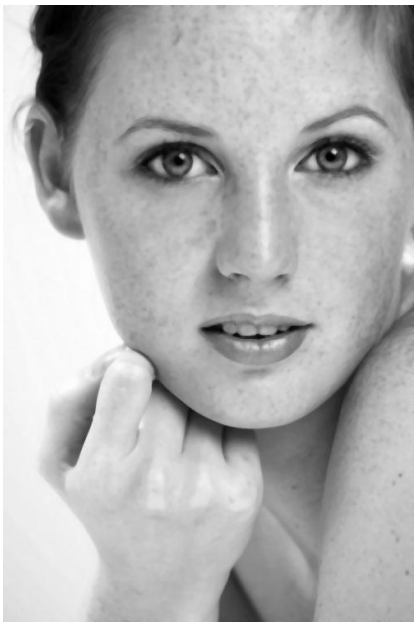
解答内容：

由上一题总结得出，均值滤波不能保护图像细节，易使图像模糊；且滤波核越大，图像越模糊；故我先选择 5x5 的中值滤波进行处理：

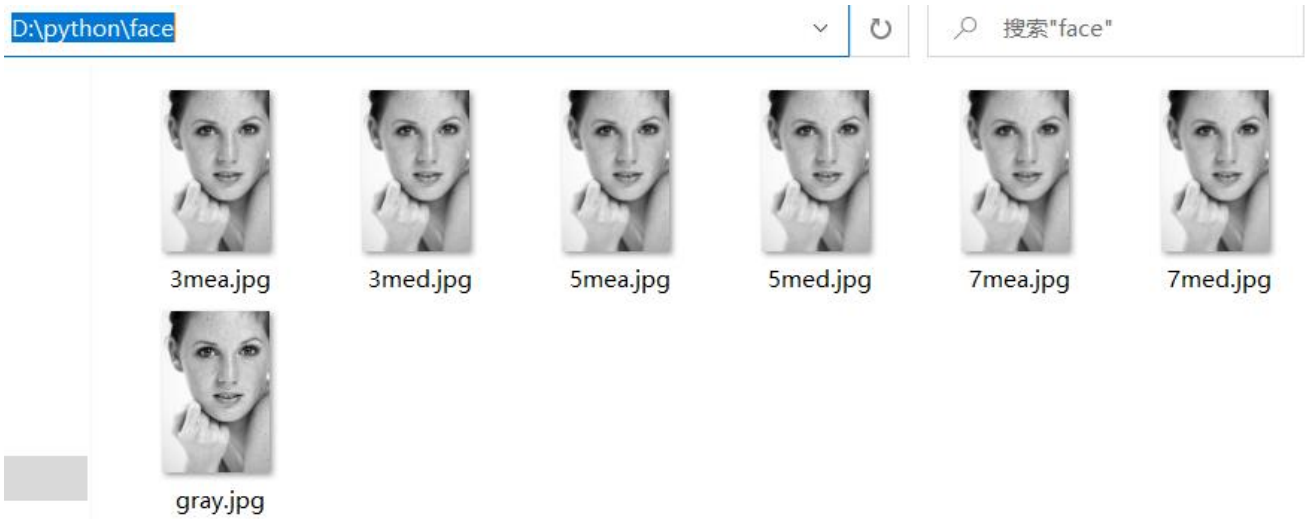
原始灰度图：

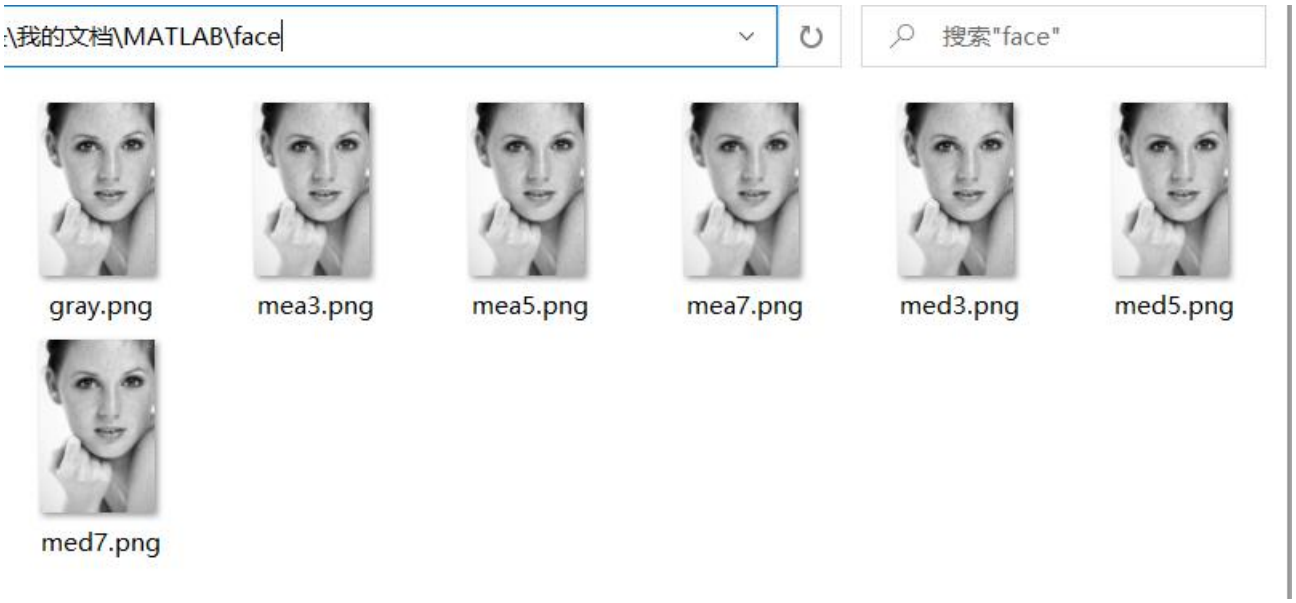
Matlab：

Python：



为了验证猜想是否正确，我将两种滤波，三种滤波核下的图片与灰度图进行对比：





肉眼观察可以验证猜想基本正确，5x5 的中值滤波处理过后的图像相对最清晰；

但由于肉眼观察具有主观性，故可计算出两种处理方法对应的 MSE、SNR、PSNR(图像去噪客观评价标准)进行对比：

SNR 于 Problem3（1）的结果分析已提及；

$$\mathrm{SNR\ (dB)} = 10 \cdot \log_{10}\left[\frac{\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}\left(f(x,y)\right)^2}{\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}\left(f(x,y)-\hat{f}(x,y)\right)^2}\right]$$

MSE 是指均分误差，均方误差法首先计算原始图像和失真像象素差值的均方值，然后通过均方值的大小来确定失真图像的失真程度。

$$MSE = \frac{1}{M \times N} \sum_{0 \le i < N} \sum_{0 \le j < M} (f_{ij} - f_{ij}')^2$$

PSNR 作为衡量图像质量的重要指标，基于通信理论而提出，是最大信号量与噪声强度的比值。由于数字图像都是以离散的数字表示图像的像素，因此采用图像的最大象素值来代替最大信号量。

$$PSNR = 10 \times \lg \frac{L \times L}{MSE}$$

中值滤波的 MSE 值更小,SNR、PSNR 值更大,对这 3 种客观评价标准来说,MSE 值越小,SNR、PSNR 值越大,代表该方法处理图像的效果越好。

去噪方法	均值滤波	中值滤波
MSE	202.2239	126.9728
SNR/dB	19.4555	21.4767
PSNR/dB	25.0725	27.0937

综上所述，中值滤波更适合用于对人像这种非单一噪声污染的图像进行美化，滤波核应结合实际需求选择，滤波核越大，美化效果越好，但同时会使图像更模糊。

（4）通过以上三道题的完成，你是否观察出 MATLAB 与 Python 的异同和各自的优劣势？请写出你的观察和总结。（10 分）

解答内容：

以上三道题分别进行了对图像加入噪声，图像去噪和人像美化，

共同点就是这三道题用 matlab 和 python 都可以实现，同一种思路只是用两种语法写出。

优劣势：

Matlab 官方相关功能的函数非常全面，例如添加噪声的 imnoise(),进行滤波的 imfilter(),在图像处理的科研工作中使用现成的函数应该更加方便，且如果是自行编写函数，由于工作区的存在各种变量内容都非常直观，不需要特地打印出来；

但无法像 python 作为开源的一样实现 cuda 显卡加速。且各个工具箱并不是默认安装，例如 imnoise()的安装就需要近 2G 空间和等待较长时间；而 python 的库只有 50M 左右。

Python 作为一款开源编程语言，越来越“无所不能”，本次作业中我使用了 OpenCV 和 Numpy 两个库，支持对像素级的操作，且底层由 C++实现，效率也较高。众多图片处理包中，Scikit-

image 更是尤为突出，几乎集合了 matlab 的所有图像处理功能。

虽然库多也能使 python 功能齐全，但这些库非官方统一开发，很多库对图像读取成数组的定义无法统一，比如颜色通道的顺序，比如长宽的顺序。在 cv2 中读取图像时就是以 BGR，如果因为常识下意识地认为是 RGB，就会导致最后的图片颜色异常。

（5）（提升题）你是否还能找出另一种更优的去噪方式是用于人脸美化并用 MATLAB 或 Python 实现？ （5 分）

解答内容：

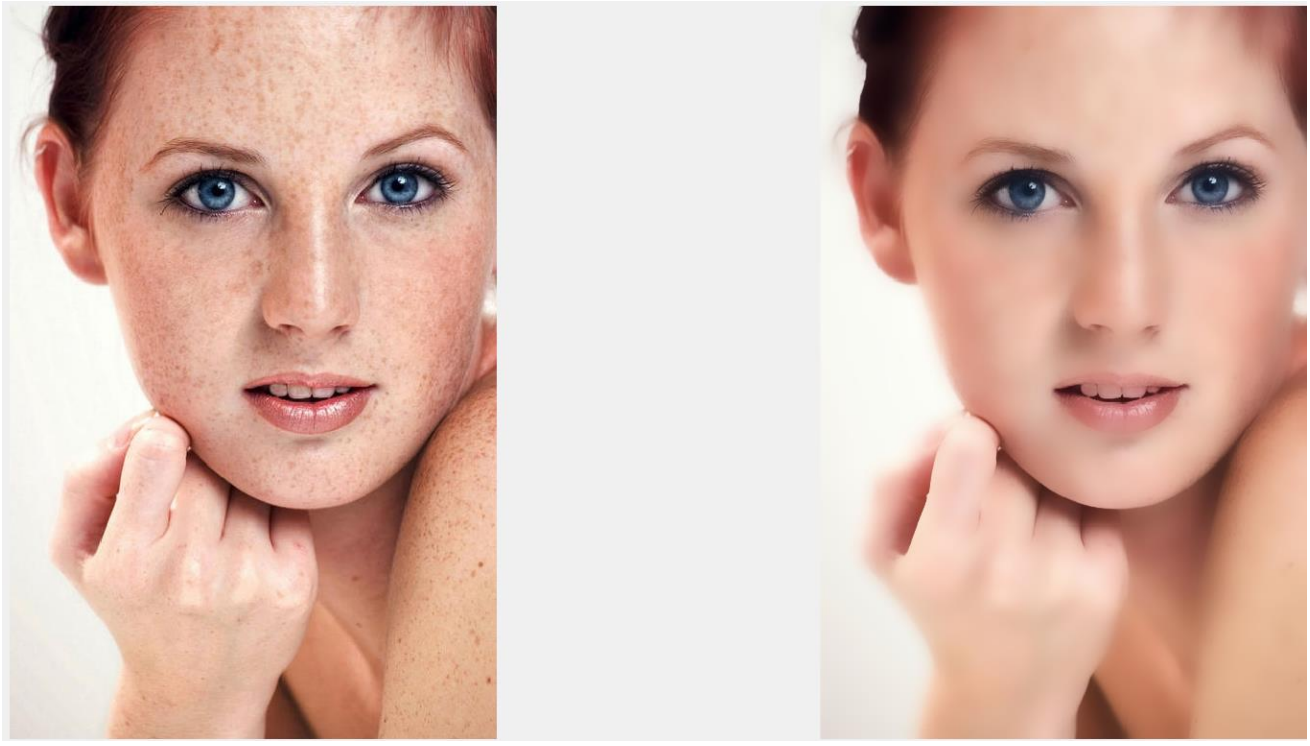
查阅资料后得知：

双边滤波，Bilateral filter，是一种保持边缘的降噪平滑滤波器。与前两种滤波相比有两个优势：

第一是保持边缘，这样对于人脸而言，可以在平滑皮肤区域的同时不破坏五官的边缘结构。

第二，降噪平滑，这样就可以抑制皮肤上的斑点和痘痘，使得皮肤变得柔和。

Matlab 对比图：



Python 对比图：

显然双边滤波的效果远远好过于中值滤波与均值滤波；

最后我猜测因为基于卷积神经网络的审美评分已经实现，基于这类评分网络用梯度上升法微调做美化可能去噪效果更好，但目前我暂时无法尝试实现。