D.O.P.:	L.D.O.S.:
Roll No: D084	Name: Somish Jain
Branch:ce	Batch:2
Date of Experiment: 7/08/2025	Date of Submission: 7/08/2025
Grade:	Faculty:

Experiment No. 4:

Aim:

Apply a suitable spatial domain filtering technique to enhance the following three images.

i. Image with Gaussian Noise.

ii. Image with salt and pepper noise.

iii. Blurred image.

Analyze the effect of various masks sizes of filter used and comprehend your findings.

Prerequisite:

1. Image Processing command.
2. MATLAB
3. Basics of Filters.

Objective:

a. Apply the smoothing filter to remove the noise

Outcome:

1. Outline the fundamentals of digital image processing system, and observing the results of filter.

Theory:

Noise:

Images are corrupted by random variations in intensity values called noise due to non-perfect camera acquisition or environmental conditions.

Common Types of Noise

- **Salt and pepper noise:** random occurrences of both black and white intensity values.
- **Impulse noise:** random occurrences of white intensity values.
- **Gaussian noise:** impulse noise but its intensity values are drawn from a Gaussian distribution. Noise Intensity Value is given by following equation:

k: random intensity value.

Effects of noise on digital images:

Presence of particular type of noise in an image may deteriorate image quality to certain level. The level of deterioration depends on the density of noise

Noise Filtering:

Basic Idea: replace each pixel intensity value with an new value taken over a neighborhood of fixed size.

The size of the filter controls degree of smoothing

- large filter » large neighborhood » intensive smoothing.

Average and weighted average Filter:

Average filter (low pass filter) removes the noise present in the image. Noise, is normally a high frequency signal and low pass filtering eliminates the noise. An image is smoothed by decreasing the disparity between pixel values by averaging nearby pixels.

Using this filter tends to retain the low frequency information within an image while reducing the high frequency information. An example is an array of ones divided by the number of elements within the kernel, such as the following 3 by 3 kernel:

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Median Filter

Replace each pixel value with the median of the gray values in the region of the pixel:

1. Take a 3 x 3 (or 5 x 5 etc.) region centered around pixel (i,j)
2. Sort the intensity values of the pixels in the region into ascending order
3. Select the middle value as the new value of pixel (i,j)

Computation of Median Values:

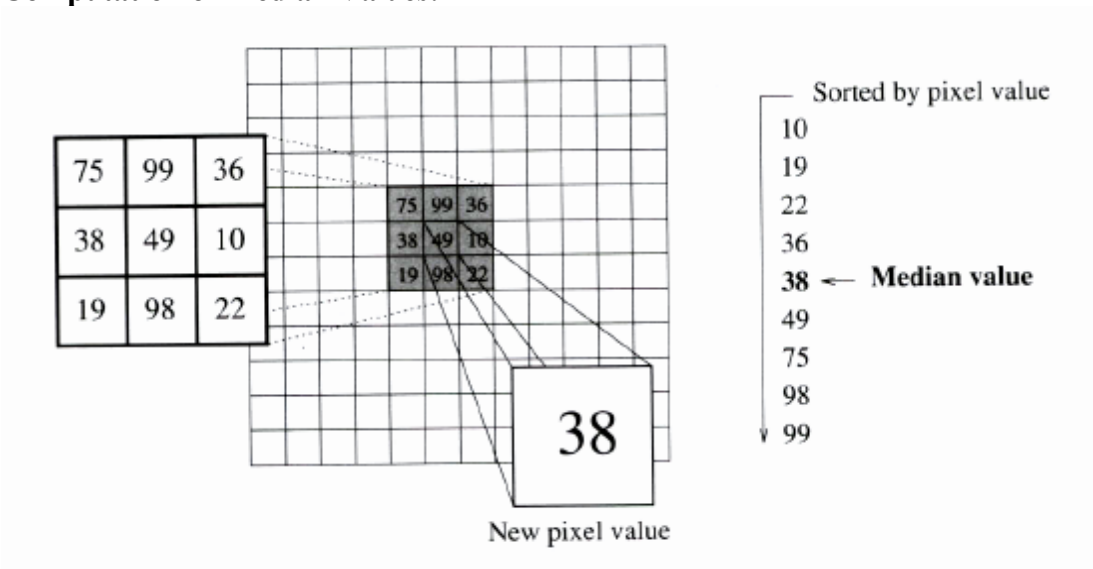


Figure 1. Computation of Median values

Procedure for smoothing filter:

1. Read the input image.

2. Add noise to the input image. (use command imnoise( )).

3. Design your own average, weighted average and Median filters

4. Apply these Filters to the image.

5. Display the original and the output image.

6. Observe the output and write observations in conclusion section.

7. Add salt and pepper noise to the input image.

8. Apply the same Filters to the image obtained in step 5.

9. Display the original and the output images.

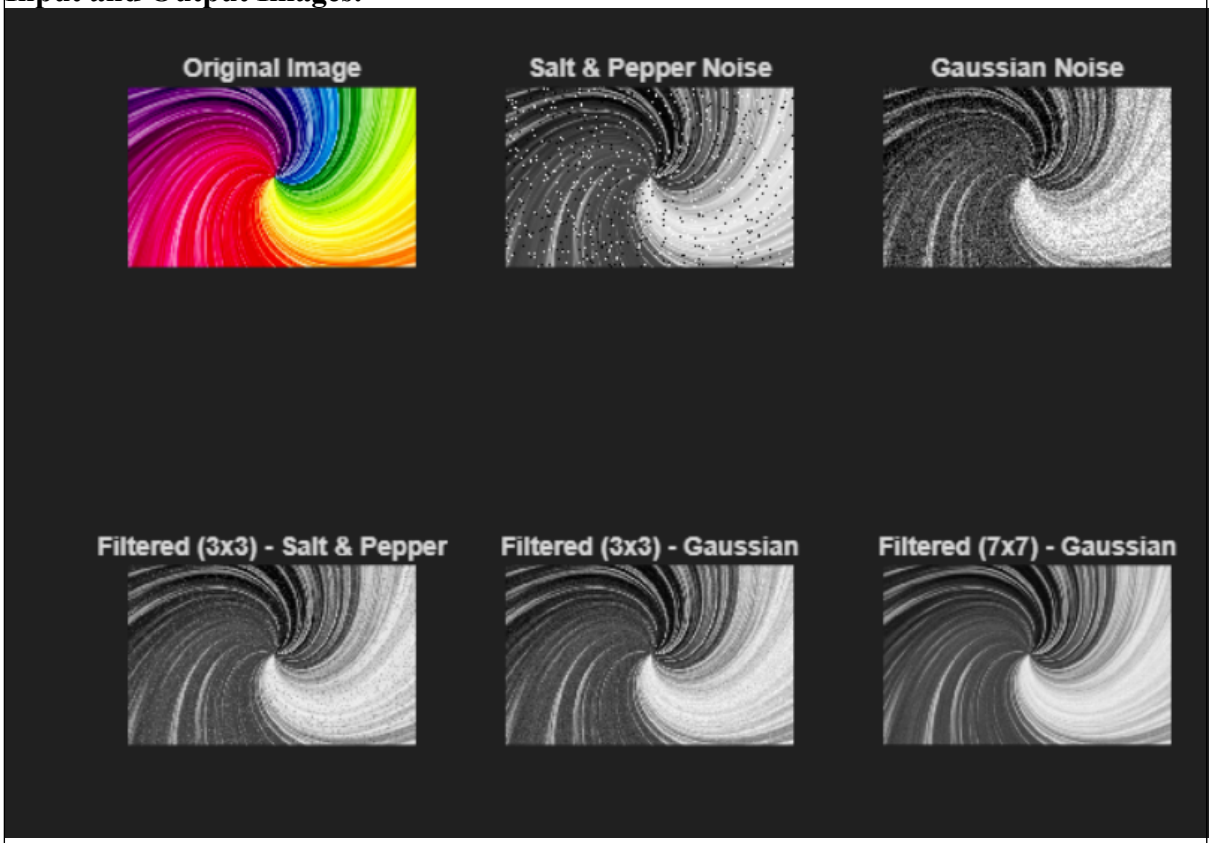
10. Observe the output and write observations in conclusion section.

Code:

```
clc;
clear;
close;

% Read the image
img = imread("pic.jpg");

% Convert to grayscale
grayimg = rgb2gray(img);

% Add noise to the grayscale image
salt_pepper_noise = imnoise(grayimg, 'salt & pepper');
gaussian_noise = imnoise(grayimg, 'gaussian');

% Define filter kernels
h1 = (1/9) * ones(3,3);  % 3x3 averaging filter
h2 = (1/49) * ones(7,7); % 7x7 averaging filter

% Apply filters
filtered_sp = imfilter(salt_pepper_noise, h1, 'same');  % Salt & Pepper filter
filtered_g1 = imfilter(gaussian_noise, h1, 'same');    % Gaussian noise filter (3x3)
filtered_g2 = imfilter(gaussian_noise, h2, 'same');    % Gaussian noise filter (7x7)

% Display results
figure;
subplot(2, 3, 1);
imshow(img);
title('Original Image');

subplot(2, 3, 2);
imshow(salt_pepper_noise);
title('Salt & Pepper Noise');

subplot(2, 3, 3);
imshow(gaussian_noise);
title('Gaussian Noise');

subplot(2, 3, 4);
imshow(filtered_sp);
title('Filtered (3x3) - Salt & Pepper');
```

```
subplot(2, 3, 5);
imshow(filtered_g1);
title('Filtered (3x3) - Gaussian');

subplot(2, 3, 6);
imshow(filtered_g2);
title('Filtered (7x7) - Gaussian');
```

Input and Output Images:



Observations:

After applying smoothing filters (like averaging or Gaussian filter), the image becomes **less noisy** and looks **blurred**.

Fine details and sharp edges are reduced, but overall image noise and graininess decrease.

The effect depends on the size of the filter: larger masks produce stronger smoothing (more blur).

Conclusion:

The smoothing operation reduces **noise and unwanted details** from the image by averaging neighboring pixel values. Though edges may become blurred, smoothing is an effective preprocessing step for applications like **denoising, background suppression, and feature extraction**.