A Comprehensive Analysis of Various Denoising Methods

Mehdi Mohammadi

Independent Researcher

Email: mmohammadi@alumni.iut.ac.ir

Independent Research, 2025

Abstract

Noise, a major cause of loss of image quality, can be attributed to a variety of factors such as sensor defects, data transmission errors, and file compression. This study aims to analyze multiple types of denoising techniques by using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Edge Preservation Index (EPI). Based on the findings, the Gaussian filter performs well at σ=1 (PSNR=23.58 dB, SSIM=0.5655, EPI=95.4%); however, it fails to preserve details. Linear Diffusion functions the most efficiently at t=1 (PSNR=22.78 dB, SSIM=0.5991, EPI=91.0%), but it has similar limitations. Increasing the σ in the Gaussian filter or the time parameter in the Linear filter lower the performance efficiency and lead to over-smoothing and the loss of edge details. The Perona-Malik diffusion method shows better results at λ=5 (PSNR=23.57 dB, SSIM=0.5778, EPI=78.7%) and λ=10 (PSNR=23.54 dB, SSIM=0.6366, EPI=68.7%). Though the EPI is lower, the Perona-Malik diffusion method produces sharper edges with higher contrast. According to the results, the numerical values of EPI alone cannot reflect the output quality of the methods accurately. The adaptive Perona-Malik diffusion can denoise and preserve high-quality edges, and compared to linear methods, it preserves structural similarity and image quality more efficiently.

Keywords: Image denoising; Gaussian filter; Linear diffusion model; Perona–Malik method; Noise intensity; Adaptive parameter

1. Introduction

The primary goal of image processing is to enhance the image quality in order to improve the perception of the image data by both humans and machines [1]. Image processing can be defined as the digitization of images and administration of certain procedures on it, in order to extract useful information with the aim of reaching a defined goal. The field of image processing has various applications in the medical field, security systems, military operations, astronomy, aerospace sciences, and in other academic, scientific, and industrial technologies [1].

Damaged images can be a result of varying factors such as the limitations of imaging equipment, environmental conditions, and the equipment technician’s lack of skills. The most common image damages, such as noise, blurring, and low contrast can reduce the image quality and cause disruptions in the humans’ eyes or the machines’ processing of the image [2].

Noise is defined as the accidental and unwanted changes that the pixels of an image undergo [1]. The presence of noise not only reduces the image quality, but it also leads to the loss of image data and details, causing undesirable results such as false edges, image artifacts, edge distortions, and image blurring. Denoising is crucial for enhancing the quality of an image and it is considered as one of the vital steps of pre-processing for many computer vision methods [2].

There are various types of noises that can be categorized into two groups of additive noise and multiplicative noise [3]. Additive noise refers to the type of noise that is added to the main image data and it includes the Gaussian noise, uniform noise, and shot noise. Gaussian noise, characterized by normal distribution, is often seen in natural images. Shot noise includes fixed pattern noise (salt-and-pepper) and random noise; this type of noise replaces the original pixels of the image with noise. Speckle noise is a type of multiplicative noise, which signifies that it varies and is multiplied by the original image pixel values. This kind of noise is often found in medical ultrasound images and satellite pictures [1, 3].

Various methods have been developed for denoising images in the spatial-frequency domain [2]. Linear Filters such as the Gaussian filter, linear diffusion models, and the Wiener Filter work by combining the neighboring pixels in a linear manner; however, this usually leads to the loss of crucial details and edges. The non-linear filters such as the Median filter, non-linear diffusion models (such as the Perona-Malik method), and adaptive methods were developed for better preservation of important structures of an image during the denoising process [1, 2].

Partial Differential Equations (PDEs)-Based Filters, such as Linear Diffusion, and the Perona-Malik diffusion, are among the more advanced methods designed for modeling the denoising process with crucial detail preservation [2]. The Linear Diffusion method aims to solve the heat diffusion function using a homogeneous diffusion coefficient; from a mathematical perspective, applying the Gaussian filter with a specific standard deviation is equivalent to the Linear Diffusion process over a corresponding time interval [1]. The Perona-Malik diffusion makes use of a gradient-based coefficient value, which controls the diffusion process in an adaptive manner; therefore, it enables strong smoothing in homogeneous parts while limiting it near the image edges [2].

One of the most important aspects of denoising is balancing the side-effects such as blurring or displacement of the edges [3]. The visual system of humans is more sensitive to the existence of noise in smooth areas of an image than the non-smooth areas. Additionally, noise has various effects on different bit-planes of an image; the most significant impact is on the bit-planes with the lowest values [1].

Achieving an efficient balance between denoising and edge and detail preservation is vital [2]. In order to conduct a comprehensive comparison, it is crucial to do a precise analysis of denoising methods using qualitative metrics, such as the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Edge Preservation Index (EPI) [3].

This study aims to investigate and compare the different functions of various denoising methods such as the Gaussian filter and PDE-based models (Linear Diffusion and Perona-Malik Diffusion) through the SSIM, PSNR, and EPI metrics. In this paper, the impact of various parameters was examined on the quality of re-construction and edge preservation in order to determine an efficient balance between denoising and detail preservation.

This paper is comprised of six segments. The first segment introduces image denoising. The previous research in the field of denoising is reviewed in the second part. In the third section, the materials and methods that were utilized for assessing denoising algorithms are described. The fourth segment goes into more details regarding the design and the application of the experiment. The results are analyzed and the various methods are compared in the fifth part. And finally, the conclusion and the recommendations for future research are provided in the sixth section.

2. Literature Review

2.1. Linear Filters

Denoising techniques are generally categorized into linear and non-linear filters. Linear filters work by combining neighboring pixels, and they often use functions such as convolution or correlation to generate the output image. Even though such methods are efficient in terms of computations, they often cause the loss of image details and edges in a real-world application.

2.1.1. Mean Filtering

Mean filtering is among the simplest linear methods of denoising; it reduces local variance in an image by replacing the central pixel value with the mean value of its neighboring values. This filter functions as a low-pass smoothing filter, suppressing the high-frequency components like the edges and producing a smooth but blurry image. Increasing the window size can improve the smoothing process; however, it significantly reduces the details of the image [4].

In order to decrease the blurring effect of the smoothing technique in mean filtering, a weighted system was introduced that assigns higher weights to central pixels, causing them to be more sensitive to the neighboring pixels. Compared to its simpler version, this filter can preserve details more efficiently under mild noise conditions; however, the window size still causes a reduction in the adaptability of edges and leads to significant blurring [5].

2.1.2. The Gaussian Smoothing Filter

The Gaussian Smoothing Filter, which is categorized as a linear method, removes the Gaussian noise efficiently by assigning weights to the neighboring pixels according to a normal 2-D distribution. In comparison with the mean filtering, this filter is more persistent in preserving the structure of the image; however, an increase in the window size and standard deviation (σ) causes more significant blurring of the edges. Previous research in the field [6] recommends this filter to serve as a standard for designing more advanced methods, as its efficiency in reducing Gaussian noise has been investigated thoroughly.

2.1.3. The Wiener Filtering

The Wiener filtering is a statistical linear filter for denoising, designed to minimize the Mean Square Error (MSE) between the noisy and the original image. This filter estimates the pixel values for noise reduction while preserving the original structure; it does so by making use of the local mean and variance of pixels. Although increasing the window size can lead to a greater amount of noise reduction, it will also cause sharpness loss and more pronounced blurring.

In this method, the adaptive Wiener filter is applied to the image’s bit-planes separately, employing the local mean and variance of each window to remove the Gaussian noise. According to the studies conducted in the field, this method functions more efficiently in regards to preserving the image structure and provides improved PSNR and RMSE values. Nevertheless, its fixed window size limits its adaptability to variable noise intensities in complex conditions [7].

2.2. Non-linear Filters

In spite of the simplicity and computational functionality of linear filters, their inherent weakness in preserving crucial image structures, especially edges, has led to the development of non-linear methods. Non-Linear methods utilize the statistical and structural properties of the image to better preserve the details while denoising; and as a result, they provide a more efficient and adaptive functionality in comparison with linear methods.

Non-linear filters, such as Median, Maximum, and Minimum filters, reduce the noise of an image by altering the image structure according to the local properties adaptively. The primary non-linear spatial filters include median, non-linear diffusion, and advanced adaptive filters. These filters and the studies related to denoising will be further discussed in the following segments.

2.2.1. Median Filtering

The Median filter is a frequently-used, non-linear, low-pass method and it removes the Salt-and-Pepper Noise efficiently. This filter replaces the pixel values with the median value of neighboring pixels and therefore, it eliminates extreme noise values (outliers) while preserving the original image data. However, uniform application of this filter across all parts of an image can displace undamaged pixels, particularly those at the edges. Adaptive versions of the filter that adjust the window size based on noise density aim to omit this limitation; however, the filter lacks the required precision for handling continuous noises like the Gaussian noise [8].

2.2.2. Modified Median Filters

In this hierarchical method, the median filter is applied gradually, starting with the most significant and then the least significant bit-planes, in order to selectively remove salt-and-pepper noise. Median selection at higher levels leads to the localization of the filter in the subsequent levels, and therefore, preventing the creation of artificial values. The main advantage of this filter lies in its ability to preserve undamaged pixels in the case of false identification of noise. Nonetheless, the limitations of the filter, such as decreased accuracy in the case of dense noise and the lack of adaptation to varying noise levels, reduce the efficiency of its performance in complex conditions [9].

1. The Modified Decision-Based Unsymmetrical Trimmed Median Filtering

This method employs Modified Decision-Based Unsymmetrical Trimmed Median Filtering (MDBUTMF) in order to reduce and remove salt-and-pepper noise while preserving the undamaged pixels. If a pixel value falls within the non-noise range, it remains unchanged; otherwise, it is replaced with the mean or the median according to the neighboring pixels pattern. This approach reduces the replacement errors and better preserves the image texture in the case of sparse noise. Nevertheless, at high noise densities, the fixed 3×3 window size reduces image sharpness and causes excessive smoothing [10].

2. Adaptive MDBUTMF

In this method, an enhanced version of the MDBUTMF method utilizes an adaptive approach according to the ratio and the percentage of noisy pixels in the window. Through adjusting the window size between 3×3 and 5×5, according to the local noise density, the filter obtains more efficient adaptability to image structures. The conditional use of the mean or median leads to the reduction of replacement errors and the filter functions effectively in the case of sparse noise. However, the filter's resistance to adapting to intense noise causes weak edges and extreme smoothness [11].

2.3. Diffusion-Based Filters

Diffusion-based filters rely on Partial Differential Equations (PDEs) for modeling the process of denoising while preserving crucial structures of the image. These methods are particularly efficient regarding images with high levels of noise and for preserving the edge details.

2.3.1. Linear Diffusion

Linear diffusion is one of the simplest PDE-based models for denoising, which functions by solving the heat diffusion equation with a homogeneous diffusion coefficient. This method uses the flow of intensity values across the entire image as a guideline for smoothing out the variance caused by the noise. The linear diffusion process is mathematically equivalent to applying a Gaussian smoothing filter with a time-dependent standard deviation. Even though this filter reduces noise by smoothing the image uniformly, it also weakens the edges and causes the loss of fine details. Previous research [12] highlights that that the model's reliance on a fixed diffusion coefficient turns it into a fundamental standard in evaluating more advanced and adaptive methods.

2.3.2. Non-linear Diffusion (Perona-Malik Model)

The Perona-Malik model is a non-linear PDE-based method that aims to improve linear diffusion by making use of a gradient-dependent diffusion coefficient. This coefficient value controls the diffusion process in an adaptive manner; allowing strong smoothing in homogeneous parts while limiting it near the image edges. This enables the filter to denoise the image without damaging the significant components of the image such as the edges and the textures.

Non-linear variations of the diffusion coefficient - such as inverse square or exponential functions - allow a better control over the balance between denoising and detail preservation. Some studies [13] emphasize the significance of precise adjustment of the threshold parameter (λ) — which helps the filter identify the edges — for achieving optimal performance. Improper adjustment of this parameter may result in excessive smoothing or residual noise. In order to enhance computational stability, the use of smoothed gradients and multi-directional application of the filter is recommended.

2.4. Hybrid Methods

2.4.1. Bit-Plane Analysis

To enhance the accuracy of the denoising process and the adaptation with the local features, some researchers have used the Bit-Plane analysis and filtering method.

In this study, a more efficient method of removing Gaussian noise in images with high contrast levels is proposed. This method functions through dividing the image into bit-planes and then, applying a moving average filter to every plane separately. Compared to the classic Mean filtering, this method enhances the RMSE and PSNR parameters. Nevertheless, this method also suffers from some limitations; for instance, the uniform application of the filter across all planes and using a fixed window size prevent this method from being adaptable to various noise intensities [14].

2.4.2 The Boolean Quantum Mean Filter

The Boolean Quantum Mean Filter performs the denoising process through focusing on bit-planes with high informational value. By filtering each channel separately and focusing on the dominant band, this filter reduces the computational complexity and removes the salt-and-pepper noise more effectively. However, the removal accuracy of this method in high-noise conditions is affected by the mask size being limited to the 3×3 size and the filter's dependency on image brightness [15].

2.4.3. The Gaussian and PCA Combination

In this hybrid algorithm, the Gaussian filter, the Principal Component Analysis (PCA), and non-linear diffusion PDE models are merged together in order to remove low-density speckle noise in ultrasound images. The Gaussian filter serves as the initial step for smoothing, which is followed by the PCA to enhance the other components. Non-linear diffusion is then applied to the image, and the impact of the iteration count on the quality of the output is analyzed. Results from quantitative studies verify the superiority of this method over wavelet thresholding techniques, although reduced accuracy at high noise levels remains a limitation [16].

2.4.4. The Homomorphic and Diffusion Filtering

In this hybrid method, the components of illumination and reflectance are decomposed by a homomorphic filter and then, diffusion filtering is applied to each component separately to reduce the speckle noise in ultrasound images. In this approach, using quadrilateral windows and modified non-linear diffusion coefficients leads to an increase in the accuracy of structural correction. However, manual adjustment of parameters and pre-existing limitations in diffusion directions influence the functionality of this method under severe noise conditions [17].

2.4.5. The Spatial and Frequency Domain Combination

In this method, linear and non-linear filters are combined in the spatial-frequency domain to identify and remove impulsive noise gradually and step-by-step. This method first detects the suspicious pixels; then, either the median or mean filtering is applied to the image according to the noise density, followed by transforming the image into the frequency domain, where a Gaussian low-pass filter is used for smoothing and a high-pass filter combined with an edge detection algorithm is applied for texture reconstruction. This multi-step approach preserves details efficiently under various noise levels, but an increased runtime and computational complexity are significant limitations of the method [18].

3. Materials and Methods

3.1. Classification of Denoising Algorithms

Image enhancement techniques can be generally categorized into two primary types of approaches [1]. Linear methods function by applying certain mathematical operations that add up the pixel values from neighboring regions using pre-determined weights. Non-linear techniques make use of more sophisticated algorithms that adapt their function based on local image properties from different parts of the image [19].

3.2. Implementation of the Gaussian Smoothing Method

The Gaussian filter is one of the most popular linear smoothing methods [1]. It is based on the bell-shaped Gaussian distribution commonly used in statistics. The filter kernel is defined by Equation (1):

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

In this equation, σ controls the amount of smoothing; smaller the value, the more preserved the fine details, and the larger the values, the higher the amount of smoothing [1]. When implemented on a computer, this bell-shaped curve is sampled within a square grid of size $(2k+1)\times(2k+1)$, where $k=\lceil 3\sigma \rceil$ captures most of the distribution weight.

The actual filter functions based on Equation (2):

$$I^{'}(i,j) = \frac{\sum_{m=-k}^{k}\sum_{n=-k}^{k} G\,(m,n,\sigma) \bullet I(i+m,j+n)}{\sum_{m=-k}^{k}\sum_{n=-k}^{k} G\,(m,n,\sigma)} \tag{2}$$

In this equation, the sums are computed from $-k$ to $+k$ in both directions over the kernel window, and the denominator is the normalization factor that ensures all kernel weights add up to one [1]. This prevents the output values from exceeding normal pixel range of 0-255.

One of the advantages of the Gaussian filter is that can be divided into two simpler operations —a horizontal and a vertical operation — helping to accelerate the mathematical operation by simplifying O(σ^2) to O(σ) [1].

3.3. Heat-Based Diffusion Processing

Linear diffusion is used to remove noise, and its function is comparable to heat diffusion in materials [20]. The image changes over time following Equation (3):

$$\frac{\partial u}{\partial t} = D\nabla^2 u \tag{3}$$

In this equation, $u(x, y, t)$ illustrates the image at the specific time identified by t; D determines the pace of diffusion, and ∇^2 is the Laplacian operator that measures the difference between a specific pixel and its neighboring pixels [20]. This process resembles the heat flowing from regions of high intensity to regions of low intensity, and in the case of images, it takes place from bright areas to dark areas.

For computer calculations, the Laplacian value is estimated using the nearby pixels according to Equation (4):

$$\nabla^2 u(i,j) = u(i-1,j) + u(i+1,j) + u(i,j-1) + u(i,j+1) - 4u(i,j) \tag{4}$$

The image is then updated step by step using Equation (5):

$$u^{n+1}(i,j) = u^n(i,j) + dt \times D \times \nabla^2 u^n(i,j) \tag{5}$$

In Equation (5), *dt* is the time step size, *D* determines the diffusion intensity, and the total evolution time is *t=n.dt*, where n is the iteration count [20].

To keep the calculations stable, it is necessary to ensure that the condition in Equation (6) is fulfilled:

$$dt \leq \frac{1}{4D} \tag{6}$$

This prevents the numerical solution from becoming unstable and producing faulty results [20].

3.4. Adaptive Perona-Malik Processing

The Adaptive Perona-Malik method improves upon the regular diffusion by being selective and smarter regarding its application of smoothing process and where it is applied [13]. This method follows Equation (7):

$$\frac{\partial u}{\partial t} = \nabla \cdot (D(|\nabla u|)\nabla u) \tag{7}$$

The key to innovation in this method is the diffusion coefficient *D*, which changes based on the local steepness levels of the image gradients [13]. This enables the algorithm to efficiently reduce noise in homogeneous regions while preserving the edges. The diffusion function is displayed in Equation (8):

$$D(s) = \frac{1}{1 + \left(\frac{s}{\lambda}\right)^2} \tag{8}$$

In this equation, $s=|\nabla u|$ represents the gradient magnitude, and λ sets the threshold for what is considered as an edge [13]. In smooth regions where gradients are small, *D* is close to 1, and diffusion takes place normally. Near the edges where gradients are large, *D* gets close to 0 and diffusion halts.

Gradients are calculated using the central differences according to Equation (9):

$$\nabla u(i,j) = \left[\frac{u(i,j+1) - u(i,j-1)}{2}, \frac{u(i+1,j) - u(i-1,j)}{2}\right] \tag{9}$$

The λ parameter is a critical element of the equation; smaller values preserve more details; bigger values lead to higher levels of smoothing [13].

3.5. The Relationship Between Diffusion and The Gaussian Filtering

A significant relationship emerges between diffusion and the Gaussian Filter when one attempts to solve the diffusion equation mathematically [21, 22]. Starting with Equation (10):

$$\frac{\partial u}{\partial t} = D\nabla^2 u \tag{10}$$

To solve Equation 10 with the initial condition $u(x, y, 0)=I(x, y)$, the Fourier Transform is employed. By applying the two-dimensional Fourier transform $\mathcal{F}\{u(x, y, t)\}=\hat{U}(\omega_x, \omega_y, t)$ to Equation (10), the frequency domain equation in Equation (11) is obtained:

$$\frac{d\hat{U}}{dt} = -D|\omega|^2\hat{U} \tag{11}$$

In this equation, $|\omega|^2$ is the squared magnitude of the frequency vector. The result of this first-order differential equation is shown in Equation (12):

$$\hat{U}(\omega_x, \omega_y, t) = \hat{U}(\omega_x, \omega_y, 0)\, exp(-D|\omega|^2 t) \tag{12}$$

Obtaining the Inverse Fourier Transform leads to Equation 13:

$$u(x, y, t) = \mathcal{F}^{-1}\{\hat{U}(\omega_x, \omega_y, 0)\, exp(-D|\omega|^2 t)\} \tag{13}$$

The component exp($-D|\omega|^2 t$) in this formula is equivalent to the Gaussian kernel calculated with $\sigma^2=2Dt$ [21]. This finally leads to the complete solution in Equation (14):

$$u(x, y, t) = I(x, y) * G(x, y, \sqrt{2Dt}) \tag{14}$$

This process demonstrates that applying linear diffusion for time (t) is exactly equal to applying a Gaussian filter with $\sigma=\sqrt{(2Dt)}$, as it is shown in Equation (15):

$$\text{Linear Diffusion } (D, t) \equiv \text{Gaussian Filter } (\sigma = \sqrt{2Dt}) \tag{15}$$

This equation helps convert the parameters between the two methods and it shows how they compare theoretically [22].

3.6. Quality Assessment Methods

3.6.1. Peak Signal-to-Noise Ratio (PSNR)

PSNR is the most common method of assessing the quality of denoised images. The ratio is calculated using Equation (16):

$$\text{PSNR} = 20\, log_{10}\left(\frac{\text{MAX}}{\sqrt{\text{MSE}}}\right) \tag{16}$$

For regular 8-bit images, the maximum possible pixel value is displayed by MAX=255. The MSE (Mean Squared Error) is calculated through Equation (17):

$$\text{MSE} = \frac{1}{M \cdot N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I(i,j) - I'(i,j)\right]^2 \tag{17}$$

In this equation, $M \times N$ represents the image dimensions, I is the original image, and I' is the processed image. The higher the PSNR score, the higher the quality of the image.

A score over 30 dB is typically considered acceptable; however, PSNR does not always correlate with the human visual perception of image quality [1].

3.6.2. Structural Similarity Index Measure (SSIM)

SSIM corresponds to the human visual perception better than PSNR [19]. It is estimated by examining the brightness, contrast, and structure of an image all together. SSIM is calculated using Equation (18):

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{18}$$

The μ_x and μ_y parameters stand for the local averages, σ_x^2 and σ_y^2 represent the local variances, and σ_{xy} refers to the local covariance; all parameters are computed using a sliding Gaussian window. Stabilizing constants are obtained using Equations (19) and (20):

$$C_1 = (K_1 \cdot L)^2 \tag{19}$$

$$C_2 = (K_2 \cdot L)^2 \tag{20}$$

In these equations, L represents the dynamic range of pixel values (typically 255), and K_1=0.01, K_2=0.03 are the default coefficient values. The SSIM score ranges from -1 to 1, with 1 representing perfect similarity [19].

3.6.3. Edge Preservation Index (EPI)

The EPI demonstrates the amount of edge preservation during denoising, which is a crucial factor for computer vision applications [23, 24]. Edge strength is calculated using the Sobel operator from Equations (21) and (22):

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{21}$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{22}$$

Then, edge strength is computed using Equation (23):

$$|\nabla I| = \sqrt{G_x^2 + G_y^2} \tag{23}$$

After thresholding in order to create binary edge maps, EPI is calculated through Equation (24):

$$\mathrm{EPI} = \frac{\sum_{i,j} E_{\mathrm{ref}}(i,j) \wedge E_{\mathrm{rec}}(i,j)}{\sum_{i,j} E_{\mathrm{ref}}(i,j)} \times 100\% \tag{24}$$

In this equation, E_{ref} and E_{rec} represent the binary values (0 or 1) in the edge masks of the original image and the reconstructed image, respectively. The numerator calculates the number of the pixels that constitute the edges in both images, while the denominator counts the edge pixels in the original image [24]. Any score above established thresholds usually suggests effective edge preservation, making this metric particularly valuable for evaluating edge-preserving methods like the Perona-Malik method.

4. Experimental Design

4.1. Experimental Structure

In order to evaluate the performance of various denoising methods, experiments were conducted on the 'office' image, which was prepared in two versions. The first version, a clean image, served as the standard for evaluation, while the second version, a noisy image, was used as the input for the algorithms. All processing was performed in grayscale with 32-bit floating-point precision to ensure high computational accuracy.

4.2. Algorithm Implementation

The two-dimensional Gaussian kernel was calculated using the 3σ rule to determine the window size, which takes 99.7% of the Gaussian distribution's energy into account. Convolution was implemented using reflective padding to manage boundaries, preventing artifacts at the edges of the image. The linear diffusion equation demonstrated in Equation (3) was solved using the explicit Euler method and finite differences, while the Laplacian was computed according to Equation (4). A time step of dt=0.2 was chosen to ensure numerical stability, as determined by the stability condition in Equation (6). The non-linear Perona-Malik diffusion equation, as demonstrated in Equation (7), was implemented using the permeability function in Equation (8). The divergence was computed using the interpolation of permeability at half-grid points to maintain numerical stability.

4.3. Design of Single-Parameter Experiments

The first experiment aimed to examine the effects of the σ parameter in the Gaussian filter. Various standard deviation values σ={0.5, 1, 2, 5, 10} were applied to the noisy image to investigate the impact of smoothing intensity on the quality of denoising and edge preservation. The second experiment analyzed the temporal evolution of linear diffusion. The diffusion coefficient was fixed at D=1, and the effects of evolution time t={1, 5, 10, 30, 100} was studied on the efficiency of noise removal. The third experiment compared linear diffusion coefficients, through examining the impact of different diffusion coefficients D={1, 5, 10} on smoothing intensity at a fixed time t=10. The fourth experiment evaluated the temporal evolution of Perona-Malik diffusion method — with the edge threshold parameter fixed at λ=0.5 — by examining the effects of evolution time t={1, 5, 10, 30, 100} on edge preservation and denoising. The fifth experiment investigated the effects of the edge threshold parameter, by assessing the impacts of the contrast parameter λ={0.5, 1, 2, 5, 10} on the balance between noise removal and edge preservation at a fixed time t=10.

4.4. Comparative Experiments Between Algorithms

The sixth experiment verified that the Gaussian filter and linear diffusion are theoretically equivalent; therefore, validating the theoretical relationship in Equation (15) by comparing the results of this empirical research. In our experiment, for time t=20 and diffusion coefficient D=1, the Gaussian filter with $\sigma=\sqrt{(2\times1\times20)}=\sqrt{40}\approx6.32$ was compared with the linear diffusion to assess the accuracy of the theoretical relationship. The seventh experiment compared the efficiency of edge preservation between the linear and non-linear diffusion methods under identical temporal conditions (t=10), and analyzed the capability of the two methods in preserving the significant structural features. Such comparisons enabled a better understanding of the strengths and weaknesses of each method, and revealed that non-linear diffusion is more efficient in edge preservation.

4.5. The Analysis of Permeability of the Perona-Malik Model

A special experiment was designed to visualize the spatial distribution of permeability in the Perona-Malik model, where the permeability function - defined as D in Equation (8) - was calculated for the clean image and displayed as a grayscale map.

This analysis illustrated that the algorithm generates high permeability in smooth areas and low permeability near edges; therefore, aiding in structural preservation.

4.6. Boundary Conditions and Numerical Parameters

For all diffusion equations, homogeneous Neumann boundary conditions ($\partial u/\partial n=0$) were applied using reflective padding, which prevents artificial variations near the boundaries and preserves the total energy of the image. The time step $dt=0.2$ was selected based on the stability condition in Equation (6), and 64-bit floating-point precision was used for intermediate diffusion calculations to ensure high numerical accuracy. The explicit Euler integration method was selected due to its simplicity of implementation and stability control, as presented in Equation (5).

4.7. Quality Assessment and Metrics

To evaluate the performance of the algorithms, three main metrics were employed: PSNR, as defined in Equation (16), SSIM, as specified in Equation (18), and EPI, as outlined in Equation (24). These metrics assess various aspects of the image quality, enabling a comprehensive comparison of the methods. PSNR measures the overall quality of reconstruction, SSIM assesses structural similarity, and EPI quantifies the amount of edge preservation.

4.8. Implementation Environment and Reproducibility

All algorithms were implemented in Python using NumPy, OpenCV, and Matplotlib libraries, with the code designed to facilitate the reproducibility of results and ease of parameter adjustments. Each function was documented with detailed comments to enable other researchers to reproduce the results and adapt the algorithms for their purposes.

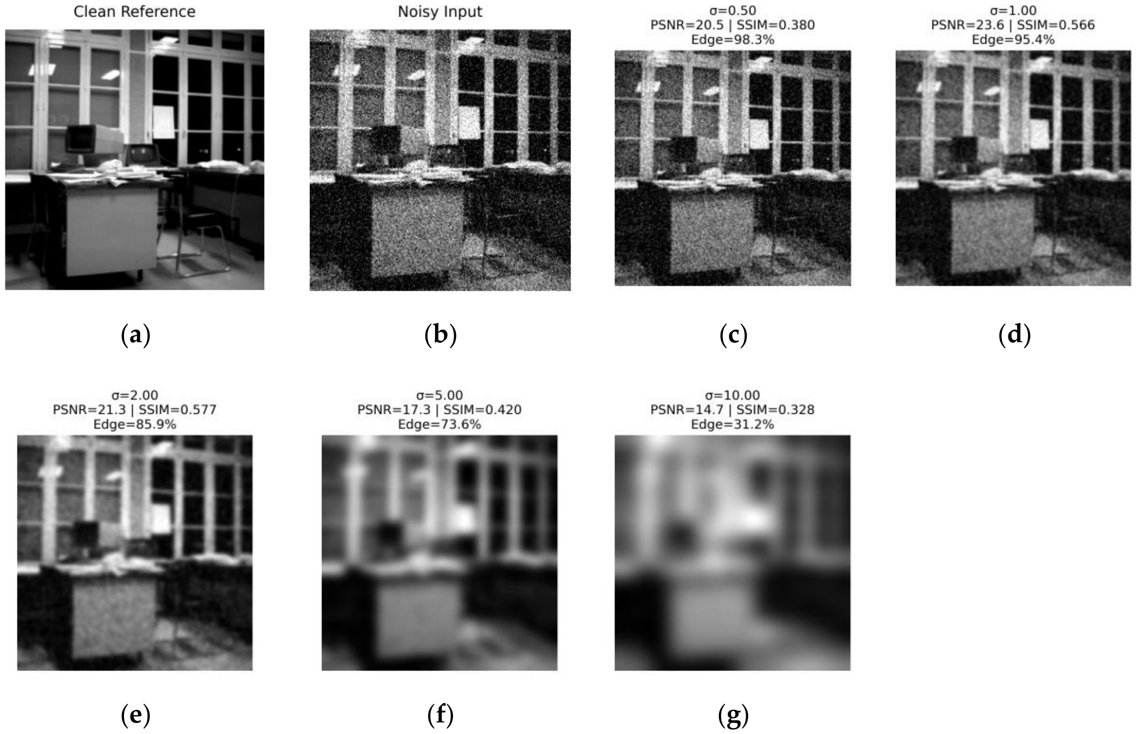
5. Results

5.1. The Analysis of the Gaussian Filter

The results presented in Table 1 demonstrate that the Gaussian filter functions differently based on values of the parameter σ, and produces varying results. The highest PSNR score of 23.58 dB is obtained at $\sigma=1$, the best SSIM score of 0.5775 is obtained at $\sigma=2$, and the maximum edge preservation of 98.3% occurs at $\sigma=0.5$. Figure 1 displays the relationship between denoising and edge preservation: by increasing the σ value, more amounts of noise is removed; however, the edges are blurred further. Additionally, the graphs in Figures 2a to 2c confirm that edge preservation decreases from 98.3% at $\sigma=0.5$ to 31.2% at $\sigma=10$, while the denoising process improves. Due to the significance of maintaining a balance between noise removal and edge preservation and overall image quality, $\sigma=1$ is identified as the optimal choice, as it offers a suitable compromise between effective noise reduction, a high PSNR, and acceptable edge preservation of 95.4%.

Table 1. Performance Quality of the Gaussian filter with varying amounts of standard deviation (σ).

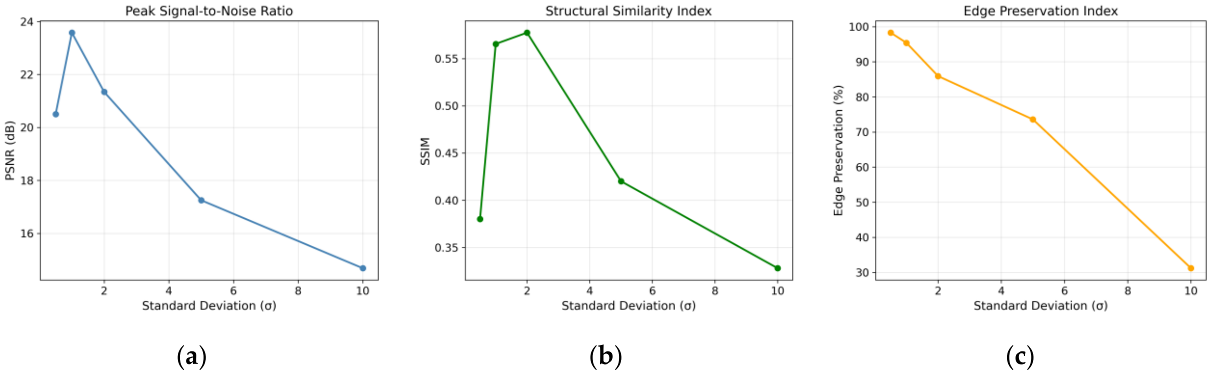
σ	PSNR (dB)	SSIM	Edge Preservation Index (%)
0.5	20.50	0.3801	98.3
1	23.58	0.5655	95.4
2	21.34	0.5775	85.9
5	17.25	0.4201	73.6
10	14.68	0.3280	31.2



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**)

Figure 1. Visual Comparison of Gaussian Smoothing Results across Various σ Values.



(**a**) (**b**) (**c**)

Figure 2. Assessment Criteria of the Gaussian Filter: (a) PSNR, (b) SSIM, (c) Edge Preservation Index (EPI) vs. σ.

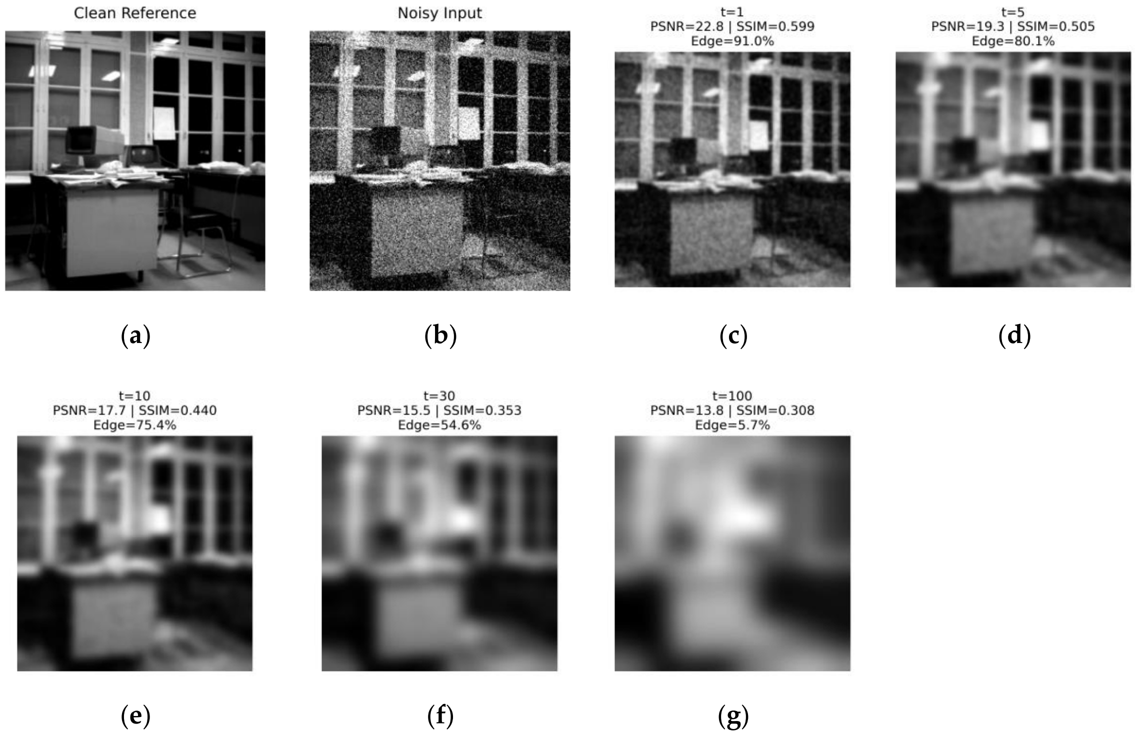
5.2. Linear Diffusion Performance Analysis

5.2.1. Time Evolution Analysis

The time evolution analysis was conducted with a fixed diffusion coefficient value of D=1; as presented in Table 2, the results reveal that linear diffusion performs poorly at longer time periods. Although longer diffusion time initially improves noise removal, it rapidly leads to excessive smoothing. Figure 3 clearly illustrates the gradual degradation of the image; the edge blurring becomes increasingly evident as both noise and important structural details are removed uniformly. The graphs in Figures 4a to 4c depict a sharp decline across all metrics: PSNR decreases from 22.78 dB to 13.82 dB, SSIM drops from 0.5991 to 0.3076, and edge preservation declines from 91.0% to 5.7%. These results indicate significant structural degradation of important image features as the diffusion time increases, demonstrating that linear diffusion fails to maintain the critical balance between noise removal and detail preservation.

Table 2. Linear Diffusion Analysis with Variable Evolution Time (*D*=1).

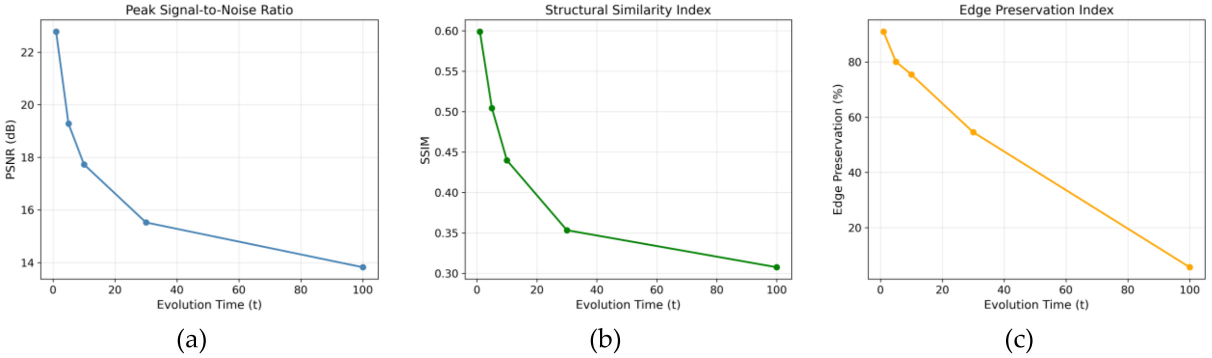
t	PSNR (dB)	SSIM	Edge Preservation Index (%)
1	22.78	0.5991	91.0
5	19.28	0.5046	80.1
10	17.73	0.4399	75.4
30	15.52	0.3534	54.6
100	13.82	0.3076	5.7



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**)

Figure 3. Visual Evolution of Linear Diffusion through Time (*D*=1).



(a) (b) (c)

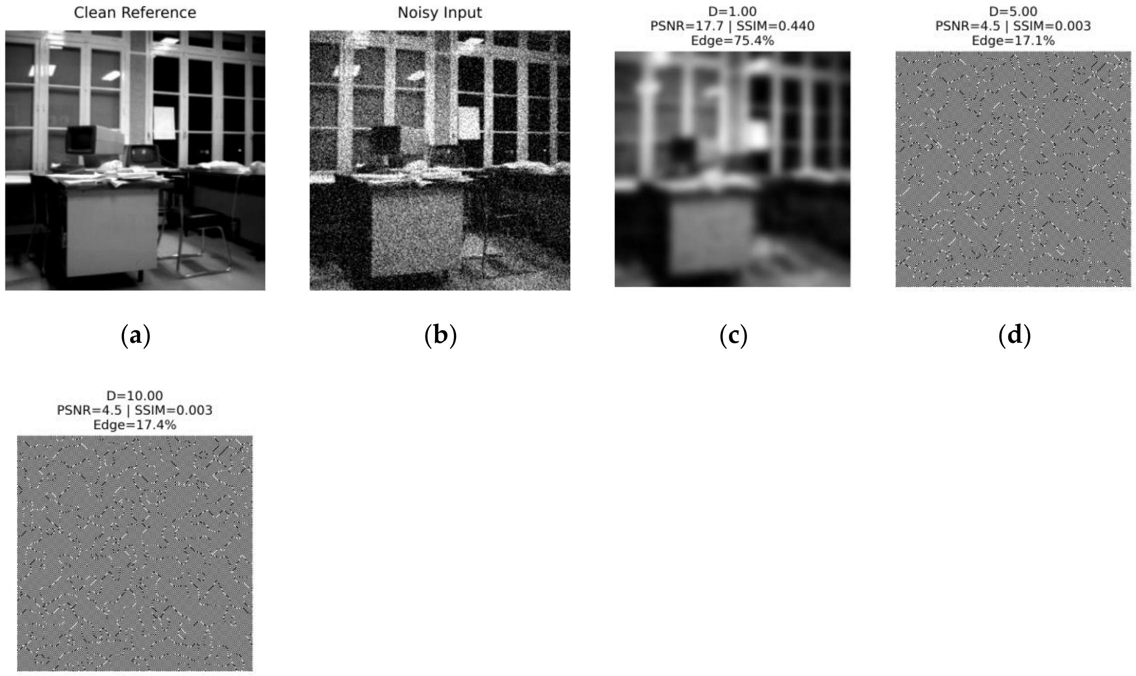
Figure 4. Analysis of Linear Diffusion Time (a) PSNR, (b) SSIM, (c) Edge Preservation Index (EPI) vs. Time.

5.2.2. The Effect of Diffusion Coefficient (*D*)

The results of the comparative experiment on the diffusion coefficient values at a fixed time step *t*=10, displayed in Table 3 and *Figure* 5, show that a rise in *D* causes a total breakdown of the function. While *D*=1 achieves a PSNR value of 17.73 dB, an SSIM value of 0.4399, and an edge preservation index of 75.4%, increasing the *D* value leads to dramatic reduction in all performance metrics. The graphs in Figures 6a to 6c illustrate a sharp decline across all metrics in real-time: Consequently, the *D*=1 and *t*=10 values are identified as the optimal parameters for linear diffusion.

Table 3. Linear Diffusion Performance with a Variable Diffusion Coefficient (*t*=10).

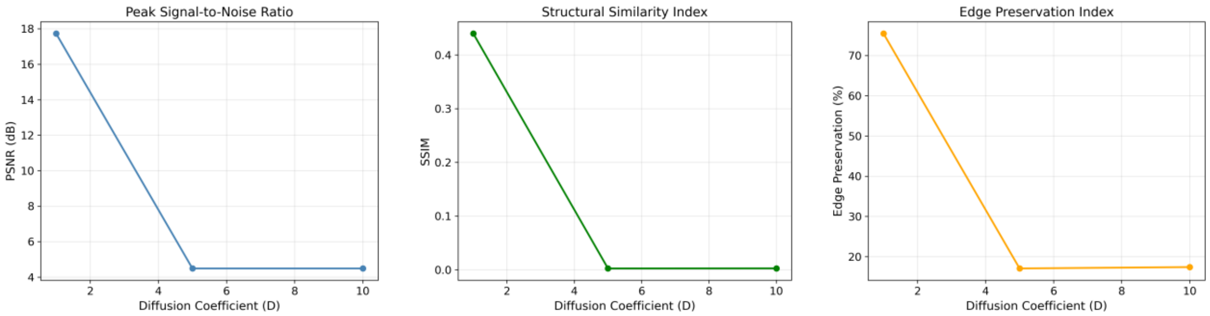
D	PSNR (dB)	SSIM	Edge Preservation Index (%)
1	17.73	0.4399	75.4
5	4.48	0.0025	17.1
10	4.48	0.0027	17.4



(**a**) (**b**) (**c**) (**d**)

(**e**)

Figure 5. Effects of Diffusion Coefficient on Linear Diffusion Results (*t*=10).



(**a**) (**b**) (**c**)

Figure 6. Analysis of Linear Diffusion Coefficient: (a) PSNR, (b) SSIM, (c) Edge Preservation Index (EPI) vs. *D*.

5.3. Perona-Malik Diffusion Analysis

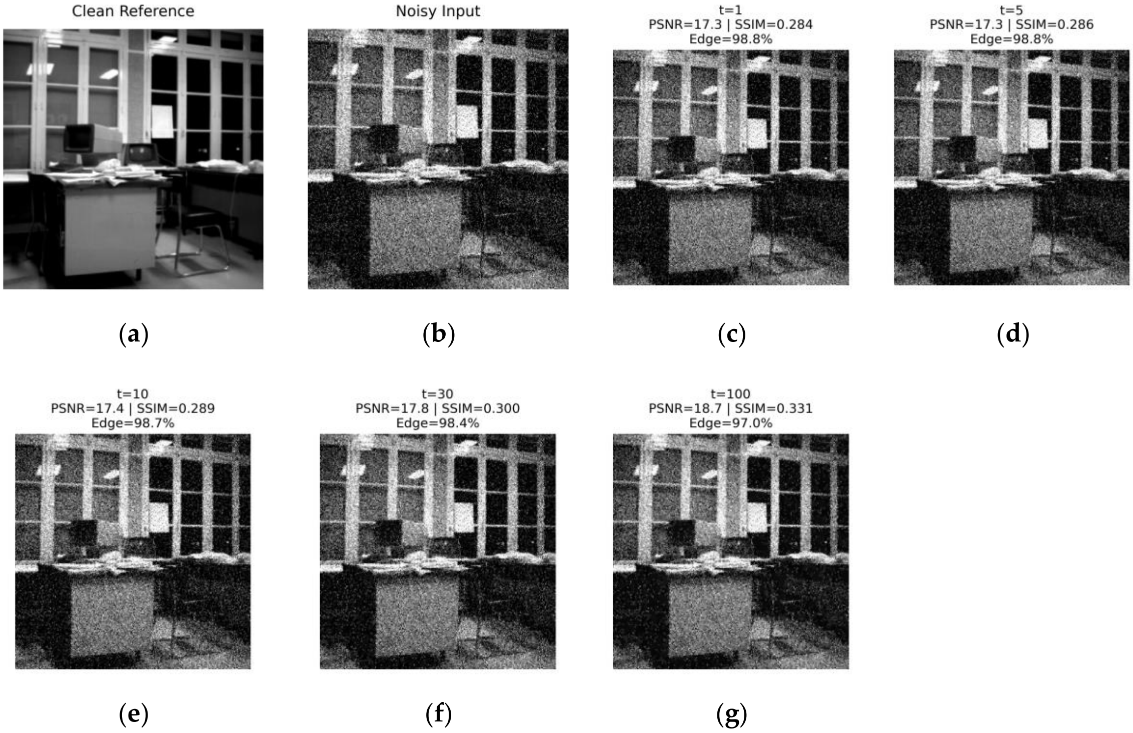
5.3.1. Time Evolution Analysis

Considering a fixed edge threshold of λ=0.5, the Perona-Malik functions completely differently compared to the previous methods. The results in Table 4 and Figure 7 show that this method provides better results with longer function times. The graphs in Figures

8a to 8c demonstrate a gradual increase in PSNR from 17.26 dB to 18.68 dB, an improvement in SSIM from 0.2835 to 0.3305, and an exceptional stability in edge preservation within the range of 97.0-98.8%. Based on this analysis, t=100 is identified as the optimal choice.

Table 4. Analysis of Perona-Malik Diffusion with Variable Evolution Time (λ=0.5).

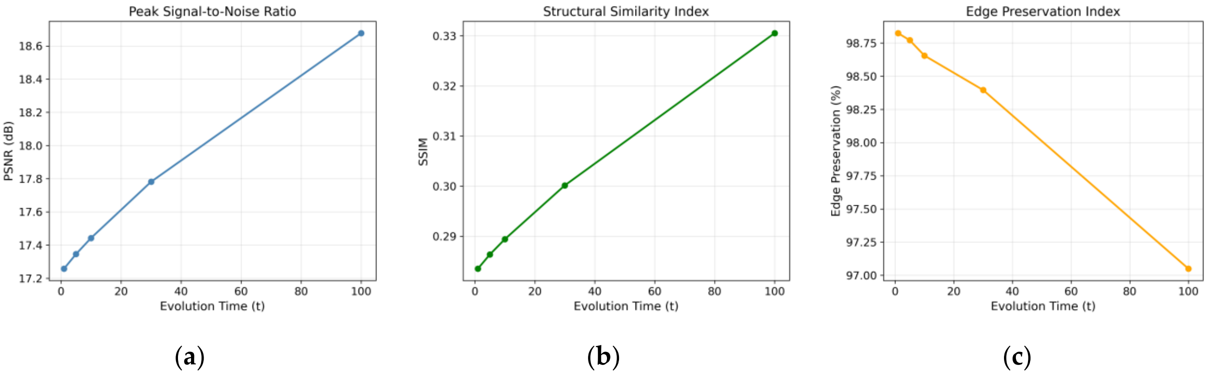
t	PSNR (dB)	SSIM	Edge Preservation Index (%)
1	17.26	0.2835	98.8
5	17.35	0.2864	98.8
10	17.44	0.2894	98.7
30	17.78	0.3002	98.4
100	18.68	0.3305	97.0



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**)

Figure 7. Visual Evolution of Perona-Malik Diffusion through Time (λ=0.5).



(**a**) (**b**) (**c**)

Figure 8. Analysis of Perona-Malik Diffusion Time: (a) PSNR, (b) SSIM, (c) Edge Preservation Index (EPI) vs. Time.

5.3.2. The Effects of λ Parameter

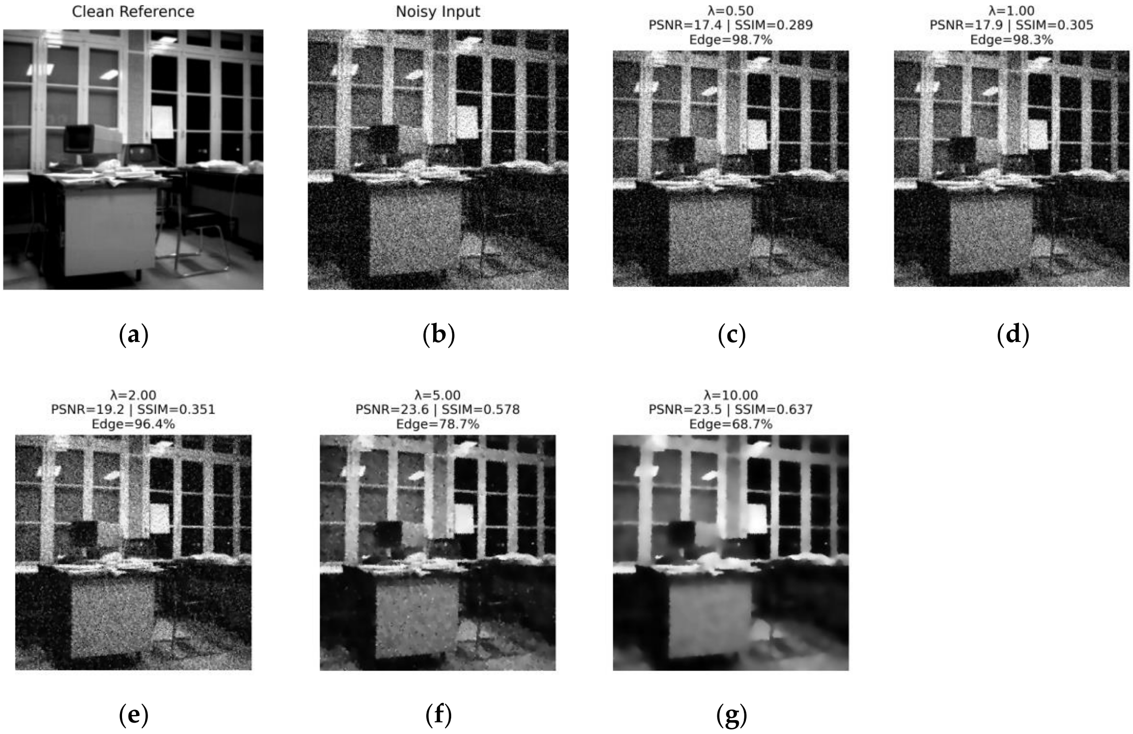
The analysis of the parameter λ was conducted at a fixed time t=10; the results presented in Table 5 and Figure 9, reveal an interesting correlation. When λ=5, the model obtains the highest PSNR (23.57 dB) and SSIM (0.5778) results, indicating excellent reconstruction quality; although edge preservation drops to 78.7%, as illustrated in Figure 9f. On the other hand, λ values below 5 ensure high levels of edge preservation (above 96%), but result in lower PSNR and lower reconstruction quality. This analysis reveals the fundamental balance in image denoising: With the increase of λ, noise removal improves significantly; however, edge preservation, particularly for weaker edges, decreases.

At λ=10, a particularly significant phenomenon is observed. While the edge preservation index decreases to 68.7%, the method demonstrates a more efficient qualitative performance regarding the edges. As shown in Figure 9g, the weaker edges, such as the chair legs, undergo an excessive amount of blurring; in contrast, the strong edges are maintained and enhanced through a better amount of contrast. This selective processing of edges leads to a clearer image with significantly lower levels of noise; additionally, it also maintains the structural integrity of the important features. The SSIM level reach its maximum amount at 0.6366, which indicates the most optimal amount of structural similarity between the original image and the processed image.

The graphs in Figures 10a to 10c illustrate this relationship: This analysis highlights the limitation of relying solely on the numerical amount of edge preservation index, as they this number may not fully reflect the perceptual quality improvements achieved by adaptive methods. After considering the general quality of the image and the balance between denoising and significant edge preservation, both amounts of λ=5 and λ=10 are revealed to be the optimal parameters; furthermore, λ=10 provides the most efficiency balance for the applications where high levels of structural preservation and denoising are required.

Table 5. Performance of Perona-Malik Diffusion with Variable Edge Threshold (t=10).

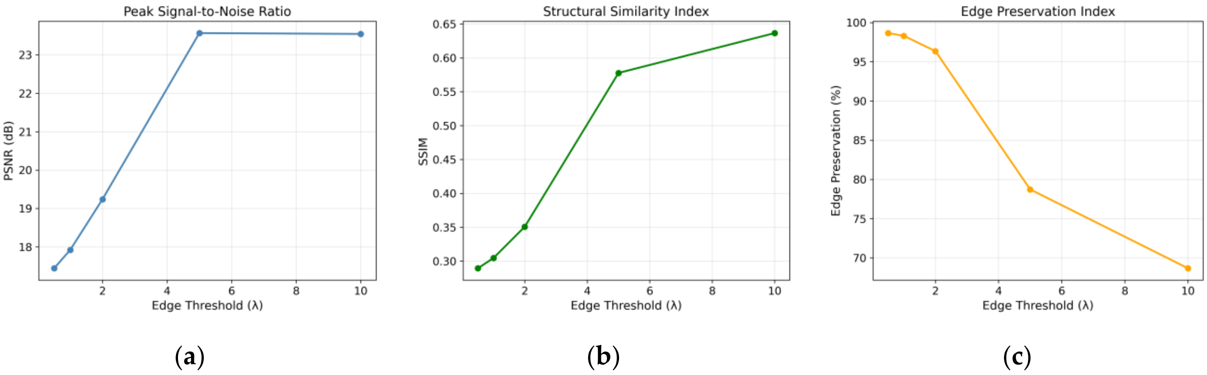
λ	PSNR (dB)	SSIM	Edge Preservation Index (%)
0.50	17.44	0.2894	98.7
1	17.92	0.3046	98.3
2	19.24	0.3508	96.4
5	23.57	0.5778	78.7
10	23.54	0.6366	68.7



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**)

Figure 9. Effect of Edge Threshold λ on the Results of Perona-Malik Diffusion (t=10).



(**a**) (**b**) (**c**)

Figure 10. Analysis of Perona-Malik Edge Threshold: (a) PSNR, (b) SSIM, (c) Edge Preservation Index (EPI) vs. λ.

5.4. Cross-Method Performance Comparison

5.4.1. The Gaussian Filter vs. Linear Diffusion

The correspondence between the Gaussian filter (σ=6.32) with the Linear Diffusion (D=1, t=20) is calculated using $\sigma=\sqrt{(2Dt)}$. The comparative tests presented in Table 6 and Figure 11 demonstrate excellent consistency between these methods with nearly identical performance metrics. The visual comparison in Figures 11c and 11d, along with the bar charts in Figures 12a to 12c, confirm the validity of this correlation, indicating that both methods produce identical results under equivalent conditions. This alignment highlights their equivalent performance in practical applications.

Table 6. Gaussian Filter vs. Linear Diffusion under Equivalent Conditions.

Method	Parameters	PSNR (dB)	SSIM	Edge Preservation Index (%)
Gaussian Filter	σ=6.32	16.29	0.3812	66.9
Linear Diffusion	D=1, t=20	16.28	0.3811	66.6
Difference (Gaussian-Linear)	**	+0.01	+0.0001	+0.03

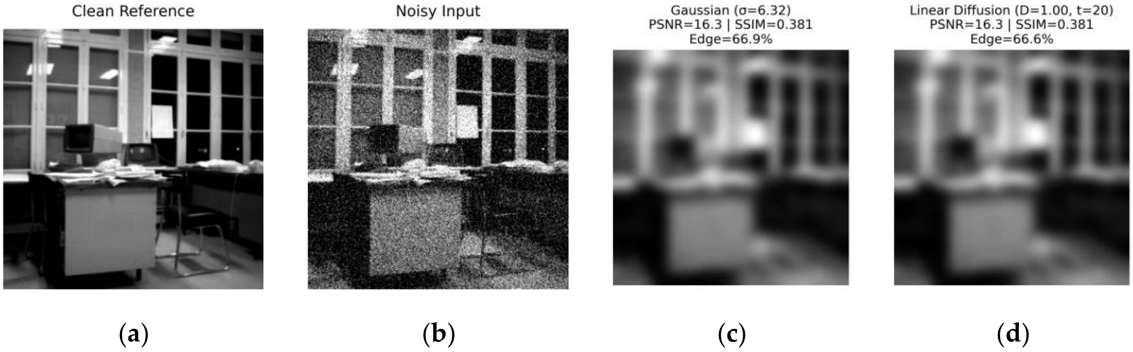


Figure 11. Visual Comparison between Gaussian Filter and Linear Diffusion under Equivalent Conditions.

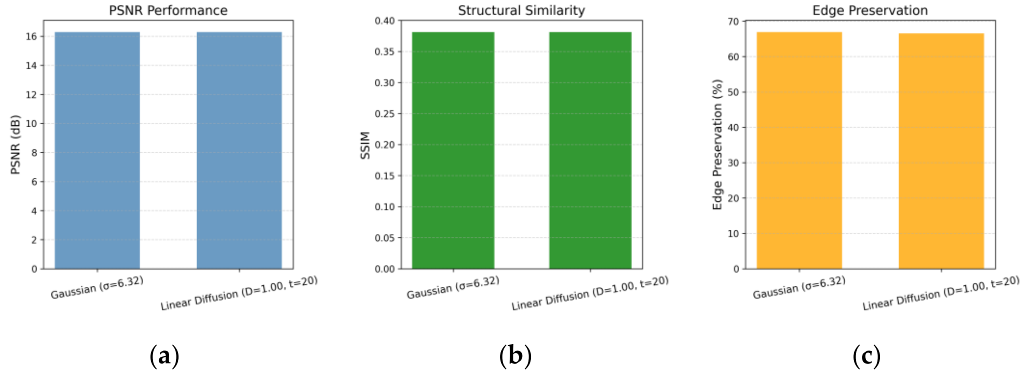


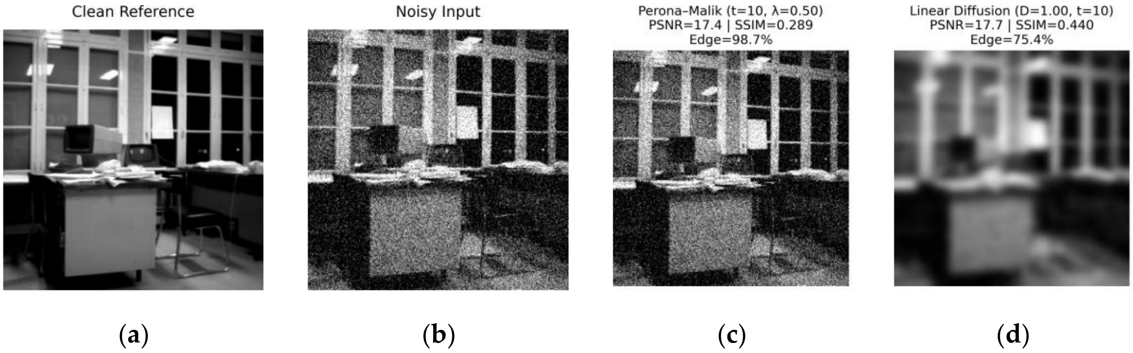
Figure 12. Gaussian Filter Performance vs. Linear Diffusion: (a) PSNR, (b) SSIM, (c) Edge Preservation Index.

5.4.2. Linear Diffusion vs. Perona-Malik diffusion

The comparison between the Perona-Malik diffusion (λ=0.5, t=10) and linear diffusion (D=1, t=10), as displayed in Table 7 and Figure 13, reveals the evident superiority of the Perona-Malik algorithm in edge preservation. While Linear Diffusion can obtain better PSNR (17.73 dB) and SSIM (0.4399) results, the Perona-Malik diffusion method provides better edge preservation results. Linear Diffusion and Perona-Malik diffusion methods result in edge preservation levels of 75.4% and 98.7%, respectively; which is a significant improvement. The visual comparisons displayed in Figures 13c and 13d, in addition to the bar charts in Figures 14a and 14c reveal this difference. Consequently, the Perona-Malik diffusion is recognized as the most efficient method for applications that prioritize edge preservation.

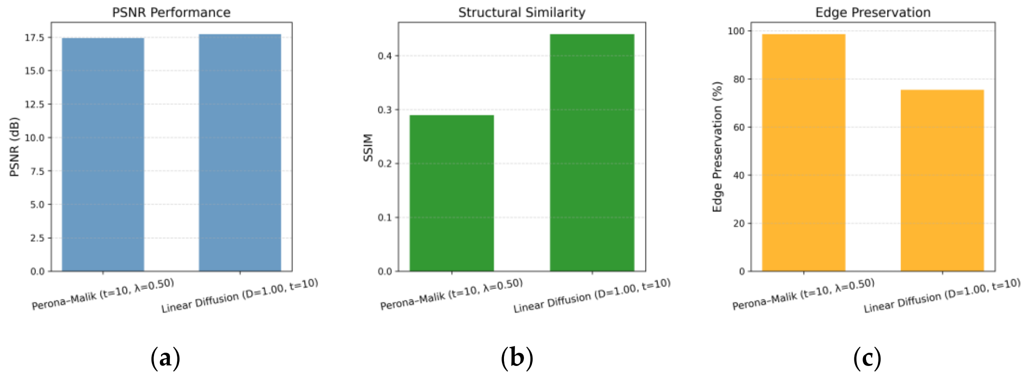
Table 7. Edge Preservation Index (EPI): Linear Diffusion vs. Perona-Malik Diffusion (t=10).

Method	Parameters	PSNR (dB)	SSIM	Edge Preservation Index (%)
Perona-Malik	λ=0.5, t=10	17.44	0.2894	98.7
Linear Diffusion	D=1, t=10	17.73	0.4399	75.4
Difference (PM-Linear)	**	-0.29	-0.1505	+23.3



(**a**) (**b**) (**c**) (**d**)

Figure 13. Difference between Linear Diffusion and Perona-Malik Diffusion in Edge Preservation.

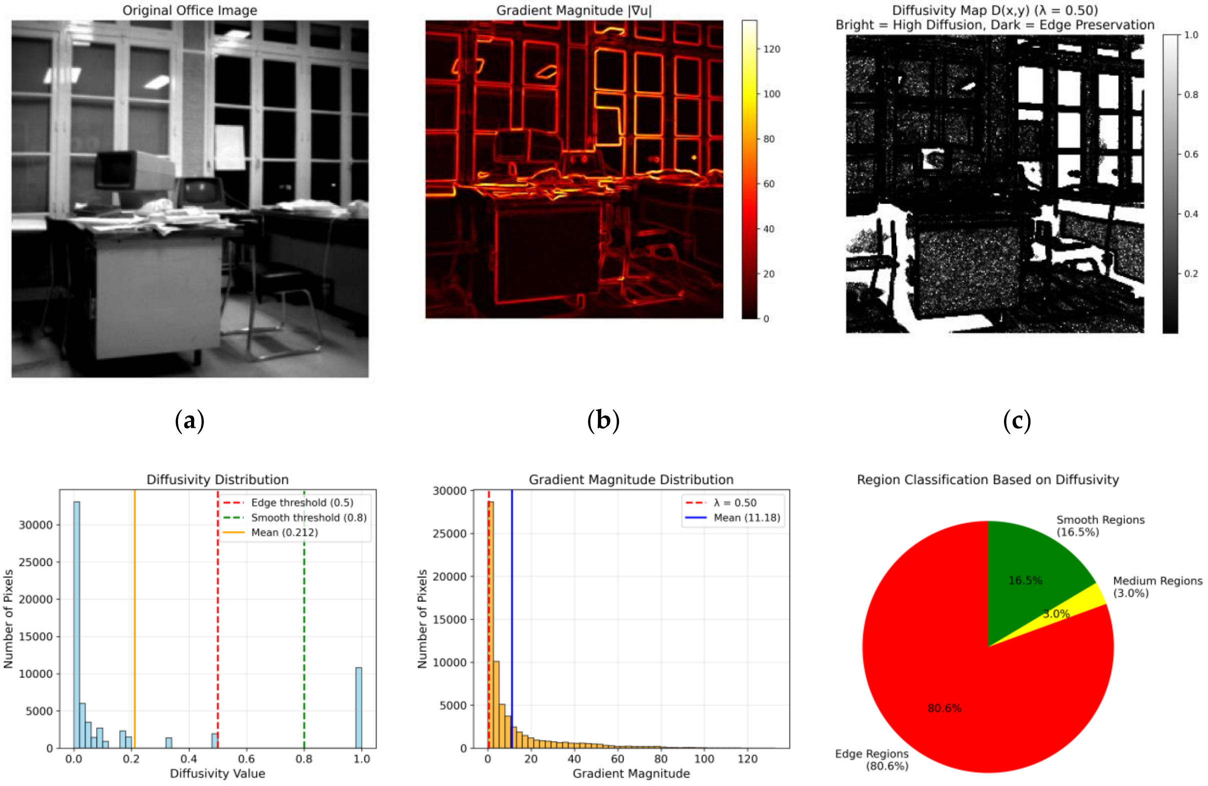


(**a**) (**b**) (**c**)

Figure 14. Linear Performance vs. Perona-Malik Diffusion (a) PSNR, (b) SSIM, (c) Edge Preservation.

5.5. The Analysis of Permeability of the Perona-Malik Model

The quantitative analysis of diffusivity displayed in Figure 15 reveals the adaptive function of the Perona-Malik algorithm. Figure 15a shows the original image, Figure 15b presents the gradient map, and Figure 15c displays the diffusivity map, where 80.6% of the image pixels lie in the edge-preserving regions (D<0.5). Figure 15d illustrates the diffusivity distribution with a mean value of 0.2119±0.3639, while Figure 15e shows the gradient distribution with a mean gradient of 11.181, and 80.6% of the pixels exceeding the threshold λ=0.5. Finally, Figure 15f provides a classification of regions: 80.6% edge-preserving, 16.5% smooth, and 3.0% intermediate. These results verify the smart and adaptive nature of the Perona-Malik diffusion process.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 15. Analysis of Permeability of the Perona-Malik Model.

When it is represented as a grayscale image, the diffusivity map highlights the anisotropic characteristics of the Perona-Malik model. In this map, the edge-preserving regions (D<0.5) appear darker, indicating low diffusivity in areas with high gradient magnitudes, which leads to the preservation of structural details. On the other hand, smooth regions (D>0.8) appear brighter, reflecting strong diffusivity in homogeneous areas, which facilitates the denoising process. Intermediate regions (0.5≤D≤0.8) are displayed in varying shades of gray, indicating a balance between smoothing and edge preservation. This representation of the Perona-Malik model further emphasizes the algorithm’s capability to adapt spatially based on the local gradient properties.

5.7. Discussion

Three image denoising approaches were analyzed in this comprehensive study: The Gaussian filtering, linear diffusion, and Perona-Malik diffusion. The findings of the study indicate that the Gaussian filter at σ=1 provides a reasonable balance between reconstruction quality (PSNR=23.58 dB), structural similarity (SSIM=0.5655), and edge preservation (95.4%); however, with small σ values, it fails to remove noise completely while preserving edges, and with larger σ values it removes more noise but causes edges to become visibly blurred.

Linear diffusion with D=1, despite its ease of use, exhibits significant structural degradation over time (from t=1 to t=100), and increasing the diffusion coefficient values (D=1 vs. D=5, D=10 at fixed t=10), leads to a significant decline in all quantitative metrics. The correlation between the Gaussian filter (σ=6.32) and linear diffusion under identical conditions (D=1, t=20, where $\sigma=\sqrt{2Dt}$) was confirmed, highlighting their similar functionality.

Nevertheless, the Perona-Malik method is more efficient due to its adaptive and smart nature. When the fixed λ=0.5 is taken into account, it not only leads to an exceptional edge preservation (97.0-98.8% across t=1 to t=100), but it also shows gradual improvement in PSNR and SSIM over time. Further analysis at fixed t=10 revealed that λ=5 yields the highest PSNR values (23.57 dB), while λ=10 achieves the best structural similarity (SSIM=0.6366) and offers selective processing— i.e. preserving strong edges while effectively suppressing noise in smoother regions.

Comparing the Perona-Malik method with linear diffusion (D=1, t=10 vs. λ=0.5, t=10) shows that Perona-Malik delivers up to 23% higher edge preservation; an analysis of the diffusivity map confirms its spatial adaptivity, with over 80% of the image pixels benefiting from edge-preserving diffusion. These findings highlight this fundamental challenge in image denoising: The Perona-Malik approach achieves the most efficient noise removal levels while preserving important structural details through its smart and adaptive approach.

Overall, these findings highlight the limitations of relying solely on numerical metrics and emphasize the necessity of qualitative assessment. The Perona-Malik approach, particularly at λ=5 or 10 with t=10, stands out as the most effective solution for applications that demand an optimal balance between denoising and the preservation of important structural details.

6. Conclusion

This study investigated the performance of three primary image noise removal methods: The Gaussian filtering (with varying σ values), linear diffusion (with fixed D=1 and varying time t, as well as fixed t=10 and varying D), and the Perona-Malik method (with fixed λ=0.5 and varying time t, as well as fixed t=10 and varying λ). Findings of our comprehensive experiments illustrate that each method has its specific advantages and limitations.

The Gaussian filter at optimal σ=1, as a classical method, is easy to use and quick, but it has limitations in preserving fine image details while removing noise, creating an unavoidable correlation between noise reduction and edge preservation. Linear diffusion, despite being theoretically equivalent to the Gaussian filter ($\sigma=\sqrt{2Dt}$), displays poor performance quality under extended temporal conditions (t>30). Conversely, the Perona-Malik method, with its adaptive characteristics and anisotropic nature, demonstrates significant superiority in preserving edges and important image structures even at extended diffusion times (t=100).

The key findings of this research include but are not limited to:

1. The confirmation of the theoretical relationship between the Gaussian filter (σ=6.32) and linear diffusion (D=1, t=20);
2. The Perona-Malik method (λ=0.5, t=10) being 23% more efficient in edge preservation compared to linear diffusion (D=1, t=10);
3. The importance of precise adjustment of the λ parameter in balancing the reconstruction quality and edge preservation;
4. The smart adaptive nature of the Perona-Malik algorithm that classifies 80.6% of the image pixels in edge-preserving regions.

These results can have important applications in various image processing fields. For applications where detail and edge preservation are prioritized, such as medical imaging or machine vision, the Perona-Malik method is recommended. Conversely, for applications requiring rapid processing, the Gaussian filter remains a suitable option.

Future research can investigate the combination of different methods to leverage the advantages of each, develop adaptive algorithms for automatic parameter adjustment, evaluate the methods performance on various types of noise and images, implement parallel programming approaches to enhance processing speed in real-time applications, and optimize the computational process for embedded systems and mobile device applications.

References

1. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 4th ed.; Pearson: New York, NY, USA, 2018.
2. Buades, A.; Coll, B.; Morel, J.M. A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.* **2005**, *4*, 490-530.
3. Motwani, M.C.; Gadiya, M.C.; Motwani, R.C.; Harris, F.C. Survey of image denoising techniques. In *Proceedings of the Global Signal Processing Expo and Conference*, Santa Clara, CA, USA, 27-30 September 2004; pp. 27-30
4. Lamba, S.; Raj, P. Edge Detection using Average filter & Thresholding. Int. J. Eng. Comput. Sci. 2017, 6, 22502-22506.
5. Zhang, P.; Li, F. A new adaptive weighted mean filter for removing salt-and-pepper noise. IEEE Signal Process. Lett. 2014, 21, 1280-1283.
6. Lindenbaum, M.; Fischer, M.; Bruckstein, A. On Gabor's contribution to image enhancement. Pattern Recognit. 1994, 27, 1-8.
7. Agarwal, R. Edge Preserving Bit-Plane Adaptive Wiener Filter for Gaussian Noise Restoration. Int. J. Comput. Appl. 2014, 108, 12.
8. Makandar, A.; Bhagirathi, H. Breast cancer image enhancement using median filter and clahe. Int. J. Sci. Eng. Res. 2015, 6, 462-465.
9. Sunitha, M.; Arunalatha, B. Implementation of median filter using bit planes for noise removal in images. 2016, 116-124.
10. Esakkirajan, S.; Veerakumar, T.; Subramanyam, A.N.; PremChand, C.H. Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. IEEE Signal Process. Lett. 2011, 18, 287-290.
11. Chandra, V.; Deokar, S.; Badhe, S.; Yawle, R. Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed adaptive median filter. Int. J. Eng. Adv. Technol. 2013, 2249-8958.
12. Nadernejad, E.; Hassanpour, H.; MaiarNaimi, H. Image Restoration using a PDE-based Approach. IJE Trans. B 2007, 20, 225-236.
13. Perona, P.; Malik, J. Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. 1990, 12, 629-639.
14. Agarwal, R. Bit plane average filtering to remove gaussian noise from high contrast images. In Proceedings of the International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 4-6 January 2012.
15. Mastriani, M. Quantum Boolean image denoising. Quantum Inf. Process. 2015, 14, 1647-1673.
16. Rahman, M.M.; Kumar, M.; Arefin, G.; Shorif Uddin, M. Speckle noise reduction from ultrasound images using principal component analysis with bit plane slicing and nonlinear diffusion method. In Proceedings of the International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 22-24 December 2012; pp. 159-163.
17. Khosravi, M.; Hassanpour, H. Image denoising using anisotropic diffusion equations on reflection and illumination components of image. Int. J. Eng. Trans. C 2014, 27, 1339-1348.
18. Momeni, M.; Noshiar, M. Spatial-frequency hybrid method in impulse noise removal and image quality enhancement. Adv. Signal Process. 2017, 2, 33-44.
19. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 2004, 13, 600-612.
20. Weickert, J. Anisotropic Diffusion in Image Processing; Teubner: Stuttgart, Germany, 1998.
21. Koenderink, J.J. The structure of images. Biol. Cybern. 1984, 50, 363-370.
22. Lindeberg, T. Scale-space theory: A basic tool for analyzing structures at different scales. J. Appl. Stat. 1994, 21, 225-270.
23. Sobel, I.; Feldman, G. A 3x3 isotropic gradient operator for image processing. In A Talk at the Stanford Artificial Intelligence Project; Stanford University: Stanford, CA, USA, 1968.
24. Rakesh, R.R.; Chaudhuri, P.; Murthy, C.A. Thresholding in edge detection: a statistical approach. IEEE Trans. Image Process. 2004, 13, 927-936.