

Article

# Image Denoising Method Relying on Iterative Adaptive Weight-Mean Filtering

Meixia Wang \*, Susu Wang, Xiaoqin Ju and Yanhong Wang

School of Intelligent Manufacturing and Information, Jiangsu Shipping College, Nantong 226010, China

\* Correspondence: wangmx@jssc.edu.cn

**Abstract:** Salt-and-pepper noise (SPN) is a common type of image noise that appears as randomly distributed white and black pixels in an image. It is also known as impulse noise or random noise. This paper aims to introduce a new weighted average based on the Atangana–Baleanu fractional integral operator, which is a well-known idea in fractional calculus. Our proposed method also incorporates the concept of symmetry in the window mask structures, resulting in efficient and easily implementable filters for real-time applications. The distinguishing point of these techniques compared to similar methods is that we employ a novel idea for calculating the mean of regular pixels rather than the existing used mean formula along with the median. An iterative procedure has also been provided to integrate the power of removing high-density noise. Moreover, we will explore the different approaches to image denoising and their effectiveness in removing noise from images. The symmetrical structure of this tool will help in the ease and efficiency of these techniques. The outputs are compared in terms of peak signal-to-noise ratio, the mean-square error and structural similarity values. It was found that our proposed methodologies outperform some well-known compared methods. Moreover, they boast several advantages over alternative denoising techniques, including computational efficiency, the ability to eliminate noise while preserving image features, and real-time applicability.

**Keywords:** salt-and-pepper noise; noise removal; different adaptive mean filters; fractional means



**Citation:** Wang, M.; Wang, S.; Ju, X.; Wang, Y. Image Denoising Method Relying on Iterative Adaptive Weight-Mean Filtering. *Symmetry* **2023**, *15*, 1181. <https://doi.org/10.3390/sym15061181>

Academic Editor: Behzad Ghanbari

Received: 20 April 2023

Revised: 18 May 2023

Accepted: 26 May 2023

Published: 1 June 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Symmetry is a fundamental concept in mathematics that pertains to the behavior of functions when subjected to specific transformations or operations. If a functional equation exhibits symmetry, applying any element from the group will result in a valid solution to the problem. This property can be advantageous in problem solving because finding one solution enables us to derive several other solutions by repeatedly applying the same transformation or operation. Recent decades have seen the widespread use of symmetry in mathematical describing of important practical problems such as adaptive control [1,2], machine learning [3], pattern recognition [4], finding analytical solutions for partial differential equations [5], signal advancement [6], passivity control [7], time series analysis [8], telecommunication network [9], 3D imaging [10], nonlinear system identification [11], stochastic processes [12], optical fiber acoustic [13], UAV-based multiple oblique image flows [14], mathematical modeling, and prediction in infectious disease epidemiology [15,16]. For see more applications, please see [17–23].

Image processing involves various processes such as image denoising [24], image mosaic [25], image stitching [26], edge detecting [27], medical image registration [28], endoscopic imaging technology [29], depth estimation [30,31], feature extraction [32,33], classifying underwater images [34], image matching [35], and image inpainting [36]. It is generally acknowledged that symmetry is also a reliable tool in image processing for various purposes such as image compression, object recognition, shape detection, and image restoration. In image compression, the symmetric properties of images are utilized

to reduce storage or transmission requirements. By exploiting symmetrical patterns, only a fraction of the original image data needs to be stored while the rest can be reconstructed using mirroring or other techniques. In addition, in object recognition and shape detection, the identification and analysis of symmetric patterns and shapes in images are essential for the accurate classification and detection of objects. For example, symmetry-based algorithms can detect deviations from expected symmetric patterns, which can help in identifying potential defects or abnormalities in an image. Symmetry can also be employed to improve the visual quality of images by removing distortions or artifacts. Symmetric image processing methods can help in restoring distorted images to their original forms by performing operations such as bilinear interpolation or mirror image reflections.

There have been notable advancements in the development of image-denoising algorithms in recent years. Noise in an image can stem from various factors, with poor lighting conditions being a frequent culprit that results in low contrast and a lack of detail. Camera settings, including high ISO values or long exposure times, can also cause noise and lead to grainy or blurry images. Furthermore, noise may be introduced during image transmission or storage. Image denoising has been a topic of interest in the field of image processing for many years [37]. With the increasing use of digital images in various fields, the need for high-quality images has become more important than ever before. However, images captured in real-world scenarios are often affected by noise, which can reduce the quality of the image and make it difficult to extract useful information [38]. In [39], a median-based filter was designed to remove SPN from digital images. The authors of [40] proposed an improved image-denoising algorithm based on the TV model. The concept of local fractional entropy was applied in [41] to design an efficient fractional-based mask in image denoising. Recently, deep learning-based methods have shown great promise in image denoising [42,43]. These methods use neural networks to learn the underlying structure of the noise and to remove it from the image. One popular approach is the use of convolutional neural networks, which are effective in removing various types of noise from images [44]. Another approach is the use of generative adversarial networks for image denoising [45]. These techniques consist of two neural networks: a generator network that generates fake images and a discriminator network that tries to distinguish between real and fake images. By training these networks together, these networks can learn to generate high-quality images that are free from noise. One of the most traditional methods for image denoising includes filters such as median filters [46] and mean filters [47] and using symmetric window masks in image processing. These filters can also be customized to target specific types of noise, such as Gaussian or salt-and-pepper noise. However, these methods have limitations when it comes to preserving important image details and textures [48–51]. Notably, the context of symmetry is often present in image processing filter masks used for image denoising. For example, the popular Gaussian filter mask [52] is rotationally symmetric, meaning that it produces the same result when rotated around its center point. This symmetry helps to ensure that the filter produces consistent results across the image and reduces the computational complexity of the denoising operation. Other denoising filters, such as median filters, may not have rotational symmetry but may have reflectional symmetry, which also helps to ensure consistent results and to reduce computational complexity.

The primary objective of this paper is to introduce and to elucidate a series of original techniques that leverage the utilization of Atangana–Baleanu fractional operators with noninteger orders for the purposes of mitigating salt-and-pepper noise from digital images. Through our study, we aim to offer a comprehensive analysis of the efficacy and potential applications of these novel techniques in the domain of image denoising. To the best of our knowledge, the approach proposed in this study has not been previously explored in the existing literature. Based on the findings of our experimental analysis, we contend that this novel method holds significant promise as a viable solution for creating effective filters in the domain of image denoising. The general structure of this article is as follows. In the next section, we will review some basic definitions related to fractional differential calculus.

Different structures for filters in image denoising that are used in this article are designed in the third section of the article. Moreover, an overview of some discretizations for the fractional integral operator of the Atangana–Baleanu type will be discussed in Section 4. The main algorithm of the paper is presented in Section 5. Numerical simulations and comparisons of results are given in Section 6. In conclusion, a summary of key findings and insights gleaned from our investigation is presented in the final section of this article. These conclusions serve to encapsulate the key takeaways from our study and offer valuable insights for future research efforts in the field of image denoising.

## 2. A Summary of Some Well-Known Fractional Operators

This section includes a short overview of some basic definitions presented in fractional calculus, which are widely used in the literature.

- The Liouville–Caputo derivative [53]:

$$D_{LC}^{\varphi} \mathcal{H}(\tau) = \frac{1}{\Gamma(1-\varphi)} \int_0^{\tau} (\tau-\phi)^{-\varphi} \dot{\mathcal{H}}(\phi) d\phi, \quad 0 < \varphi \leq 1. \quad (1)$$

- The Caputo–Fabrizio derivative [54]:

$$D_{CF}^{\varphi} \mathcal{H}(\tau) = \frac{(2-\varphi)\mathcal{J}(\varphi)}{2(1-\varphi)} \int_0^{\tau} \exp\left[-\varphi \frac{(\tau-\phi)}{1-\varphi}\right] \dot{\mathcal{H}}(\phi) d\phi, \quad 0 < \varphi < 1, \quad (2)$$

where  $\mathcal{J}(\varphi) = \varphi/(2-\varphi)$ .

- The Atangana–Baleanu fractional derivative in the Caputo sense [55]:

$$D_{ABC}^{\varphi} \mathcal{H}(\tau) = \frac{\mathcal{J}(\varphi)}{1-\varphi} \int_0^{\tau} \text{ML}_{\varphi}\left[-\varphi \frac{(\tau-\phi)^{\varphi}}{1-\varphi}\right] \dot{\mathcal{H}}(\phi) d\phi, \quad 0 < \varphi \leq 1, \quad (3)$$

where  $\text{ML}_{\varphi}(\cdot)$  stands for the well-known Mittag–Leffler function given by  $\text{ML}_{\varphi}(\tau) = \sum_{k=0}^{\infty} \frac{\tau^k}{\Gamma(\varphi k + 1)}$ .

- The Atangana–Baleanu fractional integral in the Caputo sense [55]:

$$I_{ABC}^{\varphi} \mathcal{H}(\tau) = \frac{1-\varphi}{\mathcal{J}(\varphi)} \mathcal{H}(\tau) + \frac{\varphi}{\Gamma(\varphi)\mathcal{J}(\varphi)} \int_0^{\tau} \mathcal{H}(\phi)(\tau-\phi)^{\varphi-1} d\phi, \quad 0 < \varphi \leq 1, \quad (4)$$

where  $\mathcal{J}(\cdot)$  is a function defined by  $\mathcal{J}(\varphi) = 1 - \varphi + \varphi/\Gamma(\varphi)$ .

## 3. An Overview of the Atangana–Baleanu Fractional Masks

Let us assume that a given fractional integral can be approximated at a point, using unit time-step length, in the following discrete form

$$I^{\varphi} \mathcal{H}(\tau) \approx \rho_0 \mathcal{H}(\tau) + \rho_1 \mathcal{H}(\tau-1) + \rho_2 \mathcal{H}(\tau-2) + \rho_3 \mathcal{H}(\tau-3) + \rho_4 \mathcal{H}(\tau-4) + \rho_5 \mathcal{H}(\tau-5) + \dots, \quad (5)$$

where  $\rho_1, \rho_2, \dots, \rho_5$  are the first few coefficients of the corresponding expansion of the fractional operator. Further, this idea can be also utilized in a multivariate case such as

$$\begin{aligned}
 {}^x I^\nu \mathcal{H}(x, t) &\approx \rho_0 \mathcal{H}(x, t) + \rho_1 \mathcal{H}(x - 1, y) + \rho_2 \mathcal{H}(x - 2, y) + \rho_3 \mathcal{H}(x - 3, y) + \rho_4 \mathcal{H}(x - 4, y) + \rho_5 \mathcal{H}(x - 5, y) + \dots, \\
 {}^y I^\nu \mathcal{H}(x, t) &\approx \rho_0 \mathcal{H}(x, t) + \rho_1 \mathcal{H}(x, t - 1) + \rho_2 \mathcal{H}(x, t - 2) + \rho_3 \mathcal{H}(x, t - 3) + \rho_4 \mathcal{H}(x, t - 4) + \rho_5 \mathcal{H}(x, t - 5) + \dots
 \end{aligned}
 \tag{6}$$

The obtained symmetric coefficients can be used in the design of masks in various applications of image processing. One of the possible arrangements for these masks is the following designs using different dimensions:

- For an  $3 \times 3$  fractional integral mask, we introduce the following symmetric window mask

$$\Omega^3 = [\omega_{i,j}^3] := \begin{bmatrix} \rho_1 & \rho_1 & \rho_1 \\ \rho_1 & 8\rho_0 & \rho_1 \\ \rho_1 & \rho_1 & \rho_1 \end{bmatrix}.$$

- For an  $5 \times 5$  fractional integral mask, we construct the following symmetric integral mask

$$\Omega^5 = [\omega_{i,j}^5] := \begin{bmatrix} \rho_2 & \rho_2 & \rho_2 & \rho_2 & \rho_2 \\ \rho_2 & \rho_1 & \rho_1 & \rho_1 & \rho_2 \\ \rho_2 & \rho_1 & 8\rho_0 & \rho_1 & \rho_2 \\ \rho_2 & \rho_1 & \rho_1 & \rho_1 & \rho_2 \\ \rho_2 & \rho_2 & \rho_2 & \rho_2 & \rho_2 \end{bmatrix}.$$

- In addition, for an  $7 \times 7$  fractional mask, the following symmetric structure is considered

$$\Omega^7 = [\omega_{i,j}^7] := \begin{bmatrix} \rho_3 & \rho_3 & \rho_3 & \rho_3 & \rho_3 & \rho_3 & \rho_3 \\ \rho_3 & \rho_2 & \rho_2 & \rho_2 & \rho_2 & \rho_2 & \rho_3 \\ \rho_3 & \rho_2 & \rho_1 & \rho_1 & \rho_1 & \rho_2 & \rho_3 \\ \rho_3 & \rho_2 & \rho_1 & 8\rho_0 & \rho_1 & \rho_2 & \rho_3 \\ \rho_3 & \rho_2 & \rho_1 & \rho_1 & \rho_1 & \rho_2 & \rho_3 \\ \rho_3 & \rho_2 & \rho_2 & \rho_2 & \rho_2 & \rho_2 & \rho_3 \\ \rho_3 & \rho_3 & \rho_3 & \rho_3 & \rho_3 & \rho_3 & \rho_3 \end{bmatrix}.$$

- For an  $9 \times 9$  fractional mask, the following symmetric windows mask is proposed

$$\Omega^9 = [\omega_{i,j}^9] := \begin{bmatrix} \rho_4 & \rho_4 \\ \rho_4 & \rho_3 & \rho_4 \\ \rho_4 & \rho_3 & \rho_2 & \rho_2 & \rho_2 & \rho_2 & \rho_2 & \rho_3 & \rho_4 \\ \rho_4 & \rho_3 & \rho_2 & \rho_1 & \rho_1 & \rho_1 & \rho_2 & \rho_3 & \rho_4 \\ \rho_4 & \rho_3 & \rho_2 & \rho_1 & 8\rho_0 & \rho_1 & \rho_2 & \rho_3 & \rho_4 \\ \rho_4 & \rho_3 & \rho_2 & \rho_1 & \rho_1 & \rho_1 & \rho_2 & \rho_3 & \rho_4 \\ \rho_4 & \rho_3 & \rho_2 & \rho_2 & \rho_2 & \rho_2 & \rho_2 & \rho_3 & \rho_4 \\ \rho_4 & \rho_3 & \rho_4 \\ \rho_4 & \rho_4 \end{bmatrix}.$$

- Moreover, an  $11 \times 11$  fractional mask can be constructed similarly in a symmetric form as

$$\Omega^{11} = [\omega_{ij}^{11}] := \begin{matrix} \begin{matrix} \rho_5 & \rho_5 \\ \rho_5 & \rho_4 & \rho_5 \\ \rho_5 & \rho_4 & \rho_3 & \rho_4 & \rho_5 \\ \rho_5 & \rho_4 & \rho_3 & \rho_2 & \rho_2 & \rho_2 & \rho_2 & \rho_2 & \rho_3 & \rho_4 & \rho_5 \\ \rho_5 & \rho_4 & \rho_3 & \rho_2 & \rho_1 & \rho_1 & \rho_1 & \rho_2 & \rho_3 & \rho_4 & \rho_5 \\ \rho_5 & \rho_4 & \rho_3 & \rho_2 & \rho_1 & \rho_1 & \rho_1 & \rho_2 & \rho_3 & \rho_4 & \rho_5 \\ \rho_5 & \rho_4 & \rho_3 & \rho_2 & \rho_2 & \rho_2 & \rho_2 & \rho_2 & \rho_3 & \rho_4 & \rho_5 \\ \rho_5 & \rho_4 & \rho_3 & \rho_4 & \rho_5 \\ \rho_5 & \rho_4 & \rho_5 \\ \rho_5 & \rho_5 \end{matrix} \\ \end{matrix}.$$

The coefficients  $\rho$  in these filters will all be determined according to the results of the next section. In addition, the proposed higher-order fractional filters are used in the rest of the article, especially in the case of high noise in the images.

### 4. Some Discretizations in Determining the Approximation of the AB Integral Operator

#### 4.1. Fractional Mask Based on the Grunwald–Letnikov Idea (AB1)

**Definition 1.** One of the most common discrete forms for derivatives of fractional order, which have many different applications, is the Grunwald–Letnikov (GL) derivative with the following definition [56]

$$\begin{aligned} D_{GL}^\alpha \mathcal{H}(\tau) &\approx \frac{1}{\Gamma(-\alpha)} \int_0^\tau \frac{\mathcal{H}(\phi)}{(\tau - \phi)^{1+\alpha}} d\phi \\ &= \lim_{h \rightarrow 0} h^{-\alpha} \left( \mathcal{H}(\tau) + (-\alpha)\mathcal{H}(\tau - \delta) + \frac{\alpha(\alpha + 1)}{2}\mathcal{H}(\tau - \delta) + \dots + \frac{\Gamma(\alpha + 1)}{k!\Gamma(\alpha - N + 1)}\mathcal{H}(\tau - n\delta) \right), \end{aligned} \tag{7}$$

where  $\Gamma(z) = \int_0^\infty \exp(-v)v^{z-1}dv$  is the well-known Gamma function,  $\alpha \in \mathbb{R}^+$  and  $N = [t/\delta]$ . Using Equation (7) with  $\varphi = -\alpha > 0$ , the corresponding integral definition of Grunwald–Letnikov is obtained as

$$\begin{aligned} I_{GL}^\varphi \mathcal{H}(\tau) &= \frac{1}{\Gamma(\varphi)} \int_0^\tau \frac{\mathcal{H}(\phi)}{(\tau - \phi)^{1-\varphi}} d\phi \\ &\approx \lim_{h \rightarrow 0} h^{-\varphi} \left( \mathcal{H}(\tau) + \varphi\mathcal{H}(\tau - \delta) + \frac{(-\varphi)(-\varphi + 1)}{2}\mathcal{H}(\tau - \delta) + \dots + \frac{\Gamma(-\varphi + 1)}{N!\Gamma(-\varphi - N + 1)}\mathcal{H}(\tau - n\delta) \right). \end{aligned} \tag{8}$$

Now, reconsider the definition of the AB-fractional integral defined in Equation (4), as

$$I_{AB}^\varphi \mathcal{H}(\tau) = \frac{1 - \varphi}{\mathcal{J}(\varphi)} \mathcal{H}(\tau) + \frac{\varphi}{\Gamma(\varphi)\mathcal{J}(\varphi)} \int_0^\tau \mathcal{H}(\phi)(\tau - \phi)^{\varphi-1} d\phi. \tag{9}$$

A closer look at with Equation (8) reveals that

$$I_{AB}^\varphi \mathcal{H}(\tau) = \frac{1 - \varphi}{\mathcal{J}(\varphi)} \mathcal{H}(\tau) + \frac{\varphi}{\mathcal{J}(\varphi)} I_{GL}^\varphi \mathcal{H}(\tau). \tag{10}$$

Thus, we can write

$$\begin{aligned} I_{AB}^\varphi \mathcal{H}(\tau) &= \frac{1 - \varphi}{\mathcal{J}(\varphi)} \mathcal{H}(\tau) + \frac{\varphi}{\Gamma(\varphi)\mathcal{J}(\varphi)} \int_0^\tau \mathcal{H}(\phi)(\tau - \phi)^{\varphi-1} d\phi, \\ &\approx \frac{1 - \varphi}{\mathcal{J}(\varphi)} \mathcal{H}(\tau) + \frac{\varphi}{\mathcal{J}(\varphi)} h^{-\varphi} \left( \mathcal{H}(\tau) + \varphi\mathcal{H}(\tau - \delta) + \frac{(-\varphi)(-\varphi + 1)}{2}\mathcal{H}(\tau - \delta) + \dots \right). \end{aligned} \tag{11}$$

In this way, it reads

$$\begin{aligned}
 {}^x I_{AB-1}^\varphi \mathcal{H}(x, y) &\approx \frac{1}{\mathcal{J}(\varphi)} \mathcal{H}(x, y) + \frac{\varphi^2}{\mathcal{J}(\varphi)} \mathcal{H}(x-1, y) + \frac{\varphi^3 - \varphi^2}{2 \mathcal{J}(\varphi)} \mathcal{H}(x-2, y) + \frac{\varphi^4 - 3\varphi^3 + 2\varphi^2}{6 \mathcal{J}(\varphi)} \mathcal{H}(x-3, y) \\
 &+ \frac{\varphi^5 - 6\varphi^4 + 11\varphi^3 - 6\varphi^2}{24 \mathcal{J}(\varphi)} \mathcal{H}(x-4, y) + \frac{\varphi^6 - 10\varphi^5 + 35\varphi^4 - 50\varphi^3 + 24\varphi^2}{120 \mathcal{J}(\varphi)} \mathcal{H}(x-5, y) + \dots, \\
 {}^y I_{AB-1}^\varphi \mathcal{H}(x, y) &\approx \frac{1}{\mathcal{J}(\varphi)} \mathcal{H}(x, y) + \frac{\varphi^2}{\mathcal{J}(\varphi)} \mathcal{H}(x, y-1) + \frac{\varphi^3 - \varphi^2}{2 \mathcal{J}(\varphi)} \mathcal{H}(x, y-2) + \frac{\varphi^4 - 3\varphi^3 + 2\varphi^2}{6 \mathcal{J}(\varphi)} \mathcal{H}(x, y-3) \\
 &+ \frac{\varphi^5 - 6\varphi^4 + 11\varphi^3 - 6\varphi^2}{24 \mathcal{J}(\varphi)} \mathcal{H}(x, y-4) + \frac{\varphi^6 - 10\varphi^5 + 35\varphi^4 - 50\varphi^3 + 24\varphi^2}{120 \mathcal{J}(\varphi)} \mathcal{H}(x, y-5) + \dots
 \end{aligned}
 \tag{12}$$

Hence, the coefficients of the Atangana–Baleanu fractional integral expansion are determined as follows

$$\begin{aligned}
 \rho_0 &= \frac{1}{\mathcal{J}(\varphi)}, \\
 \rho_1 &= \frac{\varphi^2}{\mathcal{J}(\varphi)}, \\
 \rho_2 &= \frac{\varphi^3 - \varphi^2}{2 \mathcal{J}(\varphi)}, \\
 \rho_3 &= \frac{\varphi^4 - 3\varphi^3 + 2\varphi^2}{6 \mathcal{J}(\varphi)}, \\
 \rho_4 &= \frac{\varphi^5 - 6\varphi^4 + 11\varphi^3 - 6\varphi^2}{24 \mathcal{J}(\varphi)}, \\
 \rho_5 &= \frac{\varphi^6 - 10\varphi^5 + 35\varphi^4 - 50\varphi^3 + 24\varphi^2}{120 \mathcal{J}(\varphi)}.
 \end{aligned}
 \tag{13}$$

■ Using coefficients in Equation (13), the so-called fractional AB1 masks of different sizes including  $\Omega_1, \Omega_2, \Omega_3, \Omega_4$  can be characterized.

4.2. Fractional Mask Based on the Toufik–Atangana Idea (AB2)

The following iterative scheme in determining the approximation of the AB integral operator is suggested as follows [57]

$$\begin{aligned}
 I_{AB}^\varphi \mathcal{H}(\tau_n) &= \frac{1-\varphi}{\mathcal{J}(\varphi)} \mathcal{H}(\tau_n) + \frac{\varphi}{B(\varphi)} \sum_{s=0}^n \left( \frac{h^\varphi \mathcal{H}(\tau_s)}{\Gamma(\varphi+2)} \left( (n+1-s)^\varphi \times (n-s+2+\varphi) - (n-s)^\varphi (n-s+2+2\varphi) \right) \right. \\
 &\quad \left. - \frac{h^\varphi \mathcal{H}(\tau_{s-1})}{\Gamma(\varphi+2)} \left( (n+1-s)^{\varphi+1} - (n-s)^\varphi (n-s+1+\varphi) \right) \right).
 \end{aligned}
 \tag{14}$$

Hence, Equation (14) can be rewritten as

$$\begin{aligned}
 I_{AB}^\varphi \mathcal{H}(\tau_n) &= \left[ \frac{(1-\varphi)\Gamma(\varphi+2) + \varphi h^\varphi (\varphi+2)}{\mathcal{J}(\varphi)\Gamma(\varphi+2)} \right] \mathcal{H}(\tau_n) + \left[ \frac{\varphi h^\varphi}{\mathcal{J}(\varphi)} \left( \frac{-2\varphi + (\varphi+3)2^\varphi - 4}{\Gamma(\varphi+2)} \right) \right] \mathcal{H}(\tau_{n-1}) + \\
 &\quad \left[ \frac{(-2^{\varphi+1}\varphi + (\varphi+4)3^\varphi + \varphi - 6 \times 2^\varphi + 2)}{\mathcal{J}(\varphi)\Gamma(\varphi+2)} \right] \mathcal{H}(\tau_{n-2}) + \dots
 \end{aligned}
 \tag{15}$$

Thus, we have the following forms

$$\begin{aligned}
 x I_{AB-2}^{\varphi} \mathcal{H}(x, y) &\approx \left[ \frac{(1-\varphi)\Gamma(\varphi+2) + \varphi(\varphi+2)}{\mathcal{J}(\varphi)\Gamma(\varphi+2)} \right] \mathcal{H}(x, y) + \left[ \frac{(\varphi^2+3\varphi)2^{\varphi} - 2\varphi^2 - 4\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi+2)} \right] \mathcal{H}(x-1, y) \\
 &+ \left[ \frac{(\varphi^2+4\varphi)3^{\varphi} + (-2\varphi^2-6\varphi)2^{\varphi} + \varphi^2 + 2\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi+2)} \right] \mathcal{H}(x-2, y) \\
 &+ \left[ \frac{(\varphi^2+5\varphi)4^{\varphi} + (-2\varphi^2-8\varphi)3^{\varphi} + (\varphi^2+3\varphi)2^{\varphi}}{\mathcal{J}(\varphi)\Gamma(\varphi+2)} \right] \mathcal{H}(x-3, y) \\
 &+ \left[ \frac{(\varphi^2+6\varphi)5^{\varphi} + (-2\varphi^2-10\varphi)4^{\varphi} + (\varphi^2+4\varphi)3^{\varphi}}{\mathcal{J}(\varphi)\Gamma(\varphi+2)} \right] \mathcal{H}(x-4, y) \\
 &+ \left[ \frac{(\varphi^2+7\varphi)6^{\varphi} + (-2\varphi^2-12\varphi)5^{\varphi} + (\varphi^2+5\varphi)4^{\varphi}}{\mathcal{J}(\varphi)\Gamma(\varphi+2)} \right] \mathcal{H}(x-5, y) + \dots, \\
 y I_{AB-2}^{\varphi} \mathcal{H}(x, y) &\approx \left[ \frac{(1-\varphi)\Gamma(\varphi+2) + \varphi(\varphi+2)}{\mathcal{J}(\varphi)\Gamma(\varphi+2)} \right] \mathcal{H}(x, y) + \left[ \frac{(\varphi^2+3\varphi)2^{\varphi} - 2\varphi^2 - 4\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi+2)} \right] \mathcal{H}(x, y-1) \\
 &+ \left[ \frac{(\varphi^2+4\varphi)3^{\varphi} + (-2\varphi^2-6\varphi)2^{\varphi} + \varphi^2 + 2\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi+2)} \right] \mathcal{H}(x, y-2) \\
 &+ \left[ \frac{(\varphi^2+5\varphi)4^{\varphi} + (-2\varphi^2-8\varphi)3^{\varphi} + (\varphi^2+3\varphi)2^{\varphi}}{\mathcal{J}(\varphi)\Gamma(\varphi+2)} \right] \mathcal{H}(x, y-3) \\
 &+ \left[ \frac{(\varphi^2+6\varphi)5^{\varphi} + (-2\varphi^2-10\varphi)4^{\varphi} + (\varphi^2+4\varphi)3^{\varphi}}{\mathcal{J}(\varphi)\Gamma(\varphi+2)} \right] \mathcal{H}(x, y-4) \\
 &+ \left[ \frac{(\varphi^2+7\varphi)6^{\varphi} + (-2\varphi^2-12\varphi)5^{\varphi} + (\varphi^2+5\varphi)4^{\varphi}}{\mathcal{J}(\varphi)\Gamma(\varphi+2)} \right] \mathcal{H}(x, y-5) + \dots
 \end{aligned} \tag{16}$$

Therefore, the coefficients used in the so-called **AB2** masks of different sizes will be determined as follows

$$\begin{aligned}
 \rho_0 &= \frac{(1-\varphi)\Gamma(\varphi+2) + \varphi(\varphi+2)}{\mathcal{J}(\varphi)\Gamma(\varphi+2)}, \\
 \rho_1 &= \frac{(\varphi^2+3\varphi)2^{\varphi} - 2\varphi^2 - 4\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi+2)}, \\
 \rho_2 &= \frac{(\varphi^2+4\varphi)3^{\varphi} + (-2\varphi^2-6\varphi)2^{\varphi} + \varphi^2 + 2\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi+2)}, \\
 \rho_3 &= \frac{(\varphi^2+5\varphi)4^{\varphi} + (-2\varphi^2-8\varphi)3^{\varphi} + (\varphi^2+3\varphi)2^{\varphi}}{\mathcal{J}(\varphi)\Gamma(\varphi+2)}, \\
 \rho_4 &= \frac{(\varphi^2+6\varphi)5^{\varphi} + (-2\varphi^2-10\varphi)4^{\varphi} + (\varphi^2+4\varphi)3^{\varphi}}{\mathcal{J}(\varphi)\Gamma(\varphi+2)}, \\
 \rho_5 &= \frac{(\varphi^2+7\varphi)6^{\varphi} + (-2\varphi^2-12\varphi)5^{\varphi} + (\varphi^2+5\varphi)4^{\varphi}}{\mathcal{J}(\varphi)\Gamma(\varphi+2)}.
 \end{aligned} \tag{17}$$

4.3. Fractional Mask Based on Euler’s Method Idea (**AB3**)

Another possible approximation for the AB- fractional integral is derived from Euler’s method in the form of an iterative scheme as [58]

$$I_{AB}^{\varphi} \mathcal{H}(\tau_n) = \frac{1-\varphi}{\mathcal{J}(\varphi)} \mathcal{H}(\tau_n) + \frac{\varphi h^{\varphi}}{\mathcal{J}(\varphi)\Gamma(\varphi+1)} \sum_{s=0}^{n-1} \theta_{n,s} \mathcal{H}(\tau_s), \tag{18}$$

where

$$\theta_{n,s} = (n - s)^\varphi - (n - s - 1)^\varphi. \tag{19}$$

Equation (18) can be reformulated in the following equivalent manner

$$I_{AB}^\varphi \mathcal{H}(\tau_n) = \left[ \frac{1 - \varphi}{\mathcal{J}(\varphi)} \right] \mathcal{H}(\tau_n) + \left[ \frac{\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)} \right] \mathcal{H}(\tau_{n-1}) + \left[ \frac{2^\varphi - 1}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)} \right] \mathcal{H}(\tau_{n-2}) + \dots \tag{20}$$

Thus, it reads

$$\begin{aligned} {}^x I_{AB3}^\varphi \mathcal{H}(x, y) &\approx \left[ \frac{1 - \varphi}{\mathcal{J}(\varphi)} \right] \mathcal{H}(x, y) + \left[ \frac{\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)} \right] \mathcal{H}(x - 1, y) + \left[ \frac{2^\varphi - 1}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)} \right] \mathcal{H}(x - 2, y) \\ &+ \left[ \frac{3^\varphi - 2^\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)} \right] \mathcal{H}(x - 3, y) + \left[ \frac{4^\varphi - 3^\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)} \right] \mathcal{H}(x - 4, y) + \left[ \frac{5^\varphi - 4^\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)} \right] \mathcal{H}(x - 5, y) + \dots, \\ {}^y I_{AB3}^\varphi \mathcal{H}(x, y) &\approx \left[ \frac{1 - \varphi}{\mathcal{J}(\varphi)} \right] \mathcal{H}(x, y) + \left[ \frac{\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)} \right] \mathcal{H}(x, y - 1) + \left[ \frac{4^\varphi - 3^\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)} \right] \mathcal{H}(x, y - 2) \\ &+ \left[ \frac{3^\varphi - 2^\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)} \right] \mathcal{H}(x, y - 3) + \left[ \frac{2^\varphi - 1}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)} \right] \mathcal{H}(x, y - 4) + \left[ \frac{5^\varphi - 4^\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)} \right] \mathcal{H}(x, y - 5) + \dots \end{aligned} \tag{21}$$

Therefore, the coefficients used in the so-called **AB3 masks** of different sizes will be determined as follows

$$\begin{aligned} \rho_0 &= \frac{1 - \varphi}{\mathcal{J}(\varphi)}, \\ \rho_1 &= \frac{\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)}, \\ \rho_2 &= \frac{2^\varphi - 1}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)}, \\ \rho_3 &= \frac{3^\varphi - 2^\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)}, \\ \rho_4 &= \frac{4^\varphi - 3^\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)}, \\ \rho_5 &= \frac{5^\varphi - 4^\varphi}{\mathcal{J}(\varphi)\Gamma(\varphi + 1)}. \end{aligned} \tag{22}$$

#### 4.4. Fractional Mask Based on the Middle Point Idea (AB4)

Let us reconsider the definition of the AB- fractional integral defined in Equation (4), as

$$I_{AB}^\varphi \mathcal{H}(t) = \frac{1 - \varphi}{\mathcal{J}(\varphi)} \mathcal{H}(\tau) + \frac{\varphi}{\Gamma(\varphi)\mathcal{J}(\varphi)} \int_0^\tau \frac{\mathcal{H}(\phi)}{(\tau - \phi)^{1-\varphi}} d\phi. \tag{23}$$

Taking  $\phi = \tau - \phi$  into account in the integral Equation (23) yields

$$I_{AB}^\varphi \mathcal{H}(t) = \frac{1 - \varphi}{\mathcal{J}(\varphi)} \mathcal{H}(\tau) + \frac{\varphi}{\Gamma(\varphi)\mathcal{J}(\varphi)} \int_0^\tau \frac{\mathcal{H}(\tau - \phi)}{\phi^{1-\varphi}} d\phi. \tag{24}$$

Now, by dividing the integral given in Equation (24), we will have

$$I_{AB}^\varphi \mathcal{H}(t) = \frac{1 - \varphi}{\mathcal{J}(\varphi)} \mathcal{H}(\tau) + \frac{\varphi}{\Gamma(\varphi)\mathcal{J}(\varphi)} \sum_{k=0}^{n-1} \int_{t_k}^{\tau_{k+1}} \frac{\mathcal{H}(\tau - \phi)}{\phi^{1-\varphi}} d\phi. \tag{25}$$

Then, applying an approximation formula gives

$$\int_{t_k}^{\tau_{k+1}} \frac{\mathcal{H}(\phi)}{\mathcal{H}(\phi)} d\phi \approx \frac{\mathcal{H}(t_k) + \mathcal{H}(\tau_{k+1})}{2} \int_{t_k}^{\tau_{k+1}} \frac{d\phi}{\mathcal{H}(\phi)}, \tag{26}$$

in Equation (25), one obtains

$$\begin{aligned}
 I_{AB}^\varphi \mathcal{H}(t) &= \frac{1-\varphi}{\mathcal{J}(\varphi)} \mathcal{H}(\tau) + \frac{\varphi}{\Gamma(\varphi)\mathcal{J}(\varphi)} \sum_{k=0}^{n-1} \frac{\mathcal{H}(\tau-t_k) + \mathcal{H}(\tau-\tau_{k+1})}{2} \int_{t_k}^{\tau_{k+1}} \frac{d\phi}{\phi^{1-\varphi}}, \\
 &= \frac{1-\varphi}{\mathcal{J}(\varphi)} \mathcal{H}(\tau) + \frac{\varphi}{\Gamma(\varphi)\mathcal{J}(\varphi)} \sum_{k=0}^{n-1} \frac{\mathcal{H}(\tau-t_k) + \mathcal{H}(\tau-\tau_{k+1})}{2\varphi} \left[ \tau_{k+1}^\varphi - \tau_k^\varphi \right].
 \end{aligned}
 \tag{27}$$

Thus, we have

$$\begin{aligned}
 I_{AB}^\varphi \mathcal{H}(\tau_n) &= \frac{1-\varphi}{\mathcal{J}(\varphi)} \mathcal{H}(\tau_n) + \frac{\varphi}{\Gamma(\varphi)\mathcal{J}(\varphi)} \sum_{k=0}^{n-1} \frac{\mathcal{H}(\tau_n-t_k) + \mathcal{H}(\tau_n-\tau_{k+1})}{2\varphi} \left[ ((k+1)h)^\varphi - (kh)^\varphi \right], \\
 &= \frac{1-\varphi}{\mathcal{J}(\varphi)} \mathcal{H}(\tau_n) + \frac{1}{\Gamma(\varphi)\mathcal{J}(\varphi)} \sum_{k=0}^{n-1} \frac{\mathcal{H}(\tau_{n-k}) + \mathcal{H}(\tau_{n-k+1})}{2} \left[ ((k+1)h)^\varphi - (kh)^\varphi \right].
 \end{aligned}
 \tag{28}$$

Upon consolidating the aforementioned findings, we can assert that [59]

$$I_{AB}^\varphi \mathcal{H}(\tau_n) = \left[ \frac{2\Gamma(\varphi) - 2\varphi\Gamma(\varphi) + 1}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(\tau_n) + \left[ \frac{\varphi 3^\varphi - \varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(\tau_{n-1}) + \left[ \frac{\varphi 4^\varphi - \varphi 2^\varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(\tau_{n-2}) + \dots
 \tag{29}$$

The equivalent form for Equation (29) in the  $x$  and  $y$  directions will be as follows

$$\begin{aligned}
 {}^x I_{AB4}^\varphi \mathcal{H}(x, y) &\approx \left[ \frac{2\Gamma(\varphi) - 2\varphi\Gamma(\varphi) + 1}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(x, y) + \left[ \frac{\varphi 3^\varphi - \varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(x-1, y) + \left[ \frac{\varphi 4^\varphi - \varphi 2^\varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(x-2, y) \\
 &+ \left[ \frac{\varphi 5^\varphi - \varphi 3^\varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(x-3, y) + \left[ \frac{\varphi 6^\varphi - \varphi 4^\varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(x-4, y) + \left[ \frac{\varphi 7^\varphi - \varphi 5^\varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(x-5, y) \dots, \\
 {}^y I_{AB4}^\varphi \mathcal{H}(x, y) &\approx \left[ \frac{2\Gamma(\varphi) - 2\varphi\Gamma(\varphi) + 1}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(x, y) + \left[ \frac{\varphi 3^\varphi - \varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(x, y-1) + \left[ \frac{\varphi 4^\varphi - \varphi 2^\varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(x, y-2) \\
 &+ \left[ \frac{\varphi 5^\varphi - \varphi 3^\varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(x, y-3) + \left[ \frac{\varphi 6^\varphi - \varphi 4^\varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(x, y-4) + \left[ \frac{\varphi 7^\varphi - \varphi 5^\varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)} \right] \mathcal{H}(x, y-5) \dots
 \end{aligned}
 \tag{30}$$

Therefore, the coefficients used in the so-called **AB4 masks** of different sizes will be determined as follows

$$\begin{aligned}
 \rho_0 &= \frac{2\Gamma(\varphi) - 2\varphi\Gamma(\varphi) + 1}{2\mathcal{J}(\varphi)\Gamma(\varphi)}, \\
 \rho_1 &= \frac{\varphi 3^\varphi - \varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)}, \\
 \rho_2 &= \frac{\varphi 4^\varphi - \varphi 2^\varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)}, \\
 \rho_3 &= \frac{\varphi 5^\varphi - \varphi 3^\varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)}, \\
 \rho_4 &= \frac{\varphi 6^\varphi - \varphi 4^\varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)}, \\
 \rho_5 &= \frac{\varphi 7^\varphi - \varphi 5^\varphi}{2\mathcal{J}(\varphi)\Gamma(\varphi)}.
 \end{aligned}
 \tag{31}$$

### 5. The Main Algorithm of the Paper

The main algorithm of the article is presented in this section. First, we assume that  $C := [c_{ij}]_{m \times n}$  is a matrix whose values are non-negative integers and are less than or equal to 255. This matrix is called an image matrix.

**Definition 2.** If  $C := [c_{ij}]_{m \times n}$  is an image matrix, we call the entries with values of 0 or 255 as noise pixels and the other entries as regular pixels of the image.

**Definition 3.** If the entries of an image matrix include noise components, then the matrix is called a noise image.

**Definition 4.** If  $C$  is the matrix corresponding to an image, then the binary matrix of  $C$  is defined as  $E := [b_{ij}]_{m \times n}$ , where

$$c_{ij} = \begin{cases} 0, & a_{ij} \in \{0, 255\}, \\ 1, & c_{ij} \notin \{0, 255\}. \end{cases}$$

**Definition 5.** Let  $C := [c_{ij}]_{m \times n}$  and  $1 \leq p \leq \min\{m, n\}$ . Then, the  $p$  symmetric padding matrix of  $C$  is a matrix of  $(m + 2p) \times (n + 2p)$  size that is defined in the following manner

$$\tilde{C}_p = \begin{bmatrix} c_{pp} & \cdots & c_{p1} & c_{p1} & c_{p2} & \cdots & c_{pn} & c_{pn} & \cdots & c_{p(n-p+1)} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{1p} & \cdots & c_{11} & c_{11} & c_{12} & \cdots & c_{1n} & c_{1n} & \cdots & c_{1(n-p+1)} \\ c_{1p} & \cdots & c_{11} & c_{11} & c_{12} & \cdots & c_{1n} & c_{1n} & \cdots & c_{1(n-p+1)} \\ c_{2p} & \cdots & c_{21} & c_{21} & c_{22} & \cdots & c_{2n} & c_{2n} & \cdots & c_{2(n-p+1)} \\ c_{3p} & \cdots & c_{31} & c_{31} & c_{32} & \cdots & c_{3n} & c_{3n} & \cdots & c_{3(n-p+1)} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{mp} & \cdots & c_{m1} & c_{m1} & c_{m2} & \cdots & c_{mn} & c_{mn} & \cdots & c_{m(n-p+1)} \\ c_{mp} & \cdots & c_{m1} & c_{m1} & c_{m2} & \cdots & c_{mn} & c_{mn} & \cdots & c_{m(n-p+1)} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{(m-p+1)p} & \cdots & c_{(m-p+1)1} & c_{(m-p+1)1} & c_{(m-p+1)2} & \cdots & c_{(m-p+1)n} & c_{(m-p+1)n} & \cdots & c_{(m-p+1)(n-p+1)} \end{bmatrix}. \tag{32}$$

**Example 1.** For  $C = \begin{bmatrix} 63 & 5 & 255 \\ 0 & 255 & 173 \\ 84 & 23 & 0 \end{bmatrix}$ , we have  $\tilde{C}_2 = \begin{bmatrix} 255 & 0 & 0 & 255 & 173 & 173 & 255 \\ 5 & 63 & 63 & 5 & 255 & 255 & 5 \\ 5 & 63 & 63 & 5 & 255 & 255 & 5 \\ 255 & 0 & 0 & 255 & 173 & 173 & 255 \\ 23 & 84 & 84 & 23 & 0 & 0 & 23 \\ 23 & 84 & 84 & 23 & 0 & 0 & 23 \\ 255 & 0 & 0 & 255 & 173 & 173 & 255 \end{bmatrix}$ .

**Definition 6.** Let  $C := [c_{ij}]_{m \times n}$  and  $1 \leq r \leq p$ . Then,  $r$ -approximate matrix of  $c_{ij}$  in  $\tilde{C}_p$  is denoted by  $C_{ij}^r$  and is as follows:

$$C_{ij}^r = [c_{ij}^r]_{(2r+1) \times (2r+1)} = \begin{bmatrix} \tilde{c}_{(i+p-r)(j+p-r)} & \cdots & \tilde{c}_{(i+p-r)(j+p+r)} \\ \vdots & \tilde{c}_{(i+p)(j+p)} & \vdots \\ \tilde{c}_{(i+p+r)(j+p-r)} & \cdots & \tilde{c}_{(i+p+r)(j+p+r)} \end{bmatrix}. \tag{33}$$

**Example 2.** Under the assumptions of Example 1, we have

$$C_{13}^1 = \begin{bmatrix} 0 & 255 & 173 \\ 63 & 5 & 255 \\ 63 & 5 & 255 \end{bmatrix}. \tag{34}$$

**Definition 7.** Let us define the matrix  $\tilde{C}_{ij}^r := [\tilde{c}_{ij}^r]_{(2r+1) \times (2r+1)}$  from  $C_{ij}^r$  as

$$\tilde{c}_{ij}^r = \begin{cases} 0, & c_{ij}^r \in \{0, 255\}, \\ c_{ij}^r, & c_{ij}^r \notin \{0, 255\}. \end{cases} \tag{35}$$

In other words, this matrix consists of all regular entries of  $C_{ij}^r$ , and zero elsewhere.

**Example 3.** Under the assumptions of Example 2, we have

$$\bar{C}_{13}^1 = \begin{bmatrix} 0 & 0 & 173 \\ 63 & 5 & 0 \\ 63 & 5 & 0 \end{bmatrix}.$$

**Definition 8.** Let  $C = [c_{ij}]_{(2r+1) \times (2r+1)}$  for  $r = 1, 2, \dots, 5$ . Then, the Atangana–Baleanu mean of  $C$  is defined as follows

$$ABm(C) := \frac{\sum_{(i,j) \in \Lambda} c_{ij} \omega_{ij}^{2r+1}}{\sum_{(i,j) \in \Lambda} \omega_{ij}^{2r+1}}, \tag{36}$$

where  $\Lambda = \{(i, j) | c_{i^*j^*} \notin \{0, 255\}\}$ , and  $\Omega^{2r+1}$ 's for  $r = 1, 2, \dots, 5$  are filters introduced in the Section 3.

**Definition 9.** Let  $C = [c_{ij}]_{m \times n}$  and  $D = [d_{ij}]_{m \times n}$  be two given matrices. The  $l_1$ -distance of them is calculated as

$$\|C - D\|_1 := \sum_{i=1}^m \sum_{j=1}^n |c_{ij} - d_{ij}|.$$

Considering the above symbols and definitions, the main denoising algorithm in this paper (Algorithm 1) is presented as follows

---

**Algorithm 1** The algorithm of the Atangana–Baleanu iterative adaptive mean filter.

---

**Input:** Obtain  $C$  as a noisy image  $C = [c_{ij}]_{m \times n}$   
**Output:** Obtain  $D$  as a denoised image  $D = [d_{ij}]_{m \times n}$   
**Step 1.** Obtain a noisy image matrix  $C := [c_{ij}]_{m \times n}$  where  $\min\{m, n\} \geq 5$ .  
**Step 2.** Change the format of matrix  $C$  from uint8 to double if needed.  
**Repeat**  
**Step 3.** Set  $D := C$ .  
**Step 4.** For  $p$  from 5 to 1  
     Construct the binary matrix  $E := [e_{ij}]_{m \times n}$  of  $C$ .  
     Construct  $\bar{C}_p$  and  $\bar{E}_p$ .  
     For  $i = 1 : m$   
         For  $j = 1 : n$   
             If  $e_{ij} = 0$   
                 For  $r$  from 1 to  $p$   
                     If  $E_{ij}^r \neq [0]$   
                         Construct  $C_{ij}^r$ .  
                         Construct  $\bar{C}_{ij}^r$ .  
                          $c_{ij} \leftarrow ABm(\bar{C}_{ij}^r)$   
                         Break  
                     End If  
                 End For  
             End If  
         End For  
     End For  
**Until**  $\|C - D\|_1 \leq \epsilon$ .  
**Step 5.**  $D$  is the denoised image matrix.  
**Step 6.** Change the format of matrix  $D$  from double to uint8.

---

The flowchart of the algorithm is also presented in Figure 1.

**Remark 1.** The main difference between the proposed algorithm in this paper and the one in [60] is that instead of the Cesàro mean in **STEP 3**, we used the Atangana–Baleanu fractional mean.

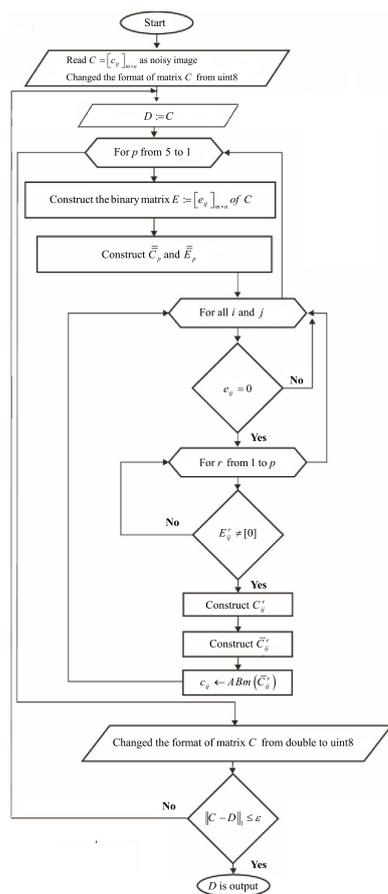


Figure 1. Flowchart of the algorithm.

### 6. Discussion

The quality of images resulting from different algorithms is measured using various criteria. One of these criteria is the calculation of the peak signal-to-noise ratio (PSNR), which can be measured by the following formula

$$PSNR = 10 \log_{10} \frac{255 \times 255}{MSE}, \tag{37}$$

where

$$MSE = \frac{1}{m \times n} \sum_{j=1}^n \sum_{i=1}^m [I^*(i, j) - I(i, j)]^2. \tag{38}$$

PSNR measures the difference between the original image and the denoised image in terms of their peak signal power and noise power. Higher PSNR values indicate better image quality. Moreover, MSE measures the average squared difference between the original and denoised images. Lower MSE values indicate better image quality.

The next known index that can be used to measure the similarity of two images is the structural similarity index measurement, which can be calculated with the following formula

$$SSIM(I_1, I_2) = \frac{(2\mu_1\mu_2 + c_1)(2\varrho_{12} + c_2)}{(\mu_1^2 + \mu_2^2 + c_1)(\varrho_1^2 + \varrho_2^2 + c_2)}. \tag{39}$$

SSIM compares the structural information of the original and denoised images. It measures the similarity in terms of luminance, contrast, and structure. Higher SSIM values indicate better image quality.

We compare the results of our proposed methods **AB1-AB4** in terms of PNSR and SSIM with those of **TSF**, **NAFSM**, **ASWMF**, **ACmF**, **NASNLM**, and **BPDF**. Each of these

algorithms mentioned above has been used in denoising of Elaine, peppers, and goldhill images contaminated with salt-and-pepper noise with intensities of 10, 30, 50, 70, and 90 percent as shown in Figures 2–16. As is evident, salt-and-pepper noise can significantly degrade the visual quality of an image and make it difficult to extract useful information from the image. Further, this kind of noise can occur in any part of an image, but it tends to be more prevalent in areas of low contrast or in regions with sharp edges.



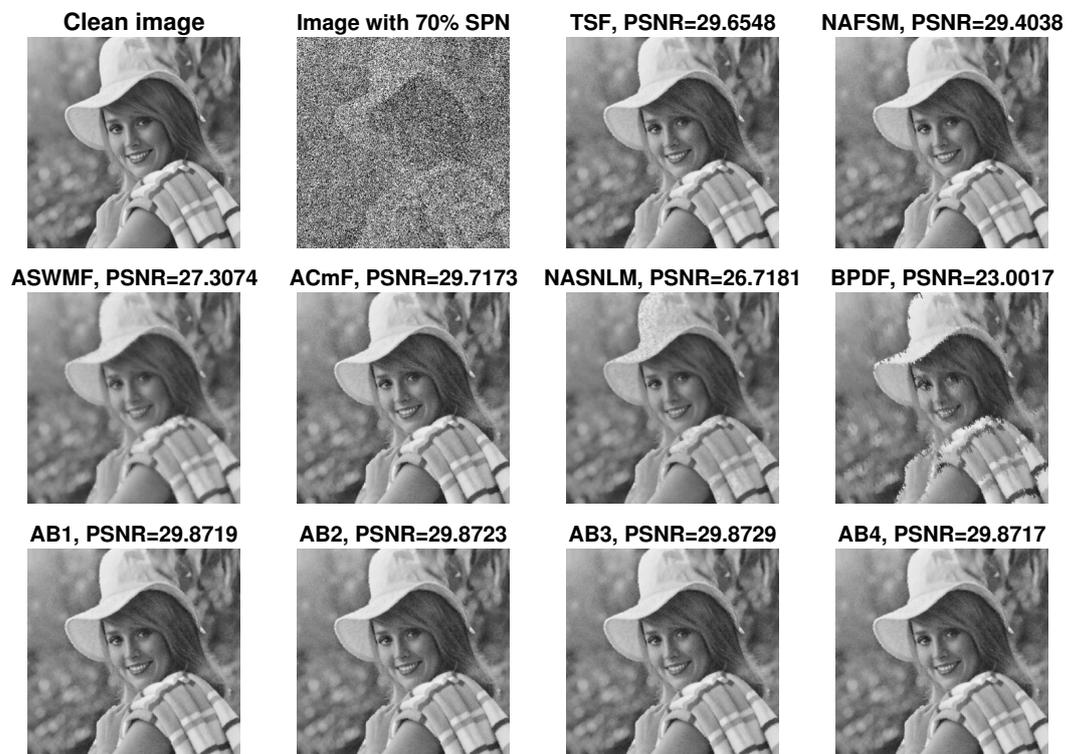
**Figure 2.** Comparison of the performance of different methods for salt-and-pepper noise ratio of 10% for Elaine.



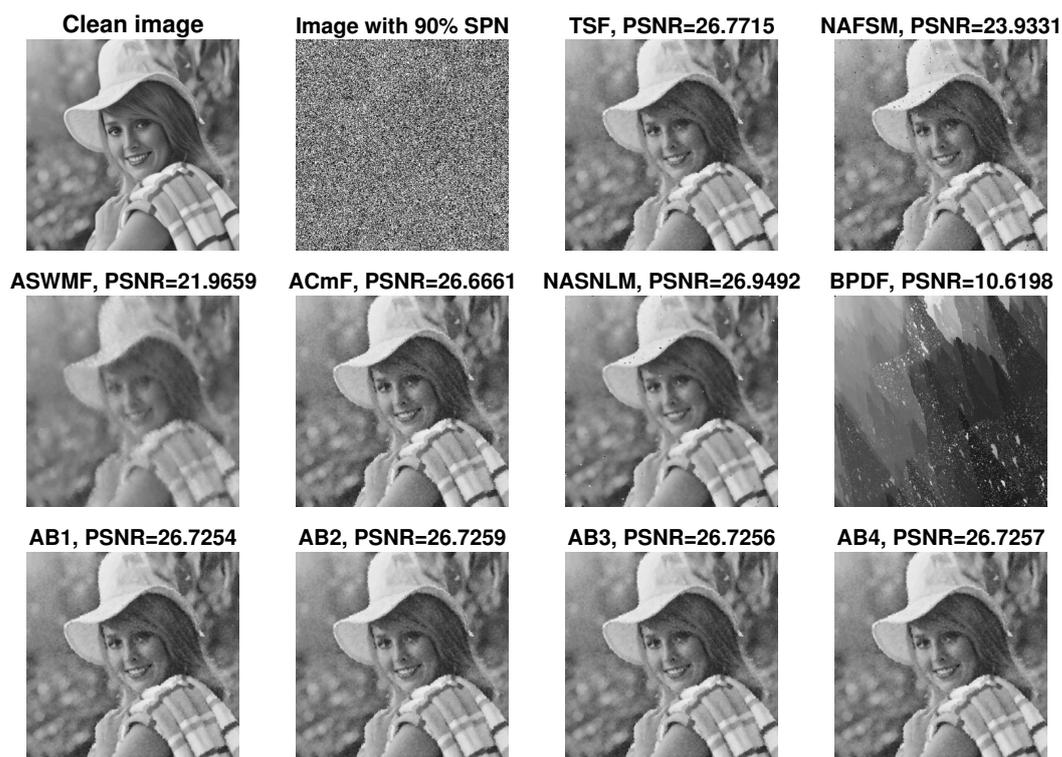
**Figure 3.** Comparison of the performance of different methods for salt-and-pepper noise ratio of 30% for Elaine.



**Figure 4.** Comparison of the performance of different methods for salt-and-pepper noise ratio of 50% for Elaine.



**Figure 5.** Comparison of the performance of different methods for salt-and-pepper noise ratio of 70% for Elaine.



**Figure 6.** Comparison of the performance of different methods for salt-and-pepper noise ratio of 90% for Elaine.



**Figure 7.** Comparison of the performance of different methods for salt-and-pepper noise ratio of 10% for peppers.



**Figure 8.** Comparison of the performance of different methods for salt-and-pepper noise ratio of 30% for peppers.



**Figure 9.** Comparison of the performance of different methods for salt-and-pepper noise ratio of 50% for peppers.

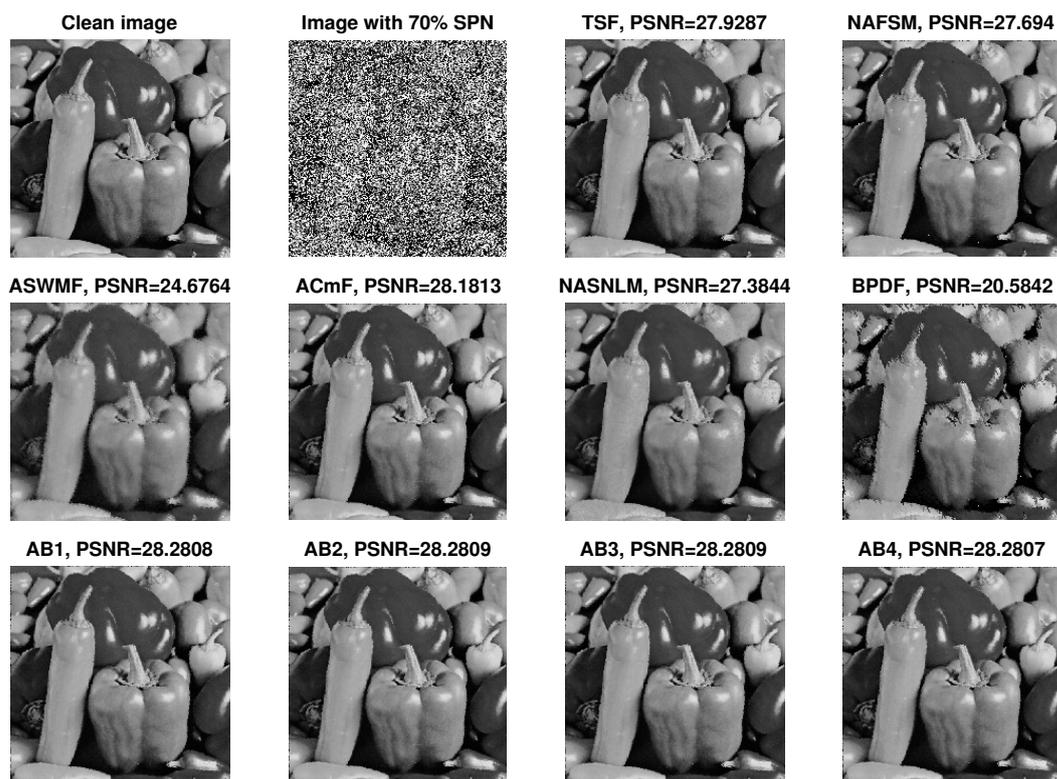


Figure 10. Comparison of the performance of different methods for salt-and-pepper noise ratio of 70% for peppers.

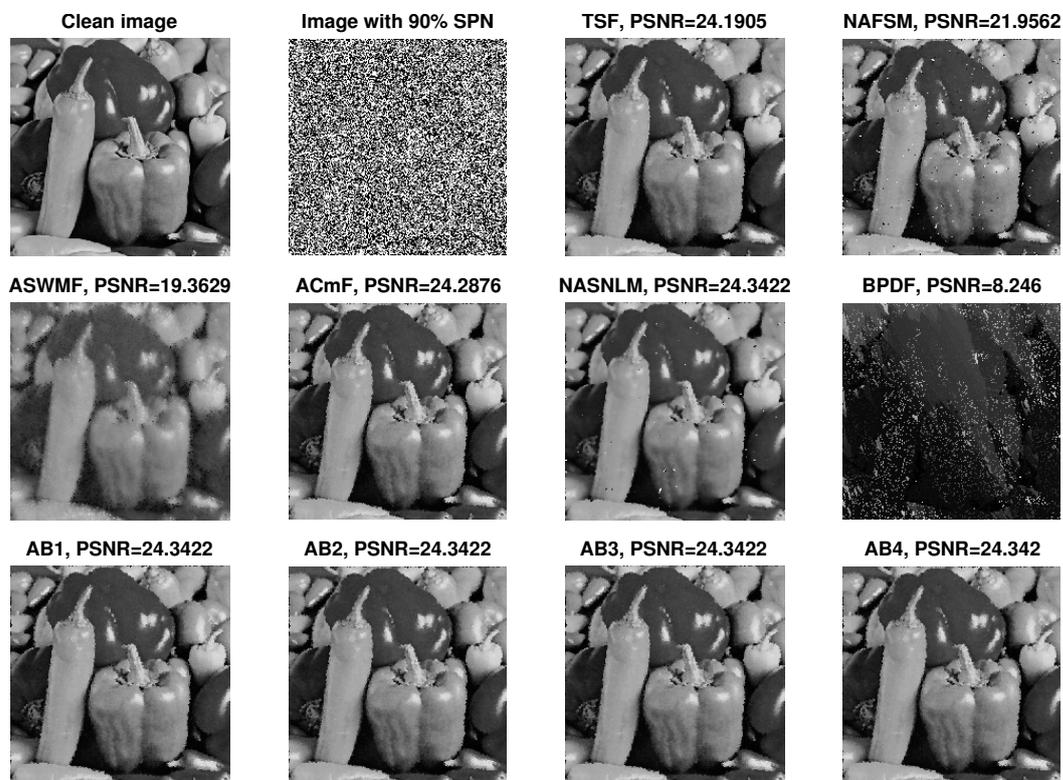


Figure 11. Comparison of the performance of different methods for salt-and-pepper noise ratio of 90% for peppers.



Figure 12. Comparison of the performance of different methods for salt-and-pepper noise ratio of 10%.



Figure 13. Comparison of the performance of different methods for salt-and-pepper noise ratio of 30%.



Figure 14. Comparison of the performance of different methods for salt-and-pepper noise ratio of 50%.

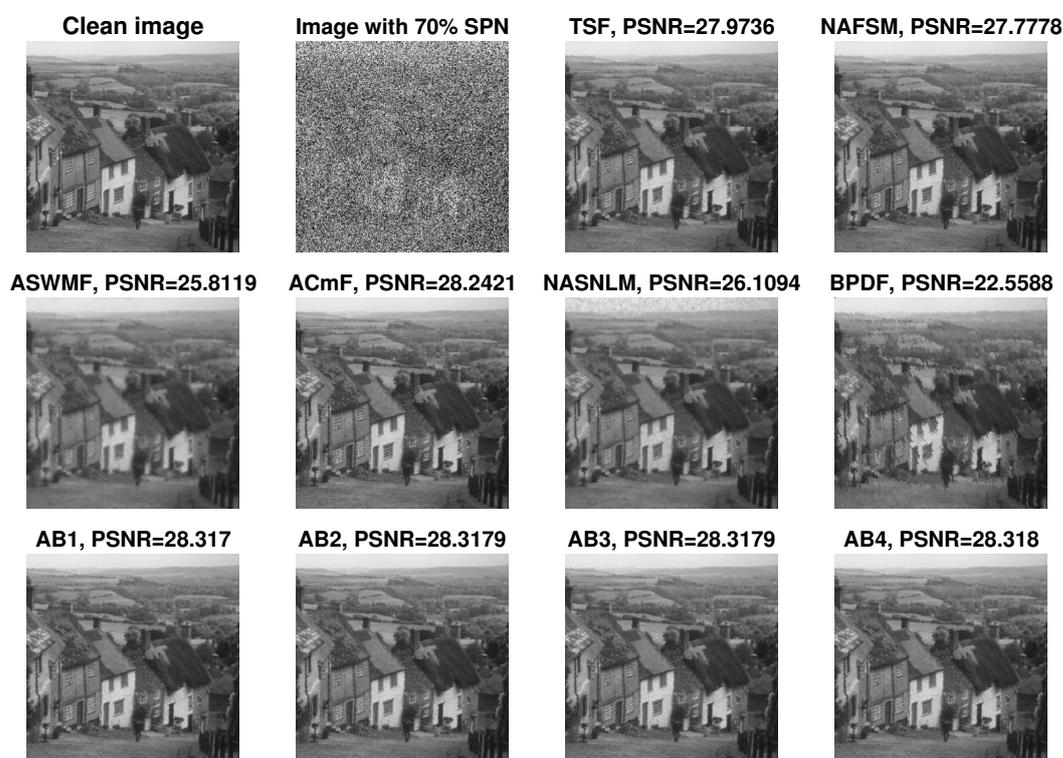
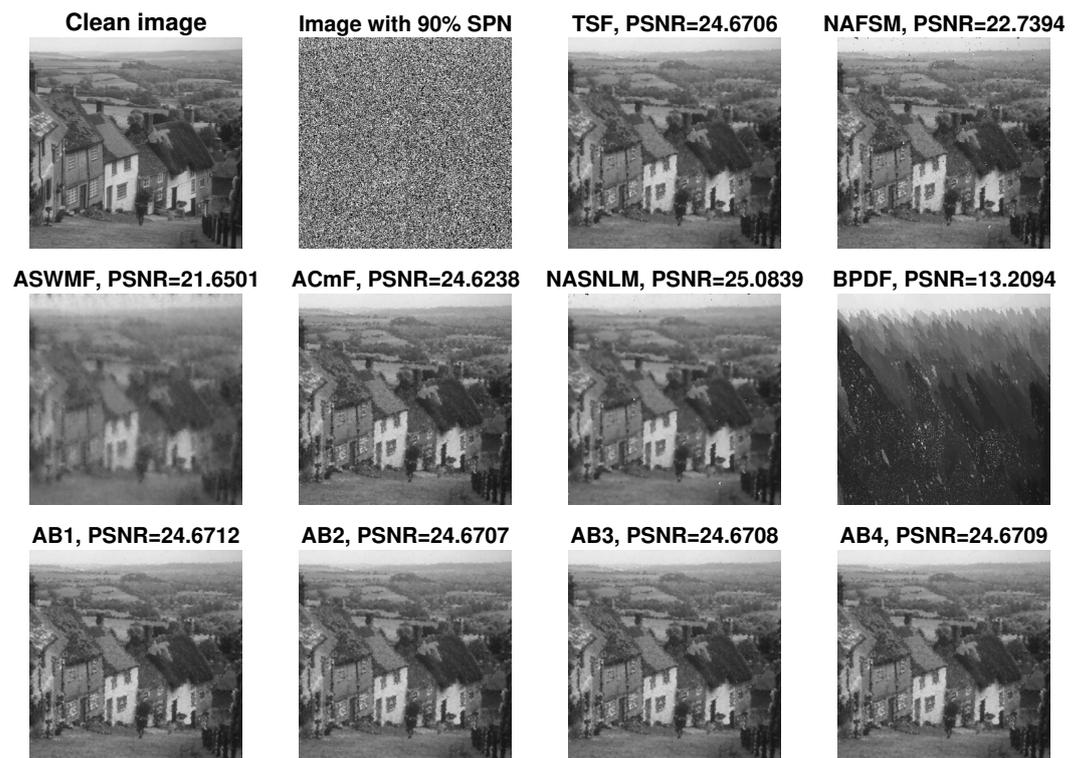


Figure 15. Comparison of the performance of different methods for salt-and-pepper noise ratio of 70%.



**Figure 16.** Comparison of the performance of different methods for salt-and-pepper noise ratio of 90%.

Further, in Tables 1–6, the amount of MSE index obtained from different algorithms in the *Elaine*, *peppers*, and *goldhill* images are reported. The results obtained in these tables confirm that the algorithms proposed in this article have a very impressive performance and have obtained the best results among other methods in most tests. The value considered for the fractional order parameter  $\varphi$  was considered equal to 0.95 in all our proposed methods while performing the experiments.

Our approach seems to be useful in applications where image quality is critical, such as medical imaging or surveillance. In medical imaging, for example, noise reduction in images is crucial for accurate diagnosis and treatment planning. Our method's ability to preserve important image details while removing noise makes it an excellent candidate for these types of applications.

**Table 1.** Comparisons of MSE obtained by different masks for Elaine.

| Noise | Noisy  | TSF    | NAFSM  | ASWMF   | ACmF   | NASNLM  | BPDF    | AB1    | AB2    | AB3    | AB4    |
|-------|--------|--------|--------|---------|--------|---------|---------|--------|--------|--------|--------|
| 10%   | 1847   | 5.499  | 5.520  | 8.874   | 5.760  | 101.134 | 7.715   | 5.302  | 5.302  | 5.301  | 5.303  |
| 30%   | 5521   | 18.480 | 18.506 | 27.718  | 18.409 | 251.650 | 30.209  | 16.946 | 16.946 | 16.948 | 16.948 |
| 50%   | 9235   | 36.470 | 36.897 | 55.834  | 35.772 | 261.184 | 88.470  | 33.260 | 33.260 | 33.260 | 33.256 |
| 70%   | 12,867 | 64.135 | 67.713 | 112.755 | 63.441 | 123.526 | 292.682 | 61.302 | 61.298 | 61.305 | 61.298 |
| 90%   | 16,595 | 126.05 | 231.62 | 380.34  | 129.49 | 109.80  | 5203.18 | 127.87 | 127.87 | 127.87 | 127.88 |

**Table 2.** Comparisons of SSIM obtained by different masks for Elaine.

| Noise | Noisy | TSF   | NAFSM | ASWMF | ACmF  | NASNLM | BPDF  | AB1   | AB2   | AB3   | AB4    |
|-------|-------|-------|-------|-------|-------|--------|-------|-------|-------|-------|--------|
| 10%   | 0.171 | 0.975 | 0.975 | 0.972 | 0.974 | 0.806  | 0.969 | 0.976 | 0.976 | 0.976 | 0.976  |
| 30%   | 0.047 | 0.919 | 0.919 | 0.912 | 0.918 | 0.726  | 0.899 | 0.924 | 0.924 | 0.924 | 0.9248 |
| 50%   | 0.022 | 0.848 | 0.848 | 0.837 | 0.847 | 0.684  | 0.804 | 0.857 | 0.857 | 0.857 | 0.857  |
| 70%   | 0.011 | 0.757 | 0.753 | 0.735 | 0.756 | 0.699  | 0.641 | 0.763 | 0.763 | 0.763 | 0.763  |
| 90%   | 0.005 | 0.640 | 0.580 | 0.558 | 0.630 | 0.692  | 0.243 | 0.632 | 0.632 | 0.632 | 0.632  |

**Table 3.** Comparisons of MSE obtained by different masks for peppers.

| Noise | Noisy  | TSF    | NAFSM  | ASWMF   | ACmF   | NASNLM  | BPDF    | AB1    | AB2    | AB3    | AB4    |
|-------|--------|--------|--------|---------|--------|---------|---------|--------|--------|--------|--------|
| 10%   | 1999   | 4.522  | 4.610  | 14.335  | 5.518  | 47.783  | 9.694   | 5.518  | 5.516  | 5.519  | 5.519  |
| 30%   | 6011   | 17.312 | 17.442 | 45.082  | 17.942 | 106.736 | 42.410  | 17.328 | 17.329 | 17.325 | 17.325 |
| 50%   | 10,102 | 43.700 | 43.642 | 89.646  | 40.793 | 121.435 | 124.066 | 39.049 | 39.052 | 39.046 | 39.053 |
| 70%   | 14,039 | 87.155 | 90.318 | 179.179 | 82.002 | 95.531  | 433.330 | 80.039 | 80.039 | 80.038 | 80.036 |
| 90%   | 18,066 | 201.78 | 327.27 | 605.09  | 200.25 | 190.14  | 8084.16 | 197.85 | 197.85 | 197.84 | 197.84 |

**Table 4.** Comparisons of SSIM obtained by different masks for peppers.

| Noise | Noisy | TSF   | NAFSM | ASWMF | ACmF  | NASNLM | BPDF  | AB1   | AB2   | AB3   | AB4   |
|-------|-------|-------|-------|-------|-------|--------|-------|-------|-------|-------|-------|
| 10%   | 0.173 | 0.987 | 0.986 | 0.977 | 0.987 | 0.882  | 0.981 | 0.987 | 0.987 | 0.987 | 0.987 |
| 30%   | 0.058 | 0.942 | 0.941 | 0.899 | 0.945 | 0.831  | 0.909 | 0.945 | 0.945 | 0.945 | 0.945 |
| 50%   | 0.028 | 0.886 | 0.886 | 0.816 | 0.893 | 0.772  | 0.806 | 0.895 | 0.895 | 0.895 | 0.895 |
| 70%   | 0.012 | 0.856 | 0.852 | 0.764 | 0.858 | 0.827  | 0.687 | 0.861 | 0.861 | 0.861 | 0.861 |
| 90%   | 0.005 | 0.751 | 0.682 | 0.572 | 0.746 | 0.784  | 0.190 | 0.748 | 0.748 | 0.748 | 0.748 |

**Table 5.** Comparisons of MSE obtained by different masks for goldhill.

| Noise | Noisy  | TSF    | NAFSM  | ASWMF   | ACmF   | NASNLM  | BPDF    | AB1    | AB2    | AB3    | AB4    |
|-------|--------|--------|--------|---------|--------|---------|---------|--------|--------|--------|--------|
| 10%   | 1865   | 6.176  | 6.206  | 14.002  | 5.953  | 68.955  | 10.217  | 6.391  | 6.389  | 6.391  | 6.390  |
| 30%   | 5711   | 24.211 | 24.248 | 45.786  | 21.818 | 190.735 | 42.992  | 22.209 | 22.213 | 22.209 | 22.211 |
| 50%   | 9421   | 48.769 | 48.918 | 83.942  | 43.474 | 221.964 | 107.896 | 42.887 | 42.887 | 42.890 | 42.895 |
| 70%   | 13,222 | 88.111 | 91.346 | 145.301 | 82.937 | 135.603 | 301.688 | 81.513 | 81.507 | 81.515 | 81.507 |
| 90%   | 17,026 | 187.60 | 310.07 | 377.12  | 190.02 | 182.89  | 2695.97 | 188.13 | 188.13 | 188.14 | 188.12 |

**Table 6.** Comparisons of SSIM obtained by different masks for goldhill.

| Noise | Noisy | TSF   | NAFSM | ASWMF | ACmF  | NASNLM | BPDF  | AB1   | AB2   | AB3   | AB4   |
|-------|-------|-------|-------|-------|-------|--------|-------|-------|-------|-------|-------|
| 10%   | 0.207 | 0.984 | 0.983 | 0.969 | 0.984 | 0.895  | 0.976 | 0.984 | 0.984 | 0.984 | 0.984 |
| 30%   | 0.028 | 0.886 | 0.886 | 0.816 | 0.893 | 0.772  | 0.806 | 0.895 | 0.895 | 0.895 | 0.895 |
| 50%   | 0.014 | 0.806 | 0.802 | 0.702 | 0.812 | 0.727  | 0.619 | 0.814 | 0.814 | 0.814 | 0.814 |
| 70%   | 0.014 | 0.806 | 0.802 | 0.702 | 0.812 | 0.727  | 0.619 | 0.814 | 0.814 | 0.814 | 0.814 |
| 90%   | 0.006 | 0.652 | 0.599 | 0.508 | 0.648 | 0.658  | 0.313 | 0.651 | 0.651 | 0.651 | 0.651 |

## 7. Conclusions

Removing noise from digital images while preserving important details is a difficult task, as the noise can be complex, and its effect on individual pixels can be unpredictable. In this paper, we propose a novel method for denoising digital images that contain salt-and-pepper noise. This type of noise appears as randomly distributed white and black pixels in an image. This paper presents a set of algorithms that are effective in removing salt-and-pepper noise from images with high efficiency. This type of noise is a type of image noise that appears as randomly distributed white and black pixels in an image, and it affects individual pixels in an image, causing them to have either the highest intensity value (white pixel) or the lowest intensity value (black pixel). The basic idea of these methods is to introduce a new weighted average based on a well-known idea in fractional calculus, called the Atangana–Baleanu fractional integral operator. Moreover, the concept of symmetry is clearly used in the proposed window mask structures in this paper. Furthermore, our proposed method has been extensively tested on various datasets to assess its effectiveness in denoising images. We compared our method with other state-of-the-art denoising techniques and found that it outperformed them in terms of the peak signal-to-noise ratio (PSNR) metric and visually as well. Our proposed methods are advantageous over other methods in image denoising because they are computationally efficient and can be easily implemented. Moreover, they remove noise from an image while preserving its important

features. Another significant advantage of our proposed method is that it can be easily implemented, which is essential for real-time applications. This feature makes it practical for use in video processing, where images are captured in rapid succession, and quick processing times are necessary to avoid delays. In conclusion, our proposed approach to image denoising is a significant step forward in the field of digital image processing. It offers several advantages over existing techniques, including its ability to remove salt-and-pepper noise while preserving important image features, computational efficiency, and ease of implementation. Overall, we believe that our method will have a significant impact on various applications that require high-quality image processing.

**Author Contributions:** M.W., S.W., X.J. and Y.W. contributed equally and significantly to writing this article. All authors have read and approved the final manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Conflicts of Interest:** The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

1. Song, F.; Liu, Y.; Shen, D.; Li, L.; Tan, J. Learning Control for Motion Coordination in Wafer Scanners: Toward Gain Adaptation. *IEEE Trans. Ind. Electron.* **2022**, *69*, 13428–13438. [\[CrossRef\]](#)
2. Meng, Q.; Lai, X.; Yan, Z.; Su, C.Y.; Wu, M. Motion planning and adaptive neural tracking control of an uncertain two-link rigid-flexible manipulator with vibration amplitude constraint. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 3814–3828. [\[CrossRef\]](#)
3. Wang, H.; Zhang, X.; Jiang, S. A Laboratory and Field Universal Estimation Method for Tire-Pavement Interaction Noise (TPIN) Based on 3D Image Technology. *Sustainability* **2022**, *14*, 12066. [\[CrossRef\]](#)
4. Zhao, C.; Cheung, C.F.; Xu, P. High-efficiency sub-microscale uncertainty measurement method using pattern recognition. *ISA Trans.* **2020**, *101*, 503–514. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Ghanbari, B.; Baleanu, D. Applications of two novel techniques in finding optical soliton solutions of modified nonlinear Schrödinger equations. *Results Phys.* **2022**, *44*, 106171. [\[CrossRef\]](#)
6. Zeng, Q.; Bie, B.; Guo, Q.; Yuan, Y.; Han, Q.; Han, X.; Chen, M.; Zhang, X.; Yang, Y.; Liu, M.; et al. Hyperpolarized Xe NMR signal advancement by metal-organic framework entrapment in aqueous solution. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 17558–17563. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Lin, X.; Wen, Y.; Yu, R.; Yu, J.; Wen, H. Improved Weak Grids Synchronization Unit for Passivity Enhancement of Grid-Connected Inverter. *IEEE J. Emerg. Sel. Top. Power Electron.* **2022**, *10*, 7084–7097. [\[CrossRef\]](#)
8. Wang, F.; Wang, H.; Zhou, X.; Fu, R. A Driving Fatigue Feature Detection Method Based on Multifractal Theory. *IEEE Sens. J.* **2022**, *22*, 19046–19059. [\[CrossRef\]](#)
9. Cao, K.; Wang, B.; Ding, H.; Lv, L.; Dong, R.; Cheng, T.; Gong, F. Improving physical layer security of uplink NOMA via energy harvesting jammers. *IEEE Trans. Inf. Forensics Secur.* **2020**, *16*, 786–799. [\[CrossRef\]](#)
10. Zhuo, Z.; Du, L.; Lu, X.; Chen, J.; Cao, Z. Smoothed Lv distribution based three-dimensional imaging for spinning space debris. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–3. [\[CrossRef\]](#)
11. Sun, L.; Hou, J.; Xing, C.; Fang, Z. A robust hammerstein-wiener model identification method for highly nonlinear systems. *Processes* **2022**, *10*, 2664. [\[CrossRef\]](#)
12. Hu, J.; Wu, Y.; Li, T.; Ghosh, B.K. Consensus control of general linear multiagent systems with antagonistic interactions and communication noises. *IEEE Trans. Autom. Control* **2018**, *64*, 2122–2127. [\[CrossRef\]](#)
13. Zhong, T.; Wang, W.; Lu, S.; Dong, X.; Yang, B. RMCHN: A Residual Modular Cascaded Heterogeneous Network for Noise Suppression in DAS-VSP Records. *IEEE D* **2023**, *20*, 7500205. [\[CrossRef\]](#)
14. Zhou, G.; Yang, F.; Xiao, J. Study on pixel entanglement theory for imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5409518. [\[CrossRef\]](#)
15. Huppert, A.; Katriel, G. Mathematical modelling and prediction in infectious disease epidemiology. *Clin. Microbiol. Infect.* **2022**, *19*, 999–1005. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Djilali, S.; Ghanbari, B. Dynamical behavior of two predators–one prey model with generalized functional response and time-fractional derivative. *Adv. Differ. Equ.* **2021**, *2021*, 235. [\[CrossRef\]](#)
17. Fan, X.; Wei, G.; Lin, X.; Wang, X.; Si, Z.; Zhang, X.; Shao, Q.; Mangin, S.; Fullerton, E.; Jiang, L.; et al. Reversible switching of interlayer exchange coupling through atomically thin VO<sub>2</sub> via electronic state modulation. *Matter* **2020**, *2*, 1582–1593. [\[CrossRef\]](#)
18. Baleanu, D.; Jajarmi, A.; Mohammadi, H.; Rezapour, S. A new study on the mathematical modelling of human liver with Caputo–Fabrizio fractional derivative. *Chaos Solitons Fractals* **2020**, *134*, 109705. [\[CrossRef\]](#)

19. Wang, G.; Zhao, B.; Wu, B.; Wang, M.; Liu, W.; Zhou, H.; Zhang, C.; Wang, Y.; Han, Y.; Xu, X. Research on the macro-mesoscopic response mechanism of multisphere approximated heteromorphic tailing particles. *Lithosphere* **2022**, *2022*, 1977890. [[CrossRef](#)]
20. Defterli, O.; Baleanu, D.; Jajarmi, A.; Sajjadi, S.S.; Alshaikh, N.; Asad, J.H. Fractional treatment: An accelerated mass-spring system. *Rom. Rep. Phys.* **2022**, *74*, 122.
21. Xu, S.; Dai, H.; Feng, L.; Chen, H.; Chai, Y.; Zheng, W.X. Fault Estimation for Switched Interconnected Nonlinear Systems with External Disturbances via Variable Weighted Iterative Learning. *IEEE Trans. Circuits Syst. II Express Briefs* **2023**, *70*, 2011–2015. [[CrossRef](#)]
22. Bai, X.; Shi, H.; Zhang, K.; Zhang, X.; Wu, Y. Effect of the fit clearance between ceramic outer ring and steel pedestal on the sound radiation of full ceramic ball bearing system. *J. Sound Vib.* **2022**, *529*, 116967. [[CrossRef](#)]
23. Huang, N.; Chen, Q.; Cai, G.; Xu, D.; Zhang, L.; Zhao, W. Fault diagnosis of bearing in wind turbine gearbox under actual operating conditions driven by limited data with noise labels. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 3502510. [[CrossRef](#)]
24. Ghanbari, B.; Atangana, A. A new application of fractional Atangana–Baleanu derivatives: Designing ABC-fractional masks in image processing. *Phys. A Stat. Mech. Appl.* **2020**, *542*, 123516. [[CrossRef](#)]
25. Zhang, Z.; Wang, L.; Zheng, W.; Yin, L.; Hu, R.; Yang, B. Endoscope image mosaic based on pyramid ORB. *Biomed. Signal Process. Control* **2022**, *71*, 103261. [[CrossRef](#)]
26. Liu, Y.; Tian, J.; Hu, R.; Yang, B.; Liu, S.; Yin, L.; Zheng, W. Improved feature point pair purification algorithm based on SIFT during endoscope image stitching. *Front. Neurobotics* **2022**, *16*, 840594. [[CrossRef](#)] [[PubMed](#)]
27. Ghanbari, B.; Atangana, A. Some new edge detecting techniques based on fractional derivatives with non-local and non-singular kernels. *Adv. Differ. Equ.* **2020**, *2020*, 435. [[CrossRef](#)]
28. Liu, S.; Yang, B.; Wang, Y.; Tian, J.; Yin, L.; Zheng, W. 2D/3D multimode medical image registration based on normalized cross-correlation. *Appl. Sci.* **2022**, *12*, 2828. [[CrossRef](#)]
29. Cao, Z.; Wang, Y.; Zheng, W.; Yin, L.; Tang, Y.; Miao, W.; Liu, S.; Yang, B. The algorithm of stereo vision and shape from shading based on endoscope imaging. *Biomed. Signal Process. Control* **2022**, *76*, 103658. [[CrossRef](#)]
30. Ban, Y.; Liu, M.; Wu, P.; Yang, B.; Liu, S.; Yin, L.; Zheng, W. Depth estimation method for monocular camera defocus images in microscopic scenes. *Electronics* **2022**, *11*, 2012. [[CrossRef](#)]
31. Wang, S.; Sheng, H.; Zhang, Y.; Yang, D.; Shen, J.; Chen, R. Blockchain-Empowered Distributed Multi-Camera Multi-Target Tracking in Edge Computing. *IEEE Trans. Ind. Inform.* **2023**. [[CrossRef](#)]
32. Zhou, W.; Lv, Y.; Lei, J.; Yu, L. Global and local-contrast guides content-aware fusion for RGB-D saliency prediction. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 3641–3649. [[CrossRef](#)]
33. Zhou, W.; Yu, L.; Zhou, Y.; Qiu, W.; Wu, M.W.; Luo, T. Local and global feature learning for blind quality evaluation of screen content and natural scene images. *IEEE Trans. Image Process.* **2018**, *27*, 2086–2095. [[CrossRef](#)]
34. Yang, M.; Wang, H.; Hu, K.; Yin, G.; Wei, Z. IA-Net: An Inception–Attention-Module-Based Network for Classifying Underwater Images from Others. *IEEE J. Ocean. Eng.* **2022**, *47*, 704–717. [[CrossRef](#)]
35. Zhou, G.; Bao, X.; Ye, S.; Wang, H.; Yan, H. Selection of optimal building facade texture images from UAV-based multiple oblique image flows. *IEEE Trans. Geosci. Remote. Sens.* **2020**, *59*, 1534–1552. [[CrossRef](#)]
36. Liu, R.; Wang, X.; Lu, H.; Wu, Z.; Fan, Q.; Li, S.; Jin, X. SCCGAN: Style and characters inpainting based on CGAN. *Mob. Netw. Appl.* **2021**, *26*, 3–12. [[CrossRef](#)]
37. Buades, A.; Coll, B.; Morel, J.M. A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.* **2005**, *4*, 490–530. [[CrossRef](#)]
38. Huang, J.; Zhao, Z.; Ren, C.; Teng, Q.; He, X. A prior-guided deep network for real image denoising and its applications. *Knowl. Based Syst.* **2022**, *255*, 109776. [[CrossRef](#)]
39. Liang, H.; Li, N.; Zhao, S. Salt and Pepper Noise Removal Method Based on a Detail-Aware Filter. *Symmetry* **2021**, *13*, 515. [[CrossRef](#)]
40. Li, M.; Cai, G.; Bi, S.; Zhang, X. Improved TV Image Denoising over Inverse Gradient. *Symmetry* **2023**, *15*, 678. [[CrossRef](#)]
41. Al-Shamasneh, A.R.; Ibrahim, R.W. Image Denoising Based on Quantum Calculus of Local Fractional Entropy. *Symmetry* **2023**, *15*, 396. [[CrossRef](#)]
42. Zhou, G.; Song, B.; Liang, P.; Xu, J.; Yue, T. Voids filling of DEM with multiattention generative adversarial network model. *Remote Sens.* **2022**, *14*, 1206. [[CrossRef](#)]
43. Tian, C.; Fei, L.; Zheng, W.; Xu, Y.; Zuo, W.; Lin, C.W. Deep learning on image denoising: An overview. *Neural Netw.* **2020**, *131*, 251–275. [[CrossRef](#)]
44. Ilesanmi, A.E.; Ilesanmi, T.O. Methods for image denoising using convolutional neural network: A review. *Complex Intell. Syst.* **2021**, *7*, 2179–2198. [[CrossRef](#)] [[PubMed](#)]
45. Zhong, Y.; Liu, L.; Zhao, D.; Li, H. A generative adversarial network for image denoising. *Multimedia Tools and Applications. Multimed. Tools Appl.* **2020**, *79*, 16517–16529. [[CrossRef](#)]
46. Rahman, S.M.; Hasan, M.K. Wavelet-domain iterative center weighted median filter for image denoising. *Signal Process.* **2003**, *83*, 1001–1012. [[CrossRef](#)]
47. Thanh, D.N.; Enginoğlu, S. An iterative mean filter for image denoising. *IEEE Access* **2019**, *7*, 167847–167859. [[CrossRef](#)]
48. Feng, Y.; Zhang, B.; Liu, Y.; Niu, Z.; Fan, Y.; Chen, X. A D-band manifold triplexer with high isolation utilizing novel waveguide dual-mode filters. *IEEE Trans. Terahertz Sci. Technol.* **2022**, *12*, 678–688.

49. Xu, K.D.; Guo, Y.J.; Liu, Y.; Deng, X.; Chen, Q.; Ma, Z. 60-GHz compact dual-mode on-chip bandpass filter using GaAs technology. *IEEE Electron Device Lett.* **2021**, *42*, 1120–1123. [[CrossRef](#)]
50. Xu, B.; Guo, Y. A novel DVL calibration method based on robust invariant extended Kalman filter. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9422–9434. [[CrossRef](#)]
51. Xu, B.; Wang, X.; Zhang, J.; Guo, Y.; Razzaqi, A.A. A novel adaptive filtering for cooperative localization under compass failure and non-gaussian noise. *IEEE Trans. Veh. Technol.* **2022**, *71*, 3737–3749. [[CrossRef](#)]
52. Chen, Z.; Zhou, Z.; Adnan, S. Joint low-rank prior and difference of Gaussian filter for magnetic resonance image denoising. *Med Biol. Eng. Comput.* **2021**, *59*, 607–620. [[CrossRef](#)]
53. Samko, G.; Kilbas, A.A.; Marichev, O.I. *Fractional Integrals and Derivatives: Theory and Applications*; Gordon and Breach: Yverdon, Switzerland, 1993. [[CrossRef](#)]
54. Caputo, M.; Fabrizio, M. A new definition of fractional derivative without singular kernel. *Prog. Fract. Differ. Appl.* **2015**, *1*, 73–85.
55. Atangana, A.; Baleanu, D. New fractional derivatives with non-local and non-singular kernel: Theory and application to heat transfer model. *Therm. Sci.* **2016**, *20*, 763–769.
56. Huading, J.; Pu, Y. Fractional calculus method for enhancing digital image of bank slip. *Proc. Congr. Image Signal Process.* **2008**, *3*, 326–330. [[CrossRef](#)]
57. Toufik, M.; Atangana, A. New numerical approximation of fractional derivative with non-local and non-singular kernel: Application to chaotic models. *Eur. Phys. J. Plus* **2017**, *132*, 444.
58. Li, C.; Zeng, F. The finite difference methods for fractional ordinary differential equations. *Num. Funct. Anal. Opt.* **2013**, *34*, 149–179. [[CrossRef](#)]
59. Pu, Y.F.; Zhou, J.L.; Yuan, X. Fractional differential mask: A fractional differential-based approach for multiscale texture enhancement. *IEEE Trans. Image Process* **2010**, *19*, 491–511. [[CrossRef](#)]
60. Pu, Y.F.; Zhou, J.L. Adaptive cesáro mean filter for salt-and-pepper noise removal. *El-Cezeri* **2020**, *7*, 304–314.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.