Design and Implementation of an Image Denoising Platform Using MATLAB App Designer

Usman Ul Haq Abbasi
Faculty of Electrical Engineering,
Ghulam Ishaq Institute of Engineering Sciences and Technology,
Topi, Pakistan.
u2021749@giki.edu.pk

Yahya Arshad
Faculty of Electrical Engineering,
Ghulam Ishaq Institute of Engineering Sciences and Technology,
Topi, Pakistan.
u2021708@giki.edu.pk

***Abstract*—This paper presents a MATLAB-based application for image denoising, developed using the MATLAB App Designer. The app enables users to load images, add adjustable noise, and apply filters with customizable levels to remove noise. It also provides performance metrics, such as Mean Squared Error (MSE), to evaluate the effectiveness of the filters, offering an intuitive platform for exploring image processing techniques.**

***Keywords*—Image Processing, Image Denoising, Denoising filters, Convolutional Neural Networks**

I. INTRODUCTION

Image noise refers to random variations in brightness or color that can degrade the quality of an image. It often originates from electronic noise in the sensors and circuitry of digital cameras or scanners, but can also result from factors such as film grain or shot noise in photon detectors. While some types of noise can be reduced or removed, others are more challenging to address. The presence of noise can interfere with subsequent image processing tasks, such as video processing [1], image analysis, and tracking [2], by distorting results and compromising accuracy. As a result, image denoising becomes a vital step in modern image processing systems, improving image quality and ensuring more accurate and reliable analyses.

Image denoising remains a significant challenge for researchers, as removing noise from an image often leads to unwanted artifacts and blurring. The main goal of denoising is to eliminate the noise while preserving the true content of the image. However, this task is complicated by the fact that noise, edges, and textures all share high-frequency components, making it difficult to differentiate between them during the denoising process. As a result, some important image details may be lost in the process. Despite these challenges, recovering meaningful information from noisy images while maintaining high quality is a critical problem in modern image processing that continues to drive research in the field [3].

In this paper, we present a MATLAB-based image denoising application designed to assist users in denoising their images. The application not only enables users to apply various filters but also provides an analytical platform to assess the effectiveness of these filters for different types of noise. By offering a practical interface for both performing and evaluating denoising tasks, the app helps users choose the most suitable filters for specific noise conditions. Additionally, we discuss the efficacy of each filter in dealing with various types of image noise, providing insights into their strengths and limitations.

II. LITERATURE REVIEW

Image denoising remains a fundamental challenge in the field of image processing, affecting various types of images, including binary, grayscale, and color formats [4]. Common noise types impacting image quality include additive white Gaussian noise (AWGN) [5], impulse noise [6], quantization noise [7], Poisson noise [8], and speckle noise [9]. AWGN typically arises from analog circuitry, whereas impulse noise, speckle noise, Poisson noise, and quantization noise result from issues such as manufacturing defects, bit errors, or insufficient photon counts [10]. To address these challenges, a wide range of filters and denoising techniques have been developed and employed.

To address these noise challenges, various types of noise reduction filters have been developed, typically categorized into six groups: linear, non-linear, adaptive, wavelet-based, partial differential equation (PDE), and total variation filters. Linear filters reduce noise by modifying output pixels based on neighboring ones through matrix multiplication. However, non-linear filters, such as the median filter (MF) [11], are preferred for their ability to preserve edges while effectively reducing noise, unlike linear filters that tend to blur edges. Adaptive filters, including least mean square [12] and recursive mean square [13], use statistical methods for real-time applications. Wavelet-based filters, which transform images into the wavelet domain, are particularly effective at reducing additive noise [14]. The PDE-based denoising method [15] excels in removing noise while preserving edges, ensuring that image details remain intact. Meanwhile, total variation denoising [16] is a highly effective edge-preserving filter that simultaneously smooths noise in flat regions while preserving edges, even at low signal-to-noise ratios.

In recent years, Deep Learning algorithms, particularly Convolutional Neural Networks (CNN) [17], have gained significant attention for image denoising, thanks to their ability to address the limitations of traditional techniques. CNNs offer several advantages, such as very deep architectures [18], which enhance their capacity to capture complex image features. Moreover, CNNs are highly effective for parallel computation on modern GPUs, significantly improving runtime performance. These advantages make CNNs highly suitable for a wide range of image denoising applications.

This paper proposes a MATLAB-based application for image denoising, specifically designed to handle Gaussian, speckle, and Salt & Pepper (impulse) noise. The application incorporates a variety of filters, including the median filter, Gaussian filter, and rank order filter, to effectively reduce noise. Additionally, the app allows users to analyze the effects of different filters on various types of noisy images and choose from a wide range of options to efficiently denoise their images. To further assess performance, the app utilizes noise metrics such as Mean Squared Error (MSE) to evaluate the level of noise and measure the effectiveness of the applied filters. Furthermore, this paper also discusses the use of pre-trained deep learning models like 'DnCNN' [19] in image denoising.

III. Methodology

We have developed a MATLAB-based application for image denoising utilizing the MATLAB App Designer. This application enables users to upload their images, apply noise with adjustable intensity, and experiment with a variety of filters at customizable levels to effectively reduce the noise. Additionally, the application provides built-in performance evaluation metrics, such as Mean Squared Error (MSE), to analyze the denoising effectiveness. The subsequent sections provide a detailed discussion of the application's design, the types of noise and filters incorporated, and the use of metrics like MSE for noise analysis.

A. Application Design

The interface of the application is shown in Fig. 1. The design is structured to provide a user-friendly experience for performing image denoising tasks. At the top, users can load their desired image into the application by clicking the 'Load Image' button.

Below this, the 'Image Noise Parameter Settings' panel allows users to select the type of noise they want to apply to the image, such as Salt & Pepper, Gaussian or Speckle, and adjust the noise intensity using a slider. Once configured, users can generate and preview the noisy image by clicking 'Load Noisy Image.' The next panel, titled 'Image Denoising Filter Settings,' provides options for selecting the desired filter type, such as the Median Filter, along with a slider to adjust the filter intensity. Users can then apply the selected filter by clicking 'Load Filtered Image' and download the processed image if needed.

On the right-hand side, the 'Image Comparison' section visually displays three versions of the image: the original image, the noisy image, and the filtered image, allowing users to observe the denoising results directly in real-time. Finally, at the bottom of the interface, the 'Image Noise Analyses' panel provides quantitative feedback on the denoising process by displaying the Mean Squared Error (MSE) values for both the noisy and filtered images, enabling users to evaluate the effectiveness of the applied filter.

The application is versatile in its use as it can function as a tool for analyzing the performance of different filters or for denoising pre-existing noisy images. This dual-purpose

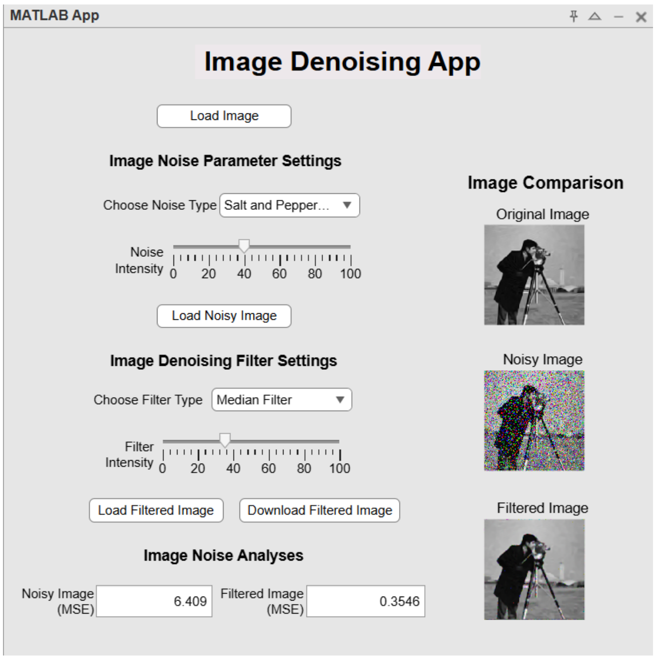

Fig. 1. Image Denoising App Interface.

design allows users to explore the effects of various noise types and filters on an image, offering an educational platform to better understand the underlying principles of image denoising and its practical implementation.

B. Noise Simulation

The application allows users to add three different types of noise to an image: Gaussian, salt-and-pepper (impulse), and speckle noise. These noises are applied using the 'imnoise()' function in MATLAB. To adjust the noise intensity, the app features a slider that enables users to set the desired noise level. This slider alters the variance for Gaussian noise, the density for salt-and-pepper noise, and the magnitude for speckle noise, providing users with a wide range of noisy images.

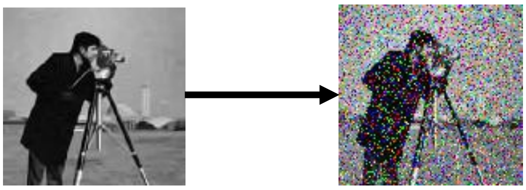

Fig. 2. Comparison of Original and Noisy Image

C. Filter Implementation

The application implements various noise reduction filters, including Gaussian, Median, Average, Rank Order, Wiener, Bilateral, Wavelet Transform, and a Hybrid Median-Wiener filter, designed to target specific noise types such as Gaussian, salt-and-pepper, and speckle noise. The Gaussian Filter reduces Gaussian noise through smoothing, while the Median Filter removes salt-and-pepper noise by replacing each pixel with the median of its neighbors. The Wiener Filter adapts to local image statistics, effectively reducing Gaussian and speckle noise, and the Wavelet Transform decomposes the image to filter out high-frequency noise, particularly speckle.

Key parameters such as kernel size, filter order, and window size are adjusted to optimize each filter's performance based on the type of noise. For example, the Gaussian Filter kernel size is modified to control smoothing, and the Median Filter window size balances noise reduction and edge preservation. The Rank-Order Filter allows customization to handle different noise patterns, while the Bilateral and Wiener Filters adaptively adjust parameters like spatial and intensity ranges to reduce noise and preserve edges. This customization ensures effective noise reduction and high-quality denoising results.

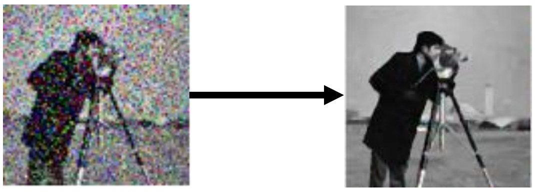

Fig. 3. Comparison of Noisy and Filtered Image

D. *Performance Metrics*

The Mean Squared Error (MSE) is employed as a performance metric to assess the effectiveness of the noise reduction filters. MSE is particularly useful because it quantifies the difference between the original and filtered images, providing insight into the amount of noise remaining after the filtering process. A lower MSE value indicates that the filtered image is closer to the original, suggesting that the filter has successfully reduced noise while preserving important image features. This makes MSE an effective tool for analyzing the efficacy of different filters.

To calculate the MSE of an image, the following equation is used:

$$MSE = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} (I(i,j) - K(i,j))^2$$

Here;

- I(i,j) represents the pixel value of the original image at position (i,j)
- K(i,j) is the pixel value of the filtered image at position (i,j)
- m and n are the dimensions of the image.

E. *App Workflow Diagram*

The following flowchart explains the working of the application:

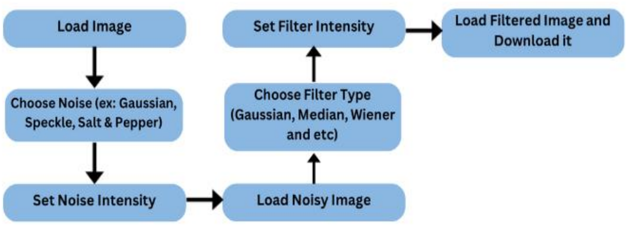

Fig. 4. Workflow Diagram of the Application

F. *Additional Work*

We further implemented denoising using a pre-trained deep learning model called 'DnCNN' on MATLAB, which works best on grayscale images with Gaussian noise. The model is trained on various types of grayscale noisy images to effectively distinguish and remove noise while preserving image details. Fig. 5 illustrates the model's impact on a Gaussian noise image, demonstrating its effectiveness in significantly reducing noise and enhancing image quality.

Fig. 5. Comparison of Original, Noisy and Filtered 'DnCNN' Images

IV. RESULTS

The performance of the image denoising application was evaluated on three types of noise: Gaussian, Salt-and-Pepper, and Speckle. For each noise type, a comparative analysis was performed using eight different filters, including Gaussian, Median, Average, and advanced techniques such as Wavelet Transform and the Hybrid Median-Wiener filter. The effectiveness of the filters was assessed quantitatively, by calculating the Mean Squared Error (MSE) between the original and denoised images. This section presents a detailed breakdown of the results for each noise type.

A. *Gaussian Noise*

The MSE of the filtered images for each of the filter are shown below in Table I.

Table I.
MSE of various gaussian noise filtered images

Type of Denoising Filter Type	MSE of filtered image when given following inputs (MSE)			
	2.5	5	7.5	10
Gaussian	0.149	0.2422	0.3375	0.4458
Median	0.3361	0.5851	0.725	0.9877
Average	0.5793	0.8866	0.93	1.124
Rank Order	0.4078	0.5829	0.9268	1.22
Wiener	0.1987	0.3895	0.5445	0.7056
Bilateral	2.482	4.977	7.498	9.966
Wavelet	0.3835	0.8883	1.909	2.96
Hybrid[b]	0.1819	0.326	0.5184	0.7135
DnCNN[a]	0.27013	0.77673	1.5992	2.6719

[a] Not integrated in the application
[b] Both Wiener and Median Filters are applied

The evaluation of denoising filters in Table I shows the Gaussian filter as the most effective, consistently achieving the lowest mean squared error (MSE) across all noise levels. This reliability makes it the preferred choice for preserving image quality during noise reduction. The Wiener and Hybrid filters also performed well, particularly at lower noise levels, though their effectiveness declined as noise intensity increased. Overall, the Gaussian filter stands out as the best option for denoising Gaussian noise images.

B. *Salt & Pepper Noise (Impulse Noise)*

The MSE of the Salt & Pepper Noise filtered images for each of the filter are shown below in Table II.

Table II.
MSE of various salt & pepper noise filtered images

Type of Denoising Filter Type	MSE of filtered image when given following inputs (MSE)			
	2.5	5	7.5	10
Gaussian	0.1397	0.2457	0.3478	0.5185
Median	0.03922	0.0792	0.2002	0.2999
Average	0.6164	0.7658	0.8957	1.173
Rank Order	0.01338	0.05551	0.194	0.3193
Wiener	0.5216	0.6144	0.77	0.9015
Bilateral	2.463	4.925	7.407	9.922
Wavelet	0.3471	1.13	1.974	2.843
Hybrid[b]	0.1003	0.1169	0.2217	0.3479
DnCNN[a]	0.33611	0.8798	1.8253	3.0576

[a] Not integrated in the application
[b] Both Wiener and Median Filters are applied

The evaluation of denoising filters for images with salt-and-pepper noise, as detailed in Table II, identifies the Rank Order and Median filters as the most effective, consistently achieving the lowest mean squared error (MSE) across all noise levels. Both filters demonstrated superior performance in preserving image quality, making them the preferred choices for this type of noise. The Gaussian and Hybrid filters showed moderate effectiveness, with the Hybrid filter displaying better resilience at various noise levels. Overall, the Rank Order and Median filters stand out as the most reliable options for denoising salt-and-pepper noise.

C. *Speckle Noise*

The MSE of the Speckle Noise filtered images for each of the filter are shown below in Table III.

Table III.
MSE of various speckle noise filtered images

Type of Denoising Filter Type	MSE of filtered image when given following inputs (MSE)			
	2.5	5	7.5	10
Gaussian	0.1517	0.2484	0.3274	0.403
Median	0.4279	0.7036	0.9912	1.211
Average	0.5446	0.7663	0.8262	0.9081
Rank Order	0.508	0.8904	1.212	1.526
Wiener	0.2583	0.4661	0.6442	0.8518
Bilateral	2.493	4.925	7.466	9.996
Wavelet	0.6426	1.461	2.295	3.516
Hybrid[b]	0.2575	0.4963	0.7253	0.9687
DnCNN[a]	0.18375	0.56623	1.1428	1.9532

[a] Not integrated in the application
[b] Both Wiener and Median Filters are applied

The evaluation of denoising filters for images with speckle noise, as shown in Table III, highlights the Gaussian filter as the most effective, consistently achieving the lowest mean squared error (MSE) across all noise levels. The Wiener, Average and Hybrid filters also performed well, maintaining relatively low MSE values and proving effective even at higher noise levels. Overall, the Gaussian filter stands out as the most reliable option for speckle noise reduction, with the Wiener, Average and Hybrid filters serving as strong alternatives.

D. *Overall effectiveness*

The performance of denoising filters varies across different types of noise. For Gaussian noise, the Gaussian filter consistently achieved the lowest mean squared error (MSE), making it the most effective option. For salt-and-pepper noise, the Median and Rank Order filters proved the most effective. In the case of speckle noise, the Gaussian filter was the best. Overall, the Gaussian filter was the most reliable across different noise types, with the Median filter excelling in salt-and-pepper noise reduction.

V. DISCUSSION

The application uses nine different filters, each with its own strengths and weaknesses for specific noise types. The Gaussian filter is effective at reducing Gaussian noise but tends to blur edges. The Median filter works very well for Salt-and-Pepper noise, preserving edges, though it is less effective for other noise types. The Average filter is simple and efficient but often causes noticeable blurring. The Rank Order filter is strong against Salt-and-Pepper noise but can be slow due to its complexity. The Wiener filter adapts to local image characteristics and performs well with Gaussian and Speckle noise but requires accurate noise estimates. The Bilateral filter reduces noise while keeping edges sharp but struggles with intense noise and is computationally demanding. The Wavelet Transform handles high-frequency noise like Speckle but can over-smooth images. The Hybrid Median-Wiener filter combines the strengths of Median and Wiener filters, performing well across multiple noise types, though it is more complex. Finally, DnCNN is highly effective for Gaussian noise and preserves details, but it is limited to grayscale images and requires significant computational power.

While these filters are effective for specific tasks, the application has some limitations. Many filters are computationally heavy, making them unsuitable for real-time use. Their performance often depends on the type of noise, reducing their versatility. Advanced models like DnCNN are not yet integrated for color images, limiting the scope of the application. The evaluation metric, Mean Squared Error (MSE), though helpful, does not account for visual quality or perceptual similarity. Additionally, the application does not handle complex noise types, such as motion blur, and cannot process videos in real time.

Future improvements can make the application more versatile and powerful. Adding advanced deep learning models, such as GANs (Generative Adversarial Networks), can improve detail retention and noise reduction. Supporting more noise types, such as motion blur and compression artifacts, would broaden its use cases. Adding evaluation metrics like Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) would provide a better understanding of filter performance. Expanding compatibility to mobile and web platforms, as well as optimizing for high-resolution and multispectral images, could make the tool more accessible and practical for a wide range of applications. Future enhancements may include exploring various more filters such as adaptive filters, PDE-based filters, deep neural networks, total variation-based filters, and others.

VI. Conclusion

This study successfully demonstrates the design and implementation of a MATLAB-based image denoising application, offering an intuitive interface and robust functionality for addressing various noise types. By integrating a range of customizable filters and quantitative performance metrics like MSE, the application provides a comprehensive tool for both practical use and educational exploration of image denoising techniques.

The impressive performance of filters, such as the Gaussian filter for Gaussian and Speckle noise, and the Median and Rank Order filters for Salt-and-Pepper noise, highlights its effectiveness. Future enhancements, including the addition of advanced deep learning models, real-time processing, and expanded filter libraries, could further elevate its potential, making it an indispensable tool for researchers and professionals in image processing.

References

[1] Wang, L., T. Liu, G. Wang, K. L. Chan, and Q. Yang, "Video tracking using learned hierarchical features," IEEE Transactions on Image Processing, vol. 24, no. 4, pp. 1424-1435, Apr. 2015.

[2] A. I. Dell, J. A. Bender, K. Branson, I. D. Couzin, G. G. de Polavieja, L. P. Noldus, A. Pérez-Escudero, P. Perona, A. D. Straw, M. Wikelski, and U. Brose, "Automated image-based tracking and its application in ecology," Trends Ecol. Evol., vol. 29, no. 7, pp. 417–428, 2014.

[3] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," Visual Computing for Industry, Biomedicine, and Art, vol. 2, no. 1, p. 7, 2019.

[4] Hance, G. A., "Computer vision and image processing software tool with application to color segmentation for skin cancer diagnosis," Master's thesis, Southern Illinois University at Edwardsville, 1996.

[5] Buades, A., B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," Multiscale Modeling & Simulation, vol. 4, no. 2, pp. 490–530, 2005.

[6] A. Awad, "Denoising images corrupted with impulse, Gaussian, or a mixture of impulse and Gaussian noise," Engineering Science and Technology, an International Journal, vol. 22, no. 3, pp. 746–753, 2019.

[7] W.-K. Bingo, Y. F. H. Charlotte, D. Qingyun, and J. D. Reiss, "Reduction of quantization noise via periodic code for oversampled input signals and the corresponding optimal code design," Digit. Signal Process., vol. 24, pp. 209–222, 2014.

[8] A. Rajagopal, R. B. Hamilton, and F. Scalzo, "Noise reduction in intracranial pressure signal using causal shape manifolds," Biomed. Signal Process. Control, vol. 28, pp. 19–26, 2016.

[9] A. E. Ilesanmi, O. P. Idowu, U. Chaumrattanakul, and S. S. Makhanov, "Multiscale hybrid algorithm for pre-processing of ultrasound images," *Biomed. Signal Process. Control*, vol. 66, p. 102396, 2021.

[10] B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, "Image denoising review: from classical to state-of-the-art approaches," Inf. Fusion, vol. 55, pp. 220–244, 2020.

[11] C. Jaspin-Jeba-Sheela and G. Suganthi, "An efficient denoising of impulse noise from MRI using adaptive switching modified decision-based unsymmetric trimmed median filter," Biomed. Signal Process. Control, vol. 55, p. 101657, 2020.

[12] P. L. Feintuch, "An adaptive recursive LMS filter," Proc. IEEE, vol. 64, no. 11, pp. 1622–1624, 1976.

[13] F. Dinga, Y. Wanga, and J. Ding, "Recursive least squares parameter identification algorithms for systems with colored noise using the filtering technique and the auxiliary model," Digit. Signal Process., vol. 37, pp. 100–108, 2015.

[14] C. Stolojescu-Crisan, "A hyperanalytic wavelet based denoising technique for ultrasound images," in Proc. Int. Conf. Bioinformatics and Biomedical Engineering, 2015, pp. 193–200.

[15] S. Kim, "PDE-based image restoration: A hybrid model and color image denoising," IEEE Trans. Image Process., vol. 15, no. 5, pp. 1163–1170, 2006.

[16] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," IEEE Trans. Image Process., vol. 18, no. 11, pp. 2419–2434, 2009.

[17] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," IEEE Trans. Image Process., vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learn. Represent., 2015.

[19] V. Murali and P. V. Sudeep, "Image denoising using DnCNN: An exploration study," in Advances in Communication Systems and Networks: Select Proceedings of ComNet 2019, Springer Singapore, 2020, pp. 847-859.