

An Image Denoising Scheme Remove Unwanted Pixel Using NLM with Sprint Deep Learning Network

D. Hemanand¹, L.Selvam², M.Arunachalam³, D.Suresh⁴

Submitted: 10/09/2022

Accepted: 20/12/2022

Abstract: An automatic image processing system is vital for detecting life-threatening diseases like tumor diagnosis. Machine learning is significant in processing medical images to detect the diseases' signs and symptoms. The essential need for image processing is the earlier detection of diseases. But due to the image complexities, the inefficiency of the conventional methods delivers poor performance in detecting the damaged cell's shape, extraction, exact size, and location. The primary goal of achieving an enhanced automatic system for preprocessing, classifying, segmenting, detecting, and sample recognition remains a challenging task. To overcome this, we proposed an improved SPRINT algorithm with NLM (Non-Local Mean filtering) algorithm. The proposed mechanism computes the original images to remove the noises, and the non-local filtering algorithm considers the high-extent redundancy of the normal images. As a result, the input images are processed using the weighted average value of the entire pixel to obtain noise-free or noise-less pixel images. Additionally, the SPRINT algorithm applies the minimum description length principle for achieving accuracy in the expected details. It contains an attribute table and histogram for holding the indexing of data records, class identification, and attribute values. Finally, our enhanced SPRINT can solve this trouble by preserving the fine details of an image while denoising. To evaluate the performance of the proposed system, a comparison work is carried out between the enhanced SPRINT algorithm with a conventional neural network (CNN) [12] and deep learning-based patch label denoising methods (LossDiff) [13]. The proposed SPRINT algorithm achieves 97% accuracy, which is far better than the CNN with 91% and LossDiff with 85% accuracy.

Keywords: Image noise, Machine learning, Deep Learning networks, non-local means algorithm, Image classification, and Segmentation

Introduction

Digital images play a vital role in several domains like signature approval, medical diagnosis, traffic monitoring, handwriting recognition, signature approvals, etc. The image sensor is a commonly used approach for collecting the image datasets. But these images are subject to various noises due to natural occurrences and defective devices. It results in delivering of poor-quality images. Image denoising is an effective approach for inspecting and removing the noises from the images. At the same time, it will retain a maximum of original details from the images. The resultant de-noised image is free from image noise. [1-5]. Image denoising is considered a major hurdle and complex task in image processing. The need for effective image processing, especially in the medical field, has grabbed much attention among researchers.

In recent days, several denoising techniques are evolved using

various algorithms. But the existing conventional machine learning is effective in resulting the accuracy. The computer-aided techniques are failed to manage this complicated process resulting in image blurring or images with fewer details [6-11]. We proposed a SPRINT algorithm with a minimum description length principle for achieving accuracy in the expected details. The proposed mechanism contains an attribute table and histogram for holding the indexing of data records, class identification, and attribute values. A comparison work is carried out between the proposed SPRINT algorithm with a conventional neural network (CNN) and the deep learning-based patch label denoising method (LossDiff).

2.Related work

Shreyasi Ghose et al. [12] proposed a conventional neural network for image denoising. About 10% of gaussian white noise is added to the input image, and the CNN model is utilized for denoising. Then the quantitative analysis is performed on the PSNR (peak signal to noise ratio), MSE (mean square error), and SSIM (structural similarity index measurement). The result shows maximum Gaussian noises are removed by restoring the image's original details.

MurtazaAshraf et al. [13] proposed a deep learning-based patch label denoising method (LossDiff) for enhancing the classification accuracy of the whole-slide image analysis. The main motto of this work is to address the patch-based label noise, which results in low classification accuracy. The proposed system

¹Professor, Department of Computer Science and Engineering, S.A. Engineering College (Autonomous) Thiruverkadu, Chennai-600077, Tamil Nadu, India.Email: drhemanandphd@gmail.com

²Professor and Head, Department of Information Technology KSR Institute for Engineering and Technology, Tiruchengode-637215, Tamil Nadu, India.Email: umaselvam_35@yahoo.com

³Professor, Department of Information technology Sri Krishna College of Engineering and Technology, Kuniamuthur, Coimbatore 641008, Tamil Nadu, India, Email: arunachalam@skcet.ac.in

⁴Professor, Department of Computer Science and Engineering, PSNA College of Engineering and Technology, Dindigul-624622, Tamil Nadu, India.Email: sureshdgl@gmail.com

is very effective in classifying the noisy patches and correctly labeled patches.

Chunwei Tian et al. [14] conducted a comparative analysis of the deep learning technologies used for image denoising. Initially, the author examines the CNN approaches various noisy images such as noisy white images, noisy hybrid images, real noisy images, and blind denoising. Then various kinds of deep learning methods are analyzed with their principles and research motivations. Finally, comparison work is conducted based on the quantitative and qualitative analyses concerning the public denoising datasets. Saeed Izadi et al. [15] researched various deep neural network-based image denoising methods. Initially, the author highlights the basic requirements in image denoising and denoising problem in handling the benchmark datasets. The author categorizes deep denoisers into supervised and unsupervised methods in this work. Each method's technical specifications and challenges facing real-time applications are discussed in detail.

Later, Cheng et al. [16] developed a novel subspace attention module for reconstructing the images with original details and removing the irrelevant information. Hu et al. [17] proposed an advanced 3D auto-correlation mechanism for instantaneously extracting horizontal, vertical, and channel-wise axes. Compared to the regular auto-correlation attention modules, the 3D module is lightweight, avoids impenetrable links, and performs optimized operations.

Zamir et al. [18] implemented a supervised attention module for a multi-stage architecture. A cross-stage information exchange module is introduced at the early stage of image denoises to enhance the feature fusion. Liu et al. [19] developed a noise component replacement mechanism through corresponding encoding. Initially, the noisy input is converted into a low-resolution clean image.

Li et al. [20] proposed a cross-patch graph convolutional approach for handling cross-patch long-range contextual needs. Similar patches are collected as primary patches, and features are extracted using ensembles for an accurate, clean patch.

3. Gaussian Filtering

Gaussian filters change the input signal using Gaussian functions to remove noise from the noisy input image. These Gaussian functions are employed in various fields, such as (i) these are used to explain the probability function of noise. (ii) Gaussian functions are also employed in mathematics (iii) these are smoothing workers. A gaussian function is given below:

$$G_u(a) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{a^2}{2\sigma^2}} \quad (1)$$

Generally, two 2-D Gaussian functions are employed for imaging functions, and it is given by:

$$G_u(a,b) = \frac{1}{2\pi\sigma^2} e^{-\frac{a^2+b^2}{2\sigma^2}} \quad (2)$$

The above function (Eqn. 2) is a simple product of two 1-Dimensional Gaussian functions. Eqn. 2 σ represents the standard deviation of the Gaussian function, a denotes the distance from the source on the x axis, and b mentions the distance from the source on the y axis. Then, a convolution matrix is formed from the values acquired from the above

distribution function. This convolution matrix is implemented to the original transmitted image. The Gaussian filters are also known as non-uniform low pass filters. It is a very effective method for noise removal, and the pixel weights afford higher significance to the edge near pixels. Gaussian filters are computationally so efficient. The noise removal takes more time, reducing the image's fine details.

4. Median filtering

The median filters are known as non-linear filters and use non-linear filtering approaches to remove image noise. Image edges are preserved during the noise removal process using this median filtering. For that reason, median filters are mostly employed in various image processing techniques for noise removal. The median filters are run through the noisy image pixel by pixel and then replace the detected noisy pixel with neighboring median pixels. The outline of adjacent pixels is known as a 'window'. The median filtering function is given by,

$$\tilde{\gamma} = \begin{cases} X_{(n+1)/2} & \text{if } n \text{ is odd} \\ \frac{1}{2}(X_{n/2} + X_{1+n/2}), & \text{if } n \text{ is even} \end{cases} \quad (3)$$

Where n represents the number of pixels present in the window, i.e., window size. If the window size is odd (i.e., it contains the odd number of pixels), then the median value is easily estimated, which is the middle value of a numerically sorted window. More than a median is present if the window possesses an even number of pixels. Median filters are a kind of linear Gaussian filtering, and it is a type of smoothing technique. It preserves sharp image edges. The calculated median measure is similar to the neighboring pixels, which will not impress the other pixels. It preserves the image boundaries. It provides poor performance in the removal of Gaussian noise. It can remove noisy pixels only if the noise-affected pixels take up less than half the neighboring region.

5. Wiener Filtering

The Wiener filter is an optimally stationary linear filter employed for additive noise and image blurring. The design function of the wiener filter assumes a dissimilar approach. The first assumes the knowledge of the master signal and the spectral noise properties. The second one is a linear time-invariant filter that produces the output close to the master signal as much as possible. At first, some assumptions have to be made to calculate the filter function: original signal and image noise (additive) are stationary linear and random processes with known spectral properties, including auto-correlation and cross-correlation functions. The requirement of this filter is it must be physically manageable/casual. The Wiener filter is given by,

$$y(t) = g(t) * [x(t) + n_a(t)] \quad (4)$$

Where $g(t)$ is the impulse response of the wiener filter, $y(t)$ is the estimated signal, $x(t)$ is the original input signal, and $n_a(t)$ is the additive noise signal. This filtering technique uses prior statistical knowledge of noise. Hence, we can obtain efficient noise removal. By using this statistical information, we can minimize the mean square error. Wiener filter cannot preserve the fine image details while image denoising takes place.

6. Non-Local Mean filtering

Non-local means is an algorithm used for image denoising in digital image processing. Some other filters, like local filters, update a pixel's value with an average value of neighboring pixels. The non-local filter also updates a pixel value by employing a weighted average of the neighboring pixels. The observed values are most similar to the local filter values. But those local smoothing approaches cannot preserve the fine details like structure and texture of the image. Thus, the image structures and details are smoothed out since they act in entire well-designed prospects as image noise. The non-local filtering algorithm attempts to accept the advantage of the high-extent redundancy of whatever normal image. This non-local means algorithm is used to compute noise-free or noise-less pixel intensity as a weighted average value of entire pixel intensities in that image. These pixel weights are proportional to the similarity within the processing pixel's local neighborhoods and those neighboring pixels' local neighborhoods. Using the given formula, the de-noised image at any pixel is calculated, ideally acting over the full image.

$$N_{lm}[U](p) = \sum_{q \in I} w_i(p, q) v_i(q) \quad (5)$$

Where the pixel weight $w_i(p, q)$ reckons on the distance between the pixels p and q which are the observed gray vector points [2], and the distance d_s is given by,

$$d_s = \|v_i(N_p) - v_i(N_q)\|_{2,c}^2 \quad (6)$$

The weight $w_i(p, q)$ [4,8] is represented as

$$w_i(p, q) = \frac{1}{Z(p)} e^{-\frac{\|v_i(N_p) - v_i(N_q)\|_{2,c}^2}{k^2}} \quad (7)$$

The NLM filter works well in any noise removal process. It provides better results when compared to other existing denoising filters such as Gaussian, Median, and Wiener filters, which leads to image smoothening.

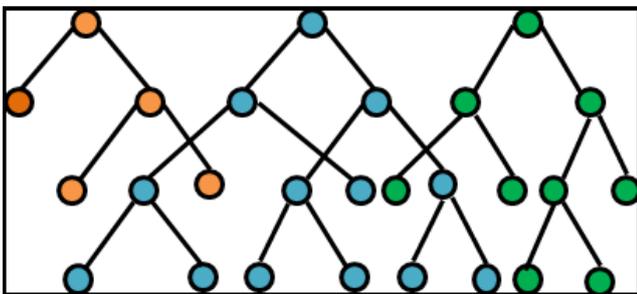


Fig.1 SPRINT structure

6.1 The Basic Idea of Improved SPRINT

The proposed SPRINT algorithm contains two data structures attribute table and histogram. The three components of the property sheet are data record indexing, attribute value, and class identification. The memory area will store only the list of attributes. The attribute table was divided with the associated

child node with node expansion. The Histograms are linked to determine the attribute node distribution type. Each node on a class distribution chart represents two histograms, such as Cbelow and Cabove, in terms of numerical attributes. The former describes the sample type taken for distribution.

Additionally, it describes the distribution type for untreated samples and the worth of the two updated samples. In the histogram, only one node's attribute is used for defining the discrete distribution class. The minimum description length principle in SPRINT pruning is implemented to enhance performance.

7. Experimental Results

This section discusses the experimental setup and obtained results in detail. The experimental work is carried out in Matlab with the dataset containing three images as shown in fig2 a Barbara image, fig 2 b Lena image, and fig2 c with Peppers image. The experimental work contains two phases. Initially, the dataset images are computed with different classifiers. The obtained resultant images after removing the noise are shown with their original images. The obtained accuracy parameter result with the proposed system is described in table 1, and table 2 shows the parameter evaluation after filtering by each respective classifier. Next, the proposed systems obtained accuracy is compared with the existing LossDiff and CNN. The comparison results are described in Table 3.



Fig. 2 : (a) (b) (c) Original Images

Figure 2 shows the original images taken for the experimental work. It contains three images such as (a) is the Barbara image, (b) is the Lena image, and (c) is the Pepper image. In these images, only noises are improved, and further classification is done with the proposed algorithm. The images are processed with the different classifiers, and its result is discussed in further sections.

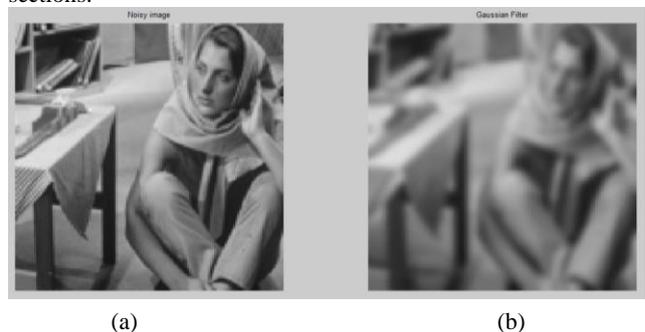


Fig. 3: (a) Noisy Image (b) Filter Image using Gaussian Filter.

Figure 3 (a) depicts the original Barbara image with noises, and figure 3 (b) depicts the filter images applying the Gaussian Filter. Figure 3 (b) is the resultant images after removing the noises from the original image. The Gaussian functions working mechanism for eliminating the noises are explained in the

Gaussian filter explanation in section 3. It applies the probability function of noise and employs a mathematics function for smoothening the images. Only the resultant images, as shown in figure 3(b), are obtained based on these functions.



Fig. 4: (a) Noisy Image (b) Filter Image using Gaussian Filter

Figure 4 (a) depicts the original Lena image with noises, and figure 4 (b) depicts the filter images applying the Gaussian Filter. Figure 4 (b) is the resultant images after removing the noises from the original image. The Gaussian functions working mechanism for eliminating the noises are explained in the Gaussian filter explanation in section 3. It applies the probability function of noise and employs a mathematics function for smoothening the images. Only the resultant images, as shown in figure 4(b), are obtained based on these functions.



Fig. 5: (a) Noisy Image (b) Filter Image using Gaussian Filter

Figure 5 (a) depicts the original Lena image with noises, and figure 5 (b) depicts the filter images applying the Gaussian Filter. Figure 5 (b) is the resultant images after removing the noises from the original image. The Gaussian functions working mechanism for eliminating the noises are explained in the Gaussian filter explanation in section 3. It applies the probability function of noise and employs a mathematics function for smoothening the images. Only the resultant images, as shown in figure 5(b), are obtained based on these functions.



Fig. 6: (a) Noisy Image (b) Filter Image using Median Filter

Figure 6 (a) depicts the original Barbara image with noises, and figure 6 (b) depicts the filter images applying Median Filter. The median filter is a non-linear filter that filters the noises pixel by pixel, and each adjacent pixel is noted as a window. The working principle of the median filter is described in section 4. The results images are obtained based on the function, as shown in figure 6(b).



Fig. 7 : (a) Noisy Image (b) Filter Image using Median Filter

Figure 7 (a) depicts the original Lena image with noises, and figure 7 (b) depicts the filter images applying Median Filter. The median filter is also a non-linear filter that filters the noises pixel by pixel, and each adjacent pixel is noted as the window. The working principle of the median filter is described in section 4. The results images are obtained based on the function, as shown in figure 7(b).



Fig. 8: (a) Noisy Image (b) Filter Image using Median Filter

Figure 8 (a) depicts the original Lena image with noises, and figure 8 (b) depicts the filter images applying Median Filter. The median filter is a non-linear filter that filters the noises pixel by pixel, and each adjacent pixel is noted as the window. The working principle of the median filter is described in section 4. The results images are obtained based on the function, as shown in figure 8(b).



Fig. 9: (a) Noisy Image (b) Filter Image using Wiener Filter

Figure 9 (a) depicts the original Barabara image with noises, and figure 9 (b) depicts the filter images applying Wiener Filter. Figure 9 (b) is the resultant images after removing the noises from the original image. Wiener filter is an optimally stationary linear filter that applies additive noise and image blurring processes, including auto-correlation and cross-correlation functions.

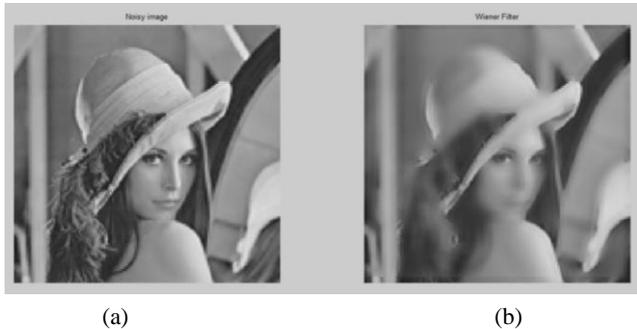


Fig. 10: (a) Noisy Image (b) Filter Image using Wiener Filter

Figure 10 (a) depicts the original Barabara image with noises, and figure 10 (b) depicts the filter images applying Wiener Filter. Figure 10 (b) is the resultant images after removing the noises from the original image. Wiener filter is an optimally stationary linear filter that applies additive noise and image blurring processes, including auto-correlation and cross-correlation functions.



Fig. 11: (a) Noisy Image (b) Filter Image using Wiener Filter

Figure 11 (a) depicts the original Barabara image with noises, and figure 11 (b) depicts the filter images applying Wiener Filter. Figure 11 (b) is the resultant images after removing the noises from the original image. Wiener filter is an optimally stationary linear filter that applies additive noise and image blurring processes, including auto-correlation and cross-correlation functions.



Fig. 12: (a) Noisy Image (b) Filter Image using NLM Filter

Figure 12 (a) depicts the original Barabara image with noises, and figure 12 (b) depicts the filter images applying NLM Filter. NLM filter applies the weighted average of the neighboring pixels and uses high-extent redundancy of the normal image. As a result, fine details like the structure and texture of the filtered image can be preserved. Based on this principle, the computed noisy image results from a noise-free or noise-less pixel image, as shown in fig 12 (b).

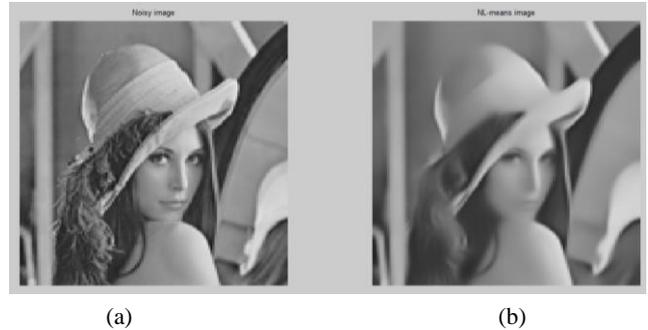


Fig. 13: (a) Noisy Image (b) Filter Image using NLM Filter

Figure 13 (a) depicts the original Lena image with noises, and figure 13 (b) depicts the filter images applying NLM Filter. NLM filter applies the weighted average of the neighboring pixels and uses high-extent redundancy of the normal image. As a result, fine details like the structure and texture of the filtered image can be preserved. Based on this principle, the computed noisy image results from a noise-free or noise-less pixel image, as shown in fig 13 (b).



Fig. 14: (a) Noisy Image (b) Filter Image using NLM Filter

Figure 14 (a) depicts the original Peppers image with noises, and figure 14 (b) depicts the filter images applying NLM Filter. NLM filter applies the weighted average of the neighboring pixels and uses high-extent redundancy of the normal image. As a result, fine details like the structure and texture of the filtered image can be preserved. Based on this principle, the computed noisy image results from a noise-free or noise-less pixel image, as shown in fig 14 (b).

7.1 Confusion Matrix

It results from a quick summary of a classification problem. There are two kinds of predictions such as correct prediction and incorrect prediction. Each class's correct and incorrect predictions are determined using the broken down and count values. Additionally, along with the error, it also describes the type of error committed by the individual classifier.

	Class 1 Predicted	Class 2 Predicted
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

Fig 15: Class Prediction

Definition of the Terms:

- Positive (P): Observation is positive
- Negative (N): Observation is not positive
- True Positive (TP): Observation is positive and is predicted to be positive.
- False Negative (FN): Observation is positive but is predicted negative.
- True Negative (TN): Observation is negative and is predicted to be negative.
- False Positive (FP): Observation is negative but is predicted positive.

Classification Rate/Accuracy: Classification Rate or Accuracy can be expressed by:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

7.2 Recall

The Recall calculates the total correctly classified examples divided by the total positive examples. The high recall values, such as the least number of FN, denotes the correctly recognized example. The recall value is calculated using the below expression;

$$Recall = \frac{TP}{TP+FN}$$

7.3 Precision

The total successfully classified positive examples with total predicted positive examples yields the precision value. The positive indicates the high precision (a small number of FP). Precision value is calculated using the below expression;

$$Precision = \frac{TP}{TP+FP}$$

7.4 F-measure

Precision and Recall are examined to determine the F-Measure. F-measure uses Harmonic Mean rather than Arithmetic Mean because it significantly excludes extreme values. F-Measure will constantly be lower than the Precision or Recall.

Table 1: Proposed algorithm obtained result

Correctly Classified Instances	675	96.5665 %
Incorrectly Classified Instances	24	3.4335 %
Kappa statistic		0.9243
Mean absolute error		0.0614
Root mean squared error		0.1659
Relative absolute error		13.582 %
Root relative squared error		34.9059 %
Total Number of Instances		699

Table 1 shows the accuracy result obtained from the resultant input images. The total instances calculated is 699, of which 97% are correctly classified, and 3% of instances are classified incorrectly. The various error parameters have described the table 1 above, such as Mean absolute error of 0.9243, Root mean squared error of 0.1659, etc.

Table 2 Detailed Accuracy by Class

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC
0.969	0.041	0.978	0.969	0.974	0.924	0.989	0.994
0.959	0.031	0.943	0.959	0.951	0.924	0.989	0.972
0.966	0.038	0.966	0.966	0.966	0.924	0.989	0.986

Table 2 shows the proposed mechanism's exact accuracy values for each image, such as Barbara, Lena & Pepper image. The evaluation metric for evaluating a classifier's accuracy are TP Rate, FP Rate, Precision, Recall, F-Measure, MCC, ROC Area, and PRC. Table 2 shows the values of each parameter obtained from the three input images.

Table 3: Existing and Proposed Filters

Image	Gaussian Filtering			Median Filtering			Wiener Filtering			NLM Filtering		
	MSE	PSNR	MAE	MSE	PSNR	MAE	MSE	PSNR	MAE	MSE	PSNR	MAE
Barbara	59.05	30.45	18.11	47.06	31.44	16.31	80.72	29.09	17.09	25.70	54.19	27.05
Lena	42.30	31.90	15.37	33.78	32.88	14.15	74.13	29.46	15.34	27.36	53.91	23.71
Peppers	51.31	31.06	16.55	35.23	32.69	14.55	82.16	29.01	16.60	26.19	53.94	35.10

Table 3 shows the comparison between the existing classifiers and the proposed classifier concerning the parameters such as MSE (Mean Squared Error), PSNR (peak signal-to-noise ratio), and MAE (Mean Absolute Error). The MSE, PSNR, and MAE are the error parameters considered to declare the image compression quality. The comparison work is conducted with three input images: Barbara, Lena, and Peppers. The NLM filtering achieved better results with minimum error values on all parameters. The detailed numerical values obtained by each algorithm concerning MSE, PSNR, and MAE are shown in table 3.

#	Name	Type	Activations	Learnable Properties	Number of Layers	States
1	imginput	Image Input	227(S) + 227(S) + 3(C) + 1(B)	-	0	-
2	conv	Convolution	227(S) + 227(S) + 32(C) + 1(B)	Wtgs: 3 x 3 x 3 - Bias: 1 x 1 x 32	896	-
3	relu	ReLU	227(S) + 227(S) + 32(C) + 1(B)	-	0	-
4	maxpool	Max Pooling	227(S) + 227(S) + 32(C) + 1(B)	-	0	-
5	softmax	Softmax	227(S) + 227(S) + 32(C) + 1(B)	-	0	-
6	fc	Fully Connected	1(S) + 1(S) + 18(C) + 1(B)	Wtgs: 18 x 1648 - Bias: 18 x 1	1648298	-
7	classoutput	Classification Output	1(S) + 1(S) + 18(C) + 1(B)	-	0	-

Fig 16: Analysis for training in deep network designer

Figure 16 shows the observation result obtained through Matlab. The results are obtained on the input dataset based on the proposed algorithm. It details input type, activations, learnable properties, the total number of learnable properties, and its resultant states.

=== Confusion Matrix ===

a b <-- classified as

444 14 |

10 231 |

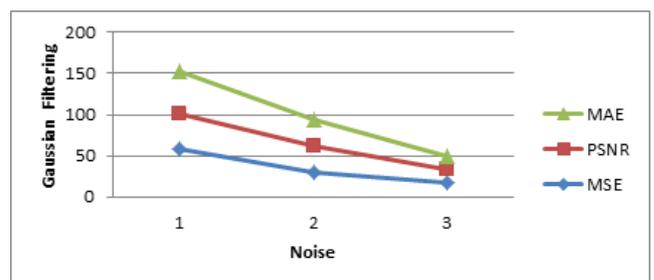


Fig. 17: Noise Vs. Gaussian Filtering

Figure 17 shows the performance achieved by the SPRINT algorithm concerning the parameters such as PSNR (peak signal-to-noise ratio), MAE (Mean Absolute Error), and MSE (Mean Squared Error). The MSE, PSNR, and MAE are the error parameters considered to declare the image compression quality. The x-axis describes the total input of noisy images, and the y-axis represents the obtained value from gaussian filtering. These graphs are plotted based on the obtained values from gaussian filtering, as in table 3.

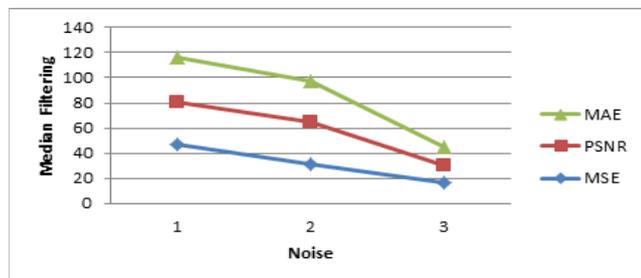


Fig. 18: Noise Vs. Median Filtering

Figure 18 shows the performance achieved by the SPRINT algorithm concerning the parameters such as MSE, PSNR, and MAE. These are the error parameters considered to declare the image compression quality. The x-axis describes the total input of noisy images, and the y-axis represents the obtained value from median filtering. These graphs are plotted based on the obtained values from median filtering, as mentioned in table 3.

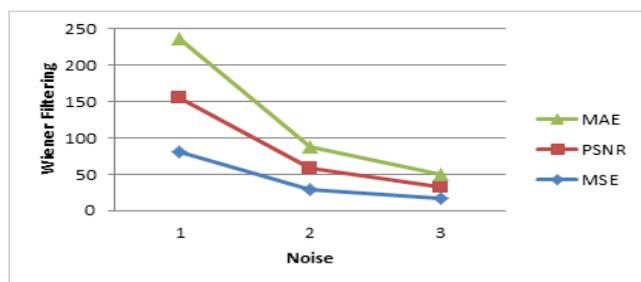


Fig. 19: Noise Vs. Wiener Filtering

Figure 19 shows the performance achieved by the SPRINT algorithm concerning the parameters such as MSE, PSNR, and MAE. These are the error parameters considered to declare the image compression quality. The x-axis describes the total input of noisy images, and the y-axis represents the obtained value from Wiener filtering. These graphs are plotted based on the obtained values from Wiener filtering, as mentioned in table 3.

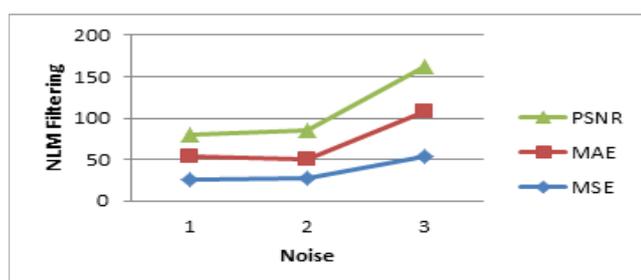


Fig. 20: Noise Vs. NLM Filtering

Figure 20 shows the performance achieved by the SPRINT algorithm concerning the parameters such as MSE, PSNR, and

MAE. These are the error parameters considered to declare the image compression quality. The x-axis describes the total input of noisy images, and the y-axis represents the obtained value from NLM filtering. These graphs are plotted based on the obtained values from NLM filtering, as mentioned in table 3.

Table 3: Accuracy comparison table

SPRINT	97%
CNN	91%
Loss Diff	85%

Table 3 shows the comparison work conducted between the proposed SPRINT algorithm with the existing CNN and LossDiff. The comparison work is carried out with the sample input images with the three algorithms. The obtained results are evaluated in performance metrics such as TP Rate, TN Rate, FP Rate, and FN Rate. From each algorithm obtained value, the accuracy is computed through the below expression;

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

The accuracy performance achieved by the proposed SPRINT algorithm is 97%, whereas CNN achieved 91% of accuracy and LossDiff achieved 85% of accuracy. It is proven that the results gained by the SPRINT algorithm are very effective than others.

8. Conclusion

In this work, a combination of the SPRINT algorithm with the NLM (Non-Local Mean filtering) algorithm is developed to remove unwanted pixels from the images. The essential factors for accuracy are TP Rate, FP Rate, Precision, Recall, F-Measure, MCC, ROC Area, and PRC. The non-local filtering algorithm considers the high-extent redundancy values of the normal images, and the SPRINT algorithm applies the minimum description length principle to enhance accuracy. The NLM filter removes the image noises and smoothens the images. In SPRINT, two data structures, such as attribute table and histogram, are maintained. The attribute table maintains the details about indexing, attribute value, and class identification. Next, the Histograms indicate the attribute node distribution type. The significant feature of the proposed system is it achieves maximum accuracy and enhanced classification accuracy without damaging the details of the original images. In the experimental work, the proposed work is compared with the existing conventional neural network (CNN) and deep learning-based patch label denoising methods (LossDiff). The SPRINT algorithm attains 97% accuracy, and it is more efficient than the others. Additionally, the NLM filtering achieved better results in MSE, PSNR, and MAE; these are the parameters considered to declare the image compression quality. Several image denoising approaches like Gaussian, Median, and Wiener filtering are discussed with their advantages and disadvantages in this paper. From this work, we conclude that our proposed SPRINT approach performs well in the process of noise removal when compared to other filtering approaches.

References

- [1] Pang, T., Zheng, H., Quan, Y., Ji, H.: Recorrputed-to-recorrputed: unsupervised deep learning for image denoising. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2043–2052 (2021). IEEE
- [2] Jo, Y., Chun, S.Y., Choi, J.: Rethinking deep image prior for denoising. In: IEEE Conference on Computer Vision (ICCV), pp. 5087–5096 (2021). IEEE

- [3] Bhat, G., Danelljan, M., Yu, F., Van Gool, L., Timofte, R.: Deep reparametrization of multi-frame super-resolution and denoising. In: IEEE Conference on Computer Vision (ICCV), pp. 2460–2470 (2021). IEEE
- [4] S. Mirjalili, J. Song Dong, A. S. Sadiq, and H. Faris, “Genetic algorithm: theory, literature review, and application in image reconstruction,” in *Nature-Inspired Optimizers. Studies in Computational Intelligence*, S. Mirjalili, D. J. Song, and A. Lewis, Eds., Vol. 811, Springer International Publishing, NY, USA, 2020
- [5] Mou, C., Zhang, J., Wu, Z.: Dynamic attentive graph learning for image restoration. In: IEEE Conference on Computer Vision (ICCV), pp. 4328–4337 (2021). IEEE
- [6] S, R. D., L. . Shyamala, and S. . Saraswathi. “Adaptive Learning Based Whale Optimization and Convolutional Neural Network Algorithm for Distributed Denial of Service Attack Detection in Software Defined Network Environment”. *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 10, no. 6, June 2022, pp. 80-93, doi:10.17762/ijritcc.v10i6.5557.
- [7] Wen, J., Xu, Y., & Liu, H. (2020). Incomplete multiview spectral clustering with adaptive graph learning. *IEEE Transactions on Cybernetics*, 50(4), 1418–1429.
- [8] Deep learning for image denoising: a survey. In *International conference on genetic and evolutionary computing* (pp. 563–572). Springer. Tian, C., Xu, Y., Li, Z., Zuo, W., Fei, L., & Liu, H. (2020).
- [9] Attention-guided cnn for image denoising. *Neural Networks*. Tian, C., Xu, Y., & Zuo, W. (2020). Image denoising using deep cnn with batch renormalization. *Neural Networks*, 121, 461–473
- [10] Gehlot, S., Gupta, A. & Gupta, R. A CNN-based unified framework utilizing projection loss in unison with label noise handling for multiple myeloma cancer diagnosis. *Med. Image Anal.* 72, 102099 (2021).
- [11] Karimi, D., Dou, H., Warfield, S. K. & Gholipour, A. Deep Learning with Noisy Labels: Exploring Techniques and Remedies in Medical Image Analysis. arXiv:191202911 Cs Less Stat (2020).
- [12] Zhang, Y., Wang, C., Wang, X., Zeng, W. & Liu, W. A Simple Baseline for Multi-Object Tracking. arXiv:200401888 Cs (2020).
- [13] S. Ghose, N. Singh and P. Singh, "Image Denoising using Deep Learning: Convolutional Neural Network," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2020, pp. 511-517, doi: 10.1109/Confluence4 7617.2020.9057895.
- [14] Garg, D. K. . (2022). Understanding the Purpose of Object Detection, Models to Detect Objects, Application Use and Benefits. *International Journal on Future Revolution in Computer Science & Communication Engineering*, 8(2), 01–04. <https://doi.org/10.17762/ijfrcsce.v8i2.2066>
- [15] Murtaza Ashraf, Willmer Rafell Quiñones Robles, Mujin Kim, Young Sin Ko, and Mun Yong Yi, "A loss-based patch label denoising method for improving whole-slide image analysis using a convolutional neural network", *Sci Rep.* 2022; 12: 1392. Published online 2022 Jan 26. doi: 10.1038/s41598-022-05001-8
- [16] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, Chia-Wen Lin, "Deep learning on image denoising: An overview", 2020 Elsevier Ltd, <https://doi.org/10.1016/j.neunet.2020.07.025>
- [17] Saeed Izadi, Darren Sutton and Ghassan Hamarneh, "Saeed Izadi, Darren Sutton and Ghassan Hamarneh", Springer Nature 2022
- [18] Cheng, S., Wang, Y., Huang, H., Liu, D., Fan, H., Liu, S.: Nbnnet: Noise basis learning for image denoising with subspace projection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4896–4906 (2021). IEEE
- [19] Hu, X., Ma, R., Liu, Z., Cai, Y., Zhao, X., Zhang, Y., Wang, H.: Pseudo 3d auto-correlation network for real image denoising. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16175–16184 (2021). IEEE
- [20] Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.-H., Shao, L.: Multi-stage progressive image restoration. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14821–14831 (2021). IEEE
- [21] Liu, Y., Qin, Z., Anwar, S., Ji, P., Kim, D., Caldwell, S., Gedeon, T.: Invertible denoising network: A light solution for real noise removal. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13365–13374 (2021). IEEE
- [22] Li, Y., Fu, X., Zha, Z.-J.: Cross-patch graph convolutional network for image denoising. In: IEEE Conference on Computer Vision (ICCV), pp. 4651–4660 (2021). IEEE
- [23] Kiran, M. S., & Yunusova, P. (2022). Tree-Seed Programming for Modelling of Turkey Electricity Energy Demand. *International Journal of Intelligent Systems and Applications in Engineering*, 10(1), 142–152. <https://doi.org/10.18201/ijisae.2022.278>