Modified Wiener Filter using Scaled Median Absolute Deviation Normalized for Gaussian Noise Reduction

G. Karyono [1]
[1] Faculty of Computer Science, Universitas Amikom Purwokerto, Central Java, 53127, Indonesia

Abstract

Modified versions of the Wiener filter that replace the mean with the median (MMWF) can better reduce Gaussian noise in images than the classical Wiener filter (WF). However, performance gradually decreases as the noise variance increases. To overcome these limitations, we propose a modified Wiener filter (MADNWF) that replaces the mean with the scaled median absolute deviation (MADN). Similar to MMWF, our modification to the Wiener filter uses a local kernel to average over each pixel. Instead of using the median alone, we replace it with MADN. We used four public datasets for the experiment: Set12, the Tampere17 noise-free dataset, TID2008, and the BSD68 dataset. The first step of this study was to generate 'degraded' images. To achieve this, we added a random amount of zero-mean Gaussian noise with noise variances ranging from 10 to 90, in increments of 10, to every image in the dataset. For comparison with the WF and MMWF, we then applied the proposed MADNWF to reduce noise in degraded images. We evaluated the resulting performance by comparing the reduced noise images with the original images using the Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) metrics. Experimental results demonstrate that MADNWF consistently outperforms the other methods. At a low noise level (variance = 10), MMWF with a 3x3 filter slightly outperforms in fine-structural preservation, achieving a peak SSIM of 0.6676. However, as the noise intensity increases (variances from 20 to 90), MADNWF achieves complete dominance across all datasets. MADNWF achieves the highest PSNR, up to 33.27 dB at low noise levels, representing an improvement of up to 0.28 dB over WF. Under extreme noise conditions (variance = 90), the MADNWF 7x7 configuration achieves superior image cleanliness (up to 28.91 dB), whereas the MADNWF 5x5 setting serves as the optimal trade-off for structure preservation, outperforming MMWF with an SSIM margin increase of up to 0.0276. In conclusion, our results demonstrate that MADNWF helps reduce the noise distribution in natural images.

Keywords: Gaussian noise reduction, Wiener filter (WF), the scaled median absolute deviation normalized (MADN), median modified Wiener filter (MMWF).

Citation: Karyono G. Modified Wiener Filter using Scaled Median Absolute Deviation Normalized for Gaussian Noise Reduction. Computer Optics 2026; 50(3): 1724. doi: 10.18287/COJ1724.

Introduction

Noise is quite often encountered in digital images [1]. It manifests as artefacts that distort image pixels [2], making images unclear [3], [4]. When noise variances in images are significant, subsequent digital image processing tasks, such as image classification [5], image recognition [6], image compression [7], edge detection [8], and image segmentation [9], may fail to produce accurate results [5], [6]. Hence, for the further processing of digital images and interpreting results, it is essential to have images with noise as low as possible [7]. Images with noise can be represented mathematically as the sum of the free-noise image and the noise:

$$I_d = I_0 + n, \tag{1}$$

where I_d is the degraded noisy image, I_0 is the unknown free-noise image, and n represents noise (with Gaussian noise as a general example).

The primary purpose of noise reduction is to decrease noise in the unknown free-noise image while maintaining the essential specific structures of the image and improving the signal-to-noise Ratio, thereby enhancing clarity and visual attractiveness [8]. This challenging topic has been the subject of active research over the last few decades, where the main challenges are as follows [9]:

- still maintain edge without blurring,
- still maintain textures,
- smooth flat areas, and
- not generate new artefacts.

Numerous noise reduction methods explicitly addressing Gaussian noise were proposed during that period and have achieved remarkable success [10]. These methods can be roughly classified into spatial domain, transform domain, and CNN-based methods [11].

With the recent rise of deep learning, CNN-based methods have emerged as powerful tools for noise reduction in images. The success of these methods for noise reduction is inseparable from leveraging Knowledge learned from pre-training data on a large-scale dataset [12]. Although CNN-based methods are developing rapidly, most are ineffective because real-world noise reduction processes often lack image pairs for training [9]. Collecting many "free-noise" training data pairs in the real world is extremely difficult and expensive. Additionally, since there is a gap between actual and synthetic noise, using synthetic noise images to train a model can significantly compromise the model's performance. Another non-negligible drawback is that CNN-based methods can only train specific models for specific types and levels of noise [13], which is not a 'real' noise [14]. Thus, they cannot be applied to blind image noise reduction [15].

Meanwhile, transform domain methods work in a transformed space (e.g., wavelet or Fourier transforms) [16]. These methods analyze frequency components to separate noise from the underlying image [17]. These separations are excellent at isolating and reducing noise, particularly high-frequency noise. However, the success of transform domain methods largely depends on several key factors, such as choosing a proper transformation base and threshold [18]. Inappropriate transformation base and threshold selection often result in poor noise reduction [9].

Spatial domain methods directly manipulate an image's neighbourhood pixels to modify its characteristics [19]. Different filter types are employed in this domain, including the Gaussian filter (GF), the median filter (MF), and the Wiener filter (WF). Each filter operates distinctly, and combining one filter with another can sometimes yield improved performance. A modified version of the Wiener filter (MMWF), proposed by Cannistraci et al. [21], is an example of such a successful combination. This filter modified the Wiener filter by replacing its mean value with median values. The success of this modified version in reducing noise is demonstrated in natural and medical images [20]–[22]. Although quite promising, the performance decreases gradually as the noise variance increases.

To address this, we present the scaled median absolute deviation normalized (MADN), an alternative and most robust measure of dispersion/scale that is easy to apply and interpret, to replace using the local kernel average around each pixel with the local kernel median and divides the results using the 0.75th quartile of the standard normal distribution for the Gaussian distribution. We select MADN from the median because it is more robust to outliers or anomalous changes in a dataset than the mean. Additionally, MADN is well-suited for zero-mean Gaussian white noise in the one-dimensional noise reduction model.

The remaining main structure of this article is as follows: the second section introduces the methods. The third section presents the experiments. The results and discussions follow in the next section. The last section is the conclusion.

1. Methods

The study proposes the MADNWF, a modified version of the Wiener filter, by replacing the local kernel mean (average) around each pixel with the local kernel scaled median absolute deviation. The method combines the complementary abilities of WF and MF to lessen noise distribution. Like MMWF, the MADNWF is a noise reduction method based on the Wiener filter, which replaces the pixel values of the mask matrix with the median values of the noise-free image. The fundamental difference is that we use the constant 0.6745, corresponding to the 0.75th percentile of the standard normal distribution, as the divisor. This MADN value is advantageous because it can remove outliers without decreasing the sharpness of the image. Before we explain MADNWF in more detail, we will first describe WF and MF in the following section, followed by MAD and MADN in Section 1.2.

1.1. Wiener and Median Filters

Wiener and Median filters are noise reduction methods that operate in nonlinear spatial domains, often used in image processing. The strategy of MF is quite simple; the operations can be divided into three steps. First, it defines a 'neighbourhood' around each pixel. Second, it arranges the pixel intensity values within that neighbourhood in numerical sequence. Finally, it replaces the original pixel with the median value from the sorted list. By consistently choosing the middle value, MF effectively smooths out erratic and extreme values typically associated with noise.

WF minimizes the MSE (mean square error) between the desired and estimated image [23], [24]. It utilizes the image and noise correlation matrix as the optimal criterion and adjusts the filtering output according to the local variance of the image [25]. WF estimates the local mean and variance around each pixel.

$$\mu = \frac{1}{NM}\sum_{n_1,n_2\epsilon\eta} a(n_1,n_2), \tag{2}$$

$$\sigma^2 = \frac{1}{NM}\sum_{n_1,n_2\epsilon\eta} \sigma^2(n_1,n_2) - \mu^2, \tag{3}$$

where μ is the average, σ^2 is the Gaussian noise variances in the image, $n_1 x n_2$ is the size of the neighborhood area η in the mask, and $a(n_1,n_2)$ represents each pixel in the area η. The WF then creates a pixel-wise Wiener using these estimates

$$b_\omega(n_1,n_2) = \mu + \frac{\sigma^2-v^2}{\sigma^2}.(\alpha(n_1,n_2)-\mu), \tag{4}$$

where v^2 is the noise variance; if the noise variance is unknown, WF uses the mean of all the local estimated variances.

1.2. Median Absolute Deviation and Scaled Median Absolute Deviation Normalized

The median absolute deviation (MAD) is a robust measure of dispersion or variability because it relies on the median to estimate the distribution's centre and the absolute difference, rather than the squared difference. The complete formula for the MAD is

$$MAD = median\left|Y_i - \tilde{Y}\right|, \quad (5)$$

where $median$ denotes the median value and $\tilde{Y} = median(Y_i) = F^{-1}(0.5)$ is the population median. For normally distributed data, the MAD equals 0.6745 times the standard deviation (SD) and 0.8453 times the average deviation [26].

Scaled median absolute deviation normalized (MADN) is an algorithm that considers a relative scale from data divergence and the median. It commonly behaves like MAD. The complete formula for the MADN is

$$MADN = MAD/0.6745, \quad (6)$$

MADN is approximately the same as the standard deviation for normally distributed data.

1.3. Modified Versions of Wiener Filter Using Scaled Median Absolute Deviation Normalized

MADNWF is a modified version of the Wiener filter that substitutes the scaled median absolute deviation normalized for the mean. As we know, WF uses the mean of all the local estimated variances, and using this mean value results in the loss of high-frequency signals, such as those of the edge region. On the other hand, MADN is a more robust scale measure, as 'the best of an inferior lot,' so it more efficiently preserves high-frequency signals while effectively reducing noise. The modification we introduced in the WF concerns the local kernel mean around each pixelμ. We replace this value with the estimate of the local kernel MADN around each pixel $\tilde{\mu}$, resulting in the following formula:

$$b_{madnwf}(n_1, n_2) = \tilde{\mu} + \frac{\sigma^2 - v^2}{\sigma^2}.(\alpha(n_1, n_2) - \tilde{\mu}). \quad (7)$$

The advantage of using MADNWF is that it can enhance degraded image quality. Due to the drop-off effect, the critical feature in the image is better preserved than in MF, WF, and MMWF. In summary, MADNWF can perform substantially better than conventional filters in denoising effect while maintaining the image's critical features.

2. Experiments

In the experiment, we implement MADNWF to reduce noise in images degraded by zero-mean Gaussian noise with variances ranging from 10 to 90, in increments of 10. In its implementation, we used the kernel sizes of 3 × 3, 5 × 5, and 7 × 7. We modified the original Wiener filter from the SciPy Python library, replacing the mean with the MADN. The SciPy Python library can be accessed at https://github.com/scipy/scipy/blob/v1.15.2/scipy/signal/_signaltools.py#L0-L1.

Four public datasets, namely Set12, the Tampere17 noise-free dataset, TID2008, and BSD68, were used in the experiment. Finally, we evaluate the resulting performance of MADNWF using the evaluation metrics and compare it with WF and MMWF. We implement the Wiener filter obtained from the original Wiener filter in the SciPy library. Likewise, we use the same library to generate MMWF by replacing the mean with the median. The following section describes datasets, the Gaussian noise model, and evaluation metrics.

2.1. Datasets

For the experiment, we used four publicly available datasets: Set12[13], Tampere17 noise-free [32], BSD68 [33], and TID2008 [34][35]. To the best of our Knowledge, those four datasets have been frequently used in the literature to benchmark state-of-the-art noise reduction methods. The following is a short description of each dataset.

The Set12 is a collection of 12 grayscale images from different scenes. This dataset has been used many times to evaluate noise reduction in images. Each image has 256 × 256 pixels in size. The set12 dataset is publicly available and can be accessed online at https://www.kaggle.com/datasets/leweihua/set12-231008. Sample images from this dataset are shown in fig. 1.

The TAMPERE17 Noise-free has 300 test images, each with a resolution of 512 × 512 pixels. The images are obtained by monitoring the noise variance without interpolation, lossy compression, or other spatial processing methods, such as denoising or sharpening. The TAMPERE17 Noise-free dataset is publicly available and can be accessed online at https://webpages.tuni.fi/imaging/tampere17/. Sample images from this dataset are shown in fig. 2.

The BSD68 is part of the Berkeley Segmentation Dataset, containing 68 high-quality images with clear resolutions of 481 × 321 pixels. This dataset is also frequently used to measure noise reduction performance. It is publicly available and can be accessed online at https://github.com/clausmichele/CBSD68-dataset. Sample images from this dataset are shown in fig. 3.

The TID2008 (Tampere Image Database) contains 25 reference distorted images, utilizing various methods, including different types of blur, noise, JPEG2000 and JPEG compression, transmission with errors, local distortions, and changes in contrast and luminance. The TID2008 dataset is publicly available and can be accessed online at https://www.ponomarenko.info/tid2008.htm. Sample images from this dataset are shown in Fig. 4.

2.2. Gaussian Noise Models

Since the experiment needs pairs of free-noisy and noisy images, while the used datasets only contain free-noisy photos, we generate the noisy image by adding Gaussian noise with a specific noise variance. In theory, Gaussian noise, often called a normal distribution, follows a normal distribution with a mean (μ) of zero and a standard deviation (σ) of one. It can be generated by generating random numbers between 0 and 1. Then, the image function value is added to the existing noise at the points affected by noise. The effect of Gaussian Noise on an image is the appearance of colored dots, whose number is the same as the percentage of noise. Mathematically, the easiest way to add Gaussian noise is by adding random values from a normal distribution with a mean of zero and a standard deviation (σ) to the input image. The probability density function (PDF) for a Normal distribution has the following formula:

$$pdf(x) = \left(1/\left(\sigma * sqrt(2 * \pi)\right)\right) * e^{\wedge}\left(-(x-\mu)^2/(2 * \sigma^2)\right), \tag{8}$$

where x is the random variable, μ is the mean, and σ is the standard deviation.

We use Python coding, as presented in https://github.com/delmarrerikaine/LPG-PCA, to generate the degraded image by adding zero-mean Gaussian noise with variances ranging from 10 to 90, in increments of 10.

2.3. Evaluation Metrics

The peak-signal-to-noise ratio (PSNR) and the structural similarity index measure (SSIM) are essential and popular quality assessment metrics for measuring the similarity between two images and identifying regions of difference between the two input images. In practice, both are often used together to evaluate image quality.

In technical terms, PSNR is a numerical representation of the ratio between the signal's maximum possible power and the noise's peak signal. Mathematically, the definition formula for PSNR is

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{MSE}\right), \tag{9}$$

where MAX is the maximum pixel intensity value, MSE is the mean squared error, namely the average of the squared differences between the original and processed images.

SSIM compares the structural similarity between two images, not only the similarity of pixel intensities. The key idea is that pixels are strongly correlated, remarkably so, when they are spatially close. SSIM works in three aspects: luminance, contrast, and structure. The formula for each of these three aspects in sequence is as follows.

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}, \tag{10}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}, \tag{11}$$

$$s(x,y) = \frac{\sigma_{\mu_x\mu_y} + c_3}{\sigma_x\sigma_y + c_1}. \tag{12}$$

The SSIM measure between x and y is:

$$SSIM(x,y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{(\mu_x^2 + \mu_y^2 + c_1)(\mu_x^2 + \mu_y^2 + c_2)}, \tag{13}$$

where x and y are two images to compare, μ_x and μ_y are the luminance of two images, σ_x and σ_y are the contrast of two images, σ_{xy} is the covariance between two images, and c_1 and c_2 are constants used to avoid division by zero.

We use the scikit-image Python library to implement the PSNR and SSIM metrics.

3. Results and Discussions

This section presents the quantitative and qualitative results of the proposed MADNWF and its comparison with WF and MMWF on four publicly available datasets: Set12, Tampere17, BSD68, and TID2008, focusing on reducing noise in degraded images by zero-mean Gaussian noise with noise variances ranging from 10 to 90 in increments of 10.

3.1. Qualitative Evaluation

The noise reduction results on the images are highly focused on visual appearance, making visual assessment a critical aspect. We present the qualitative evaluation results of the proposed MADNWF and comparison references with WF and MMWF, obtained from image samples on the Set12, Tampere17, BSD68, and TID2008 datasets, in Figs. 1, 2, 3, and 4, respectively. One can see that the proposed MADNWF is comparable to WF and MMWF under almost all conditions. Although the difference is not yet very significant, the proposed MADNWF reduces noise more effectively while maintaining sharper images without introducing over-smoothing.

3.2. Quantitative Evaluation

In addition to considering the image quality of the noise reduction results based on their visual appearance, it is also necessary to perform a quality assessment benchmark of the noise reduction results using quantitative evaluation. As mentioned earlier, the measurement metrics used here are the PSNR and SSIM. The following section presents the measurement results of both metrics for each dataset.

3.2.1. Noise Reduction Results on the Set12 Dataset

The characteristics of the images from the set12 dataset are mostly homogeneous regions. We first experimented with this dataset. The goal is to evaluate the performance of the proposed MADNWF in reducing Gaussian noise in degraded images with homogeneous region characteristics similar to those found in the Set12 dataset. Then we compare it with the MMWF and WF to determine the most efficient and effective solution for that task. The PSNR and SSIM measurement results for these three methods are presented in Tabls. 1 and 2, respectively. The PSNR and SSIM measurement results for these three methods are presented in Tabls. 1 and 2, respectively.

As shown in Tab. 1, the MADNWF achieves higher average PSNR values than WF and MMWF when applied at almost all noise variances, except for noise variances of 30 and 40. These results are affected by the kernel size, and choosing the proper size has always been a challenging decision. Related to this, kernel sizes that can be recommended based on PSNR results presented in Tab. 1 are 3 × 3 for a noise variance of 10, 5 × 5 for a noise variance of 20, and 7 × 7 for a noise variance of 30 and above. These kernel sizes align with the existing fact that higher noise variances require a larger kernel size. In contrast, a smaller kernel size is more suitable for lower noise variances, as it helps prevent the loss of fine image details.

Tab. 1. The PSNR results of WF, MMWF, and MADNWF on the Set12 dataset

Noise variances	The WF			The MMWF			The MADNWF		
	3 × 3	**5 × 5**	**7 × 7**	**3 × 3**	**5 × 5**	**7 × 7**	**3 × 3**	**5 × 5**	**7 × 7**
10	32.99	32.26	31.54	33.19	33.12	32.59	**33.27**	32.91	32.28
20	30.79	31.34	31.10	30.58	31.36	31.33	30.81	**31.40**	31.22
30	29.64	30.48	30.61	29.51	30.36	**30.63**	29.70	30.48	30.61
40	29.05	29.82	**30.15**	28.97	29.72	30.10	29.13	29.87	30.13
50	28.70	29.37	**29.76**	28.65	29.29	29.69	28.78	29.44	**29.76**
60	28.47	29.05	29.43	28.43	28.99	29.38	28.54	29.14	**29.46**
70	28.32	28.82	29.18	28.29	28.78	29.13	28.38	28.92	**29.24**
80	28.20	28.64	28.97	28.18	28.61	28.95	28.26	28.75	**29.06**
90	28.13	28.51	28.82	28.11	28.48	28.79	28.17	28.61	**28.91**

Tab. 2. The SSIM results of WF, MMWF, and MADNWF on the Set12 dataset

Noise variances	The WF			The MMWF			The MADNWF		
	3 × 3	**5 × 5**	**7 × 7**	**3 × 3**	**5 × 5**	**7 × 7**	**3 × 3**	**5 × 5**	**7 × 7**
10	0.6649	0.6017	0.5292	**0.6844**	0.6640	0.6158	0.6799	0.6401	0.5845
20	0.5448	0.5306	0.4817	0.5292	**0.5488**	0.5221	0.5447	0.5349	0.4929
30	0.4422	0.4677	0.4362	0.4222	**0.4638**	0.4513	0.4502	0.4626	0.4285
40	0.3609	0.4112	0.3968	0.3425	0.3998	0.3984	0.3788	**0.4093**	0.3826
50	0.2971	0.3613	0.3611	0.2818	0.3494	0.3565	0.3218	**0.3679**	0.3483
60	0.2467	0.3185	0.3283	0.233	0.3082	0.3223	0.275	**0.3341**	0.3212
70	0.2066	0.2824	0.2992	0.1947	0.2735	0.2933	0.2357	**0.3053**	0.2992
80	0.1741	0.2515	0.2736	0.1635	0.2437	0.2683	0.2023	0.2801	**0.2804**
90	0.1475	0.2251	0.2510	0.1382	0.2180	0.2461	0.1743	0.2580	**0.2641**

The SSIM measurement results, as presented in Tab. 2, differ from the PSNR measurement results. The tab. indicates that MADNWF outperforms MMWF and WF when applied at noise variances of 40 and up. It appears that the recommended kernel sizes also differ. We found that the required kernel size is smaller than the PSNR measurement results. A kernel size of 3 × 3 seems suitable for noise variances of 20 and down. Meanwhile, a kernel size of 5 × 5 is used for noise variances between 30 and 70, and a kernel size of 7 × 7 is used for noise variances of 80 and above.

The proposed MADNWF is more suitable for applying to image cases, such as those presented in Fig. 1(a). In this case, according to the PSNR measurement results, the proposed MADNWF outperforms WF and MMWF in all noise variance ranges. When the performance of MMWF is lower than that of WF, the proposed MADWF makes the difference. Although it is not very significant, the performance is superior to MMWF and even WF. Meanwhile, for the image case, which features larger homogeneous textures and many sharp image details, as shown in Fig. 1(e), the superiority of the proposed MADNWF is not observed in all noise variances.

Meanwhile, Figs. 1 (d) and (h) show the SSIM measurement results. These results indicate that MADNWF remains superior to WF and MMWF, although the superiority is found not to be in all noise variances. The cameraman image yields one of the worst SSIM results, where we discovered that the SSIM results are lower than those of WF in all noise variances. However, the performance is still better than MMWF.

3.2.2. Noise Reduction Results on the Tampere17 Noise-free Dataset

Images from the Tampere17 noise-free dataset comprise many highly textured images, images with only part of the image in focus, images featuring self-similar textures, and images with large homogeneous regions, among others.

Tables 3 and 4 present the measurement results of PSNR and SSIM when MADNWF, MMWF, and WF are applied to reduce Gaussian noise in images with those characteristics.

As shown in Tab. 3, the average PSNR values of the proposed MADNWF are higher than those of MMWF and WF across all noise variances. The recommended kernel sizes are similar to the results on Set12, where a 3x3 kernel size is suitable for noise variances of 10 and below, a 5 × 5 kernel size for noise variances of 20, and a 7 × 7 kernel size for noise variances of 30 and above. The best performs on images containing fine details, such as those presented in Fig. 2 (a). In contrast, the proposed MADNWF does not work well when applied to images containing fine details and homogeneous texture regions, such as those presented in Figs. 2 (e). In this image case, the proposed MADNWF yields lower PSNR values than MMWF but still higher than WF in all noise variances. Other PSNR results show MADNWF are superior only for a specific noise variance.

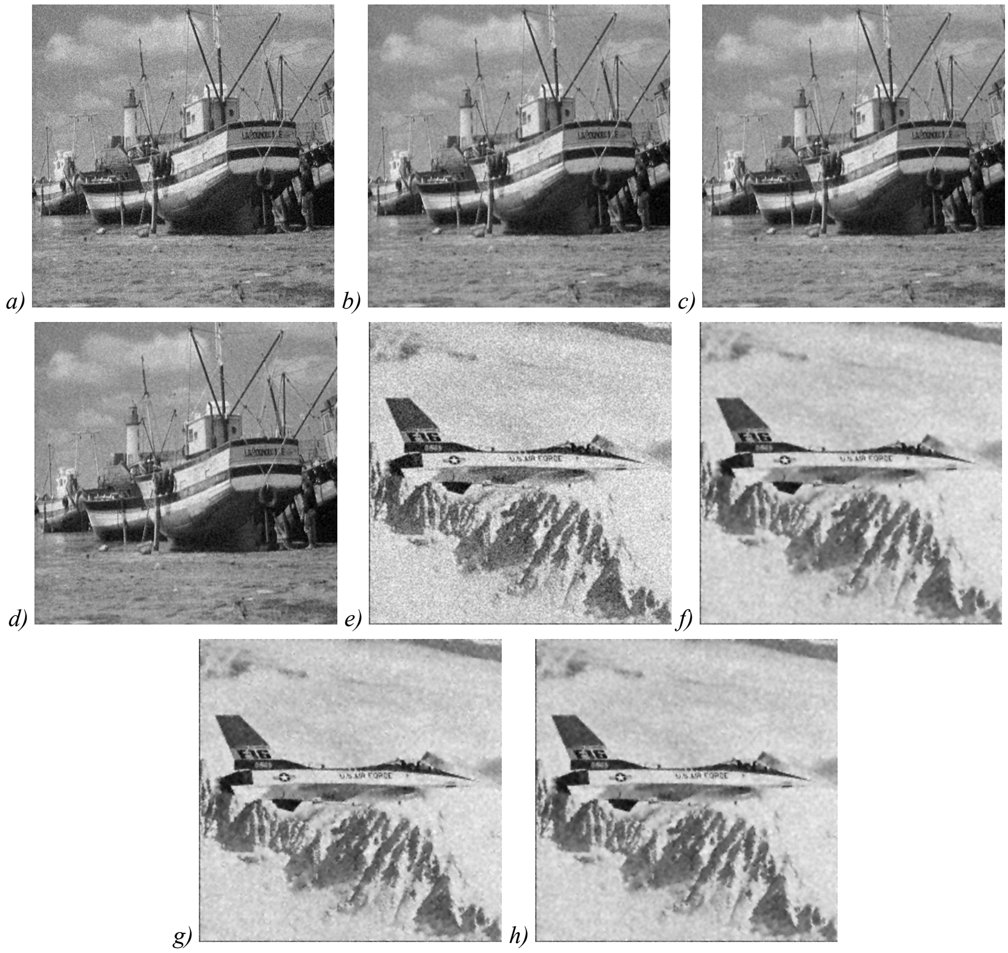

Fig. 1. Visual comparison results of WF, MMWF, and MADNWF on image samples from the Set12 dataset. a) A boat image with a Gaussian noise [σ = 20, μ = 0], (b)WF [PSNR = 30.6706, SSIM = 0.7479], (c)MMWF [PSNR = 30.4883, SSIM = 0.7291], (d)MADNWF [PSNR = 30.6989, SSIM = 0.7493], (e) A jet image with a Gaussian noise [σ = 20, μ = 0], (f)WF [PSNR = 30.9067, SSIM = 0.8242], (g)MMWF [PSNR = 30.6446, SSIM = 0.8027], (h)MMWF [PSNR = 30.8753, SSIM = 0.8243]

Meanwhile, as shown in Tab. 4, MADNWF also has average SSIM values higher than those of MMWF and WF in all noise variances. Based on the SSIM results, using smaller kernel sizes than those in the PSNR results is recommended. Those kernel sizes are 3 × 3 for noise variances of 10 and below, 5 × 5 for noise variances between 20 and 40, and 7 × 7 for noise variances of 50 and above. For images with characteristics such as those presented in Figs. 2. (a) and (e), MADNWF obtained the SSIM results similar to the PSNR results.

3.2.3. Noise Reduction Results on the BSD68 Dataset

Tables 5 and 6 present the PSNR and SSIM average values obtained when MADNWF, MMWF, and WF are applied to reduce Gaussian noise on the BSD68 dataset. The dataset comprises diverse patterned images from natural and object-specific categories, including people, food, and plants. Through an experiment on the BSD68 dataset, we test the resilience of MADNWF in handling Gaussian noise with image characteristics similar to those in the dataset.

As shown in Tab. 5, the PSNR average values of MADNWF are higher than the PSNR average values of MMWF and WF when handling noise variances above 40. In this, the recommended kernel sizes are 7 × 7. Meanwhile, a 5 × 5 kernel size is suitable for a noise variance of 20, while a 3 × 3 kernel size is ideal for noise variances of 10 and below. We found the best performance of MADNW on the BSD68 dataset when applied to the image sample shown in Fig. 3(a). In this image case, the PSNR average values of the MADNWF are higher than the PSNR average values of MMWF and WF in all noise variances. In contrast, we observe the lowest PSNR average values of the MADNWF when the MADNWF is applied to images containing complex fine details, as shown in Fig. 3(e). In this image case, the MADNWF is not superior to MMWF in all noise variances.

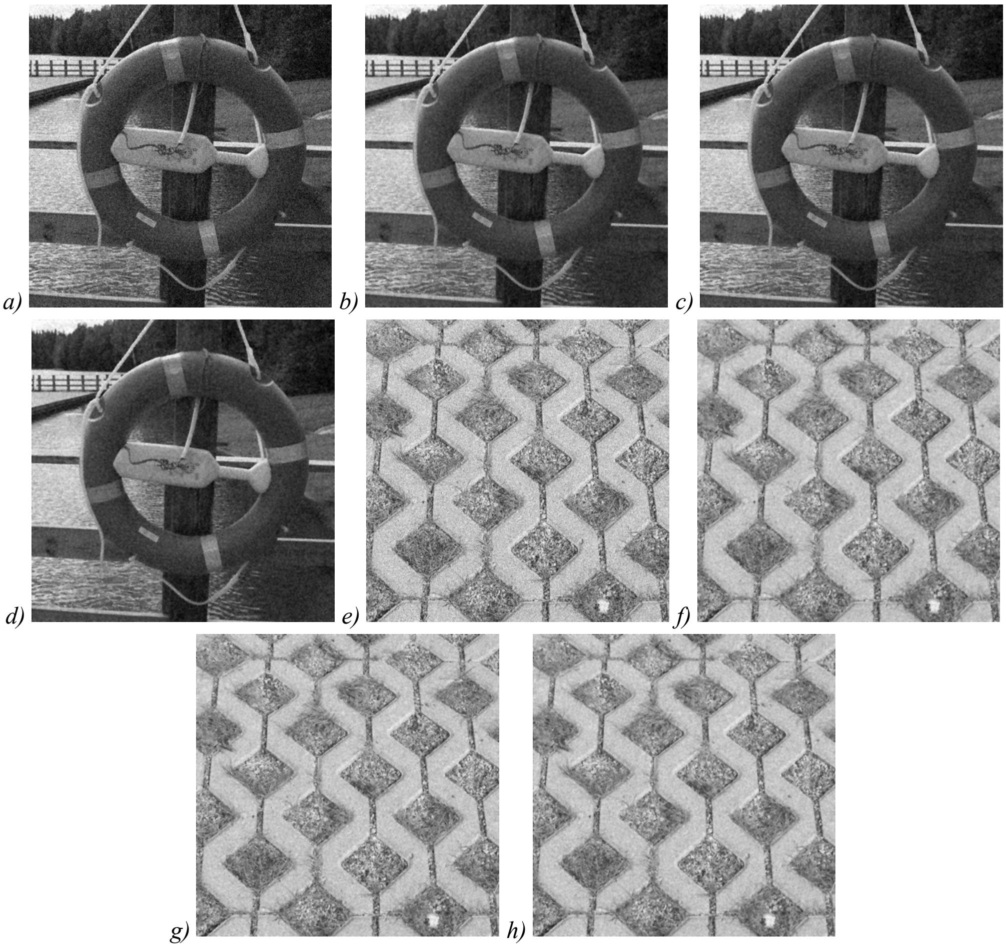

Fig 2. Visual comparison results of WF, MMWF, and MADNWF on image samples from the Tampere17 noise-free dataset. (a) The t023 image with a Gaussian noise [σ = 20, μ = 0], (b)WF [PSNR = 30.8539, SSIM = 0.7399], (c) MMWF [PSNR = 30.6279, SSIM = 0.7091], (d) MADNWF [PSNR = 30.872, SSIM = 0.7426], (e) The t007 image with a Gaussian noise [σ = 20, μ = 0](f)WF [PSNR = 29.613, SSIM = 0.7046], (g)MMWF [PSNR = 29.6737, SSIM = 0.7335], (h)MADNWF [PSNR = 29.5898, SSIM = 0.6961]

Tab. 3. The PSNR results of WF, MMWF, and MADNWF on the Tampere17 noise-free dataset

Noise variances	**The WF**			**The MMWF**			**The MADNWF**		
	3 × 3	**5 × 5**	**7 × 7**	**3 × 3**	**5 × 5**	**7 × 7**	**3 × 3**	**5 × 5**	**7 × 7**
10	32.45	31.94	31.43	32.58	32.47	32.07	**32.59**	32.27	31.82
20	30.48	31.02	30.94	30.34	30.99	31.03	30.51	**31.03**	30.96
30	29.48	30.20	30.44	29.40	30.10	30.39	29.55	30.22	**30.42**
40	28.97	29.61	29.97	28.91	29.54	29.90	29.05	29.68	**29.98**
50	28.65	29.22	29.59	28.61	29.17	29.53	28.73	29.32	**29.63**
60	28.44	28.95	29.29	28.41	28.91	29.25	28.52	29.05	**29.37**
70	28.30	28.75	29.07	28.27	28.72	29.03	28.36	28.86	**29.16**
80	28.19	28.59	28.89	28.17	28.57	28.87	28.24	28.70	**29.00**
90	28.12	28.47	28.75	28.10	28.45	28.73	28.16	28.57	**28.86**

Meanwhile, the SSIM average results of the MADNWF, as shown in Tab. 6, indicate that the MADNWF outperforms MMWF and WF in all noise variances. These results are similar to the SSIM measurement results when applying the MADNWF to the Tampere17 noise-free dataset. The similarity is not only reflected in the SSIM measurement results, but also in the recommended kernel size. For noise variances of below 40, a 3 × 3 kernel size is more recommended. Meanwhile, for noise variances above 50, the SSIM results recommended a 5 × 5 kernel size. For images with characteristics similar to those presented in Figs. 3 (a) and (e), the SSIM average results of the MADNWF are comparable to its PSNR results.

Tab. 4. The SSIM results of WF, MMWF, and MADNWF on the Tampere17 noise-free dataset

Noise variances	The WF			The MMWF			The MADNWF		
	3 × 3	5 × 5	7 × 7	3 × 3	5 × 5	7 × 7	3 × 3	5 × 5	7 × 7
10	0.6274	0.5192	0.4283	**0.6577**	0.5923	0.5219	0.6410	0.5460	0.4612
20	0.5201	0.4599	0.3847	0.5170	0.4908	0.4359	**0.5188**	0.4530	0.3787
30	0.4234	0.4076	0.3480	0.4114	0.4163	0.3740	**0.4299**	0.3911	0.3240
40	0.3434	0.3594	0.3166	0.3301	0.3587	0.3281	**0.3598**	0.3468	0.2877
50	0.2798	0.3162	0.2883	0.2671	0.3122	0.2924	0.3026	**0.3122**	0.2619
60	0.2295	0.2785	0.2626	0.2178	0.2735	0.2631	0.2553	**0.2834**	0.2422
70	0.1895	0.2459	0.2394	0.1789	0.2408	0.2384	0.2159	**0.2585**	0.2261
80	0.1575	0.2179	0.2187	0.1480	0.2128	0.2170	0.1830	**0.2363**	0.2122
90	0.1316	0.1937	0.2002	0.1232	0.1887	0.1981	0.1553	**0.2163**	0.1998

Tab. 5. The PSNR results of WF, MMWF, and MADNWF on the BSD68 dataset

Noise variances	The WF			The MMWF			The MADNWF		
	3 × 3	5 × 5	7 × 7	3 × 3	5 × 5	7 × 7	3 × 3	5 × 5	7 × 7
10	32,48	31,94	31,48	**32,69**	32,54	32,13	32,68	32,28	31,83
20	30,51	31,05	30,99	30,38	31,04	31,09	30,55	**31,07**	31,01
30	29,50	30,24	**30,49**	29,41	30,15	30,45	29,57	30,27	30,48
40	28,98	29,64	30,02	28,92	29,57	29,95	29,06	29,71	**30,03**
50	28,66	29,24	29,63	28,62	29,19	29,56	28,74	29,34	**29,68**
60	28,45	28,96	29,32	28,42	28,92	29,28	28,52	29,07	**29,41**
70	28,31	28,76	29,09	28,28	28,73	29,06	28,36	28,87	**29,19**
80	28,20	28,60	28,91	28,18	28,58	28,88	28,25	28,71	**29,02**
90	28,12	28,48	28,77	28,11	28,46	28,75	28,16	28,58	**28,88**

Tab. 6. The SSIM results of WF, MMWF, and MADNWF on the BSD68 dataset

Noise variances	The WF			The MMWF			The MADNWF		
	3 × 3	5 × 5	7 × 7	3 × 3	5 × 5	7 × 7	3 × 3	5 × 5	7 × 7
10	0,6446	0,5463	0,4719	0,6676	0,6208	0,5661	**0,6588**	0,5794	0,5096
20	0,5228	0,4745	0,4177	0,5152	0,5039	0,4672	**0,5237**	0,4719	0,4154
30	0,4214	0,4137	0,3729	0,4075	0,4213	0,3965	**0,4293**	0,4010	0,3531
40	0,3406	0,3604	0,3347	0,3261	0,3592	0,3444	**0,3573**	0,3503	0,3104
50	0,2769	0,3144	0,3013	0,2635	0,3101	0,3041	0,3000	**0,3113**	0,2792
60	0,2267	0,2751	0,2719	0,2144	0,2699	0,2715	0,2528	**0,2797**	0,2550
70	0,1870	0,2416	0,2460	0,1760	0,2362	0,2442	0,2139	**0,2530**	0,2354
80	0,1552	0,2131	0,2231	0,1455	0,2079	0,2209	0,1810	**0,2300**	0,2188
90	0,1295	0,1887	0,2032	0,1210	0,1837	0,2007	0,1537	**0,2097**	0,2044

3.2.4. Noise Reduction Results on the TID2008 Dataset

Images from the TID2008 dataset exhibit various texture characteristics and a percentage of homogeneous regions, details, and edges. We last experimented with this dataset. The goal is to evaluate the performance of the proposed MADNWF in reducing Gaussian noise in degraded images with characteristics similar to those found in the TID2008 dataset, including a comparison of the resulting performance with that of the MMWF and WF. The PSNR and SSIM measurement results for each method are presented in Tabls. 7 and 8.

As shown in Tab. 7, the proposed MADNWF has average PSNR values higher than those of MMWF and WF in all noise variances. These results are similar to those obtained on the Tampere17 noise-free dataset, including the recommended kernel sizes: 3 × 3 for noise variances of 10 and below, 5 × 5 for noise variances of 20, and 7 × 7 for noise variances of 30 and above. We found the best performance of the proposed MADNW on the TID2008 dataset when applied to the image sample shown in Fig. 4(a). In this image case, the PSNR values of the proposed MADNWF are higher than those of MMWF and WF in almost all noise variances except for 10. We found the lowest PSNR values of the proposed MADNWF when applied to the house image. In this case, the proposed MADNWF is not superior to MMWF in all noise variances. However, the performance is still better than WF.

As presented in Tab. 8, the SSIM average values for the proposed MADNWF are higher than those of MMWF and WF when applied for noise variances from 30 and above. The performance is quite promising, with the recommended kernel sizes being 3x3 for noise variances of 40 and below and 5 × 5 for noise variances of 50 or higher. Compared to the kernel sizes based on the PSNR results, the recommended ones based on the SSIM results appear smaller. The goal is to reduce the loss of fine details in the image. For images with characteristics similar to those presented in Figs. 4 (a) and (e), the proposed MADNWF yields SSIM results comparable to those obtained with PSNR.

3.3. Runtime Evaluation

Finally, we evaluate the runtime of the MADNWF and compare it with the runtimes of WF and MMWF when they are actively running or executing to reduce Gaussian noise on the same four publicly available datasets: namely, Set12, Tampere17, BSD68, and TID2008. It involves collecting and analyzing real-time data to calculate performance metrics during execution. Tab. 9 demonstrates the runtime results obtained from those. When executed on the set12 and TID2008 datasets, particularly for noise variances of 10 and 20, the proposed MADNWF has better runtime than WF and MMWF. In addition to those results, it does not have a better runtime than WF, although it is still better than MMWF. We also found that larger kernel sizes required longer processing times.

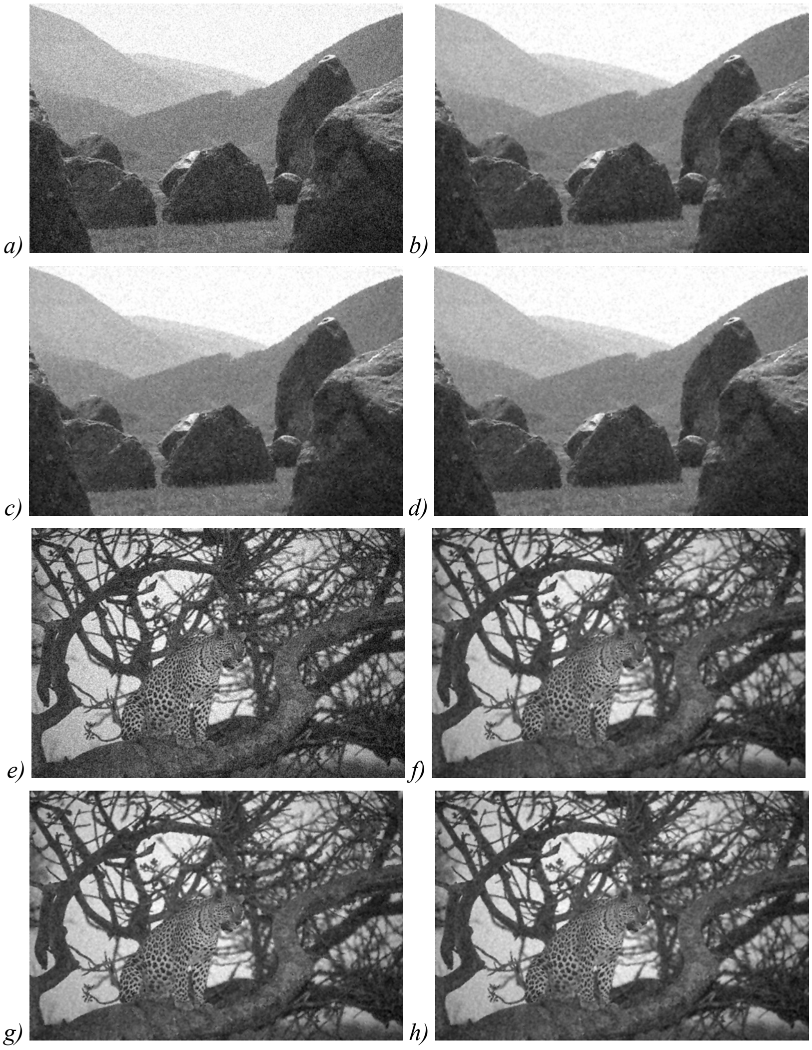

Fig. 3. Visual comparison results of WF, MMWF, and MADNWF on image samples from the BSD68 dataset. (a) A test049 image with a noise variance of 20, (b)WF [PSNR = 31.0591, SSIM = 0.5941], (c)MMWF [PSNR = 30.7792, SSIM=0.5684], (d)MADNWF [PSNR = 31.1299, SSIM = 0.5985], (e) A test016 image with a noise variance of 20, (f)WF [PSNR = 30.2387, SSIM = 0.8371], (g)MMWF [PSNR = 30.3694, SSIM = 0.8355], (h)MADNWF [PSNR = 30.2413, SSIM = 0.8386]

Conclusion

This study proposes a modified version of the Wiener filter (MADNWF), which replaces the mean value with the scaled median absolute deviation normalized (MADN). Our modification to the Wiener filter involves using a local kernel to average around each pixel. We replace this value with MADN. We used four public datasets for the experiment: Set12, the Tampere17 noise-free dataset, TID2008, and the BSD68 dataset. The first step of this study was to generate 'degraded' images. To achieve this, we added a random amount of zero-mean Gaussian noise with noise variances ranging from 10 to 90, in increments of 10, to every image in the dataset and stored it in a separate file from the clean images. Together with the WF and MMWF for comparison, we then applied MADNWF to reduce noise in degraded images. We evaluated the resulting performance by comparing the reduced noise images with the original images using the Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) metrics. The experiment results show that MADNWF improved image quality better than WF and MMWF. In conclusion, our results demonstrate that MADNWF helps reduce the noise distribution in natural images.

Tab. 7. The PSNR results of WF, MMWF, and MADNWF on the TID2008 dataset

Noise variances	The WF			The MMWF			The MADNWF		
	3 × 3	**5 × 5**	**7 × 7**	**3 × 3**	**5 × 5**	**7 × 7**	**3 × 3**	**5 × 5**	**7 × 7**
10	32,64	32,05	31,60	32,90	32,75	32,34	**32,92**	32,47	32,00
20	30,59	31,16	31,11	30,45	31,17	31,24	30,64	**31,20**	31,14
30	29,53	30,31	30,60	29,44	30,22	30,56	29,61	30,35	**30,59**
40	28,99	29,69	30,10	28,93	29,62	30,03	29,07	29,77	**30,12**
50	28,67	29,27	29,69	28,62	29,22	29,63	28,75	29,38	**29,75**
60	28,45	28,99	29,37	28,42	28,95	29,33	28,52	29,10	**29,46**
70	28,30	28,77	29,13	28,28	28,75	29,09	28,37	28,89	**29,23**
80	28,19	28,62	28,94	28,18	28,59	28,91	28,25	28,73	**29,05**
90	28,12	28,49	28,79	28,10	28,47	28,77	28,16	28,59	**28,91**

Tab. 8. The SSIM results of WF, MMWF, and MADNWF on the TID2008 dataset

Noise variances	The WF			The MMWF			The MADNWF		
	3 × 3	**5 × 5**	**7 × 7**	**3 × 3**	**5 × 5**	**7 × 7**	**3 × 3**	**5 × 5**	**7 × 7**
10	0,6165	0,5026	0,4272	**0,6377**	0,5875	0,5355	0,6290	0,5473	0,4803
20	**0,4968**	0,4380	0,3806	0,4866	0,4710	0,4378	0,4964	0,4400	0,3870
30	0,3988	0,3807	0,3395	0,3834	0,3903	0,3681	**0,4060**	0,3704	0,3242
40	0,3211	0,3307	0,3038	0,3061	0,3313	0,3173	**0,3371**	0,3211	0,2810
50	0,2603	0,2879	0,2724	0,2469	0,2851	0,2781	0,2819	**0,2836**	0,2495
60	0,2126	0,2512	0,2449	0,2008	0,2474	0,2471	0,2369	**0,2534**	0,2256
70	0,1750	0,2204	0,2210	0,1646	0,2162	0,2212	0,1998	**0,2284**	0,2065
80	0,1450	0,1941	0,2001	0,1358	0,1897	0,1993	0,1689	**0,2068**	0,1907
90	0,1209	0,1716	0,1817	0,1126	0,1673	0,1805	0,1430	**0,1879**	0,1772

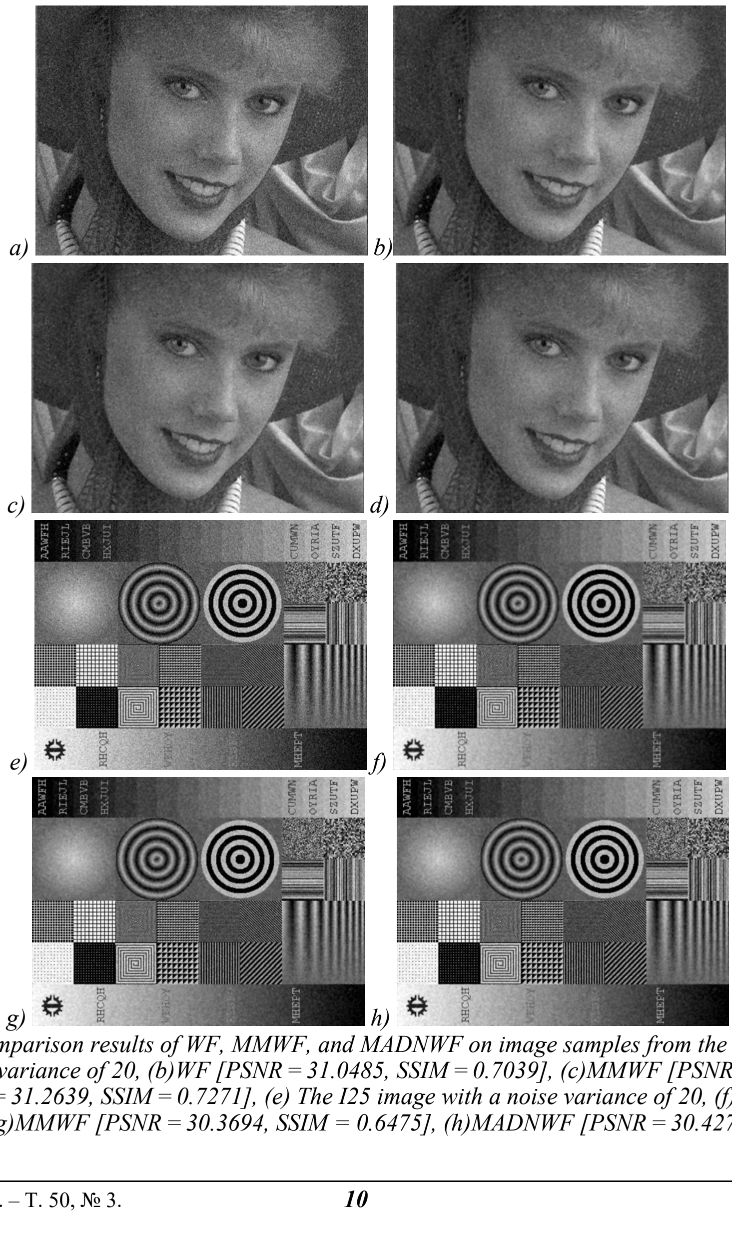

Fig. 4. Visual comparison results of WF, MMWF, and MADNWF on image samples from the TID2008 dataset. The I04 image with a noise variance of 20, (b)WF [PSNR = 31.0485, SSIM = 0.7039], (c)MMWF [PSNR = 30.8398, SSIM = 0.6774], (d)MADNWF [PSNR = 31.2639, SSIM = 0.7271], (e) The I25 image with a noise variance of 20, (f)WF [PSNR = 30.0140, SSIM = 0.7299], (g)MMWF [PSNR = 30.3694, SSIM = 0.6475], (h)MADNWF [PSNR = 30.4277, SSIM = 0.6822]

Tab. 9. The runtime results of WF, MWF, and MADMWF on Set 12, Tampere 17, BSD68, and TID2008 datasets

Dataset Name	Noise variances	The WF			The MMWF			The MADNWF		
		3 × 3	5 × 5	7 × 7	3 × 3	5 × 5	7 × 7	3 × 3	5 × 5	7 × 7
	10	42.8	116.4	89.4	43.9	116.2	87.9	**40.5**	121.2	89.4
	20	52.0	232.9	97.2	53.7	159.8	89.3	**51.5**	168.8	89.8
	30	**54.6**	94.8	95.4	57.6	97.6	92.1	55.6	97.4	89
	40	**54.3**	122.5	125.4	55.3	89.9	125.3	57.2	91.9	157.7
Set12	50	**38.9**	87.3	78.7	39.0	89.8	81.4	40.6	89.9	82.6
	60	**48.1**	82.2	81.1	49.1	84.1	83.4	51.8	86.2	85.0
	70	**45.1**	93.2	81.4	51.0	86.5	84.3	50.5	87.7	85.0
	80	**38.4**	87.7	86.2	47.0	88.8	85.2	47.5	92.1	85.5
	90	**37.7**	82.5	83.2	37.1	84.9	81.5	34.7	85.5	83.8
	10	77.6	**54.7**	72.1	80.5	57.9	75.1	83.1	58.8	76.8
	20	80.8	**55.3**	74.5	84.4	58.4	77.6	85.9	59.2	75.9
	30	**85.8**	91.8	87.1	89.3	95.3	93.3	91.2	97.8	92.9
	40	**73.9**	178.4	84.5	78.6	184.7	88.0	77.7	187.9	89.8
Tampere17	50	**198.0**	224.4	78.0	207.2	237.9	79.8	209.0	232.8	81.1
	60	**132.3**	247.8	88.0	143.0	265.1	90.3	142.3	275.8	93.3
	70	**64.3**	255.2	105.8	68.5	269.5	110.7	68.9	277.4	113.5
	80	**62.1**	280.3	108.8	66.5	295.8	115.8	67.5	296.4	117.6
	90	**55.5**	128.8	121.9	59.7	133	128	60.4	135.6	130.9
	10	**37.4**	38.7	60.9	39.8	41.6	64.0	40.5	43.2	64.2
	20	37.2	**36.4**	58.8	39.0	39.0	62.7	39.0	39.7	62.0
	30	**35.3**	37.3	61.3	38.5	39.0	64.2	38.7	39.8	65.4
	40	**39.9**	**35.5**	47.9	42.2	37.8	50.6	41.9	37.9	50.4
BSD68	50	**40.4**	**36.9**	64.4	42.9	40.4	67.8	45.0	41.2	68.7
	60	**38.5**	**37.0**	91.4	40.9	38.7	93.6	41.4	39.5	93.8
	70	46.9	36.9	**36.2**	48.8	39.4	38.0	49.8	39.3	38.2
	80	60.4	42.6	**41.4**	62.7	44.8	44.5	60.9	44.6	44.1
	90	**43.0**	45.7	37.7	45.4	48.3	39.9	44.8	49.4	39.0
	10	44.0	98.7	117.5	45.5	99.0	127.4	**43.4**	99.4	133.3
	20	42.4	54.8	64.0	42.6	55.4	62.4	**41.9**	53.8	63.7
	30	**45.3**	50.5	52.7	47.1	55.9	59.3	47.4	54.2	54.3
	40	**45.2**	49.3	83.6	47.5	54.8	87	45.9	55.1	85.6
TID2008	50	**40.5**	54.2	92.9	42.2	58.2	96.5	46.5	56.7	99.2
	60	**40.9**	72.4	111.8	44.5	74.1	117.2	46.6	72.3	113.8
	70	**42.7**	71.1	76.7	46.5	75.3	84.6	47.1	73.7	83.7
	80	**46.0**	51.1	61.8	46.5	56.7	65.6	48.0	56.2	65.1
	90	**44.4**	58.6	61.9	47.5	61.9	60.8	49.2	62.3	60.8

References

[1] Abdulah CSK, Rahman R, Ahmad F, et al. Review study of image de-noising on digital image processing and applications. J Adv Res Appl Sci Eng Technol 2023; 30(1): 331-343. doi:10.37934/araset.30.1.331343.

[2] Orazaev A, Lyakhov P, Baboshina V, Kalita D. Neural network system for recognizing images affected by random-valued impulse noise. Appl Sci 2023; 13(3): 1585. doi:10.3390/app13031585.

[3] Hussain D, Gu YH. Exploring the impact of noise and image quality on deep learning performance in DXA images. Diagnostics 2024; 14(13): 1328. doi:10.3390/diagnostics14131328.

[4] Honnatteppanavar J. Analyzing filtering techniques to eliminate Gaussian image noise. Natl Qual Res J 2021; 19(9): 1077-1084. doi:10.48047/nq.2021.19.9.NQ21182.

[5] Jiang X, Xu S, Wu J, Zhou C, Ji S. Boosting noise reduction effect via unsupervised fine-tuning strategy. Appl Sci 2024; 14(5): 1742. doi:10.3390/app14051742.

[6] Momeny M, Mohammad A, Agha M, Sheikhpour R. A noise-robust convolutional neural network for image classification. Results Eng 2021; 10: 100225. doi:10.1016/j.rineng.2021.100225.

[7] Wang T, Hu Z, Guan Y. An efficient lightweight network for image denoising using progressive residual and convolutional attention feature fusion. Sci Rep 2024; 14: 12345. doi:10.1038/s41598-024-60139-x.

[8] Hu J, Xia B, Li B, Yang W. Low-trace adaptation of zero-shot self-supervised blind image denoising. arXiv preprint 2024; arXiv:2403.12382. Available at: http://arxiv.org/abs/2403.12382.

[9] Fan L, Zhang F, Fan H, Zhang C. Brief review of image denoising techniques. Vis Comput Ind Biomed Art 2019; 2(1): 7. doi:10.1186/s42492-019-0016-7.

[10] Milanfar P, Delbracio M. Denoising: a powerful building-block for imaging, inverse problems, and machine learning. arXiv preprint 2024; arXiv:2409.06219. Available at: http://arxiv.org/abs/2409.06219.

[11] Yadav M, Tiwari N. Performance comparison of image restoration techniques using CNN and their applications. In: Proc 5th Int Conf Comput Methodol Commun (ICCMC); 2021. pp. 1146-1151. doi:10.1109/ICCMC51019.2021.9418340.

[12] Zhang K, Zuo W, Zhang L. FFDNet: toward a fast and flexible solution for CNN-based image denoising. IEEE Trans Image Process 2018; 27(9): 4608-4622. doi:10.1109/TIP.2018.2839891.
[13] Anwar S, Barnes N. Real image denoising with feature attention. In: Proc IEEE Conf Comput Vis Pattern Recognit (CVPR); 2019. pp. 3155-3164. doi:10.1109/CVPR.2019.00327.
[14] Kim Y, Soh JW, Park GY, Cho NI. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In: Proc IEEE/CVF Conf Comput Vis Pattern Recognit (CVPR); 2020. pp. 3482-3492. doi:10.1109/CVPR42600.2020.00355.
[15] Chen Y, Zhou G, Li L, Chen A, Wang Y, Li L. A high-quality self-supervised image denoising method based on SDDW-GAN and CHRNet. Expert Syst Appl 2024; 258: 125157. doi:10.1016/j.eswa.2024.125157.
[16] Chen Y, Chen F. Application in insulator infrared image denoising with dual-tree complex wavelet transform. In: Proc 2020 12th Int Conf Meas Technol Mechatronics Autom (ICMTMA); 2020. pp. 249-252. doi:10.1109/ICMTMA50254.2020.00063.
[17] Fan H, Zhang Y, Wang W, Li T. Suppressing seismic random noise based on non-subsampled shearlet transform and improved FFDNet. Front Earth Sci 2024; 12: 1408317. doi:10.3389/feart.2024.1408317.
[18] Jiang N, Zhang Y, Li Q, Yan F. An infrared thermal image denoising method focusing on noise feature learning. Opt Laser Technol 2025; 184: 112475. doi:10.1016/j.optlastec.2025.112475.
[19] Mohan R, Chokkalingam SP, Ganapathy K, Rama A. Comparative image quality analysis of spatial filters for pre-processing of CT abdominal images. Webology 2021; 18(Special Issue 4): 1449-1469. doi:10.14704/WEB/V18SI04/WEB18283.
[20] Park CR, Kang SH, Lee Y. Median modified Wiener filter for improving the image quality of gamma camera images. Nucl Eng Technol 2020; 52(10): 2328-2333. doi:10.1016/j.net.2020.03.022.
[21] Ju S, Kang SH, Lee Y. Optimization of mask size for median-modified Wiener filter according to matrix size of computed tomography images. Nucl Instrum Methods Phys Res A 2021; 1010: 165508. doi:10.1016/j.nima.2021.165508.
[22] Lee S, Lee Y. Performance evaluation of median-modified Wiener filter algorithm in high-resolution CMOS radio-magnetic X-ray imaging system: an experimental study. Nucl Instrum Methods Phys Res A 2021; 1010: 165509. doi:10.1016/j.nima.2021.165509.
[23] Zhou Z, Qu Y, Zhu B, Zhang B. Detection of typical transient signals in water by XGBoost classifier based on shape statistical features: application to the call of southern right whale. J Mar Sci Eng 2024; 12(9): 1596. doi:10.3390/jmse12091596.
[24] Yan P, Jeong J, Sagan N, Weissman T. Universal discrete filtering with lookahead or delay. arXiv preprint 2025; arXiv:2501.10609. Available at: http://arxiv.org/abs/2501.10609.
[25] Xiuyu Y, Liyong T, Feng C. Thermal infrared imaging for conveyor roller fault detection in coal mines. PLoS One 2024; 19(7): e0307591. doi:10.1371/journal.pone.0307591.
[26] Howell DC. Median absolute deviation. In: Encyclopedia of statistics in behavioral science. Vol. 3. Chichester: Wiley; 2005. pp. 1193-1193. doi:10.1002/9781118445112.stat06232.

Author's information

Giat Karyono (b. 1983) graduated from the University of Technical Malaysia Melaka in 2024, majoring in Information and Communication Technology with a focus on image processing. He currently works as a lecturer at Universitas Amikom Purwokerto, Indonesia. My research interests include image processing, computer vision, and programming. E-mail: *giatkaryono@amikompurwokerto.ac.id*

Received April 29, 2025. Final version – September 07, 2025