LAB Report

Experiment No: 01

Experiment Name: Image Filtering and Noise Removal

Question: Apply different noise types (Gaussian, Salt & Pepper) to an image and use the following filters to remove noise:

- Mean Filter
- Median Filter
- Gaussian Filter

Compare the effectiveness of each method. Which filter works best for which type of noise?

Objectives

1. To apply different noise types (Gaussian and Salt & Pepper) to an image.
2. To implement different filtering techniques: Mean Filter, Median Filter, and Gaussian Filter.
3. To compare the effectiveness of each filtering technique in noise removal.

Theory

1. Noise in Images:
Noise in images refers to random variations in pixel intensity, often caused by external factors like sensor interference or transmission errors. Common noise types include:

- **Gaussian Noise:** A statistical noise having a normal distribution, often seen in natural images and defined by the equation:

  $$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

  where μ is the mean and σ^2 is the variance.

- **Salt & Pepper Noise:** A type of noise where random pixels are set to maximum or minimum intensity, appearing as white and black specks. It follows a random distribution where:

$$P(x) = \begin{cases} p_s, & x = I_{max} \\ p_p, & x = I_{min} \\ 0, & \text{otherwise} \end{cases}$$

where p_s and p_p are probabilities of salt and pepper noise, respectively.

2. Image Filtering Techniques:
To remove noise from images, various filtering methods can be applied:

- **Mean Filter:** A simple averaging filter that smoothens the image by replacing each pixel with the mean of its neighborhood pixels. It is effective for Gaussian noise but may blur edges. The mathematical representation is:

  $$I'(x,y) = \frac{1}{N}\sum_{i=-k}^{k}\sum_{j=-k}^{k} I(x+i, y+j)$$

  where is the total number of pixels in the filter window.

- **Median Filter:** A non-linear filter that replaces each pixel with the median value of its surrounding pixels. It is highly effective for Salt & Pepper noise as it preserves edges. It is computed as:

  $$I'(x,y) = \text{median}\{I(x+i, y+j) | -k \leq i, j \leq k\}$$

- **Gaussian Filter:** A smoothing filter that applies a weighted average, giving more importance to central pixels. It is effective for Gaussian noise while maintaining edge details. The filter uses a Gaussian function:

  $$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

  where σ controls the amount of smoothing.

3. Properties of Filters:

- **Mean Filter:** Simple and fast, but it blurs the image.
- **Median Filter:** Good for removing impulse noise without blurring.
- **Gaussian Filter:** Best for reducing Gaussian noise while preserving edges.

Code Implementation by MATLAB

```
clc;
```

```
clear all;
close all;

% Read image and convert to grayscale
image = im2double(imread('image.jpg'));

% Add Gaussian noise
gaussian_noisy = imnoise(image, 'gaussian', 0, 0.01);

% Add Salt & Pepper noise
sp_noisy = imnoise(image, 'salt & pepper', 0.02);

% Apply filters
mean_filter = fspecial('average', [5 5]);
mean_gaussian = imfilter(gaussian_noisy, mean_filter);
median_gaussian = medfilt2(gaussian_noisy, [5 5]);
gaussian_filtered_gaussian = imgaussfilt(gaussian_noisy, 1);

mean_sp = imfilter(sp_noisy, mean_filter);
median_sp = medfilt2(sp_noisy, [5 5]);
gaussian_filtered_sp = imgaussfilt(sp_noisy, 1);

% Display results
figure;
subplot(3,4,1), imshow(image), title('Original');
subplot(3,4,2), imshow(gaussian_noisy), title('Gaussian Noise');
subplot(3,4,3), imshow(sp_noisy), title('Salt & Pepper Noise');

subplot(3,4,5), imshow(mean_gaussian), title('Mean Filter (G)');
subplot(3,4,6), imshow(median_gaussian), title('Median Filter (G)');
subplot(3,4,7), imshow(gaussian_filtered_gaussian), title('Gaussian Filter (G)');

subplot(3,4,9), imshow(mean_sp), title('Mean Filter (SP)');
subplot(3,4,10), imshow(median_sp), title('Median Filter (SP)');
subplot(3,4,11), imshow(gaussian_filtered_sp), title('Gaussian Filter (SP)');
```

Output: