IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 11, No. 10, October, 2024, pp. 2715-2725
Journal homepage: http://iieta.org/journals/mmep

Development of Adaptive Gaussian Filter Based Denoising as an Image Enhancement Technique

Aarthi D[1], Panimalar A[2], Santhosh Kumar S[1*], Anitha K[3]

[1] Department of Mathematics, Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore 641020, India
[2] Department of Mathematics, KGiSL Institute of Technology, Coimbatore 641035, India
[3] Department of Mathematics, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Chennai 601103, India

Corresponding Author Email: santhoshkumars@rmv.ac.in

https://doi.org/10.18280/mmep.111013

Received: 21 May 2024
Revised: 10 August 2024
Accepted: 22 August 2024
Available online: 31 October 2024

Keywords:
image denoising, filtration, gaussian filter, PSNR, RMSE

ABSTRACT

Image denoising is crucial for enhancing image quality, especially in medical applications where noise can significantly impact the accuracy of analysis and interpretation. This paper presents the development of an adaptive Gaussian filter-based denoising technique that effectively enhances images corrupted by various types of noise. By incorporating the adaptive adjustment of filter parameters based on local image characteristics, the proposed method achieves superior denoising performance. The algorithm analyzes the noisy image to estimate the noise characteristics, dynamically adjusting the Gaussian filter parameters to ensure optimal preservation of image details while effectively suppressing noise artifacts. Optimized strategies for parameter selection and filtering operations are employed to ensure computational efficiency. A comparative analysis demonstrates that the adaptive Gaussian filter outperforms traditional methods, achieving a higher Peak Signal-to-Noise Ratio (PSNR) and a lower Root Mean Square Error (RMSE). The technique also exhibits robustness against different noise distributions, making it a versatile solution for various image enhancement applications. These findings highlight the potential of the adaptive Gaussian filter to significantly improve image quality, facilitating more accurate and reliable analysis across diverse domains.

1. INTRODUCTION

Medical imaging is indispensable in modern healthcare, offering non-invasive insights into internal structures for clinical analysis and intervention. It facilitates detailed visualization and monitoring of organs and tissues, aiding accurate diagnoses and treatment plans. The continual evolution of imaging technology significantly contributes to enhanced patient care and treatment outcomes, making medical imaging integral part of modern medicine.

The synergy between mathematics and the medical field is a powerful force, leveraging abstract analytical tools to address practical healthcare challenges. Quantitative methods, including statistical analysis and mathematical modeling, are crucial for understanding disease patterns, predicting outcomes, and optimizing healthcare processes. This symbiotic relationship not only deepens our understanding of biological systems but also drives innovation in diagnostic techniques, treatment strategies, and overall healthcare delivery.

Image processing, situated at the crossroads of computer science, mathematics, and engineering, plays a pivotal role in medical imaging by applying algorithms for tasks like reconstruction, segmentation, and feature extraction. Denoising [1], a critical technique within image processing, aims to reduce unwanted noise and enhance image quality. In medical imaging, denoising [2] directly influences diagnostic accuracy and treatment planning, contributing to more reliable clinical assessments. The continuous development of denoising methods ensures the production of high-fidelity images across various applications, from healthcare to computer vision and scientific research. Fuzzy logic [3] plays a significant role in image processing, providing a framework for handling uncertainty and imprecision inherent in image data. In image processing, fuzzy logic [4] allows for the development of algorithms that can effectively deal with vague or subjective information, such as edge detection, segmentation, and pattern recognition. By incorporating fuzzy sets, fuzzy inference systems, and fuzzy clustering techniques, image processing tasks can be performed with greater robustness and flexibility, enabling the extraction of meaningful information from complex and noisy image datasets. Fuzzy logic-based [5, 6] image processing techniques have been successfully applied in various fields, including medical imaging, remote sensing, object recognition, and computer vision, contributing to advancement in automated image analysis and decision-making systems.

2. LITERATURE SURVEY

Zaynidinov et al. [7] described a technique that uses two-dimensional Haar wavelets to digitally compress an image, to find the recovery coefficients, and show the modified image in a higher quality than the original. One of the most common issues with image compression is figuring out and implementing a workable solution that enables you to display every kind of pixel (dot) in a condensed form. This issue was resolved by using a two-dimensional Haar wavelet [8] modification, which led to the compression of the image and improved quality of the processed image over the original. Also, Zhu [9] investigation delves into a total variation-based image denoising model aimed at addressing the staircasing phenomenon inherent in the Rudin-Osher-Fatemi model. This variational model is optimized through the Augmented Lagrangian Method (ALM). A convergence analysis of the proposed algorithm is provided, demonstrating the characteristics of the model and the efficiency of the suggested numerical technique through numerical experiments. Siddig et al. [10], suggested a fourth order image denoising model such that on applying the fixed-point theorem an entropy solution exists and is unique. Also, numerical experiments based on the fast explicit diffusion scheme (FED) demonstrates the efficiency of the proposed method in image denoising. The proposed method was compared with three other models namely, modern mean curvature (MC) model, the You and Kaveh (YK) model, and the Lysaker, Lundervold, and Tai (LLT) model. Their performance was analyzed and is more efficient in reducing noise and maintaining image.

Evaluation by Palma et al. [11] provides a comprehensive overview of anisotropic diffusion filtering, a popular technique used for image denoising and enhancement. The focus on MRI evaluation suggests that the method examines the effectiveness of anisotropic diffusion filtering specifically in the context of MRI images. This valuable contribution on MRI images often suffers from noise and artifacts, and the performance of denoising techniques can vary depending on the characteristics of MRI data. But there is no optimize parameters to address specific artifacts and to validate the technique in clinical practice. Yuan and He [12] explored the use of an anisotropic diffusion-based preprocessing filtering algorithm for segmenting high-resolution remote sensing images. The authors introduce anisotropic diffusion filtering, a technique used for image enhancement and noise reduction while preserving edges and features.

Shahin et al. [13] introduced a novel enhancement technique for improving the quality of pathological microscopic images by employing neutrosophic similarity score scaling. It provides an introduction to neutrosophic similarity score scaling, a method used for image enhancement that considers the neutrosophic similarity between pixels. Neutrosophic logic deals with uncertainty, indeterminacy, and inconsistency, making it suitable for handling the complexities of pathological microscopic images.

Khan et.al. [14] emphasized on denoising in complex fuzzy environments suggests that it explores scenarios where signals are affected by multiple sources of uncertainty, ambiguity, or imprecision. This could include environments with uncertain noise characteristics, fuzzy boundaries between signal and noise, or imprecise measurement conditions. But fails in adaptation of denoising techniques to dynamic or evolving environments. In many real-world applications, signal characteristics and noise properties may change over time, requiring adaptive denoising algorithms that can dynamically adjust to changing conditions. Ali [15] investigated the performance of three different completely filtering methods tested with different noises on Magnetic Resonance Imaging (MRI) images. The median filter algorithm is modified, and Gaussian noise and salt-and-pepper noise are added to the MRI image. The proposed Median Filter (MF), Adaptive Median Filter (AMF), and Adaptive Wiener Filter (AWF) are implemented.

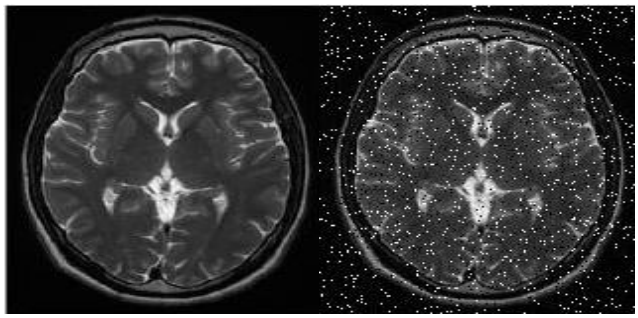
The existing literature highlights various image compression and denoising techniques but lacks a robust solution that adapts effectively to varying noise types and image conditions. While methods like anisotropic diffusion and neutrosophic scaling show promise, they often fall short in dynamic or complex environments. This research gap underscores the need for an adaptive Gaussian filter, which can dynamically adjust to different noise characteristics, offering a more versatile and reliable solution for image denoising.

3. IMAGE

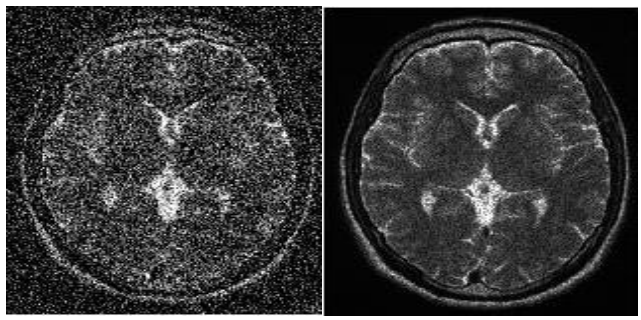
An image [16] can be assumed to be an encoded form of matrix with grey-level or color pixel intensity values as its elements. In case of grey scale images, it can be referred to as pixel value in two-dimension (x, $u(x)$) where $u(x)$ is the pixel intensity value at location x. An image noise model can roughly be approximated as:

$$f(x) = u(x) + n(x), x \subset X, X \subset Z^2$$

where, $u(x)$ denotes the original pixel matrix and $n(x)$ denotes noise at location x. The image used is an MRI brain image (Figure 1).



(a) Original image (b) Salt and pepper



(c) Gaussian noise (d) Speckle noise
With Noise Density 0.06

Figure 1. MRI image displaying original image and noised images

4. GAUSSIAN MEMBERSHIP FUNCTION

The Gaussian membership function is a bell-shaped curve used in fuzzy logic to determine the degree of membership of an element in a fuzzy set. Its formula is given by:

$$\mu(x) = exp\left(-\frac{(x-c)^2}{2\sigma^2}\right)$$

where, $\mu(x)$ is the membership value for an element x, c is the centre or peak of the curve, σ is the standard deviation controlling the spread of the curve.

In the context of image processing, particularly denoising of pixel matrices, the Gaussian membership function can be applied to assign membership values [17] to pixel intensities. The peak of the Gaussian curve represents the intensity value with full membership, while values farther from the peak have lower membership. This function is useful for capturing the gradual transition between different intensity levels, allowing for a smooth representation of uncertainties in pixel values. When denoising an image, the Gaussian membership function helps in preserving important details while smoothing out noise, providing a balanced and effective approach to enhance image quality.

5. METHODOLOGY

Filtering techniques are essential tools in image processing that play a crucial role in enhancing image quality and extracting relevant information. These techniques involve the application of filters, which are mathematical operations applied to pixel values in an image. Filtering techniques [18, 19] find widespread use in various fields, including medical imaging, computer vision, and remote sensing. They are employed for tasks such as noise reduction [20], edge enhancement, and feature extraction. The choice of a filtering technique depends on the specific characteristics of the image and the objectives of the image processing task. While some filters are designed to smoothen the image and reduce noise, others are tailored for edge detection or sharpening details. The continuous evolution of filtering techniques contributes to advancements in image processing, facilitating improved analysis and interpretation of visual data in diverse applications.

Gaussian filters, both 1D and 2D, are vital for medical image processing, particularly MRI, since they effectively reduce noise while maintaining important imagine characteristics. By using fuzzy logic to address uncertainties in picture data, the fuzzy Gaussian filters improve this capability even more. Standard and fuzzy Gaussian filters smooth images while maintaining edges, providing a balanced approach to noise reduction. They are computationally efficient, can tolerate a wide range of noise kinds, and serve as the foundation for numerous sophisticated image processing methods. While traditional filters such as wiener and median filters have their advantages, Gaussian filters are often more versatile, computationally efficient, and able to handle a wider range of noise types.

When it comes to context-aware smoothing, adaptive filters are excellent at distinguishing between different kinds of image regions and applying the right amount of smoothing to improve overall image quality and minimize abnormalities. Adaptive Gaussian filters are particularly helpful in situations where preserving minute structural details is crucial, such medical imaging, is required for accurate diagnosis and analysis, due to their versatility and effectiveness.

This paper presents four types of adaptive gaussian filter [21, 22] and their performance analysis.

5.1 Adaptive gaussian filter 1D and derivative of gaussian 1D (AGD-1D)

It is a filter whose impulse response is gaussian function. A Gaussian filter [23] is a popular image processing technique used for smoothening and reducing noise in images. It is based on the Gaussian distribution [24, 25] and operates by convolving the image with a Gaussian kernel. The mathematical formula for a one-dimensional Gaussian function is given by:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (1)$$

where, $G(x)$ is the value of the Gaussian function at position x, σ is the standard deviation, determining the width of the gaussian curve. The standard deviation and the constant value are given by:

$$\sigma = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(A(i,j)-m)^2}\ ;\propto = \frac{A}{\max(|A|)} \quad (2)$$

where, max($|A|$) is the maximum absolute value of any element in the matrix. Derivative [26, 27] of Gaussian 1D filter can indirectly contribute to denoising by emphasizing edges and suppressing noise. The steep response of the filter to intensity changes can help highlight significant structures in the image while minimizing the influence of random noise. The one-dimensional mathematical formula for the derivative of Gaussian 1D filter is given by:

$$\frac{\partial G(x)}{\partial x} = -\frac{x}{\sigma^3} exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (3)$$

where, $\frac{\partial G(x)}{\partial x}$ is the derivative of the Gaussian function with respect to x, x is the spatial coordinate, σ is the standard deviation, controlling the width of the Gaussian distribution.

$$F(x,\sigma) = G(x) + \alpha.\frac{\partial G(x)}{\partial x} \quad (4)$$

where, α is a constant multiplier for the derivative term. The combine equation represents a linear combination of a Gaussian Function and its derivative.

$$\begin{aligned} F(x,\sigma) &= \frac{1}{\sqrt{2\pi}\sigma} exp\left(\frac{-x^2}{2\sigma^2}\right) + \alpha.\left[-\frac{x}{\sigma^2}\frac{1}{\sqrt{2\pi}\sigma} exp\left(\frac{-x^2}{2\sigma^2}\right)\right] \\ &= \frac{1}{\sqrt{2\pi}\sigma}\left[1 - \alpha.\frac{x}{\sigma^2}\right] exp\left(\frac{-x^2}{2\sigma^2}\right) \end{aligned} \quad (5)$$

5.2 Adaptive gaussian 2D filter and derivative of 2D filter (AGD-2D)

A two-dimensional Gaussian filter, the formula is the product of two one-dimensional Gaussians along the rows and columns, forming a 2D kernel. The filter effectively reduces high-frequency noise in the image while preserving its overall structure. It is based on the mathematical formulation of a two-dimensional Gaussian distribution and operates by convolving the image with a Gaussian kernel. The mathematical formula for a two-dimensional Gaussian function is given by:

$$G(x, y, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} exp\left(-\frac{x^2}{2{\sigma_x}^2} - \frac{y^2}{2{\sigma_y}^2}\right) \quad (6)$$

where, $G(x, y, \sigma_x, \sigma_y)$ is the value of the two-dimensional Gaussian function at positions x and y, σ is the standard deviation determining the width of the Gaussian distribution. Also, the two-dimensional Gaussian derivative filter is given by:

$$G_x(x, y, \sigma_x, \sigma_y) = -\frac{x}{{\sigma_x}^2}\frac{1}{2\pi\sigma_x\sigma_y} exp\left(-\frac{x^2}{2{\sigma_x}^2} - \frac{y^2}{2{\sigma_y}^2}\right) \quad (7)$$

$$G_y(x, y, \sigma_x, \sigma_y) = -\frac{y}{{\sigma_y}^2}\frac{1}{2\pi\sigma_x\sigma_y} exp\left(-\frac{x^2}{2{\sigma_x}^2} - \frac{y^2}{2{\sigma_y}^2}\right) \quad (8)$$

where, $G_x(x, y, \sigma_x, \sigma_y)$ represents $\frac{\partial G(x,y)}{\partial x}$, $G_y(x, y, \sigma_x, \sigma_y)$ represents $\frac{\partial G(x,y)}{\partial y}$ are the partial derivatives of the Gaussian function with respect to x and y, σ is the standard deviation controlling the width of the Gaussian distribution, x and y are the spatial coordinates.

$$\begin{aligned} F(x, y, \sigma_x, \sigma_y, \alpha, \beta) &= G(x, y, \sigma_x, \sigma_y) \\ &+\alpha. G_x(x, y, \sigma_x, \sigma_y) + \beta. G_y(x, y, \sigma_x, \sigma_y) \end{aligned} \quad (9)$$

where, α, β are constant multiplier for the partial derivative in the x and y directions respectively. The final combined equation is:

$$\begin{aligned} F(x, y, \sigma_x, \sigma_y, \alpha, \beta) &= \frac{1}{2\pi\sigma_x\sigma_y} exp\left(-\frac{x^2}{2{\sigma_x}^2} - \frac{y^2}{2{\sigma_y}^2}\right) \\ &+\alpha.\left[-\frac{x}{{\sigma_x}^2}\frac{1}{2\pi\sigma_x\sigma_y} exp\left(-\frac{x^2}{2{\sigma_x}^2} - \frac{y^2}{2{\sigma_y}^2}\right)\right] \\ &+\beta\left[-\frac{y}{{\sigma_y}^2}\frac{1}{2\pi\sigma_x\sigma_y} exp\left(-\frac{x^2}{2{\sigma_x}^2} - \frac{y^2}{2{\sigma_y}^2}\right)\right] \\ F(x, y, \sigma_x, \sigma_y, \alpha, \beta) &= \frac{1}{2\pi\sigma_x\sigma_y}\left[1 - \alpha.\frac{x}{{\sigma_x}^2} - \beta.\frac{y}{{\sigma_y}^2}\right] exp\left(-\frac{x^2}{2{\sigma_x}^2} - \frac{y^2}{2{\sigma_y}^2}\right) \end{aligned} \quad (10)$$

where,

$\sigma_x = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(A(i,j) - m_x)^2}$ for each i;

$\sigma_y = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(A(i,j) - m_y)^2}$ for each j;

$\propto = \frac{A(i,j)}{\max(|A(i,j)|)_{row}}$ for each i, j;

$\beta = \frac{A(i,j)}{\max(|A(i,j)|)_{column}}$ for each i, j.

5.3 Adaptive fuzzy gaussian 1D filter and derivative fuzzy gaussian 1D filter (AFGD-1D)

Let the function be represented as:

$$F(x, \sigma, m) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) \quad (11)$$

where, $F(x, \sigma, m)$ is Fuzzy Gaussian 1D, x is a variable, σ is the standard deviation and m is a fuzziness parameter. Also, the derivative of Eq. (11) (i.e.) fuzzy gaussian function with respect to x. The derivative is given as:

$$F'(x, \sigma, m) = -\left(\frac{x-m}{\sigma^2}\right)\frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) \quad (12)$$

The final combination function is given by:

$$G(x, \sigma, m, \alpha) = F(x, \sigma, m) + \alpha. F'(x, \sigma, m)$$

where, σ is a constant multiplier and is calculated using (2):

$$\begin{aligned} G(x, \sigma, m, \alpha) &= \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) \\ &+\alpha.\left[-\left(\frac{x-m}{\sigma^2}\right)\frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-m)^2}{2\sigma^2}\right)\right] \\ G(x, \sigma, m, \alpha) &= \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-m)^2}{2\sigma^2}\right)\left[1 - \alpha.\left(\frac{x-m}{\sigma^2}\right)\right] \end{aligned} \quad (13)$$

where, $m = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} A(i,j)$.

5.4 Adaptive fuzzy gaussian 2D filter and derivative of fuzzy gaussian 2D (AFGD-2D)

Consider a 2D fuzzy Gaussian function $F(x, y, \sigma_x, \sigma_y, m_x, m_y)$ where x and y are the variables $\sigma_x, \sigma_y, m_x, m_y$ are the standard deviation and mean along the respective axis. The function is given by:

$$\begin{aligned} &F(x, y, \sigma_x, \sigma_y, m_x, m_y) \\ &= \frac{1}{2\pi\sigma_x\sigma_y} exp\left[-\frac{(x-m_x)^2}{2{\sigma_x}^2} - \frac{(y-m_y)^2}{2{\sigma_x}^2}\right] \end{aligned} \quad (14)$$

The fuzzy partial derivative of this 2D fuzzy gaussian function with respect to x and y. The partial derivatives are given by:

$$\begin{aligned} &F_x(x, y, \sigma_x, \sigma_y, m_x, m_y) \\ &= -\left(\frac{x-m_x}{{\sigma_x}^2}\right)\frac{1}{2\pi\sigma_x\sigma_y} exp\left[-\frac{(x-m_x)^2}{2{\sigma_x}^2} - \frac{(y-m_y)^2}{2{\sigma_y}^2}\right] \end{aligned} \quad (15)$$

$$\begin{aligned} &F_y(x, y, \sigma_x, \sigma_y, m_x, m_y) \\ &= -\left(\frac{y-m_y}{{\sigma_y}^2}\right)\frac{1}{2\pi\sigma_x\sigma_y} exp\left[-\frac{(x-m_x)^2}{2{\sigma_x}^2} - \frac{(y-m_y)^2}{2{\sigma_y}^2}\right] \end{aligned} \quad (16)$$

Combining all above equations, we get:

$$\begin{aligned} &G(x, y, \sigma_x, \sigma_y, m_x, m_y, \alpha, \beta) \\ &= F(x, y, \sigma_x, \sigma_y, m_x, m_y) \\ &+\alpha. F_x(x, y, \sigma_x, \sigma_y, m_x, m_y) \\ &+\beta. F_y(x, y, \sigma_x, \sigma_y, m_x, m_y) \\ &G(x, y, \sigma_x, \sigma_y, m_x, m_y, \alpha, \beta) \\ &= \frac{1}{2\pi\sigma_x\sigma_y} exp\left[-\frac{(x-m_x)^2}{2{\sigma_x}^2} - \frac{(y-m_y)^2}{2{\sigma_y}^2}\right] \\ &+\alpha.\left\{-\left(\frac{x-m_x}{{\sigma_x}^2}\right)\frac{1}{2\pi\sigma_x\sigma_y} exp\left[-\frac{(x-m_x)^2}{2{\sigma_x}^2} - \frac{(y-m_y)^2}{2{\sigma_y}^2}\right]\right\} \\ &+\beta.\left\{-\left(\frac{y-m_y}{{\sigma_y}^2}\right)\frac{1}{2\pi\sigma_x\sigma_y} exp\left[-\frac{(x-m_x)^2}{2{\sigma_x}^2} - \frac{(y-m_y)^2}{2{\sigma_y}^2}\right]\right\} \\ &G(x, y, \sigma_x, \sigma_y, m_x, m_y, \alpha, \beta) \\ &= \frac{1}{2\pi\sigma_x\sigma_y} exp\left[-\frac{(x-m_x)^2}{2{\sigma_x}^2} - \frac{(y-m_y)^2}{2{\sigma_y}^2}\right]\left[1 - \alpha.\left(\frac{x-m_x}{{\sigma_x}^2}\right) - \right. \\ &\left. \beta.\left(\frac{y-m_y}{{\sigma_y}^2}\right)\right] \end{aligned} \quad (17)$$

For 2D the constants are calculated using formulas for a given matrix A of size m x n: $m_x = \frac{1}{n}\sum_{j=1}^{n} A(i,j)$ for each i; $m_y = \frac{1}{m}\sum_{i=1}^{m} A(i,j)$ for each j.

5.5 Numerical example of proposed methodology

In order to display the numerical calculation of the above designed filters, consider a sample pixel 5×5 matrix from the image given in Figure 1.

Let

$$A = \begin{bmatrix} 102 & 168 & 199 & 209 & 195 \\ 158 & 195 & 202 & 190 & 172 \\ 197 & 209 & 197 & 174 & 158 \\ 208 & 206 & 189 & 166 & 157 \\ 190 & 181 & 172 & 159 & 154 \end{bmatrix}$$

be the 5×5 pixel matrix and the numerical calculation to obtain denoised matrix using the above-described formulas are demonstrated below. The calculation is done using MATLAB and the following values are obtained:

$$\text{Image}_{\text{filtered}} = \text{Image}_{\text{original}} + \text{Convolution}_{\text{matrix}}$$

$$\propto = \begin{bmatrix} 0.4880 & 0.8038 & 0.9522 & 1 & 0.9330 \\ 0.7560 & 0.9330 & 0.9665 & 0.9091 & 0.8230 \\ 0.9426 & 1 & 0.9426 & 0.8325 & 0.7560 \\ 0.9952 & 0.9856 & 0.9043 & 0.7943 & 0.7512 \\ 0.9091 & 0.8660 & 0.8230 & 0.7608 & 0.7368 \end{bmatrix};$$

$$\beta = \begin{bmatrix} 0.4904 & 0.7596 & 0.9471 & 1 & 0.9135 \\ 0.8038 & 0.9330 & 1 & 0.9856 & 0.8660 \\ 0.9851 & 1 & 0.9752 & 0.9356 & 0.8515 \\ 1 & 0.9091 & 0.8325 & 0.7943 & 0.7608 \\ 0.9135 & 0.8821 & 0.8103 & 0.8051 & 0.7897 \end{bmatrix}$$

$$m_x = \begin{bmatrix} 171 \\ 191.8 \\ 191.8 \\ 179.6 \\ 167.2 \end{bmatrix} m_y = \begin{bmatrix} 174.6 \\ 183.4 \\ 187 \\ 185.2 \\ 171.2 \end{bmatrix};$$

$$\sigma_x = \begin{bmatrix} 38.9204 \\ 18.1769 \\ 19.0221 \\ 21.3897 \\ 13.9628 \end{bmatrix}; \sigma_y = \begin{bmatrix} 38.7536 \\ 16.1196 \\ 18.4065 \\ 20.6436 \\ 13.3776 \end{bmatrix};$$

$$\sigma = 24.0807;\ m = 180.28$$

Using AGD-1D:

$$Convolution\ matrix\ (C_x) = \begin{bmatrix} 25.83 & 34.54 & 42.28 & 34.73 & 25.67 \\ 35.37 & 46.56 & 56.59 & 45.71 & 33.43 \\ 43.97 & 57.55 & 69.84 & 55.91 & 40.78 \\ 36.50 & 46.87 & 56.23 & 44.08 & 31.65 \\ 27.70 & 35.17 & 42.01 & 32.46 & 23.11 \end{bmatrix}$$

$$A_{AGD-1D\ Filter} = A + C_x$$

$$= \begin{bmatrix} 127.83 & 202.54 & 241.28 & 243.73 & 220.67 \\ 193.37 & 241.56 & 258.59 & 235.71 & 205.43 \\ 240.97 & 266.55 & 266.84 & 229.91 & 198.78 \\ 244.50 & 252.87 & 245.23 & 210.08 & 188.65 \\ 217.70 & 216.17 & 214.01 & 191.46 & 177.11 \end{bmatrix}$$

$$\cong \begin{bmatrix} 128 & 203 & 241 & 244 & 221 \\ 193 & 242 & 259 & 236 & 205 \\ 241 & 267 & 267 & 230 & 199 \\ 245 & 253 & 245 & 210 & 189 \\ 218 & 216 & 214 & 191 & 177 \end{bmatrix}$$

$$= \begin{bmatrix} 128 & 203 & 241 & 244 & 221 \\ 193 & 242 & 255 & 236 & 205 \\ 241 & 255 & 255 & 230 & 199 \\ 245 & 253 & 245 & 210 & 189 \\ 218 & 216 & 214 & 191 & 177 \end{bmatrix}$$

Using AGD-2D:

$$C_x = \begin{bmatrix} 0.56 & 0.77 & 0.95 & 0.80 & 0.60 \\ 0.78 & 1.04 & 1.28 & 1.04 & 0.77 \\ 1.19 & 1.58 & 1.94 & 1.59 & 1.17 \\ 1.18 & 1.53 & 1.85 & 1.46 & 1.06 \\ 0.90 & 1.15 & 1.38 & 1.07 & 0.77 \end{bmatrix}$$

$$A_{AGD-2D\ Filter}$$

$$= \begin{bmatrix} 102.56 & 168.77 & 199.95 & 209.80 & 195.60 \\ 158.78 & 196.04 & 203.28 & 191.04 & 172.77 \\ 198.19 & 210.58 & 198.94 & 175.59 & 159.17 \\ 209.18 & 207.53 & 190.85 & 167.46 & 158.06 \\ 190.90 & 182.15 & 173.38 & 160.07 & 154.77 \end{bmatrix}$$

$$\cong \begin{bmatrix} 103 & 169 & 200 & 210 & 196 \\ 159 & 196 & 203 & 191 & 173 \\ 198 & 211 & 199 & 176 & 159 \\ 209 & 208 & 191 & 167 & 158 \\ 191 & 182 & 173 & 160 & 155 \end{bmatrix}$$

Using AFGD-1D:

$$C_x = \begin{bmatrix} 52.94 & 37.06 & 38.10 & 25.80 & 31.08 \\ 38.56 & 24.69 & 10.14 & 30.13 & 27.11 \\ 27.07 & 37.11 & 23.97 & 19.16 & 40.83 \\ 29.06 & 19.09 & 45.54 & 23.82 & 56.50 \\ 13.75 & 49.57 & 20.61 & 20.49 & 41.07 \end{bmatrix}$$

$$A_{AFGD-1D\ Filter}$$

$$= \begin{bmatrix} 154.94 & 168.06 & 199.10 & 209.10 & 195.08 \\ 158.56 & 195.09 & 202.14 & 190.13 & 172.11 \\ 198.07 & 209.11 & 197.17 & 174.16 & 158.13 \\ 208.06 & 206.09 & 189.14 & 166.12 & 157.10 \\ 190.75 & 181.07 & 172.11 & 159.09 & 154.07 \end{bmatrix}$$

$$\cong \begin{bmatrix} 155 & 168 & 199 & 209 & 195 \\ 159 & 195 & 202 & 190 & 172 \\ 198 & 209 & 197 & 174 & 158 \\ 208 & 106 & 189 & 166 & 157 \\ 191 & 181 & 172 & 159 & 154 \end{bmatrix}$$

Using AFGD-2D:

$$C_x = \begin{bmatrix} 0.26 & 0.48 & 1.48 & 0.98 & 0.83 \\ 0.26 & 058 & 1.47 & 0.29 & 0.92 \\ 0.24 & 0.33 & 0.44 & 0.93 & 0.27 \\ 0.92 & 0.57 & 0.08 & 0.63 & 1.98 \\ 0.85 & 1.93 & 0.81 & 1.02 & 1.05 \end{bmatrix}$$

$$A_{AFGD-2D\ Filter}$$

$$= \begin{bmatrix} 102.26 & 168.48 & 200.48 & 209.98 & 195.83 \\ 158.26 & 195.58 & 203.47 & 190.29 & 172.92 \\ 197.24 & 209.33 & 197.44 & 174.93 & 158.27 \\ 208.92 & 206.57 & 189.08 & 166.63 & 158.98 \\ 190.85 & 182.93 & 172.81 & 160.02 & 155.05 \end{bmatrix}$$

$$\cong \begin{bmatrix} 102 & 168 & 200 & 210 & 196 \\ 158 & 196 & 203 & 190 & 173 \\ 197 & 209 & 197 & 175 & 158 \\ 209 & 207 & 189 & 167 & 159 \\ 191 & 182 & 173 & 160 & 155 \end{bmatrix}$$

6. PERFORMANCE ANALYSIS

The performance analysis process in image processing is a critical step for evaluating the effectiveness of various techniques and algorithms. This multifaceted assessment involves the application of quantitative metrics to gauge the quality of processed images. Common performance metrics include the Root Mean Squared Error (RMSE) and Peak Signal-to-Noise Ratio (PSNR), which measure the difference and similarity between the original and processed images, respectively. Comparative studies between different filters or algorithms provide insights into their strengths and weaknesses, aiding practitioners in making informed choices

for specific applications. The performance analysis process is crucial for advancing the field, enabling the identification of optimal solutions and fostering continuous improvement in image processing methodologies.

6.1 Root Mean Squared Error

Root Mean Squared Error (RMSE) is a widely used metric in image processing and various other fields to quantify the difference between predicted or processed values and the actual or reference values. It is particularly useful for assessing the accuracy and fidelity of reconstructed or filtered images. The RMSE is calculated by taking the square root of the mean of the squared differences between corresponding pixel values in the original and processed images. The mathematical formula for RMSE is as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(I_i - \hat{I}_i\right)^2} \tag{18}$$

where, N is the total number of pixels in the image, I_i is the intensity value of the i[th] pixel in the original image, $\hat{I}_i$ is the intensity value of the i[th] pixel in the processed or reconstructed image. In Figure 2, the work flow is shown as flow chart.

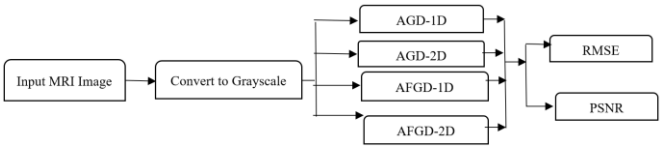


Figure 2. Explains the proposed work flow

6.2 Peak Signal-to-Noise Ratio (PSNR)

Peak Signal-to-Noise Ratio (PSNR) is a widely used metric in image processing to assess the quality of a reconstructed or processed image by comparing it to a reference or original image. PSNR is expressed as a ratio of the peak signal level to the Root Mean Squared Error (RMSE) between corresponding pixel values in the original and processed images. It provides a quantitative measure of the fidelity and similarity between the two images. The mathematical formula for PSNR is given by:

$$PSNR = 10.log_{10}\left(\frac{Peak\ Signal\ Value^2}{Mean\ Squared\ Error}\right) \tag{19}$$

where, *Peak Signal Value* is the maximum possible pixel value, *Mean Squared Error* is the mean of the squared differences between corresponding pixel values in the original and processed image.

7. EXPERIMENTAL RESULTS

The major objective of this work is to design an Adaptive Gaussian Filter by combining the gaussian function and its derivative and the resultant is used as filter for denoising the image. Also, to compare the performance of the adaptive gaussian filter with each other. The filtered images results are shown in Figure 3. The images taken for the proposed work is the secondary data taken from Kaggle.

Algorithm: Adaptive Gaussian Filter-Based Denoising

Input: Noisy Image (I)
Output: Denoised Image (I_denoised)
1. Initialize Parameters:
  a. Read the image as grayscale image.
  b. Set initial Gaussian filter parameters
2. Analyze the Noisy Image:
  a. Estimate local noise characteristics (e.g., noise level, variance)
3. Adaptive Adjustment of Filter Parameters:
  a. Adjust Gaussian filter parameters based on the estimated local noise characteristics.
5. Evaluate Denoising Performance:
  a. Calculate quality metrics (e.g., PSNR, RMSE) to assess the performance of the denoising process.
End Algorithm

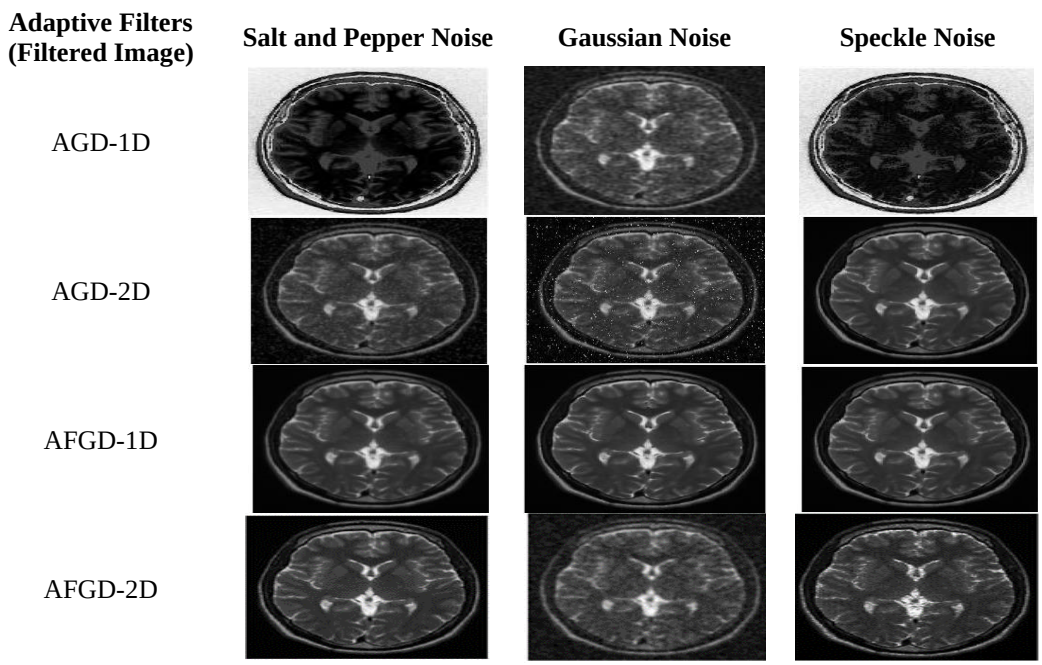


Figure 3. Filtered image after using adaptive filters for noise density =0.06

Table 1. RMSE value for speckle noise

Noise Density	1D Gaussian	2D Gaussian	1D Derivative	2D Derivative	AGD-1D	AGD-2 D	AFGD-1D	AFGD-2D
0.01	7.109147881	9.51227493	48.35614443	46.49438233	4.4315	7.0499	**3.6751**	13.0718
0.02	7.671172547	9.835913216	48.22640418	46.40444893	4.7316	7.1425	**3.9752**	13.096
0.03	8.210896484	10.14732279	48.0989115	46.28819597	5.0585	7.2675	**4.3021**	13.1154
0.04	8.707606813	10.42382294	48.02646318	46.22959589	5.3075	7.3423	**4.5511**	13.1333
0.05	9.080211021	10.61466951	47.94066323	46.15142981	5.5821	7.4365	**4.8257**	13.1506
0.06	9.60487158	10.90986684	47.8783036	46.11420211	5.8534	7.5641	**5.097**	13.1673
0.07	9.990753023	11.14869392	47.79922848	46.0557318	6.1168	7.6674	**5.3604**	13.1814
0.08	10.34709934	11.3494867	47.7208218	45.96744576	6.3838	7.7944	**5.6274**	13.1929
0.09	10.66651743	11.49004254	47.67081004	45.93751735	6.5836	7.8929	**5.8272**	13.208
0.1	11.08357009	11.77125603	47.60464728	45.87470046	6.8535	8.0119	**6.0971**	13.2192

Table 2. PSNR values of speckle noise

Noise Density	1D Gaussian	2D Gaussian	1D Derivative	2D Derivative	AGD-1D	AGD-2 D	AFGD-1D	AFGD-2D
0.01	31.09445264	28.56511573	14.44177029	14.78279396	35.2	31.1673	**56.6179**	36.0645
0.02	30.43356858	28.27450984	14.46510598	14.79961122	34.6307	31.0539	**56.4605**	35.4952
0.03	29.84299207	28.00377409	14.48809865	14.82139851	34.0504	30.9033	**56.3343**	34.9149
0.04	29.3328274	27.77026309	14.50119151	14.83240167	33.6331	30.8143	**56.2182**	34.4976
0.05	28.96888478	27.61267407	14.51672285	14.8471004	33.195	30.7035	**56.1065**	34.0595
0.06	28.48097236	27.37441462	14.52802852	14.85410964	32.7828	30.5557	**55.9989**	33.6473
0.07	28.13883915	27.18632376	14.54238587	14.86512986	32.4004	30.438	**55.9082**	33.2649
0.08	27.83443124	27.0312792	14.55664533	14.88179616	32.0294	30.2953	**55.8337**	32.8939
0.09	27.57035066	26.92437088	14.56575297	14.8874532	31.7617	30.1861	**55.7366**	32.6262
0.1	27.23721017	26.71434749	14.57781657	14.89933878	31.4126	30.0562	**55.665**	32.2771

Table 3. RMSE value for salt and pepper noise

Noise Density	1D Gaussian	2D Gaussian	1D Derivative	2D Derivative	AGD-1D	AGD-2 D	AFGD-1D	AFGD-2D
0.01	11.11702861	10.21946771	48.61942732	46.54211052	6.285	7.7701	**4.1974**	13.0604
0.02	14.44717898	11.5344193	48.63665398	46.43533838	7.9623	8.5877	**5.8747**	13.0778
0.03	17.50551308	12.93244987	48.67265104	46.3265297	9.3603	9.3822	**7.2727**	13.0931
0.04	19.63295042	14.04488319	48.69226808	46.26227233	10.7906	10.2824	**8.703**	13.1083
0.05	22.05272295	15.36976576	48.7141882	46.17514423	12.0355	11.1279	**9.9479**	13.1243
0.06	23.9057489	16.46080945	48.72247308	46.10760899	13.1609	11.9145	**11.0733**	13.1369
0.07	25.82604815	17.64797129	48.74933894	46.06733781	14.4544	12.9025	**12.3668**	13.149
0.08	27.51779109	18.68339019	48.7829182	46.06149024	15.4735	13.6981	**13.3859**	13.1661
0.09	29.32424472	19.93126336	48.80346442	46.0127109	16.5842	14.6072	**14.4966**	13.1775
0.1	31.16517892	21.20821542	48.87401754	46.02980714	17.7571	15.555	**14.6695**	13.1899

Table 4. PSNR value of salt and pepper noise

Noise Density	1D Gaussian	2D Gaussian	1D Derivative	2D Derivative	AGD-1D	AGD-2 D	AFGD-1D	AFGD-2D
0.01	27.21102914	27.9422381	14.39460683	14.77388215	32.1649	30.3224	**56.6922**	33.0625
0.02	24.93514255	26.89088891	14.39152982	14.79383132	30.1101	29.4534	**56.5788**	31.0077
0.03	23.26730672	25.89718754	14.38510358	14.81420824	28.7051	28.6848	**56.4793**	29.6027
0.04	22.27109221	25.18044098	14.38160352	14.82626439	27.47	27.889	**56.3803**	28.3676
0.05	21.26155918	24.39745863	14.37769422	14.8426384	26.5216	27.2026	**56.2767**	27.4192
0.06	20.56075654	23.80177986	14.37621712	14.85535158	25.7452	26.6094	**56.1949**	26.6428
0.07	19.88964448	23.19690784	14.37142899	14.8629413	24.9309	25.9174	**56.1168**	25.8285
0.08	19.33853222	22.70168993	14.36544808	14.86404392	24.3391	25.3977	**56.0063**	25.2367
0.09	18.7862669	22.14010706	14.36179056	14.87324719	23.737	24.8395	**55.9329**	24.6346
0.1	18.25741112	21.60072109	14.3492428	14.8700205	23.1435	24.2935	**55.8533**	24.0411

Table 5. RMSE values of Gaussian noise

Noise Density	1D Gaussian	2D Gaussian	1D Derivative	2D Derivative	AGD-1D	AGD-2 D	AFGD-1D	AFGD-2D
0.01	14.45374646	13.49420509	47.75976741	45.82554865	9.2092	9.8082	**6.3197**	13.2281
0.02	19.34256992	16.83674224	47.47599651	45.51995165	12.4663	12.2058	**9.5768**	13.3152
0.03	22.99063698	19.4753713	47.34050229	45.36223612	15.0327	14.2439	**12.1432**	13.3733
0.04	26.17116478	21.83692253	47.2459628	45.23099926	17.1824	16.9282	**14.2929**	13.4173
0.05	28.94662331	23.98696408	47.19347487	45.1444431	19.0999	17.6862	**15.2104**	13.4539
0.06	31.55328362	26.06022579	47.13545949	45.02103329	21.082	19.4344	**16.1925**	13.4786
0.07	33.91014375	27.91011346	47.13329082	45.01382967	22.7703	20.9124	**19.8808**	13.5032
0.08	36.08758531	29.63753666	47.1157946	44.99458047	24.3868	22.389	**21.4973**	13.5209
0.09	38.07428237	31.28258924	47.10518741	44.93996311	25.8149	23.6889	**22.9254**	13.5402
0.1	40.03755472	32.91757745	47.1501727	44.96015887	27.3009	25.0351	**24.4114**	13.5549

Table 6. PSNR value of gaussian noise

Noise Density	1D Gaussian	2D Gaussian	1D Derivative	2D Derivative	AGD-1D	AGD-2 D	AFGD-1D	AFGD-2D
0.01	24.93119496	25.52785748	14.54955955	14.90865014	28.8464	28.2991	**55.608**	30.6094
0.02	22.4005201	23.60564234	14.60132182	14.96676777	26.2161	26.3995	**55.0532**	27.9791
0.03	20.89978353	22.34108869	14.62614638	14.99691451	24.5901	25.0583	**54.6862**	26.3531
0.04	19.77434257	21.34697494	14.64350954	15.02207996	23.4291	24.0331	**54.41**	25.1921
0.05	18.89884542	20.53129791	14.65316449	15.03871761	22.5102	23.1781	**54.1808**	24.2732
0.06	18.14991238	19.81124012	14.66384871	15.06249444	21.6526	22.3594	**54.0272**	23.4156
0.07	17.52421099	19.21557157	14.66424835	15.06388434	20.9835	21.7227	**53.8746**	22.7465
0.08	16.98364714	18.69396153	14.66747321	15.06759947	20.3877	21.1301	**53.7645**	22.1507
0.09	16.51816908	18.22474976	14.66942889	15.07814938	19.8934	20.6399	**53.6455**	21.6564
0.1	16.0814527	17.78224629	14.66113785	15.07424686	19.4073	20.1598	**53.5545**	21.1703

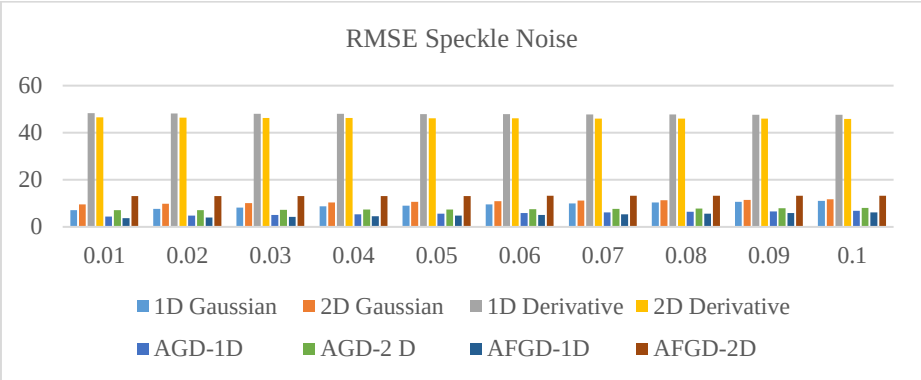


Figure 4. RMSE of different filters adding speckle noise

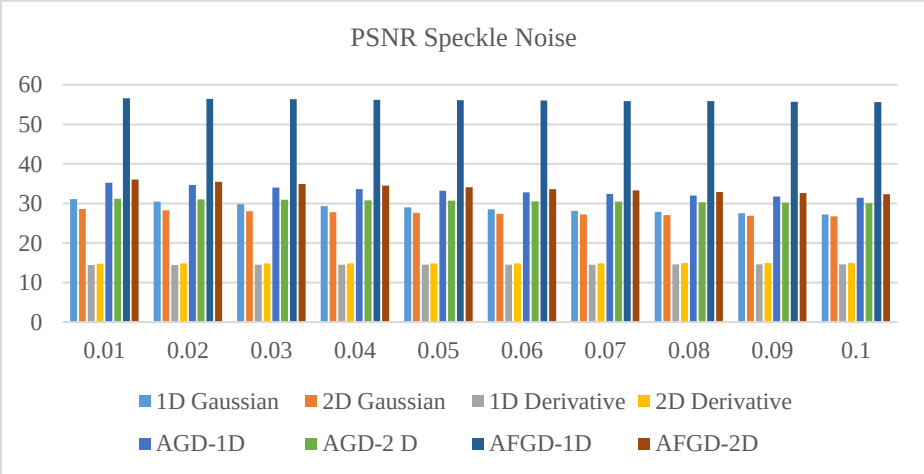


Figure 5. PSNR of different filters added speckle noise

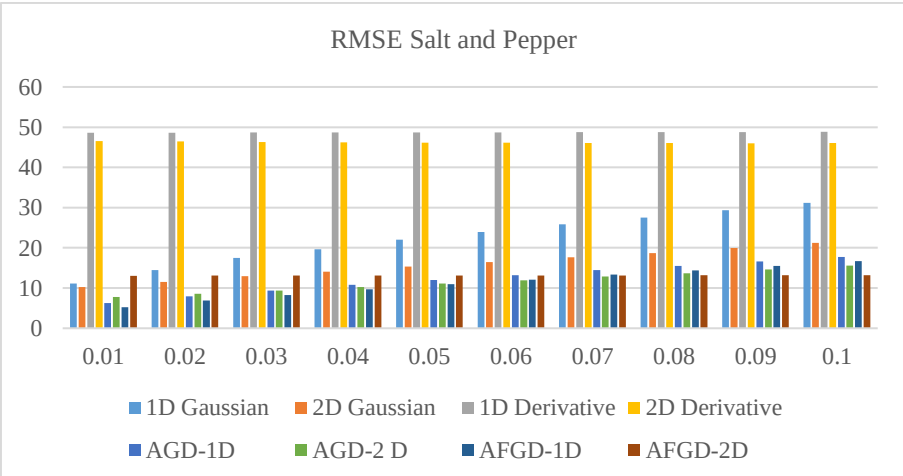


Figure 6. RMSE of different filters added salt and pepper noise

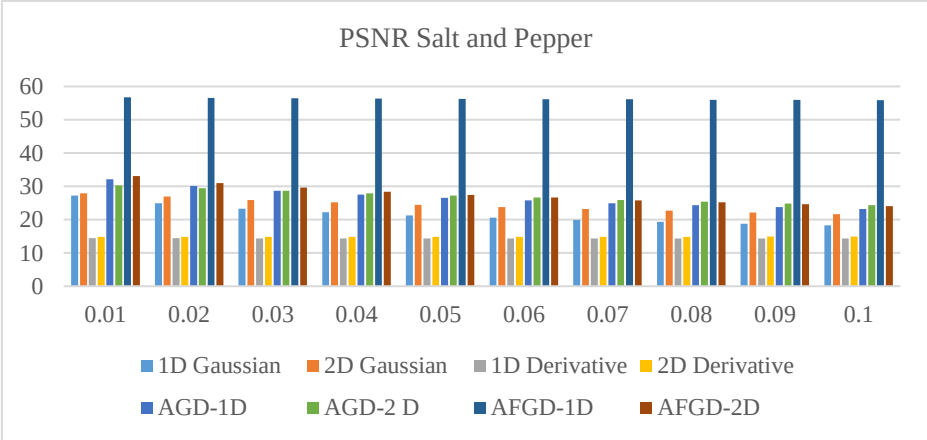


Figure 7 PSNR of different filters added salt and pepper noise

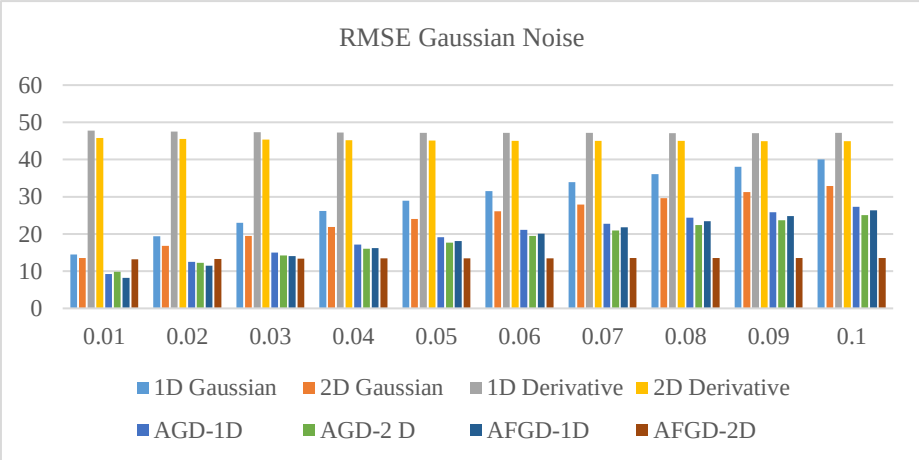


Figure 8. RMSE of different filters added gaussian noise

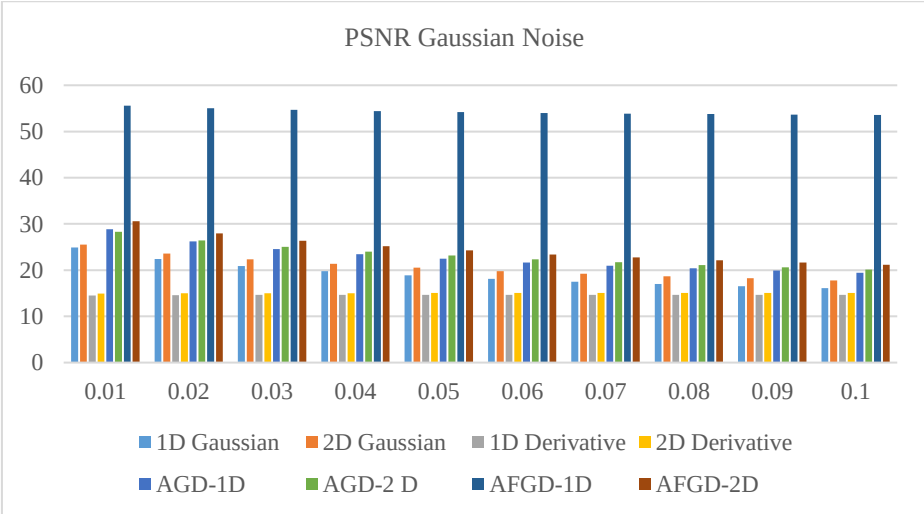


Figure 9. PSNR of different filters added salt and gaussian noise

An MRI image was taken and different noises were added and the RMSE and PSNR value where calculated and the results are shown in Tables 1-6 and their corresponding graphical representation is shown in Figures 4-9. The MATLAB software was used for the execution of algorithm. In this, four Adaptive gaussian filter is proposed namely AGD-1D, AGD-2D, AFGD-1D, AFGD-2D. The performance of the filter is calculated using the statistical method PSNR and RMSE. The calculated value is compared with some basic gaussian filter namely Gaussian 1D filter, Gaussian 2D filter. Derivative of Gaussian 1D filter and Derivative of Gaussian 2D filter. The results are shown in the Tables 1-6. From the tabulation it can be concluded that the adaptive filter works effectively. Also, the performance of AFGD-1D gives the best results at different level of noises. It can also be noticed that the AFGD-1D filter works better in different noises.

8. CONCLUSION AND FUTURE WORK

The major objective of this work is to design an Adaptive Gaussian Filter for denoising technique and to compare the performance of the four adaptive gaussian filter with each other and also with gaussian traditional filters. An MRI image is considered for this proposed methodology. The denoising filters are used at different noise intensity level. This paper mainly focuses on the pre-processing level of image processing. The performance of the filters was analyzed using the statistical values like RMSE and PSNR. From the results, it is evident that the adaptive filter namely AFGD-1D works better in comparison of other gaussian filters at different noise intensity level. Further this can be extended to other level of image processing like image classification and image segmentation. Also more fuzzy logic techniques can be applied in other stages for reducing the uncertainty.

REFERENCES

[1] Suneetha, A., Srinivasa Reddy, E. (2020). Robust Gaussian noise detection and removal in color images using modified fuzzy set filter. Journal of Intelligent Systems, 30(1): 240-257. https://doi.org/10.1515/jisys-2019-0211

[2] Kaur, A., Dong, G. (2023). A complete review on image denoising techniques for medical images. Neural Processing Letters, 55(6): 7807-7850. https://doi.org/10.1007/s11063-023-11286-1

[3] Atanassov, K. (1986). Intuitionistic fuzzy sets. Fuzzy Sets and Systems, 20: 87-96.

[4] Kumar, S.S. (2018). Medical image segmentation by modified fuzzy logic algorithm. International Journal of Advanced Research Trends in Engineering and Technology, 5.

[5] Harishvijey, A., Raja, J.B. (2022). Automated technique for EEG signal processing to detect seizure with optimized variable gaussian filter and fuzzy RBFELM classifier. Biomedical Signal Processing and Control, 74: 103450. https://doi.org/10.1016/j.bspc.2021.103450

[6] Kulikova, M.V., Kulikov, G.Y. (2023). On derivative-free extended Kalman filtering and its Matlab-oriented square-root implementations for state estimation in continuous-discrete nonlinear stochastic systems. European Journal of Control, 73: 100886. https://doi.org/10.1016/j.ejcon.2023.100886

[7] Zaynidinov, H., Juraev, J., Juraev, U. (2020). Digital image processing with two-dimensional Haar wavelets. International Journal of Advanced Trends in Computer Science and Engineering, 9(3): 38932020. https://doi.org/10.30534/ijatcse/2020/38932020

[8] Kumar, S.S. (2015). Analysis of brain tumor using undecimated wavelet transform and neural network. International Journal of Applied Engineering Research

[9] Zhu, W. (2019). A first-order image denoising model for staircase reduction. Advances in Computational Mathematics, 45(5): 3217-3239. https://doi.org/10.1007/s10444-019-09734-5

[10] Siddig, A., Guo, Z., Zhou, Z., Wu, B. (2018). An image denoising model based on a fourth-order nonlinear partial differential equation. Computers & Mathematics with Applications, 76(5): 1056-1074. https://doi.org/10.1016/j.camwa.2018.05.040

[11] Palma, C.A., Cappabianco, F.A., Ide, J.S., Miranda, P.A. (2014). Anisotropic diffusion filtering operation and limitations-magnetic resonance imaging evaluation. IFAC Proceedings Volumes, 47(3): 3887-3892. https://doi.org/10.3182/20140824-6-ZA-1003.02347

[12] Yuan, J., He, G. (2008). Application of an anisotropic diffusion based preprocessing filtering algorithm for high resolution remote sensing image segmentation. In 2008 Congress on Image and Signal Processing, Sanya, China, pp. 629-633. https://doi.org/10.1109/CISP.2008.318

[13] Shahin, A.I., Amin, K.M., Sharawi, A.A., Guo, Y. (2018). A novel enhancement technique for pathological microscopic image using neutrosophic similarity score scaling. Optik, 161: 84-97. https://doi.org/10.1016/j.ijleo.2018.02.026

[14] Khan, M., Rehman, R.F.U., Anis, S., Zeeshan, M. (2022). Denoising data in signal processing under the complex fuzzy environment. Preprint. https://doi.org/10.21203/rs.3.rs-1580891/v1

[15] Ali, H.M. (2018). MRI medical image denoising by fundamental filters. High-Resolution Neuroimaging-Basic Physical Principles and Clinical Applications, 14: 111-124.

[16] Kadhim, M.A. (2021). Restoration medical images from speckle noise using multifilters. In 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, pp. 1958-1963. https://doi.org/10.1109/ICACCS51430.2021.9441814

[17] Fan, L., Zhang, F., Fan, H., Zhang, C. (2019). Brief review of image denoising techniques. Visual Computing for Industry, Biomedicine, and Art, 2(1): 7.

[18] Simoncelli, E.P. (1994). Design of multi-dimensional derivative filters. In Proceedings of 1st International Conference on Image Processing, Austin, TX, USA, pp. 790-794. https://doi.org/10.1109/ICIP.1994.413423

[19] Zhang, J., He, X., Zhou, D. (2016). Generalised proportional–integral–derivative filter. IET Control Theory & Applications, 10(17): 2339-2347. https://doi.org/10.1049/iet-cta.2015.0610

[20] Kopparapu, S.K., Satish, M. (2011). Identifying optimal gaussian filter for gaussian noise removal. In 2011 Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, Hubli, India, pp. 126-129. https://doi.org/10.1109/NCVPRIPG.2011.34

[21] Deng, G., Cahill, L.W. (1993). An adaptive Gaussian filter for noise reduction and edge detection. In 1993 IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference, Francisco, CA, USA, pp. 1615-1619. https://doi.org/10.1109/NSSMIC.1993.373563

[22] D'Haeyer, J.P.F. (1989). Gaussian filtering of images: A regularization approach. Signal Processing, 18(2): 169-181. https://doi.org/10.1016/0165-1684(89)90048-0

[23] Keilmann, A., Godehardt, M., Moghiseh, A., Redenbach, C., Schladitz, K. (2023). Improved anisotropic gaussian filters, computer vision and pattern recognition. arXiv:2303.13278. https://doi.org/10.48550/arXiv.2303.13278

[24] Singh, A.K. (2020). Major development under Gaussian filtering since unscented Kalman filter. IEEE/CAA Journal of Automatica Sinica, 7(5): 1308-1325. https://doi.org/10.1109/JAS.2020.1003303

[25] Paul, S. (2024). A gaussian filter-based window for

efficient SAR image matching. Journal of Spatial Science, 69(2): 681-697. https://doi.org/10.1080/14498596.2023.2274998
[26] Shanmugam, A., Devi, S.R. (2020). A fuzzy model for noise estimation in magnetic resonance images. IRBM, 41(5): 261-266. https://doi.org/10.1016/j.irbm.2019.11.005
[27] Zhao, W., Dong, M., Wang, Y., Tan, R., Wu, T. (2024). A novel spectral super-resolution network with dominant information between spatial and spectral domains. Neurocomputing, 590: 127753. https://doi.org/10.1016/j.neucom.2024.127753