International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Optimal Denoising System for Medical Images Using Recurrent Neural Network and SVM

Vidhya V[1], Rajavarman V.N.[2], S. Kevin Andrews[3]

1 Research Scholar, Department of Computer Applications, Dr. M. G. R. Educational and Research Institute, Chennai, Tamilnadu, India. Email: vidhyaaug7@gmail.com
2 Professor, Department of Computer Science and Engineering, Dr. M. G. R. Educational and Research Institute, Chennai, Tamilnadu, India. Email: rajavarman.vn@drmgrdu.ac.in
3 Professor, Department of Computer Science and Engineering, Dr. M. G. R. Educational and Research Institute, Chennai, Tamilnadu, India. Email: kevin.mca@drmgrdu.ac.in

Abstract

Image denoising serves as a crucial preprocessing step in the realm of medical image analysis, with the primary objective of faithfully reconstructing the original image from its noisy counterpart. This process is essential for maintaining the integrity of vital details, such as edges and textures, within the denoised image. Innovatively addressing this challenge, our proposed system introduces a novel approach that seamlessly integrates Recurrent Neural Network (RNN) and Support Vector Machine (SVM). This powerful combination is adept at efficiently eliminating various types of noise, including gaussian, white noise, salt and pepper noise, and speckle noise, from intricate lung CT images. To enhance both learning accuracy and training efficiency, we have incorporated batch normalization in conjunction with residual learning. Notably, batch normalization is executed with the support of Long Short-Term Memory (LSTM). This strategic integration aids in the gradual separation of image structure from the noisy observations, a pivotal aspect in achieving optimal denoising outcomes. This approach not only enhances the accuracy of denoising but also contributes to reducing the overall training time, making it a valuable advancement in medical image preprocessing.

Keywords: Recurrent Neural Networks, Support vector Machine, image denoising, Long Short-Term Memory

1. Introduction

The intersection of medical science and technology relies heavily on the effective utilization of images to visually represent the inherent components of the body. These images serve a crucial purpose in laboratory testing analysis, revealing structures hidden beneath the skin and bone. Furthermore, they play a decisive role in making informed decisions about the treatment of diseases. In the examination of patients' ailments, medical imaging assumes a pivotal position, proving itself indispensable in recent times for disease diagnosis. Over the past couple of decades, various medical imaging modalities have been designed to cater to diverse applications, as noted by Manjón et al. (2008) [1] and Mitiche et al. (2013) [2]. These modalities facilitate the acquisition of images of anatomical structures within the body without the need for invasive procedures. Notable modalities include X-rays, Computed Tomography (CT), nuclear imaging, Magnetic Resonance Imaging (MRI), and Ultrasound (US), each serving a specific purpose in diagnosing various diseases.

However, a significant challenge faced by these imaging modalities is the presence of noise. Noise manifests as variations in the intensity values of pixels within the image, deviating from the true pixel values. In essence, noise represents undesirable effects in medical images, posing a hurdle to accurate interpretation. The removal of noise becomes imperative in medical imaging applications, aiming to enhance and simultaneously preserve intricate information within the images to the highest possible extent. This process ensures that the diagnostic images remain reliable and contribute to informed medical decisions.

1.1 Type of Noises

Throughout the stages of image processing, including acquisition, transmission, reception, storage, and retrieval, diverse forms of noise can introduce distortions to an image. The impact of noise on the image varies depending on the nature of the distortion it introduces.

1.1.1 Gaussian Noise

Gaussian noise is characterized by its uniform distribution across the signal. This means that in a noisy image, each pixel results from the summation of the actual pixel values and a randomly distributed Gaussian noise component, as outlined by Lone et al. [3] in 2018. Notably, the intensity of the pixel value at any given point does not influence the noise, emphasizing its evenly distributed nature. A specific instance of Gaussian noise is white Gaussian noise, where values at different times follow a similar distribution and demonstrate statistical independence. The term "white" in this context is derived from white light, symbolizing the even distribution of frequencies. The primary sources of Gaussian noise in digital images are often encountered during the acquisition process. For example, sensor noise may arise due to factors such as inadequate illumination or high temperature during image capture. Additionally, transmission processes can introduce Gaussian noise, further emphasizing the need for robust noise reduction techniques in image processing to enhance image quality and preserve the accuracy of the underlying information.

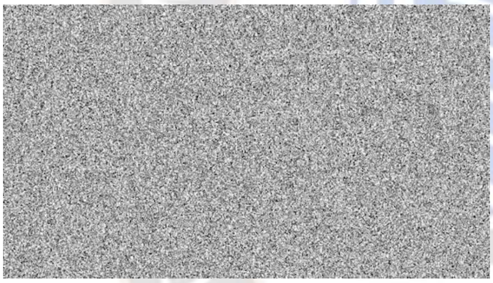

Figure 1 Gaussian noise

1.1.2 Salt and pepper noise

Salt and pepper noise, categorized as impulse noise, manifests as spikes in intensity, as indicated by Kaur and Kaur [4] in 2014. This type of noise emerges due to errors in data transmission, leading to abrupt variations in the image signal. In images affected by salt and pepper noise, noisy pixels exhibit either the maximum or minimum values within the dynamic range.In an 8-bit image, it is commonly observed that pepper noise is represented by a pixel value of 0, while salt noise is represented by a pixel value of 255. This distinctive noise pattern often stems from malfunctions in pixel elements within camera sensors, defective memory locations, or timing errors during the digitization process. Figure 2 visually represents the characteristic appearance of salt and pepper noise in an image.

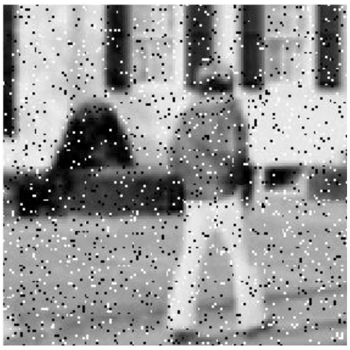

Figure 2 Salt and Pepper Noise

1.1.3 Speckle noise

Speckle noise is a granular type of noise inherent in active radar and Synthetic Aperture Radar (SAR) images, leading to a degradation in image quality. This form of noise is particularly prevalent in certain biomedical applications such as Ultrasonic Imaging and emergency applications like Synthetic Aperture Radar (SAR) imaging. Notably, speckle noise is characterized by higher intensity when the magnitude of image pixels is high, indicating its dependence on the signal, as highlighted by Verma and Ali [5] in 2013.

Speckle noise is multiplicative in nature, stemming from the initial transmission of a signal to an object and the subsequent recording of the reflected signal. During signal transmission, additive noise in the channel may corrupt the signal. Additionally, variations in the reflectance of the object's surface contribute to changes in the magnitude of the reflected signal, impacting the nature and magnitude of the noise. The multiplicative nature of speckle noise results in higher noise magnitude when the signal magnitude is high.

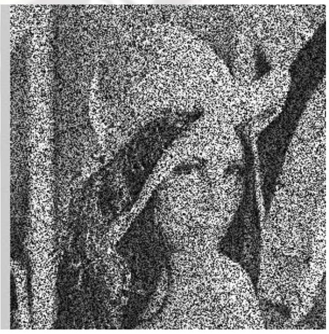

Figure 3 Speckle Noise

1.2 Source of Noises

In any physical measurement, it's common for the measured quantity to be affected by various types of noise. The sources and nature of this noise depend on the specific characteristics of the physical measurement being conducted. Noise often originates from sources that are not identical to the one being measured, and sometimes it arises due to the measurement process itself. Drawing parallels with images, read-out noise in digital cameras exemplifies noise arising from a source different from the intended measurement, while speckle noise in Synthetic Aperture Radar (SAR) images exemplifies noise caused by the measurement process. Noise can also stem from the mathematical interpretation of a signal, such as in image deconvolution or image compression. In many cases, measurements are distorted by multiple sources of noise, making it challenging to fully characterize each one. In an ideal scenario, efforts are made to eliminate noise by refining the image acquisition process. However, when such manipulations are not feasible, denoising algorithms become essential.

The characteristics of noise are intricately linked to the image acquisition process, encompassing both digital and analog cameras of various types, whether for visible or infrared light. Additionally, noise affects radar imagery like Synthetic Aperture Radar (SAR), as well as medical imaging modalities such as Magnetic Resonance Imaging (MRI), Computer Tomography (CT), Positron Emission Tomography (PET), ultra-sonography, electron microscopy, among others. Recognizing and addressing noise in these diverse contexts is crucial for obtaining accurate and reliable measurements and images.

1.3 Overview of Medical Image Noising

The dynamic landscape of biomedical imaging presents both significant challenges and opportunities for image processing tasks, as noted by Mredhula&Dorairangasamy[6] in 2013, Manjón et al. [7] in 2010, and Omer et al. in 2018. Accurately modeling various biomedical imaging modalities, such as X-ray, ultrasound, magnetic resonance imaging (MRI), and computed tomography (CT), proves challenging due to their intricate nature and evolving technologies. In ultrasound imaging, speckle noise exhibits non-static characteristics, varying based on ultrasound attenuation and the presence of sub-resolution scatterers in the tissue, as explained by Kaur et al. [8] in 2007. For CT, factors like tube current-time product and slice thickness influence image noise. Meanwhile, in MR images, the Signal-to-Noise Ratio (SNR) depends on a multitude of variables, including pulse sequence, radiofrequency coil(s), magnetic field strength, and acquisition parameters.

Understanding the physical and image characteristics specific to each medical imaging modality becomes crucial for effective image denoising. Realistic applications often demand image enhancement and reconstruction. Corruption due to Additive White Gaussian Noise (AWGN) may occur in images extracted in poor quality conditions, those in noisy environments, or due to inherent noise in communication channels. While linear filtering and smoothing operations are common for image reconstruction due to their simplicity, they assume static image signals generated through linear systems, limiting their overall efficiency. Real-world medical images, on the other hand, often possess non-static statistical features, created through nonlinear systems where intensity distribution is the product of object reflectance and illumination distribution.To address this complexity, adaptive and nonlinear image restoration techniques consider local statistical features, achieving enhanced reconstruction while preserving high-frequency features like edges. These advanced techniques prove invaluable in coping with the diverse and dynamic nature of biomedical imaging data.

1.4 Supervised Learning Models

Supervised learning is a machine learning paradigm focused on learning a function that can map input data to corresponding outputs based on labeled training examples. This approach involves inferring a function from a set of training data, where each example is a pair comprising an input object (usually a vector) and its corresponding desired output value, often referred to as the supervisory signal. The algorithm learns patterns and relationships within the labeled data to make predictions or classify new, unseen examples.

In supervised learning, the algorithm processes the labeled training data to create an inferred function. This function can then be applied to map inputs to outputs for previously unseen examples. The goal is to generalize the learning from the training data to accurately predict outcomes for new, similar instances.

Support Vector Machine (SVM) is a specific type of supervised learning method widely used for classification and regression analysis. SVM aims to find the optimal hyperplane that best separates different classes in the input space, making it a powerful tool for tasks such as image classification, speech recognition, and various other pattern recognition applications. SVMs can handle both linear and non-linear relationships within the data, offering flexibility in addressing complex problems in supervised learning scenarios.

***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

1.4.1 Introduction to Support Vector Machines (SVM)

In machine learning, support-vector machines (SVMs), also known as support vector networks, are supervised learning models equipped with learning algorithms designed for classification and regression analysis. These models analyze labeled data to build a robust framework for categorizing new examples into predefined classes or predicting numeric values. When given a set of training examples, each labeled as belonging to one of two categories, an SVM training algorithm constructs a model that effectively separates the examples into distinct categories. SVMs operate as non-probabilistic binary linear classifiers, meaning they classify instances into two categories without providing probabilistic estimates. However, methods like Platt scaling exist to adapt SVMs for probabilistic classification scenarios, as highlighted by Meyer & Wien in 2015 and Harris [9] in 2015.The SVM model represents training examples as points in a multidimensional space, carefully mapped to ensure a clear gap between points of different categories. The objective is to create a gap that is as wide as possible, enhancing the model's ability to discriminate between classes. Once the model is established, new examples are mapped into the same space, and their category assignment is predicted based on which side of the gap they fall. SVMs have proven to be versatile and effective tools in various machine learning applications, offering robust classification capabilities in both linear and non-linear scenarios.

1.4.3Firefly Algorithm (FA)

The Firefly Algorithm is a nature-inspired, metaheuristic, and swarm intelligence algorithm that draws inspiration from the characteristic behaviors of fireflies. It leverages the concept of the flashing lights of fireflies to optimize solutions in a search space. Here are some key characteristics and principles of the Firefly Algorithm:

1. Nature-Inspired Optimization:
   - The algorithm is designed to mimic the behavior of fireflies in nature, where the flashing lights serve as a means of communication and attraction.
2. Swarm Intelligence:
   - Similar to other swarm intelligence algorithms, the Firefly Algorithm capitalizes on the collective behavior of a swarm of individuals (fireflies) to collectively explore and optimize a solution space.
3. Light Intensity and Attraction:
   - The intensity of the light emitted by a firefly is directly linked to its attractiveness.
   - Fireflies are attracted to each other, and the one emitting brighter light attracts others more strongly than those emitting dimmer light.
4. Optimization Principle:
   - The movement of fireflies is guided by the principle of moving toward brighter and more attractive locations in the search space.
   - Brightness is often associated with the quality of a solution in the context of optimization problems.
5. Random Movement:
   - If a firefly cannot find a neighboring firefly with higher brightness, it moves randomly to explore other regions of the search space.
6. Objective Function and Brightness:
   - In the mathematical model of the algorithm, the brightness of a firefly is typically determined by the objective function relevant to the optimization problem being solved.

The Firefly Algorithm has been applied to various optimization problems, including mathematical functions, engineering design, and other domains where finding optimal solutions is crucial. Its simplicity, efficiency, and ability to handle optimization tasks in complex solution spaces make it a popular choice in the field of metaheuristic algorithms.

2. Hybrid Recurrent Neural Networks with Support Vector Machines (HRNN-SVM)

In this research propose hybrid recurrent neural networks with support vector machines to improve the discrimination ability of RNNs. Given an input image sequence (lung CT images) X 1:T={x_1 , … . x_T}, the simple RNNs compute the hidden vector {h_1 , … , h_T} by iterating the following equations from t = 1,…, T,

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (2.1)$$

where W is the weight matrix, e.g.,W_{xh} is the input-hidden weight matrix, b is the bias vector, e.g., b_h is the hidden bias vector, and H(.) is the recurrent hidden layer function. Denote the corresponding state labels as s = {s_1, … ,}, the state-label posterior for pixel t is computed by the softmax

$$P(s_t|X_{1:t}) = \frac{\exp{(w_{st}^T h_t + b_{st})}}{\sum_{s_t=1}^{N} \exp{(w_{st}^T h_t + b_{st})}} \quad (2.2)$$

Where, N is the total number of labels, w_{st} is the weight vector connecting the hidden layer to the output state s_t. The proposed system replaces the softmax layer in RNN with

Support Vector Machines (SVMs). The parameters of RNNs and SVMs are jointly learned using sequence-level max margin criteria, instead of cross entropy. For multiclass SVM, the classification function is

$$arg_{s}^{max}\, w_s^T \varphi(x_t) \qquad (2.3)$$

Where, ( ) is the predefined pixels of SVM and ws is the weight parameter for class/state s. If RNNs are used to derive the pixel, e.g., ( ) ≜h$_t$, decoding of SVMs (Guenther& Schonlau 2016) and RNNs are the same. This inspires us to replace the softmax layer in RNNs with SVM.The RNNs can be trained using pixel-level cross entropy, connection is t temporal classification. The first step is to estimate the parameters of SVM in the last layer using the quadratic programming. The second step is to update the parameters of RNN in all previous layers using sub gradient approaches (Mehr et al. 2019) [10] ; (Mohammadi et al. 2015) [11].

2.1 Pixel -level training using Recurrent SVM

In the pixel-level max-margin training, given training inputs and state labels, , $\{(x_t\ ,\ s_t)\}$ t=1 T , let $\varphi(x_t\ )$ ≜h$_t$ as the pixel space derived from RNN recurrent states, the parameters of the last layer are first estimated using the SVM training algorithm,

$$\min_{w_s,\xi_t} \frac{1}{2} \sum_{s=1}^{N} ||w_s||_2^2 + C \sum_{t=1}^{T} \xi_t^2 \qquad (2.4)$$

s.t. for every training pixel t = 1,…., T

for every competing states $\bar{s} \in \{1, \dots, N\}\ t$:

where ξt> 0 is the slack variable which penalizes the data points that violate the margin requirement. The equation (2.4) basically says that, the score of the correct state label, $w_{st}Th$, has to be greater than the scores of any other states, $w_{st}T\ h_t$, by a margin. Substituting ξt from the constraints into the objective function, equation (2.4) can be reformulated as minimizing

$$F_{frm}(w_s, W_*) = \frac{1}{2} \sum_{s=1}^{N} ||w_s||_2^2 + C \sum_{t=1}^{T} \Big[1 - \xi_{tw_s^T h_t}^2 + {}^{max}_{s_t} w_{s_t}^T h_t \geq \Big]^2 \qquad (2.5)$$

where h$_t$ depends on all the weights W_* in previous layers. [x]+ = max(0; x) is a hinge function. Note the maximum of a set of linear functions is convex, thus equation (4.5) is convex w.r.t.ws. The first step of recurrent SVM training is to optimize w$_s$ by minimizing (2.5). The second step of recurrent SVM training is to optimize the parameters in previous layers. These parameters can be updated by back propagating the gradients from the top layer.

$$\frac{\partial F_{frm}}{\partial W_{ih}} = \sum_{t=1}^{T} \left(\frac{\partial F_{frm}}{\partial h_t} \frac{\partial h_t}{\partial W_{ih}} \right) \qquad (2.6)$$

The key here is to compute the derivative of F$_{frm}$w.r.t. the ht. However, equation (2.5) is not differentiable because of the max(.). To handle this, the sub gradient method is applied.

2.2 Long Short-Term Memory with Batch Normalization

Certainly, the Long Short-Term Memory (LSTM) is a widely used recurrent neural network (RNN) architecture that effectively addresses the vanishing gradient problem, a common issue in vanilla RNNs. LSTMs utilize gating functions to control information flow and maintain memory over long sequences. Here's an overview of the computation at each time step t in an LSTM:

$$i_t = sigmoid(W_{hi}h_{t-1} + W_x x_t)$$

$$f_t = sigmoide(W_{hf}h_{t-1} + W_{hf}x_t)$$

$$c_t = f_t \odot c_{t-1} + i_t\, tanh\odot\, (W_{hc}h_{t-1} + W_{xc}x_t)$$

$$o_t = sigmoide(W_{ho}h_{t-1} + W_{hx}x_t + W_{co}c_t)$$

$$h_t = o_t \odot tanh(\,c_t)$$

Where sigmoid(·) is the logistic sigmoid function, tan h is the hyperbolic tangent function, W_h are the recurrent weight matrices and W_x are the input to hidden t weight matrices. , and o$_t$ are respectively the input, forget and output gates, and c_t is the cell (Voigtlaender et al. 2016) [12].

2.3 Batch size selection using FireFly Algorithm

In this proposed research work, an effective batch size selection is performed by using firefly algorithm. From equation 2.8, an analogous way to apply batch normalization to an RNN would be as follows:

$$h_t = \phi(BN(W_h h_{t-1} + W_x x_t)) \qquad (2.8)$$

However, in this experiments, when batch normalization was applied in this fashion, it didn't help the training procedure. Instead we propose to apply batch normalization only to the input-to-hidden transition ($W_x x_t$), i.e. as follows:

$$h_t = \phi(W_h h_{t-1} + BN(W_x x_t)) \qquad (2.9)$$

This idea is similar to the way dropout can be applied to RNNs batch normalization is applied only on the vertical

Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

connections (i.e. from one layer to another) and not on the horizontal connections (i.e. within the recurrent layer). We use the same principle for LSTMs: batch normalization is only applied after multiplication with the input-to-hidden weight matricesWx. Based on the above procedures, the images are normalized.

Batch size selection using FA

In this work, the presented methods implement a FA algorithm to ideally identify the batch size. Firefly algorithm is one among the effective optimization techniques (Laurent et al. 2016) [13] ;(Pereira& Yang 2016)[14]. The FA is a metaheuristic, nature inspired, optimization algorithm which is based on the social flashing behaviour of fireflies, or lighting bugs. The FA uses the following three rules:

- A firefly is attracted to other fireflies regardless of their sex, because all fireflies are unisex.
- Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less− bright one will move towards the brighter one. Both attractiveness and brightness are decreasing as the distance between the two fireflies increases. If no one is brighter than a particular firefly, then it moves randomly.
- The brightness or light intensity of a firefly is determined by the objective function of the optimization problem.

In this work, number of pixels is considered as fireflies. The light intensity (denoising quality) thus attractiveness is inversely proportional with the particular distance r from the light source. Thus the light and attractiveness is decrease as the distance increase.

$$I(r) = I_o e^{-\gamma r^2} \tag{2.10}$$

Where, γ -light absorption coefficient

Attractiveness is proportionally to the light intensity seen by another fireflies, thus attractiveness is

$$\beta = \beta_0 e^{-\gamma r2} \tag{2.11}$$

Where, β_0 - attraction at r=0

The distance between two fireflies can define using Cartesian distance $r_{i,j}$ is the distance amid any two fireflies i and j, that are at positions x_i and $x_j$correspondingly. The Cartesian distance is provided by the equation

$$r_{ij} = |x_i - x_j| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2} \tag{2.12}$$

Here , is known as the kth element of the spatial coordinate xi of the firefly i and d is known as the amount of dimensions. The movement of a firefly i in the direction of more another (brighter) firefly j is provided by [15]

$$x_i = x_i + \beta_0 e^{-\gamma r_i^2 j} (x_j - x) + \alpha \varepsilon_i \tag{2.13}$$

For the attractiveness, second component is utilized and for randomization, third component is utilized with α being the randomization parameter, and ε_i is known as a vector of random numbers being derived from a Gaussian distribution or else uniform distribution interval [0, 1].

3. Experimental Results

The designed research work was implemented using MATLAB. The experimental work is performed on SIMBA dataset. Based on the white noise, salt and pepper noise, gaussian and speckle noises the experimental results are presented. The performance of the proposed HRNN-SVM scheme is compared with the existing image denoising schemes such as CNN, DnCNN and RNN in terms of Peak to Signal Noise Ratio (PSNR), Mean Squared Error (MSE) and accuracy.

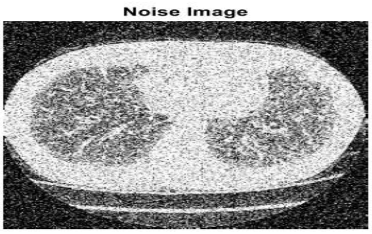

Figure 4. Input image with white noise

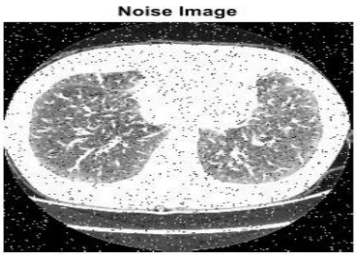

Figure 5. Input image with salt and pepper noise

***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

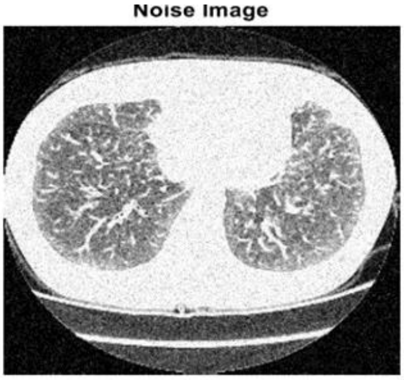

Figure 6. Input image with Gaussian noise

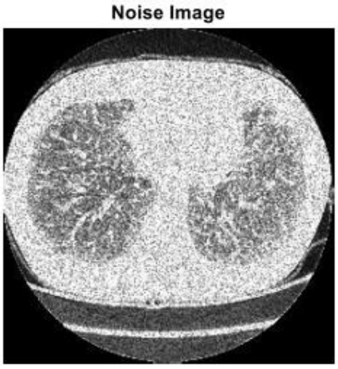

Figure 7. Input image with speckle noise

The various types of noises such as white noise, gaussian, salt and pepper noise, and speckle noise are corrupted with input images. The image with noises is shown in figure 4 to 7.

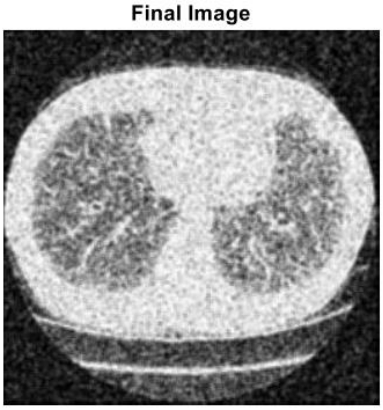

Figure 8. Denoised image for white noise

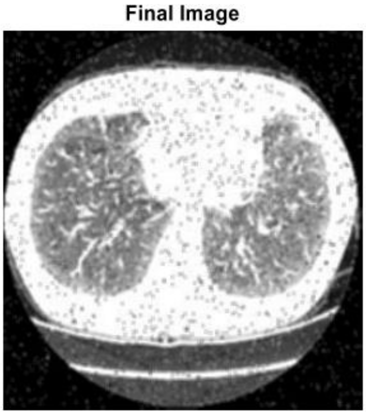

Figure 9. Denoised image for salt and pepper noises

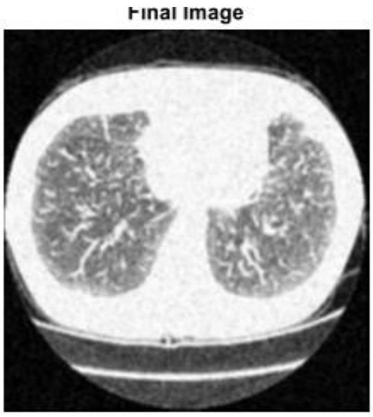

Figure 10. Denoised image for Gaussian noise

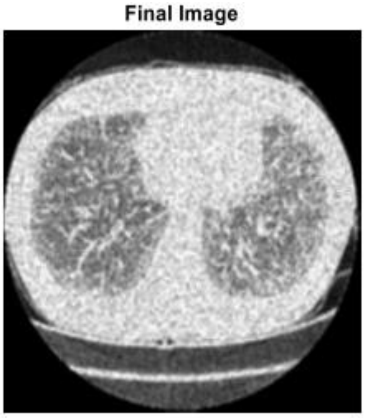

Figure 11. Denoised image for speckle noise

Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

The denoised image output is shown in figure 8 to 11. With the help of Recurrent Neural Network (RNN) with Support Vector Machine (SVM) the white noise, salt and pepper noise, Gaussian and speckle noise are efficiently removed from lung CT images. In order to improve the denoising quality the last layer of RNN is replaced with SVM classifier.

The CT lung images are denoised and the PSNR evaluated. The PSNR value is represented in Table 1.

Table 1 PSNR comparison

No of images	PSNR comparison			
	CNN	Dn-CNN	RNN	HRNN-SVM
Image 1	38.56	39.63	44.38	46.25
Image 2	36.85	38.56	43.63	45.96
Image 3	37.25	39.87	41.96	43.52
Image 4	39.02	40.10	42.21	44.32
Image 5	35.12	36.65	37.65	40.54
Image 6	35.85	37.12	39.33	41.21
Image 7	34.51	36.12	38.54	40.24
Image 8	34.02	35.95	37.41	40.00

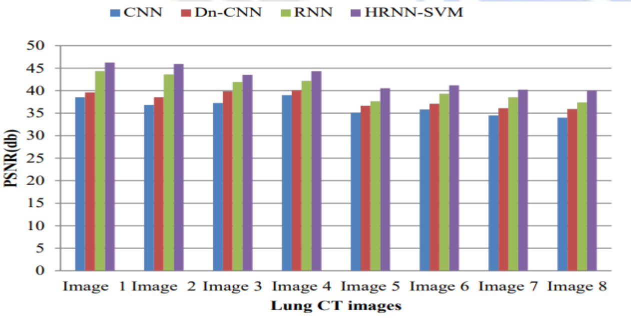

Figure 12. Experimental results of PSNR comparison for lung CT images

The PSNR of the proposed and existing methods are compared in figure 12. In x-axis number of images is taken and PSNR is taken as y-axis. In this proposed research work, softmax layer in RNN is replaced by using Support Vector Machines. Here the parameters of RNNs and SVMs are jointly learned. In addition, the firefly algorithm is utilized to select an optimal batch size selection. This hybrid framework achieves the higher quality image as the output. The result shows that the proposed HRNN-SVM achieves 45.96 db when the other method such as CNN, DnCNN, and RNN achieves 36.85db, 38.56db and 43.63 db respectively for image 2.

The MSE performance of the proposed HRNN-SVM scheme is compared with the other methods. The values are tabulated in Table 2.

Table 2 MSE comparison

No of images	MSE comparison			
	CNN	Dn-CNN	RNN	HRNN-SVM
Image 1	9.8	9.2	8.8	7
Image 2	9.6	9.4	8.9	7.5
Image 3	9.5	9	8.5	8
Image 4	9.4	8.9	8.2	7.9
Image 5	9.3	8.6	7.8	7.5
Image 6	9.1	8.3	7.3	7.0
Image 7	9.0	8.2	7.0	6.5
Image 8	8.9	8.0	6.8	6.1

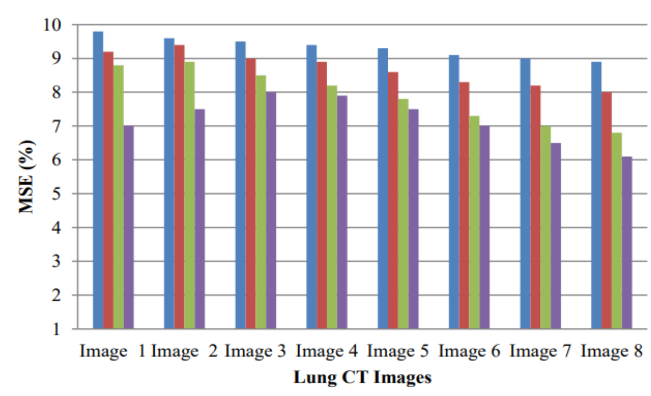

Figure 12.Experimental results of MSE comparison for lung CT images

Fig 12 illustrates the MSE comparison of the proposed HRNN-SVM and existing approaches. In x-axis, number of images is taken and MSE is taken as y-axis. From this figure, it can be noticed that the proposed HRNN-SVM attains 7.5 % of MSE whereas other methods such as CNN, DnCNN and RNN achieves 9.6%, 9.4% and 8.9% respectively for image 2.

The CT lung images are denoised and the accuracy is evaluated. The accuracy value is represented in Table 3.

Table 3 Accuracy comparison

No of images	Accuracy comparison			
	CNN	Dn-CNN	RNN	HRNN-SVM
Image 1	87	88	93	93.5
Image 2	88	89	94	94.2
Image 3	89	90	95	95.3
Image 4	90	92	96	96.2
Image 5	92	93	97.5	98
Image 6	93	94	98	98.4
Image 7	94	95	98.2	98.9
Image 8	95	96	98.5	99.3

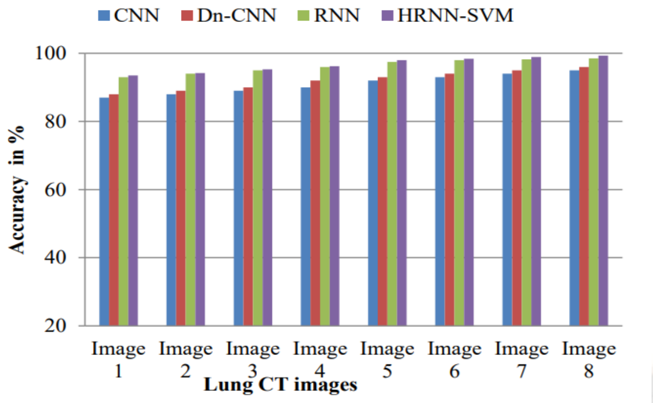

Figure 13. Experimental results of accuracy comparison for lung CT images

The proposed HRNN-SVM based denoising is compared with the existing CNN, DnCNN and RNN based image denoising schemes. In this research work, denoising image accuracy is improved with the help of HRNN-SVM based approach. From figure 13, it can be concluded that the proposed system achieves 97% of accuracy when the other methods such CNN, DnCNN and RNN achieves 88%, 89%, and 94% respectively for image 2.

4. Conclusion

In this paper, a new image denoising and classification technique using Recurrent Neural Networks (RNN) with Support Vector Machines (SVMs) is proposed for efficiently removing noise from lung CT images. In this work, three noises are used, gaussian, white, salt & pepper and speckle noises. In this paper, first the SVM is used in the last layer of RNN to improve the performance. To improve the learning accuracy and reduce the 20 40 60 80 100 Image 1 Image 2 Image 3 Image 4 Image 5 Image 6 Image 7 Image 8 Accuracy in % Lung CT images CNN Dn-CNN RNN HRNN-SVM 90 training time, batch normalization is integrated with residual learning. The PSNR and MSE results show that the proposed HRNN-SVM attained better results compared than other denoising schemes like DnCNN, CNN and RNN. Comparison of experimental results verify the denoising ability of the proposed method and indicate that it provides an effective solution to image denoising.

References

[1] Manjón, JV, Carbonell-Caballero, J, Lull, JJ, García-Martí, G, MartíBonmatí, L & Robles, M 2008, 'MRI denoising using non-local means', Medical image analysis, vol. 12, no. 4, pp. 514-523.

[2] Mitiche, L, Adamou-Mitiche, ABH & Naimi, H 2013, 'Medical image denoising using dual tree complex thresholding wavelet transform', IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), pp. 1-5.

[3] Lone, AH, Siddiqui, AN & Nazir, N 2018, 'Noise models in digital image processing', Advance and Innovative Research, vol. 5, no. 4, pp. 66-305.

[4] Kaur J & Kaur R 2014, 'Digital Image De-Noising Filters A Comprehensive Study', International Journal of Research In Computer Applications And Robotics, vol. 2, no. 4, pp. 105-111

[5] Verma, R & Ali, J 2013, 'A comparative study of various types of image noise and efficient noise removal techniques', International Journal of advanced research in computer science and software engineering, vol. 3, no. 10, pp. 617-622.

[6] Mredhula, L &Dorairangasamy, MA 2013, 'An extensive review of significant researches on medical image denoising techniques', International Journal of Computer Applications, vol. 64, no. 14, pp. 1-12.

[7] Manjón, JV, Coupé, P, Martí-Bonmatí, L, Collins, DL & Robles, M 2010, 'Adaptive non-local means denoising of MR images with spatially varying noise levels', Journal of Magnetic Resonance Imaging, vol. 31, no. 1, pp. 192-203.

[8] Kaur, L, Gupta, S, Chauhan, RC & Saxena, SC 2007, 'Medical ultrasound image compression using joint optimization of thresholding quantization and best-basis selection of wavelet packets', Digital Signal Processing, vol. 17, no. 1, pp. 189-198.

[9] Meyer, D & Wien, FT 2015, 'Support vector machines', The Interface to lib svm in package, pp. 1-8.

[10] Mehr, AD, Nourani, V, Khosrowshahi, VK & Ghorbani, MA 2019, 'A hybrid support vector regression–firefly model for monthly rainfall forecasting', International Journal of Environmental Science and Technology, vol. 16, no. 1, pp. 335-346.

[11] Mohammadi, K, Shamshirband, S, Danesh, AS, Zamani, M &Sudheer, C 2015, 'Horizontal global solar radiation estimation using hybrid SVM-firefly and SVM-wavelet al.gorithms: a case study', Natural Hazards, pp. 1-18.

[12] Voigtlaender, P, Doetsch, P& Ney, H 2016, ‘Handwriting recognition with large multidimensional long short-term memory recurrent neural networks’, International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 228-233

[13] Laurent, C, Pereyra, G, Brakel, P, Zhang, Y & Bengio, Y 2016, ‘Batch normalized recurrent neural networks’, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2657-2661.

[14] Pereira, C & Yang, XS 2016, ‘Learning parameters in deep belief networks through firefly algorithm’, Proceedings Artificial Neural Networks in Pattern Recognition: 7th IAPR TC3 Workshop, pp. 1-138.

[15] Wang, H, Wang, W, Sun, H &Rahnamayan, S 2016, ‘Firefly algorithm with random attraction’, International Journal of Bio-Inspired Computation, vol. 8, no. 1, pp. 33-41.