The Seybold Report

ISSN 1533-9211

DOI 10.17605/OSF.IO/CJU6V

DE-NOISING AND OPTIMIZATION OF MEDICAL IMAGES USING DEEP LEARNING TECHNIQUES

VIDHYA V[1], Dr. RAJAVARMAN V. N[2] and Dr. 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

Advanced image securing frameworks envelop various optical and electronic gadgets whose deficiencies cause clamor in the gained image. Clinical image handling involves algorithms and systems for tasks like image upgradation, compression of images, and so on. A few cutting-edge systems that convert human organs into computerized images for treatment incorporates X ray, Computed Tomography (CT), Magnetic Resonance Imaging (MRI). The major drawback of these medical images acquired by the said modalities is the errors in the images because of image acquisition and alteration in the images. These errors in images are called as noise. So, all the clinical images require some kind of De-noising techniques to improve the quality of the images being considered by experts to perform analysis and diagnosis. This work proposes an efficient De-noising approach by using Recurrent Neural network with Long-short-term memory (LSTM) and genetic optimization to eliminate Gaussian, and Salt & Pepper Noises.

Keywords: De-Noising of Medical Images; Image Noises; Optimization of Images; Deep Learning Techniques; Batch Normalization; Recurrent Neural Networks; Long Short Term Memory (LSTM)

1. INTRODUCTION

Machine learning techniques mainly focuses on finding patterns of interest in the data to uncover the hidden information. The performance of the machine learning techniques always relies on the quality of data that is used for analysis. Pre-processing is done in order to convert the data into a perfect data for analysis and feature extractions, so that interesting patterns for prediction can be obtained. It is very important to have a good feature extraction model to build a high performing machine learning tool which will be less complex to use [1]. As many new techniques have been developed in recent years, it is inevitable to use technology in health sector. In this, medical imaging is a technique that is used to visualize the bodily organs to help medical diagnosis and provide appropriate treatment. The organs that are normally imaged are bones, tissues and skeletons etc. The objective of the visualization and the further analysis is to diagnose the diseases in the patients and ensure an effective medical treatment for it. The investigation of medical images is research that has tremendously improved the abilities of the medical practitioners. The huge number of medical care associations and patients has given way to build and utilize computer-aided clinical diagnosis and decision support system [2]. Medical image analysis helps to interpret the contents of the image. Apart from image analytics, some techniques are more important for managing medical image like gathering, storing, compressing, sharing etc.

ISSN 1533-9211

DOI 10.17605/OSF.IO/CJU6V

The main objective of this research work is to build an efficient De-noising technique to eliminate noises such as Gaussian, salt pepper and white noises from the medical images. The proposed work is planned to develop an Enhanced Recurrent Neural Network (RNN) based De-noising method to reduce several types of noises affecting the medical images.

The medical image analysis faces similar kind of challenges compared to other Big Data analysis. However, because medical images can yield various types of data in each individual case, additional steps are necessary in this analysis. Data with high dimensionality and large in size poses a great challenge for Big Data analysis. Performance of most of the Big Data algorithms and approaches are good with medium sized data, but the performance degrades with sudden increase in dimensionality.

2. RELATED WORKS

Image Denoising is a fundamental and highly important technique in image processing that remains a challenging and actively researched area for scholars. Images are a necessary requirement in various fields, including geosciences, aerospace, education, agriculture, entertainment, surveillance and others, either in print or electronic form. Images are often affected by noise, but there has been a considerable research effort to address this issue and various techniques have been developed for resolving it. This section includes a review of optimization techniques for image denoising, an overview of machine learning methods for image denoising and a description of various image denoising methods. These techniques have been characterized on the basis of techniques utilized.

2.1 Reviews on Optimization Methods for Image Denoising

Pedada et al. (2009) showcased a variety of cost functions to assess noise removal quality and structural information preservation in denoised images [3]. Strength Pareto Evolutionary Algorithm 2 (SPEA2) was used to concurrently optimize these cost functions by altering the parameters linked with the denoising techniques. The Algorithm's efficiency was demonstrated by utilizing the proposed optimization method to improve image denoising outcomes with block matching and 3D collaborative filtering. The results of the experiments indicated that the performance of image denoising techniques could be significantly enhanced in both noise removal and preservation of structural information through the use of the proposed optimization algorithm.

Subashini et al. (2013)'s primary objective was to utilize wavelet techniques optimized by Ant Colony Optimization (ACO) for speckle filtering [4]. The study conducted three stages of work aimed at achieving three main objectives in speckle filtering: removing noise in uniform regions, preserving and enhancing edges and image features, and producing a visually appealing output. The first stage involved transforming the noisy image into the frequency domain, the second stage involved the manipulation of coefficients, and the third stage involved reconverting the resultant coefficients back into the original spatial domain. The results of the study revealed that statistical wavelet shrinkage filters effectively reduced speckle, but they also resulted in the loss of important feature details.

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/CJU6V

Tayee's (2013) main objective was to process noisy images using the particle swarm optimization (PSO) technique, which draws a formal analogy with physical systems [5]. By postulating that swarm motion behaves similarly to quantum and classical particles, the study establishes a direct connection between separate fields of study, making it possible to employ the quantum PSO algorithm to denoise images. The physical theory of PSO (particle swarm optimization) is utilized to propose several enhancements in algorithm itself. This work presents a comprehensive overview of the applications of both classical and quantum Particle Swarm Optimization (PSO) in solving challenging engineering problems, particularly in the field of image processing. The focus is on providing a multi-disciplinary perspective, with a unified framework that encompasses swarm dynamics. The proposed PSO algorithm is utilized to determine the optimal degree for de-noising in image processing. The simulation results indicate that the proposed method outperforms conventional approaches, which tend to introduce excessive smoothness in the processed images.

Ahamed et al. (2009) illustrated a new technique of hybrid filter for denoising digital images corrupted by mixed noise [6]. The hybrid filter design proposed in this study combines spatial domain filtering with the Neuro Fuzzy Network concept. To achieve noise reduction, the method employs an improved adaptive Wiener filter for reducing white Gaussian noise and an adaptive median filter for removing Impulse noise. The choice of filters used in this study is based on the effectiveness of the impulse noise detection process. An edge detector is utilized to identify edges in filtered images that may have been blurred due to various filtering operations. The optimization of the Neuro Fuzzy Network's training and internal parameters is achieved through the use of both synthetic and natural images.

Om & Biswas (2014) proposed an image denoising method that utilizes local parameters of neighboring coefficients of pixels to be denoised in noisy images [7]. This technique introduces two novel shrinkage factors and thresholds at each decomposition level, which result in significantly improved visual quality. The study also establishes a relationship between the two shrinkage factors. The performance of their proposed technique was compared with that of Neigh Shrink and Visu Shrink by including various variants. The simulation results demonstrated that their method achieved higher peak signal-to-noise ratio and superior visual quality of images as compared to conventional methods such as Modi Neigh Shrink, LAWML, IAWDMBNC, Weiner filter, Visu Shrink, Neigh Block, Neigh Shrink, Sure Shrink, and IIDMWT methods.

Kushwaha and Singh (2017) proposed an efficient denoising technique for ultrasound images that can effectively address noise, particularly speckle noise that is commonly acquired during image acquisition [8]. This study employs Anisotropic Diffusion as a method for image denoising. Furthermore, the study provides a more detailed analysis of the denoising process using Teaching-Learning Based Optimization techniques to achieve more accurate and targeted results. In addition, the study employs the HSOA algorithm and compares the results obtained from both techniques. The method for denoising proposed in this study significantly enhances the denoising process in ultrasound images of the liver. However, although the proposed system

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/CJU6V

is considered effective, the denoising performance achieved by the TLBO algorithm suggests that the image was over-filtered, leading to additional loss of data.

Chan & Chen (2006) proposed a rapid multilevel approach that utilizes primal relaxations to perform total variation image denoising and analyze the convergence of the method [9]. It is known that the fundamental primal relaxation method can become trapped at a non-stationary point, which is close to a local minimum of the minimization problem. This solution is considered "non-smooth" in the space of functions that have bounded variation. Their concept was to utilize coarse level corrections, conquering deadlock in the fundamental primal relaxation system leading to significant improvements over the relaxation method. Furthermore, to attain a global minimizer, additional refinement of the multilevel technique is necessary. They anticipated non-regular coarse level sourced on the concept of patch-detection, which involves hemivariateness, to enhance and correct the standard multilevel method. Both analytical and algorithmic results, along with numerical experiments on one- and two-dimensional images, were presented.

Cristin et al. (2020) [10] offered a hybrid approach that combines cuckoo search algorithm with artificial neural network to effectively remove noise from images using an adaptive non-linear Zernike filter. The simulation results demonstrate that the proposed algorithm is capable of preserving edges and other important features of the image while removing noise, resulting in a significant improvement in image denoising. The quality of the resulting image is evaluated using metrics such as Root Mean Square Error and Peak Signal to Noise Ratio (PSNR).

3. RECURRENT NEURAL NETWORK WITH NORMALIZATION FOR IMAGE DENOISING

In digital image acquisition systems, noise can be introduced due to the shortcomings of various optical and electronic devices. In medical image processing, various algorithms and procedures are used for operations such as image enhancement and compression. The digital forms of human organs produced by state-of-the-art equipment used for medical treatment include X-ray-based techniques such as radiography, CT (Computed Tomography), MRI (Magnetic Resonance Imaging), PET (Positron Emission Tomography), SPECT (Single Photon Emission Computed Tomography), and various methods in optical imaging. [11][12].

Medical imaging systems encounter significant challenges due to imperfect acquisition and transmission errors that alter the visual images obtained. The presence of noise in medical images has two major drawbacks, namely the degradation of image quality and the masking of important information that is crucial for accurate diagnosis. Consequently, denoising algorithms are required for all medical imaging devices to improve the quality of the images and aid medical professionals in making prompt and efficient diagnoses.

3.1 Recurrent Neural Network (RNN)

A Recurrent Neural Network (RNN) is a type of artificial neural network in which nodes are connected in a directed graph across a temporal sequence, allowing it to exhibit temporal dynamic behavior. Unlike feed forward neural networks, RNNs utilize their internal state or

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/CJU6V

memory to process sequences of inputs, making them ideal for tasks such as unsegmented, connected handwriting recognition or speech recognition. The term "recurrent neural network" is used broadly to refer to two classes of networks that share a similar structure, one being finite impulse and the other infinite impulse [13]. Both classes exhibit temporal dynamic behavior [14]. Finite impulse recurrent networks are directed acyclic graphs that can be unrolled and replaced with strictly feed forward neural networks, while infinite impulse recurrent networks are directed cyclic graphs that cannot be unrolled. Both types can have additional stored state, which can be directly controlled by the neural network.

3.2 Types of Recurrent Neural Networks (RNN)

3.2.1 Fully recurrent neural network (FRNN)

A new neural network called FRNN was developed that can learn temporal sequences, either in batch mode or online. The FRNN consists of two layers: an input layer with linear units and an output layer with non-linear units. The units in the input layer are fully connected to every unit in the output layer through adjustable weights. Each unit has a time-varying activation function with real-valued inputs. The output units have information about their previous activations, which are fed back to the input layer units. Learning in FRNNs involves mapping input sequences and activations to another set of output sequences, and this process continues to feedback to input sequences over multiple time steps, allowing for the discovery of abstract representations over time.

3.2.2 Recursive Neural network

Recursive Neural Networks are created using a differentiable graph like structure by applying the same set of weights recursively in a topological order. These networks can be trained using automatic differentiation in reverse mode and typically have a linear chain structure. They are commonly used in natural language processing to process distributed representations of structure. Another variation of this type of network is the Recursive Neural Tensor Network, which utilizes a tensor-based composition function at each node in the network.

3.2.3 Hopfield Neural Network

The Hopfield network is a neural network where all connections are symmetric. It is comprised of a set of N interconnected neurons that update their activation values asynchronously and independently of other neurons. On applying new input, the output is calculated and fed back to the input layers, and this process continues until the output becomes constant. The network can be learned using either the Hebbian Learning rule or the Storkey Learning rule. An adaptation of the Hopfield network is the Bidirectional Associative Memory (BAM), which is often used for Content Addressable Memory (CAM).

3.2.4 Elman Networks and Jordan Networks or Simple Recurrent Network (SRN)

The Elman network is a 3-layer neural network that includes additional context units. It consists of an input layer, an output layer, and a middle-hidden layer that is connected to the context unit with a weight of 1. At each time step, the input unit is applied to the learning unit and fed forward. The context units act as back-connections to the network, coming from the hidden

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/CJU6V

layer, and save the previous values of hidden units. The Jordan network is similar to the Elman network, with the only difference being that instead of a feedback context unit from the hidden layer, context units (also known as the state layer) are fed back from the output layer.

3.2.5 Echo State Network

An echo state network is a type of neural network that has sparse connectivity in its hidden layer, typically only 1% of the neurons are connected. The connections between the hidden neurons are fixed and its weights are randomly assigned. The weights of the output neurons can be learned so that the network can (re)produce specific time-based patterns. The weights which are modified during training are connected from hidden-to-output neurons.

3.2.6 Neural history compressor

It is an unsupervised network composed of stacked RNNs. The input to each subsequent level is predicted based on the learned representation of the previous layer. Inputs that cannot be predicted are passed on to the next higher level, where more hidden units are added to learn their representation. Each higher level contains inputs that were unpredicted by the lower levels and compressed information representation of RNN. The Network can be considered as having two levels: (conscious) chunker at higher level, (subconscious) automatizer at lower level. It can partially solve vanishing gradient problem of automatic differentiation or back propagation in Neural Networks.

3.2.7 Long short-term memory (LSTM)

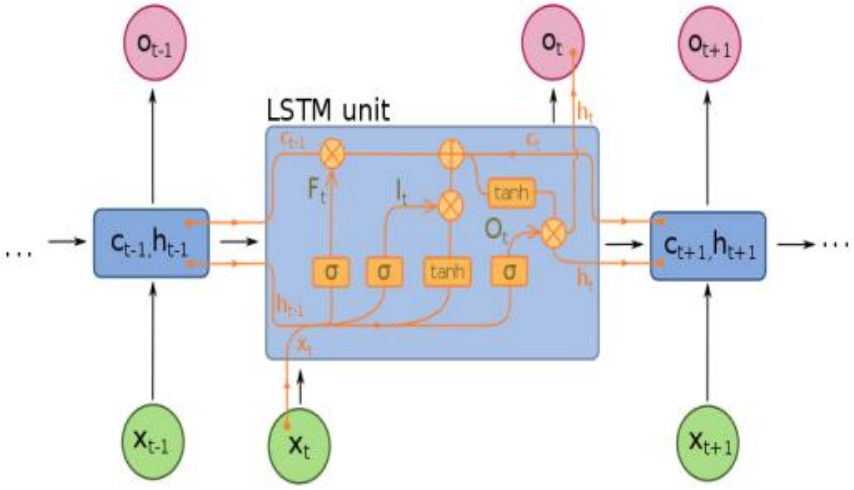
LSTM (Long Short-Term Memory) is a deep learning system that overcomes the vanishing gradient problem and is capable of learning various tasks. It incorporates recurrent gates called ‘forget’ gates to enhance its performance. To learn tasks, LSTM requires a memory of events that have occurred in the past. One way to train LSTM is by using Connectionist Temporal Classification (CTC), which can achieve both alignment and recognition for the weights.

3.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is an advanced type of artificial recurrent neural network used in deep learning that can not only process single data points (like images) but also entire sequences of data (like speech or video). Unlike traditional neural networks, LSTM has feedback connections that enable it to compute anything a machine can, making it a "general purpose computer". It is often used for tasks such as speech recognition and handwriting recognition. It consists of a cell, input gate, output gate, and forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. LSTM Networks are well-suited for classifying, processing and making predictions based on time series data since there can be lags of unknown duration between important events in a time series. LSTMs were created to address the issues of exploding and vanishing gradients that are commonly encountered when training traditional RNNs. Compared to RNNs, hidden Markov models, and other sequence learning techniques, LSTM offers the benefit of being less sensitive to gap length. This advantage has made LSTMs a popular choice for various applications.

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/CJU6V

Figure 3.1: Long Short-Term Memory



3.4 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a stochastic optimization technique that utilizes a population-based approach, developed by Kennedy and Eberhart in 1995. The concept of PSO is inspired by the social behavior of organisms, such as bird flocking and fish schooling, to describe a self-evolving system. In PSO, each single candidate solution is represented by a particle in the search space, similar to an individual bird in a flock.

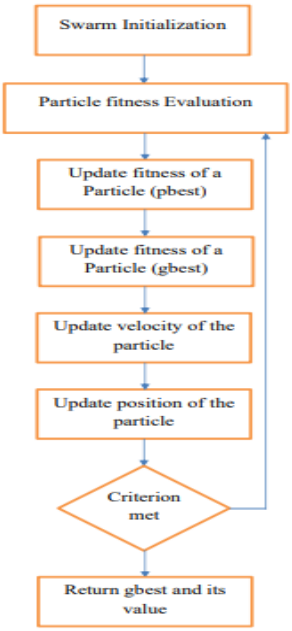
By utilizing its individual memory and the collective knowledge of the swarm, each particle strives to find the best solution. The fitness of each particle is determined by evaluating its fitness function, and the velocity of each particle guides its movement towards the optimal solution [16].

As particles move in PSO, they adapt their positions based on their personal experience as well as the experience of nearby particles. They use the best position encountered by themselves and their neighbors to guide their movement towards the optimal solution. The particles move through the problem space by following a stream of optimal particles. The PSO algorithm begins by initializing a group of random particles (solutions) and then updates generations to search for optima. In each iteration, every particle is updated by considering two "best" values.

The first best value is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is known as pbest. The other best value tracked by the particle swarm optimizer is the best value obtained by any particle in the population, known as gbest. If a particle considers only a subset of the population as its neighbors, then the best value is a local best called lbest. Once these two best values are determined, the particle updates its velocity and position.

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/CJU6V

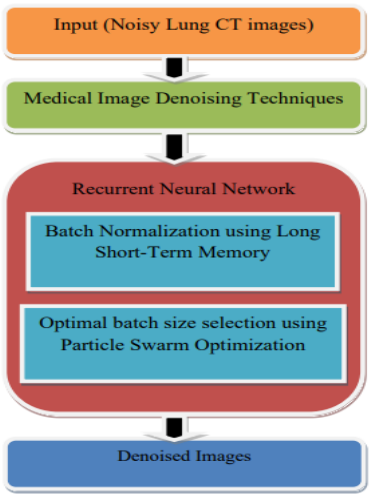
Figure 3.2: Flowchart of the PSO algorithm



4. PROPOSED METHODOLOGY

This proposed research work aims to improve medical image denoising by designing an effectual Recurrent Neural Network with Long Short-Term Memory that incorporates Batch Normalization. The optimization process involves two steps. Firstly, Long Short-Term Memory-based batch normalization is utilized to increase the efficiency of the optimization. Secondly, an effective batch size selection is performed using Particle Swarm Optimization algorithm. The overall flow diagram of the proposed work is shown in figure 3.3.

Figure 4.1: Flow diagram of the proposed work



The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/CJU6V

Initially, Lung CT images were obtained from the SIMBA Database to use as inputs for the experiments. In the experiments conducted, the artificially degraded images are simulated by applying the degradation model in order to design the training set for the neural network. The lung CT image is first convoluted with a selected noise operation such as Gaussian white noise or salt and pepper noise and then noise is added to it at various rates of occurrence.

A corrupted medical image can be given as:

$$\mathbf{y}(i, j) = \mathbf{x}(i, j) + n(i, j) \quad (1)$$

Where y(i,j) is the observed value, x(i,j) is the true (original) value (lung CT image) and n(i,j) is the noise perturbation at pixel (i,j) (added noise). In the learning phase of the recurrent neural network, the goal is to enable it to capture the inherent spatial relationships between degraded pixels and correlate them to their non-degraded pixels. This is achieved through supervised learning, whereby the degraded image data is fed as input to the recurrent neural network, and the non-degraded image is provided as the corresponding output.

4.1 Batch Normalization of Input Variables

In the optimization process, whitening is a commonly used technique that has been shown to reduce convergence rates (LeCun et al. 1998). When applied to deep neural networks, it can be understood as each arbitrary layer receiving samples from a distribution shaped by the layer preceding it. As the training progresses, this distribution changes, making each layer, except for the first, responsible for not only learning a good representation but also adapting to the changing input distribution. This phenomenon is referred to as Internal Covariate Shift, and reducing it is believed to facilitate the training procedure (Ioffe&Szegedy 2015).

Whitening each layer of the network is a potential solution for mitigating internal covariate shift. However, this approach can be computationally intensive. Instead, batch normalization is used as an approximation by standardizing the intermediate representations using the current minibatch's statistics. Specifically, given a mini-batch x, the system can calculate the sample mean and sample variance of each feature k along the mini-batch axis.

$$\bar{X}_k = \frac{1}{m}\sum_{i=1}^{m} x_{i,k}$$

$$\sigma_k^2 = \frac{1}{m}\left(x_{i,k} - \bar{x}_k\right)^2$$

Where m is the size of the mini-batch. The mini-batch is optimally selected by using PSO to improve the PSNR performance in RNN.

4.2 Particle Swarm Optimization

In this study, the Particle Swarm Optimization (PSO) algorithm is used to optimally select the batch size for training the recurrent neural network. In PSO, a group of birds fly and communicate with each other. Each bird has a specific location which communicates

ISSN 1533-9211

DOI 10.17605/OSF.IO/CJU6V

collectively and recognizes the bird in the best location. Each bird then flies in the direction of the best bird using a velocity based on its current position. All the birds explore the search space from their new local location, and the process is repeated until the flock reaches a desired destination. Notably, this process involves both social interaction and intelligence, allowing the birds to learn from their own experience called as local search and the experiences of others around them known as global search. The process begins with a set of random particles, N, where the i^{th} particle is represented by its position as a point in S-dimensional space, where S represents the number of variables. During the process, each particle i observes three values: its current position (X), the best position it has reached in previous cycles (Pi), and its flying velocity (Vi).

These three values are denoted as follows:

$$\text{Current position } X_i = (x_{i1}, x_{i2}, \dots, x_{iS})$$

$$\text{Best previous position } P_i = (p_{i1}, p_{i2}, \dots, p_{iS})$$

$$\text{Flying velocity } V_i = (v_{i1}, v_{i2}, \dots, v_{iS}$$

At each time interval or cycle, the best particle (g) is determined as the particle with the best fitness and its position (Pg) is computed. Each particle updates its velocity (Vi) based on the position of the best particle (g) in order to get closer to it. The velocity update process is as follows.

$$New\ V_i = \omega \times current\ V_i + c_1 \times rand() \times (P_i - X_i) + c_2 \times Rand()\ \times (P_i - X_i)$$

After the velocity of each particle (Vi) is updated based on the position of the best particle (g), the particle's position is updated accordingly. The updated position of the particle can be expressed as

$$New\ position\ X_i = current\ position\ X_i + New\ V_i$$

$$V_{max} \geq V_i \geq -V_{max}$$

where $c1$ and $c2$ represent two positive constants named learning factors (usually $c_1 = c_2 = 2$); rand ( ) and Rand ( ) denote two random functions in the range [0, 1], V_{max} is an upper limit on the maximum change of particle velocity, ω denotes an inertia weight employed as an enhancement to manage the influence of the previous history of velocities on the current velocity. The parameter ω is used to balance the global search and local search in the Particle Swarm Optimization (PSO) algorithm.

It is gradually decreased linearly with time, starting from a value of 1.4 and ending at 0.5. This enables the algorithm to prioritize global search initially with a large weight, and then shift towards local search as time progresses.

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/CJU6V

Detailed pseudo-code of PSO algorithm:

1. A population of agents (Number of pixels) is created randomly.

$$X_i = (P_1, P_2, P_3, \ldots \ldots \ldots, P_N)$$

2. Each particle's position is evaluated based on the objective function. In this case, it is the total operational cost given by C. The goal is to minimize this objective function, and particles are evaluated based on how well they achieve this objective. This evaluation process helps to determine the fitness of each particle.
3. Cycle =1
4. Repeat
5. Update the velocity of the particles according to the formula

$$V_i(t) = V_i(t-1) + C_i r_i\big(\text{pbest}(t) - x_i\,(t-1)\big) + C_2 r_2\big(\text{gbest}(t) - x_i\,(t-1)\big)$$

   Where c = acceleration factor. r = random values between 1 and 0
6. Evaluate the velocity to ascertain if it is the range of

$$V_{max} \leq V_i \leq V_{min}$$

7. Move particles to their new position

$$X_i\,(t) = X_i\,(t-1) + V_i(t)$$

8. Evaluate to ensure that limits have not been exceeded.
9. Compare the particle's fitness evaluation with its previous pbest. If the current value is better than the previous pbest, then set the pbest value equal to the current value and the pbest location equal to the current location in the N dimensional search space.
10. Compare the best current fitness evaluation with the population gbest. If the current value is better than the population gbest, then reset the gbest to the current best position and the fitness value to current fitness value.
11. Check if stopping criterion had been met. If not update the cycle and go back to step (5).
12. End when the stopping criterion, which here is the number of iterations, has been met (produce the optimal batch size).

4.3 Long Short-Term Memory

The Long Short-Term Memory (LSTM) is a frequently used recurrent architecture that overcomes the vanishing gradient issue that is commonly present in vanilla RNNs by incorporating gating functions into its state dynamics. In an LSTM, at each time step, there are two key vectors that are maintained: a hidden vector h and a cell vector c, which are responsible for regulating state updates and outputs.

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/CJU6V

Specifically, the system computes its state at time step t as follows: (Gers et al. 2002)

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

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

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

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

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

Where sigmoid (·) is the logistic sigmoid function, tanh is the hyperbolic tangent function, W_h· are the recurrent weight matrices and W_x· are the input-to-hidden t weight matrices. i_t, $f_t$and o_t are respectively the input, forget and output gates, and c_t is the cell.

5. EXPERIMENTAL RESULTS

The Proposed technique was implemented using MATLAB and tested on medical images using various types of noises such as Gaussian, white, salt and pepper, speckle, and Poisson, with zero mean and different variance values. The objective was to assess the effectiveness of the proposed method and compare it with existing methods. The performance comparison of the DnCNN, CNN, RNN, HRNN-SVM and HRNN-TSVM methods are analysed in terms of PSNR, MSE and accuracy.

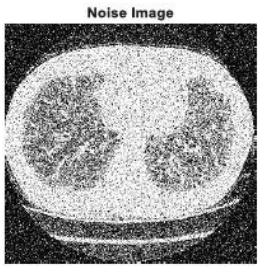


Figure 5.1: Input image with white noise

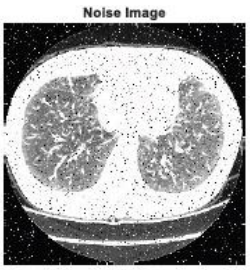


Figure 5.2: Input image with salt and pepper noises

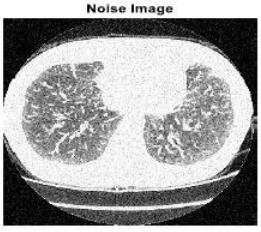


Figure 5.3: Input image with Gaussian noise

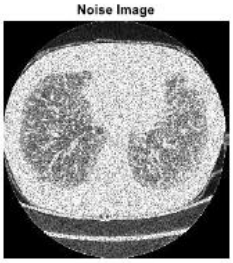
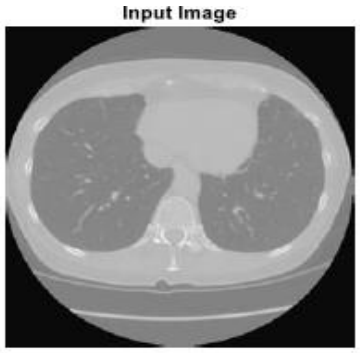


Figure 5.4: Input image with speckle noise

ISSN 1533-9211

DOI 10.17605/OSF.IO/CJU6V

Figure 5.5: Input image with Poisson noise



The white noise, Gaussian, salt and pepper noise, speckle noise and Poisson noise are initially added into CT lung images to perform denoising. The images with noises are shown in Figure 5.1 to 5.5.

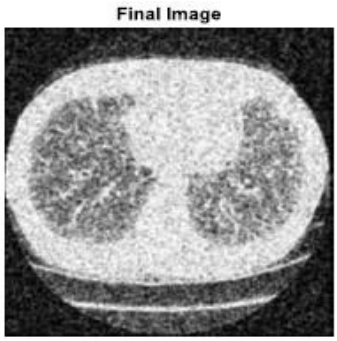


Figure 5.6: De-noised image for white noise

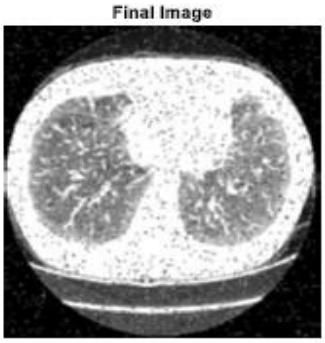


Figure 5.7: De-noised image for salt and pepper noises

Figure 5.8: Denoised image for Gaussian noise

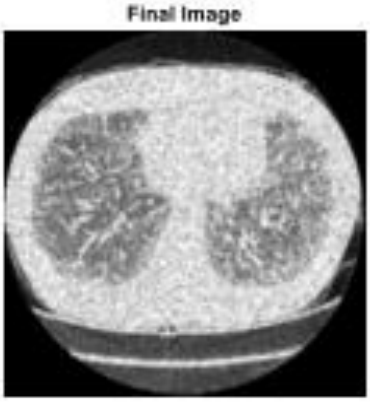


Figure 5.9: Denoised image for speckle noise

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/CJU6V

Figure 5.10 Denoised Image for Poisson Noise

Figure 5.6 to 5.10 shows the denoised image. The denoising process is done with the help of hybrid Recurrent Neural Networks (RNN) with Transductive Support Vector Machines (TSVMs). It can able to detect the white noise, gaussian, salt and pepper noise, speckle noise and Poisson noises efficiently.

5.1 Peak to Signal Noise Ratio (PSNR)

The PSNR evaluation of the proposed method HRNN-TSVM and other methodologies such as DnCNN, CNN, RNN, and HRNN-SVM are represented in Table 5.2.

Table 5.2: PSNR comparison

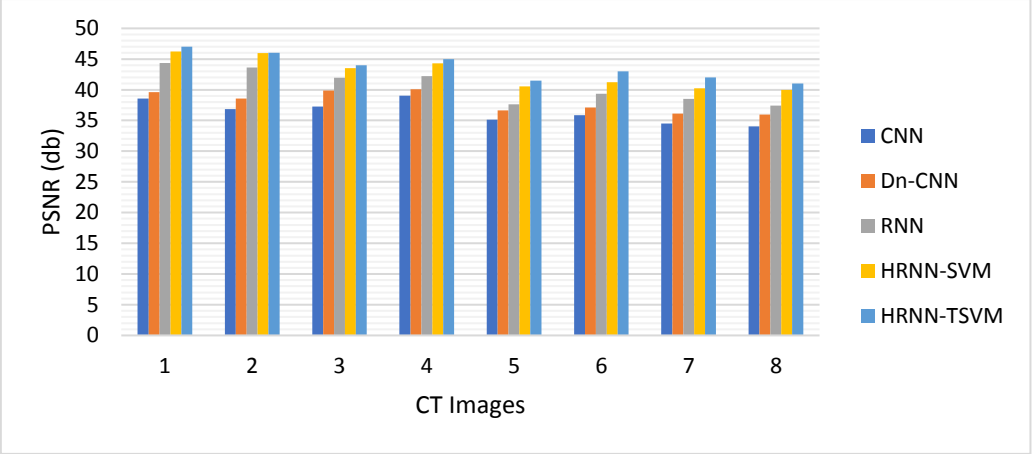
No of images	PSNR comparison				
	CNN	Dn-CNN	RNN	HRNN-SVM	HRNN-TSVM
Image 1	38.56	39.63	44.38	46.25	47
Image 2	36.85	38.56	43.63	45.96	46
Image 3	37.25	39.87	41.96	43.52	44
Image 4	39.02	40.10	42.21	44.32	45
Image 5	35.12	36.65	37.65	40.54	41.5
Image 6	35.85	37.12	39.33	41.21	43
Image 7	34.51	36.12	38.54	40.24	42
Image 8	34.02	35.95	37.41	40.00	41

The PSNR comparison of the proposed method HRNN-TSVM and other methodologies such as DnCNN, CNN, RNN, HRNN-SVM has been represented graphically in Figure 5.11. The CT lung images are plotted on the x-axis, while the y-axis represents the corresponding PSNR values. The proposed research work involves replacing the last layer of the RNN with TSVM to enhance the denoising performance. Furthermore, an optimal batch size is chosen using FA, which leads to an improvement in PSNR.

ISSN 1533-9211

DOI 10.17605/OSF.IO/CJU6V

Figure 5.11: PSNR Comparison for Medical CT Images

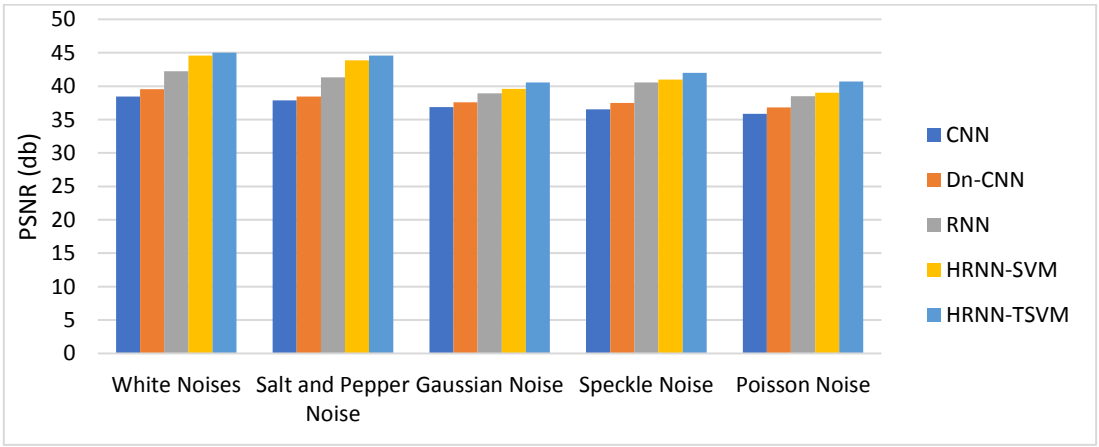


The experimental results prove that the proposed denoising method based on HRNN-TSVM yields a higher PSNR compared to other techniques such as DnCNN, CNN, and HRNN-SVM. The PSNR comparison is analyzed for various types of noises which are represented in table 5.3.

Table 5.3 PSNR Comparison for Noises

Noises	PSNR comparison				
	CNN	Dn-CNN	RNN	HRNN-SVM	HRNN-TSVM
White noise	38.43	39.56	42.23	44.56	45
Salt and pepper noise	37.85	38.43	41.34	43.84	44.58
Gaussian noise	36.86	37.58	38.95	39.58	40.58
Speckle noise	36.52	37.49	40.58	40.98	42
Poison noise	35.86	36.8	38.5	39	40.68

Figure 5.12 PSNR comparison between various noises



ISSN 1533-9211

DOI 10.17605/OSF.IO/CJU6V

The performance of the proposed denoising method based on HRNN-TSVM and existing techniques, including DnCNN, CNN, and HRNN-SVM, are evaluated for different types of noises such as White, Salt and Pepper, Speckle, and Poisson. The graph indicates that the proposed approach outperforms the existing methods in terms of achieving higher PSNR values for all noises

5.2 Mean Squared Error (MSE)

Table 5.4 presents the analysis of MSE values for CT lung images using the proposed HRNN-TSVM and existing techniques such as CNN, DnCNN, RNN, and HRNN-SVM.

Table 5.4: Mean Square Error comparison

No of images	MSE comparison				
	CNN	Dn-CNN	RNN	HRNN-SVM	HRNN-TSVM
Image 1	9.8	9.2	8.8	7	6.8
Image 2	9.6	9.4	8.9	7.5	6.2
Image 3	9.5	9	8.5	8	6.9
Image 4	9.4	8.9	8.2	7.9	7.3
Image 5	9.3	8.6	7.8	7.5	7.0
Image 6	9.1	8.3	7.3	7.0	6.8
Image 7	9.0	8.2	7.0	6.5	6.2
Image 8	8.9	8.0	6.8	6.1	6.0

Figure 5.13 MSE Comparison for Medical CT Images

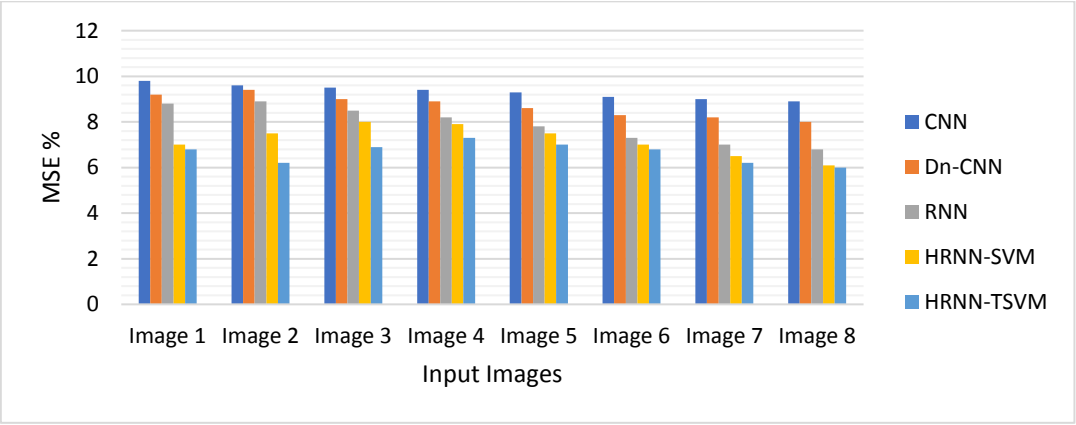


Figure 5.13 displays the MSE values of the CT lung images during the denoising process, where the x-axis represents the CT lung images and the y-axis represents the corresponding PSNR values. The graph clearly shows that the proposed HRNN-TSVM approach achieves an MSE value of 6.2% for image 2, while other techniques such as CNN, DnCNN, RNN, and HRNN-SVM achieve MSE values of 9.6%, 9.2%, 8.9%, and 7.5%, respectively. Based on

ISSN 1533-9211

DOI 10.17605/OSF.IO/CJU6V

these results, it can be concluded that the proposed method yields better PSNR compared to the existing approaches.

5.3 Accuracy

Table 5.5 presents the comparison of accuracy values for CT lung images, where the medical images are corrupted with various types of noise and then denoised using existing and proposed methods.

Table 5.5: Accuracy comparison

No of images	Accuracy comparison				
	CNN	Dn-CNN	RNN	HRNN-SVM	HRNN-TSVM
Image 1	87	88	93	93.5	94
Image 2	88	89	94	94.2	94.8
Image 3	89	90	95	95.3	96
Image 4	90	92	96	96.2	96.5
Image 5	92	93	97.5	98	98.2
Image 6	93	94	98	98.4	98.8
Image 7	94	95	98.2	98.9	99.2
Image 8	95	96	98.5	99.3	99.5

Figure 5.14: Accuracy Comparison for Medical CT Input Images

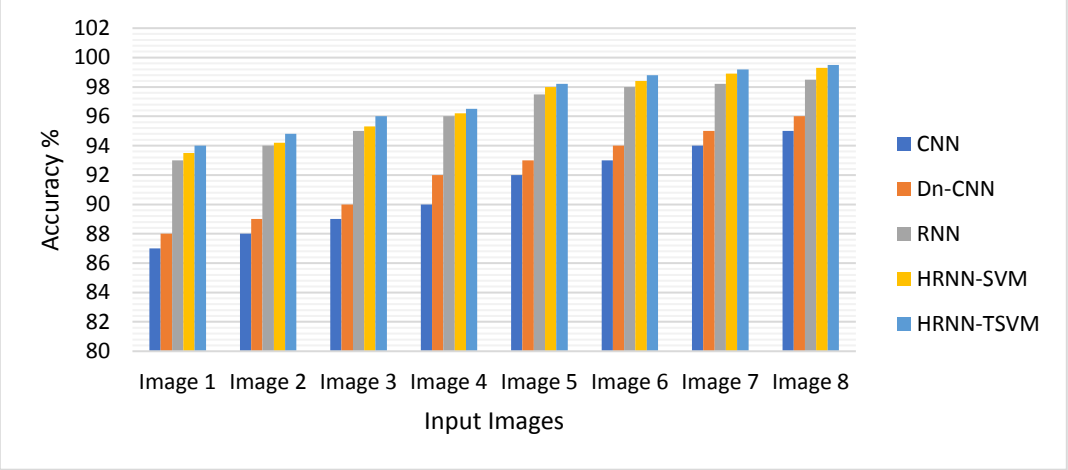


Figure 5.14 provides a graphical representation of the accuracy comparison between the proposed HRNN-TSVM approach and other methodologies, including DnCNN, CNN, HRNN-SVM, for denoising CT lung images. The x-axis represents the images, and the y-axis represents the corresponding PSNR values. Based on the results, the proposed HRNN-TSVM approach achieves an accuracy of 94.8% for image 2, while other techniques such as CNN, DnCNN, RNN, and HRNN-SVM achieve accuracy values of 88%, 89%, 94%, and 94.2%, respectively.

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/CJU6V

Hence, the proposed system is designed to improve the denoising performance and has achieved better results compared to other denoising schemes in terms of PSNR, MSE, and accuracy for various types of noise, including gaussian, white noise, salt and pepper noise, speckle noise, and Poisson noise. For example, based on Figure 5.16, it can be concluded that the proposed HRNN-TSVM approach achieves an accuracy of 94.8% for image 2, while other techniques such as CNN, DnCNN, RNN, and HRNN-SVM achieve accuracy values of 88%, 89%, 94%, and 94.2%, respectively.

6. CONCLUSION

The aim of this study is to develop an efficient denoising system to remove noise from medical images. To achieve this, a denoising scheme is proposed that combines Long Short-Term Memory (LSTM) based Batch Normalization and Recurrent Neural Network (RNN) techniques to remove gaussian, white, salt and pepper noises. The use of LSTM based batch normalization is aimed at reducing internal covariate shift in neural networks. Particle Swarm Optimization (PSO) algorithm is used to select an optimal batch size to improve the denoising image quality. The proposed technique is implemented using MATLAB, and the experimental results showed that it achieved 42.38 dB of PSNR and 8.9% of MSE, which outperforms existing systems such as CNN and DnCNN.

References

1) Tahmassebi, A., \ideeple: Deep learning in an ash," in [Disruptive Technologies in Information Sciences], 10652, 106520S, International Society for Optics and Photonics (2018).

2) Maes, F., Robben, D., Vandermeulen, D. and Suetens, P., 2019. The role of medical image computing and machine learning in healthcare. In Artificial Intelligence in Medical Imaging (pp. 9-23). Springer, Cham.

3) Pedada, R., Kugu, E., Li, J., Yue, Z. and Shen, Y., 2009, March. Parameter optimization for image denoising based on block matching and 3D collaborative filtering. In Medical Imaging 2009: Image Processing (Vol. 7259, pp. 713-724). SPIE.

4) Subashini, P., Krishnaveni, M., Ane, B.K. and Roller, D., 2013. Wavelet based image denoising using ant colony optimization technique for identifying ice classes in SAR imagery. In Soft Computing Models in Industrial and Environmental Applications: 7th International Conference, SOCO'12, Ostrava, Czech Republic, and September 5th-7th, 2012 (pp. 399-407). Springer Berlin Heidelberg.

5) Tayee, A.A.W.F.A., 2013. Denoising An Image Based On Particle Swarm Optimization (PSO) Algorithm. Journal of University of Babylon, 21(5).

6) Ahamed, J.N. and Rajamani, V., 2009. Design of hybrid filter for denoising images using fuzzy network and edge detecting. American Journal of Scientific Research, 3, pp.5-14.

7) Om, H. and Biswas, M., 2014. MMSE based map estimation for image denoising. Optics & Laser Technology, 57, pp.252-264.

8) Kushwaha, S. and Singh, R.K., 2017. An efficient approach for denoising ultrasound images using anisotropic diffusion and teaching learning based optimization. Biomedical and Pharmacology Journal, 10(2), pp.805-816.

9) Chan, T.F. and Chen, K., 2006. An optimization-based multilevel algorithm for total variation image denoising. Multiscale Modeling & Simulation, 5(2), pp.615-645.

ISSN 1533-9211

DOI 10.17605/OSF.IO/CJU6V

10) Cristin, R., Kumar, B.S., Priya, C. and Karthick, K., 2020. Deep neural network based Rider-Cuckoo Search Algorithm for plant disease detection. Artificial intelligence review, 53, pp.4993-5018.

11) Pizurica, A. and Philips, W., 2006. Estimating the probability of the presence of a signal of interest in multiresolution single-and multiband image denoising. IEEE Transactions on image processing, 15(3), pp.654-665.

12) Chang, SUN & Chao, ZHANG 2018, 'Progress in medical image denoising technology based on local and nonlocal filtering', Chinese Journal of Medical Physics.

13) Santoro, A, Faulkner, R, Raposo, D, Rae, J, Chrzanowski, M, Weber, T & Lillicrap, T. 2018, 'Relational recurrent neural networks', Advances in Neural Information Processing Systems, pp. 7299-7310.

14) Toderici, G, Vincent, D, Johnston, N, Jin Hwang, S, Minnen, D, Shor, J & Covell, M 2017, 'Full resolution image compression with recurrent neural networks', Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5306-5314.

15) Marini, F & Walczak, B 2015, 'Particle swarm optimization (PSO)', a tutorial. Chemo metrics and Intelligent Laboratory Systems, pp. 153-165.

16) Du, KL & Swamy, MNS 2016, 'Particle swarm optimization', In Search and optimization by Meta heuristics, pp. 153-173.