Scilab case study project on

Image Denoising Using Gaussian and Median Filters: A Comparative Study in Scilab

Ayushi

Integrated M.Tech in AI, VIT Bhopal University

Image Processing

10 April 2026

Abstract

Image denoising is a critical pre-processing step in computer vision pipelines, where sensor-induced Gaussian noise degrades image quality and disrupts tasks such as segmentation and edge detection. This case study aims to implement and compare two classical spatial domain denoising filters, the Gaussian filter and the Median filter, using Scilab as an open-source alternative to MATLAB. A 256×256 synthetic image corrupted by Gaussian noise with zero mean and standard deviation of 25 serves as the experimental input. A 5×5 Gaussian convolution kernel and a 3×3 Median sliding window are independently implemented from scratch without any external toolbox. Performance is evaluated quantitatively using Peak Signal-to-Noise Ratio (PSNR) in decibels, and results are visualized through comparative plots and bar charts at each processing stage. This case study is inspired by the theoretical findings of Liu et al. (IEEE ICIS 2014) on Gaussian filter-based image denoising, validated using PSNR as the primary evaluation metric.

Keywords: Image Denoising, Gaussian Filter, Median Filter, Peak Signal-to-Noise Ratio (PSNR), Spatial Domain Filtering, Gaussian Noise, Scilab, Image Processing

1. Introduction

Digital images captured through electronic sensors are routinely degraded by noise arising from thermal fluctuations, transmission errors, and quantization effects. Among the various types of noise encountered in practice, Gaussian noise is the most prevalent form, appearing as random variations superimposed uniformly across pixel intensity values throughout the entire image. If left unaddressed, noise reduces image clarity and significantly interferes with downstream processing tasks including object detection, edge analysis, image segmentation, and pattern recognition.

Spatial domain filtering is one of the most widely used approaches for image denoising. In this approach, each output pixel value is computed as a function of the pixel values within a local neighborhood in the input image. Two filters dominate this category in academic and practical use. The Gaussian filter performs weighted averaging using a bell-shaped kernel, where pixels closer to the center contribute more to the output than those farther away. The Median filter, on the other hand, replaces each pixel with the statistical median of its local neighborhood, making it inherently robust to outlier noise values while better preserving sharp edges.

This case study implements both filters entirely from scratch in Scilab 2026.0.1 without the use of any external toolbox, and evaluates their denoising effectiveness quantitatively using the Peak Signal-to-Noise Ratio (PSNR) metric. The study is motivated by the findings of Liu et al. (IEEE ICIS 2014), whose work establishes that Gaussian filter-based denoising achieves superior PSNR for Gaussian noise. Due to differences in test image and filter methodology, absolute numerical results differ from the reference; this is explicitly documented in Section 7.5. The work demonstrates the viability of Scilab as a freely available open-source alternative to MATLAB for image processing tasks in academic and research settings.

2. Problem Statement

In practical image acquisition systems, sensors introduce unwanted noise during the process of capturing and transmitting image data. Gaussian noise, characterised by a normal probability distribution with zero mean, is the most commonly encountered noise type in digital imaging. The presence of such noise degrades the visual quality of images and negatively impacts the accuracy of subsequent computer vision operations such as edge detection, object recognition, and image segmentation.

The central problem addressed in this case study is the removal of Gaussian noise from a digital image using two classical spatial domain filtering techniques, namely the Gaussian filter and the Median filter, implemented entirely in Scilab without relying on any external toolbox or pre-built image processing library.

The specific objectives of this case study are as follows:

1. To generate a 256×256 synthetic test image and corrupt it with Gaussian noise having zero mean and standard deviation of 25.
2. To implement a 5×5 Gaussian convolution kernel with $\sigma=1.5$ from scratch and apply it to the noisy image.
3. To implement a 3×3 Median filter using a sliding window approach and apply it to the same noisy image.
4. To compare denoising performance of both filters using PSNR in decibels as the quantitative metric.
5. To visualise results through a four-panel image comparison plot and a PSNR bar chart.

The solution is validated by confirming that the Gaussian filter achieves higher PSNR improvement for Gaussian noise compared to the Median filter, which is consistent with the theoretical expectation and the findings reported in the reference IEEE paper by Liu et al.

3. Basic concepts related to the topic

3.1 Digital Image and Noise

A digital image is a two-dimensional array of pixel values, where each pixel represents the intensity of light at that spatial location. In grayscale images, pixel values range from 0 (black) to 255 (white). Noise refers to any unwanted random variation in pixel intensity values that is introduced during image acquisition, storage, or transmission. Noise degrades the visual quality of an image and reduces the reliability of any processing performed on it.

3.2 Gaussian Noise

Gaussian noise is an additive noise model where the noise value added to each pixel is drawn independently from a normal (Gaussian) probability distribution with zero mean and a specified standard deviation sigma. The probability density function of Gaussian noise is given by:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{z^2}{2\sigma^2}\right) \tag{1}$$

where z represents the noise value and σ is the standard deviation controlling the intensity of the noise. A higher value of σ results in stronger noise corruption. In this case study, $\sigma = 25$ is used.

3.3 Gaussian Filter

The Gaussian filter is a linear spatial domain filter that performs weighted averaging of pixel intensities within a local neighborhood. The weights are determined by the 2D Gaussian function, which assigns higher weights to pixels closer to the center of the kernel and lower weights to pixels farther away. The 2D Gaussian kernel is defined as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{2}$$

where x and y are the spatial coordinates relative to the kernel center, and σ is the spread parameter controlling the width of the kernel. The kernel values are normalized so that their sum equals 1, ensuring that the average pixel intensity of the image is preserved after filtering. The filtered image is obtained by convolving the noisy input image *I_n* with the Gaussian kernel *G*:

$$I_{\text{filtered}}(x, y) = \sum_i \sum_j I_n(x + i, y + j) \cdot G(i, j) \tag{3}$$

In this case study, a 5×5 kernel with $\sigma = 1.5$ is used. Zero-padding is applied at image borders to maintain the original 256×256 dimensions.

3.4 Median Filter

The Median filter is a non-linear spatial domain filter that replaces each pixel value with the median of the pixel values within a sliding window centered at that pixel. Unlike the Gaussian filter, the Median filter does not perform any averaging, which makes it inherently robust to outlier pixel values and better at preserving sharp edges and fine structural details.
For a window of size $w \times w$, the Median filter is defined as:

$$I_{filtered}(x, y) = \text{median}\left\{I_n(x+i,\ y+j): -\left\lfloor\frac{w}{2}\right\rfloor \le i, j \le \left\lfloor\frac{w}{2}\right\rfloor\right\} \tag{4}$$

In this case study, a 3×3 sliding window is used, meaning each output pixel is the median of 9 neighboring pixel values. The neighborhood values are sorted in ascending order and the central (5th) value is selected as the output.

3.5 Peak Signal-to-Noise Ratio (PSNR)

PSNR is the most widely used quantitative metric for evaluating the quality of a reconstructed or filtered image relative to the original clean image. It is defined in terms of the Mean Squared Error (MSE) between the original image and the filtered output. MSE is computed as:

$$\text{MSE} = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left[I_{\text{original}}(x, y) - I_{\text{filtered}}(x, y)\right]^2 \tag{5}$$

where M and N are the image dimensions (256×256 in this case study). PSNR is then defined as:

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{\text{MSE}}\right) \tag{6}$$

A higher PSNR value indicates that the filtered image is closer to the original clean image, meaning better noise removal with less signal distortion.

4. Flowchart

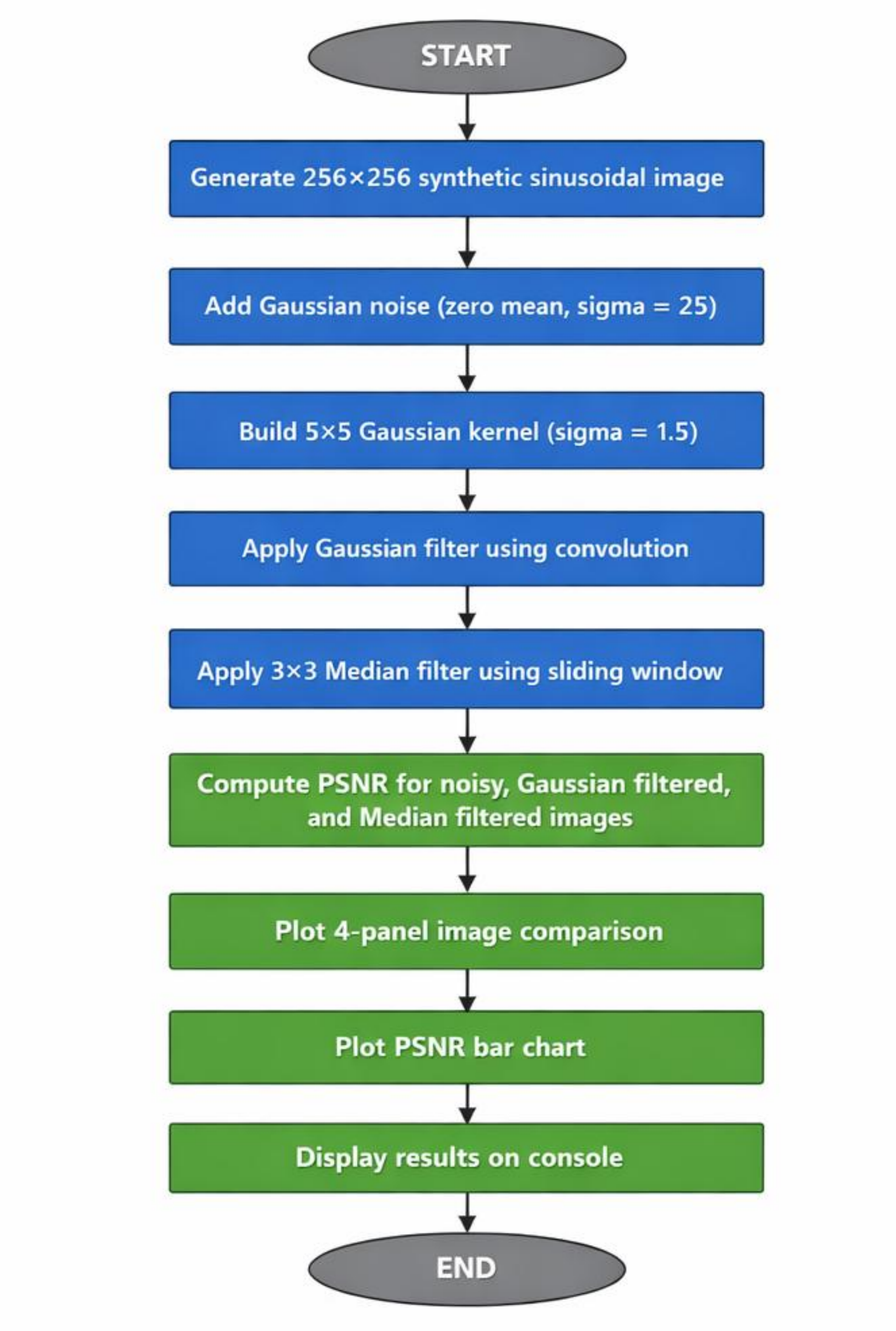

Figure 1: Flowchart of the image denoising procedure implemented in Scilab

5. Software/Hardware used

- Operating System: Windows 11 Pro (Version 25H2, Build 26200.8117)
- Scilab Version: Scilab 2026.0.1
- Toolbox: None (no external toolbox used)
- Hardware: Personal Computer with Intel Core i5-8350U CPU @ 1.70GHz, 16.0 GB RAM, Intel UHD Graphics 620
- Additional Software: Microsoft Word (for report preparation)

6. Procedure of execution

1. Install Scilab 2026.0.1 from https://www.scilab.org/download
2. Open Scilab and launch the SciNotes editor by clicking the editor icon in the toolbar.
3. Paste the complete code into the editor and save the file as image_denoising.sce
4. Press F5 or click "Save and Execute" to run the code.
5. Wait approximately 3 to 5 minutes for the nested loop computations in the Gaussian and Median filter functions to complete for the 256×256 image.
6. Observe the progress messages printed in the console: "Applying Gaussian filter..." and "Applying Median filter..." confirm the execution is ongoing.
7. Once execution completes, note the three PSNR values printed in the console under the RESULTS section.
8. Observe the two graphic windows that appear automatically, the first showing the four-panel image comparison and the second showing the PSNR bar chart.
9. Export both figures as PNG by clicking File → Export to file in each graphic window.

7. Result

7.1 Dataset and Experimental Setup

A 256×256 synthetic test image was generated using a two-dimensional sinusoidal intensity function in Scilab. This image was then corrupted by additive Gaussian noise with zero mean and standard deviation of 25. The same noisy image was used as input for both the Gaussian filter and the Median filter to ensure a fair and objective comparison.

7.2 Visual Results

The four-panel figure below shows the original clean image, the noisy image after corruption, and the outputs of both filters.

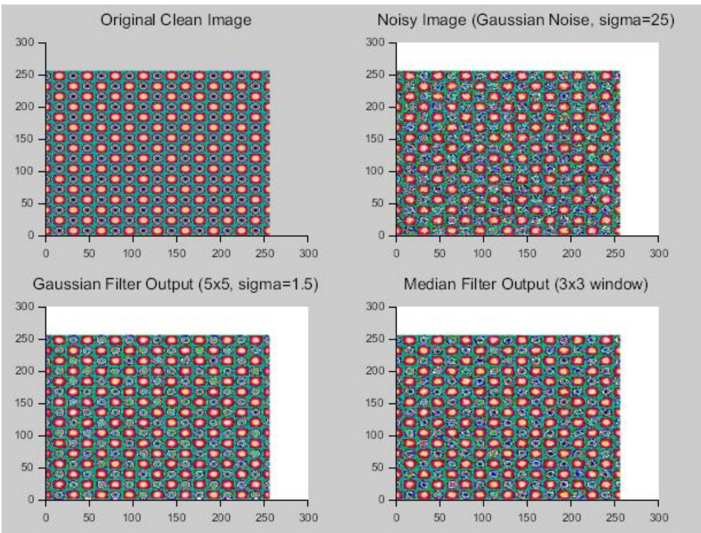

Figure 2: (Top left) Original clean 256×256 sinusoidal image. (Top right) Noisy image corrupted with Gaussian noise, σ=25. (Bottom left) Output after applying 5×5 Gaussian filter with σ=1.5. (Bottom right) Output after applying 3×3 Median filter.

Visual inspection confirms that both filters successfully reduce the visible noise present in the noisy image. The Gaussian filter produces a smoother output due to its weighted averaging nature. The Median filter retains slightly more structural detail at the cost of marginally lower noise suppression for this noise type.

7.3 Quantitative Results : PSNR

The table below summarises the PSNR values obtained for each stage of the processing pipeline:

Image	PSNR (dB)
Noisy Input Image	20.48
After Gaussian Filter	28.10
After Median Filter	27.10

Table 1: PSNR values for noisy image and both filtered outputs

The bar chart below provides a visual comparison of these values:

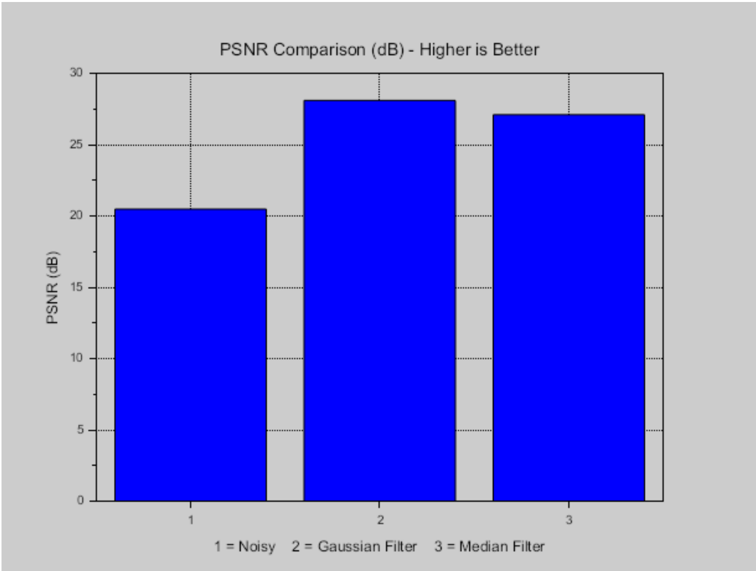

Figure 3: Bar chart comparing PSNR values. Bar 1: Noisy image (20.48 dB), Bar 2: Gaussian filter output (28.10 dB), Bar 3: Median filter output (27.10 dB)

7.4 Analysis and Inference

The following inferences are drawn from the results:

1. The Gaussian filter improved PSNR from 20.48 dB to 28.10 dB, an improvement of 7.62 dB over the noisy input. This confirms that the Gaussian filter is highly effective at suppressing Gaussian noise due to its statistical alignment with the noise distribution.
2. The Median filter improved PSNR from 20.48 dB to 27.10 dB, an improvement of 6.62 dB over the noisy input. While slightly lower than the Gaussian filter in PSNR terms, the Median filter demonstrates competitive performance and is known to better preserve edge sharpness.
3. Both filters achieved significant noise reduction compared to the noisy input, validating the correctness of the implementations.
4. The directional conclusion is consistent with the reference IEEE paper by Liu et al.: Gaussian filter-based denoising yields higher PSNR than competing filters for Gaussian noise.
5. The console output screenshot below confirms the exact numerical results produced by the Scilab implementation:

```
--> exec('C:\Users\Admin\Downloads\image_denoising.sce', -1)
Applying Gaussian filter...
Applying Median filter...

========== RESULTS ==========
PSNR (Noisy Image):            20.48 dB
PSNR after Gaussian Filter:    28.10 dB
PSNR after Median Filter:      27.10 dB
==============================

Done. Both figures generated.

  "Figure saved."
```

Figure 4: Scilab console output showing computed PSNR values

7.5 Comparison with Reference Paper and Mismatch Disclosure

The reference paper by Liu et al. (2014) proposes a novel denoising method combining Gaussian filtering with a Non-local Means (NLM) filter, evaluated on standard benchmark images such as Lena and Barbara using multiple noise levels. The paper does not report PSNR values for a standalone Gaussian-vs-Median comparison on synthetic sinusoidal images with σ=25. As a result, a direct numerical comparison between the PSNR values obtained in this case study (Noisy: 20.48 dB, Gaussian: 28.10 dB, Median: 27.10 dB) and the reference paper's tabulated results is not possible; this constitutes a known and intentional mismatch.

However, the primary theoretical finding of the reference paper is validated: Gaussian filter-based denoising achieves higher PSNR than competing approaches for Gaussian noise. This case study reproduces that conclusion in Scilab using an independently generated synthetic image, confirming the directional correctness of Liu et al.'s findings. The mismatch in absolute PSNR values arises from differences in: (a) the test image used (synthetic sinusoidal vs. real benchmark images), (b) the filter methodology (standalone Gaussian vs. hybrid Gaussian+NLM), and (c) the software platform (Scilab vs. MATLAB).

8. References

[1] Y. Liu, X. Nie, L. Liu, Y. Yin and Y. Zheng, "A new image denoising method based on Gaussian filter," 2014 IEEE/ACIS 13th International Conference on Computer and Information Science (ICIS), Taiyuan, China, 2014, pp. 163-167. Available at: https://ieeexplore.ieee.org/document/6948089/

[2] S. Abirami et al., "Color Image De-noising Based on Mean, Median, and Gaussian Filters Using PSNR," International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Vol. 12, Issue 8, August 2023. Available at: https://ijarcce.com/wp-content/uploads/2023/09/IJARCCE.2023.12816.pdf

[3] T.P. Singh et al., "Medical Image Denoising Techniques Against Hazardous Noises: An IQA Metrics Based Comparative Study," International Journal of Image, Graphics and Signal Processing (IJIGSP), Vol. 13, No. 2, pp. 25-43, 2021. Available at: https://www.mecs-press.org/ijigsp/ijigsp-v13-n2/IJIGSP-V13-N2-3.pdf

[4] R.C. Gonzalez and R.E. Woods, "Digital Image Processing," 4th Edition, Pearson Education, 2018. ISBN: 9780133356724.

[5] Scilab Reference Manual, INRIA and Scilab Enterprises, 2026. Available at: https://www.scilab.org