



## Enhanced Speckle Noise Reduction in Breast Cancer Ultrasound Imagery Using a Hybrid Deep Learning Model

Nagireddy Venkata Raja Sekhar Reddy<sup>1\*</sup>, Chengamma Chitteti<sup>2</sup>, Sreeraman Yesupadam<sup>3</sup>, Venkata Subbaiah Desanamukula<sup>4</sup>, Sai Srinivas Vellela<sup>5</sup>, Naga Jagadesh Bommagani<sup>6</sup>

<sup>1</sup> Department of Information Technology, MLR Institute of Technology, Hyderabad 500043, India

<sup>2</sup> Department of Data Science, School of Computing, Mohan Babu University, Tirupati 517102, India

<sup>3</sup> Department of CSE, School of Technology, The Apollo University, Chittoor 517127, India

<sup>4</sup> Department of Computer Science and Engineering, Lakireddy Bali Reddy College of Engineering, Mylavaram 521230, India

<sup>5</sup> Department of CSE-Data Science, Chalapathi Institute of Technology, Guntur 522016, India

<sup>6</sup> School of Computer Science and Engineering, VIT-AP University, Vijayawada 522237, India

Corresponding Author Email: [drrajasekhar@mlrinstitutions.ac.in](mailto:drrajasekhar@mlrinstitutions.ac.in)

<https://doi.org/10.18280/isi.280426>

### ABSTRACT

**Received:** 19 May 2023

**Revised:** 26 July 2023

**Accepted:** 23 August 2023

**Available online:** 31 August 2023

#### Keywords:

*breast cancer, fundamental convolutional neural network ultrasound images, improved noise removal, speckle noise, Fundamental Convolutional Neural Network (FCNN)*

Ultrasonic imaging serves as a pivotal tool in mitigating overdiagnosis of breast cancer in women, owing to its high sensitivity, low false-positive rate, and ability to reduce unnecessary biopsies. Nevertheless, these images are impaired by speckle noise, which appears as granular interference obscuring tissue boundaries and diminishing image contrast. This noise impedes subsequent image processing tasks such as edge detection, segmentation, feature extraction, and classification. Existing strategies for speckle noise reduction in ultrasonic images either compromise on effectiveness or demand substantial processing time, presenting challenges in preserving fine edge details. Addressing these issues, we propose an innovative hybrid deep learning model, FCNN-IDOA, which synergizes a Fundamental Convolutional Neural Network (FCNN) with an optimization algorithm. Our FCNN model is built upon the framework of GoogLeNet, enhanced with fifteen additional layers to augment its expressiveness. Subsequently, an Improved Dragonfly Optimization Algorithm (IDOA) is deployed to optimize FCNN's parameters, thereby improving the computational efficiency of the model. The suggested model has demonstrated superior performance, outstripping previous models in terms of accuracy. During experimental validation, the model achieved an average t(s) value of 84.764421, a PSNR value of 66, an MSE value of 54.9143, an RMSE value of 0.491631, and a final t(s) value of 83.759067. The results indicate that this novel model significantly outperforms the BC models, rendering it a promising solution for speckle noise reduction in breast cancer ultrasound images.

## 1. INTRODUCTION

Breast cancer is a globally prevalent disease, with early detection being pivotal in reducing mortality [1]. The efficacy of treatments is significantly amplified when tumors are detected at an incipient stage. A precise diagnostic tool capable of distinguishing benign from malignant tumors is therefore indispensable for early detection. Traditionally, mammography has been the gold standard for early detection and diagnosis of breast cancer [2]. However, its applicability is circumscribed, especially in detecting cancer in young women with dense breast tissue. Moreover, both patients and radiologists are exposed to potentially harmful ionizing radiation during mammography.

As an alternative, Breast Ultrasound (BUS) imaging is increasingly being harnessed due to its non-invasive, non-radioactive, and cost-effective characteristics, making it more compatible with mass breast cancer screening and diagnosis [3]. For women under 35, ultrasound is particularly advantageous as it can detect abnormalities in dense breasts more effectively than mammography. However, ultrasound images are susceptible to various types of noise, the most

significant of which is speckle noise [4].

Speckle noise, a high-frequency artifact, is a random noise pattern formed by a large number of waves scattered from tissues with varying phases [5]. Interference between these scattering waves can have deleterious effects, such as the creation of speckles and mottled B-scan noises, or potentially beneficial effects, such as the generation of strong noise [6]. Speckle noise detrimentally affects the perceived quality of the ultrasound image by introducing artificial structures and obscuring the true tissue boundaries, complicating subsequent tasks in the image processing pipeline, such as edge feature extraction [7]. Compounding the issue, speckle is multiplicative in nature, unlike most noises which are additive, posing challenges in its removal from ultrasound images [8].

There are five distinct families of speckle-reduction algorithms, including different types of image processing filters, the Generalised Likelihood Method (GLM) filter, and Wavelet-Based filters. Two common issues with local adaptive filters are their sensitivity to noise and their propensity to artificially enhance high contrast regions of images [9]. Anisotropic diffusion filters require extensive parameter tuning and degrade small structures in addition to

reducing image resolution [10]. Algorithms based on multiscale approaches are computationally demanding and require additional constraints. Non-local means filters result in a similar increase in computational time due to weighted averaging, while hybrid approaches degrade image quality and lead to blurred edges.

While existing filtering algorithms are somewhat effective in reducing speckle noise, they often compromise on image fidelity key for breast cancer detection [11]. These filters tend to erase the finer edge features, which are crucial for diagnosis, along with the speckles, thereby blurring the exact boundaries of tumors [12]. Most algorithms employ a locally adaptive restoration paradigm, in which the restoration of a pixel's value is determined based on neighboring pixels. Non-local approaches are not solely dependent on neighboring pixels, but they require more processing or computation time [13].

This study explores the implementation of a deep learning (DL)-based technique designed to augment the efficacy and performance of existing algorithms for speckle noise removal in ultrasound images. The proposed methodology, an innovative hybrid FCNN-IDOA model, leverages DL techniques for hyperparameter tuning and incorporates a Softmax classification layer to ensure flexibility. Rigorous testing against a diverse range of DL models, using a publicly accessible dataset, reveals the proposed model to outperform state-of-the-art methods in terms of accuracy.

In the realm of breast cancer ultrasound imaging, the presence of speckle noise poses considerable challenges. To counteract this issue, several image processing techniques are typically employed, with the following methods being the most commonly utilized:

- **Median Filtering:** This is a non-linear filtering technique wherein the value of each pixel is replaced with the median value within its immediate vicinity. It demonstrates efficacy in speckle noise reduction whilst preserving the edges and details of the image.
- **Wiener Filtering:** As a statistical-based method, Wiener filtering aims to minimize the mean square error between the original and filtered image by estimating the noise power spectrum and applying a frequency-dependent filter to reduce the noise.
- **Anisotropic Diffusion:** This technique harnesses the process of diffusion to eliminate noise while preserving edges. It selectively diffuses the image based on gradient information, allowing for noise reduction without compromising important image features.
- **Non-local Means Denoising:** This method capitalizes on the redundancy in the image to reduce noise. It compares similar patches from different regions of the image to estimate the noise-free pixel value.

The structure of the present paper is as follows: Section 2 provides a detailed literature review, followed by an explanation of the proposed model in Section 3. An experimental analysis and validation of the model are given in Section 4, and a final analysis of the model is presented in Section 5.

## 2. RELATED WORKS

In the realm of image denoising, numerous novel methodologies have been proposed. Vimala et al. [14] employed a hybrid deep learning approach for the removal of local speckle noise from breast ultrasound images. Their

technique entailed enhancing the contrast of ultrasound images with logarithmic and exponential transformations prior to the application of guided filter algorithms to amplify details in glandular ultrasound images. They further modified the Logical-Pool (LPRNN) with edge-sensitive terms to filter local noise while preserving the integrity of the image boundaries. Evidence of successful training of the LPRNN was indicated by a mean square rate of less than 1.1% after one iteration. Moreover, its Signal-to-Noise Ratios (SNRs) of 65 dB, peak, and rapid decay rates highlighted its effectiveness.

Huang et al. [15] introduced a unique denoising method, the Dual Deep Denoising Convolutional Neural Networks (D3CNNs), designed to eliminate both random and striped noise. They formulated the inverse problem as a constrained optimization problem, solvable through iterative methods, by introducing two auxiliary variables for the image and stripe noise. They trained the image variable with the residual CNN (RCNN), and their experimental results showcased the superior effectiveness of their approach, with outcomes equivalent to state-of-the-art techniques, both subjectively and objectively.

Li et al. [16] proposed an adaptive iterative non-sampled shearlet transform (NSST) technique based on enhanced soft thresholding to address the issues induced by hard thresholding discontinuity and soft thresholding constant deviation. The technique mitigates the effects of oversmoothing and restores the image to its original state. Experimental evidence supports that their method outperforms other deep learning denoising techniques in terms of Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) measurements.

In an effort to address practical image denoising, Zhang and Zhou [17] presented a unique Contrast-Aware Dual-Task (CADT) unit and Secondary Noise Extractor (SNE) block-based Denoise Transformer. The CADT unit comprises a local branch that focuses on extracting features from nearby pixels with narrow receptive fields and a global branch that employs a Transformer encoder to capture global details. They used a hierarchical network built with CADT as basic components to quickly learn noise distribution via residual learning, generating the first stage output. The SNE was then employed to efficiently remove secondary global noise with minimal computational effort. The final output of the Denoise Transformer was collected, and blind spots were reconstructed to complete the image.

Hu et al. [18] proposed a method to prevent model overfitting on the noisy image and enhance performance. Their approach, TripleDIP, demonstrated a significant improvement over the original Deep Image Prior (DIP) and current supervised models such as SwinIR and Restormer on the Set12 dataset. The main driver of this achievement was the implementation of a two-branch noise learning strategy that produced stable noise without constraining the optimization process of the content learning branch.

Xu et al. [19] introduced a proposed Deep Unfolding Multi-scale Residual Network (DUMRN) for image denoising that includes image denoising in the feature space explicitly. At its core, a Feature Denoising Module (FDM) operates by directly removing noise from the deep feature space. The DUMRN, built by layering FDMs and trained from scratch, demonstrated superior performance compared to state-of-the-art methods using both synthetic and real-world benchmarks.

Yan et al. [20] proposed a Transfer Learning Dense Convolutional Denoising (TLD-CDL) framework to improve

the resolution of denoising results by integrating a convolutional neural network (CNN). Their approach started with equipping the network with dense connections and a structural design, followed by training a pre-model on a dataset of natural images. Finally, they applied transfer learning to adapt the model for post-processing of low-dose computed tomography (LDCT) images. They further applied a perceptual loss to guide the training. Their experimental results showed impressive effectiveness in both quality and quantity, maintaining a balanced performance between noise suppression and detail preservation.

### 2.1 Problem statement

In the field of ultrasound imaging, speckle noise is a pervasive issue, invariably compromising the quality of the images and hindering subsequent analyses. In this context, the present study introduces a novel denoising algorithm meticulously designed for the reduction of speckle noise in ultrasound images. Speckle noise, typified by its multiplicative interference, is known to degrade visual clarity and obstruct the accurate interpretation of ultrasound images.

The algorithm proposed in this study leverages a combination of multi-scale analysis and non-local means filtering, aiming to effectively eliminate speckle noise whilst preserving critical image features. A series of experimental results, derived from a comprehensive dataset of ultrasound images, underscore the superior performance of the proposed algorithm in terms of both speckle reduction and preservation of structural details.

Speckle noise, besides degrading image quality, also exerts a detrimental impact on subsequent processes in the image processing pipeline, including but not limited to, edge recognition, segmentation, feature extraction, and classification. By mitigating the effects of speckle noise, the algorithm proposed herein contributes significantly to enhancing the accuracy and reliability of these subsequent image analysis tasks.

The proposed algorithm, therefore, holds considerable promise for augmenting the diagnostic value of ultrasound imaging across a diverse range of medical applications. This paper provides a comprehensive literature review on this topic, showing how the proposed algorithm can make a significant contribution to the field.

## 3. PROPOSED METHODOLOGY

### 3.1 The denoising process

They can emphasize any novel architectural designs, algorithmic advancements, or data processing techniques employed in their Hybrid Deep Fundamental CNN. For example, they could highlight the use of specific layers, activation functions, or loss functions that have not been extensively explored in previous works. A comparative analysis of their Hybrid Deep Fundamental CNN with existing methods, both qualitatively and quantitatively. They can demonstrate how their methodology outperforms previous approaches in terms of noise reduction effectiveness, preservation of important image features, computational efficiency, or other relevant evaluation metrics. Comparative results, such as visual comparisons and performance metrics, can provide strong evidence of the superiority of the proposed

methodology.

K-means clustering using Hu's moment invariants is used for pre-classification after the data has been filtered using a Gaussian filter. Gaussian filter smooths the picture while keeping following operations scale invariant, and improves the proposed model by selecting qualified candidates for the weighted averaging job. The suggested technique employs a K-means clustering algorithm that utilises Hu's moment group like candidates into clusters. To explain the rationale behind choosing a specific filter size in the pooling layer. They can discuss how the chosen filter size affects the down sampling process, feature representation, or computational efficiency. Additionally, authors can refer to prior studies or empirical evidence that support the use of a particular filter size for the given task or dataset. We have chosen the Iterative Divided Optimization Algorithm (IDO) for feature optimization, they can explain why this specific algorithm was suitable for their research problem. They can discuss the advantages of IDO over other optimization algorithms, such as faster convergence, better exploration-exploitation trade-off, or robustness to noise. Authors can also highlight any prior studies or applications that have successfully employed IDO for similar tasks.

During the noisy input picture is smoothed down using a Gaussian filter. The high Gaussian value and the fact that the filter's effectiveness diminishes with increasing pixel distances both contributed to Gaussian's selection as the optimal filter. The resulting blur is more effective at protecting borders and boundaries than those produced by unchanging blurring filters. The Gaussian filter may also be used to make visual processes scale invariant, which is necessary when dealing with potentially varying sizes of picture data. This is because there is no guarantee that the object's distance from the acquisition technique will be constant, and hence the picture size will be consistent. Some of the key features of the Gaussian blur that make it suitable in our instance are: linearity invariance.

Consider a square mask with dimensions  $(2m + 1) \times (2m + 1)$ , a centre  $(0, 0)$ , and  $x, y$  ranges of  $(-m, m)$  to  $(m, m)$ . Eq. (1) provides the mask's component:

$$G_{\sigma}(x, y) = e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \tag{1}$$

where,  $\sigma$  represents the Gaussian distribution's standard deviation. We used  $Sum_{\sigma}$ , to perform normalisation in order to maintain the image's balance in terms of brightness level in Eq. (2) as in Eq. (3):

$$sum_{\sigma} = \sum_{x=-m}^m \sum_{y=-m}^m G_{\sigma}(x, y) \tag{2}$$

$$G_{k\sigma}(x, y) = \frac{G_{\sigma}(x, y)}{Sum_{\sigma}} \tag{3}$$

where,  $G_{k\sigma}$  is the normalised Gaussian filter used to create the output image with the Gaussian blur (G) as in Eq. (4).

$$G_b = G_{k\sigma} * v \tag{4}$$

where,  $n$  is the amount of noise current in the original image and is the convolution function. We do not utilise a greater value of  $s$  in this study since doing so generates additional artefacts and we want to image. The noisy input image is first processed using a Gaussian filter, and the resulting blurred image is sent into a clustering-based pre-classification step.

After applying a Gaussian filter, the picture is segmented into patches for use in following steps. For all of the patches, we calculate Hu's moment characteristics. For this research, we relied on Hu's 2, and we employed row vectors of size (1 7) to characterise each patch. Think about the NN picture and the nn patch centred on  $i$  ( $i = 1, 2, \dots, NN$ ). A 1 by 7 row vector is constructed to hold the calculated moment invariants and feature metrics ( $_1, _2, \dots, _7$ ) for each patch. Therefore, there are NN feature vectors for the entire image. The K-means clustering algorithm is fed these vectors as input, and it divides the N by N vectors into K clusters according to the goal function in Eq. (5).

$$\underset{c}{\operatorname{argmin}} \sum_{i=1,2,\dots,N \times N}^K H(gb(i)) \epsilon Hm_k |H(Gb(i)) - \mu_k|^2 \quad (5)$$

where,  $Gb(i)$  is a image cover with center  $i$ . The  $H(\cdot)$  stretches the input patch's vector, whereas  $\mu_k$  is the cluster,  $Hm_k$ . So, we find K clusters,  $\{Hm_1, Hm_2, \dots, Hm_k\}$ , each cluster labelled "Hm"  $_i$  with  $l_i$  feature vectors included. This means that the length  $l_i$  of each cluster varies. In most cases, a feature vector that has already been categorised may be expressed using k and l indices as  $Hm_{kl}$ , where the directories span:  $k = 1, \dots, K; l = 1, \dots, L$ . Here, k represents the various clusters, and l represents the various patches that form those clusters [21].

### 3.2 Noise removal using FCNN-IDOA

The FCNN-IDOA model is consistent with the standard CNN structure. Training a challenging undertaking that might take weeks or months. Therefore, rather of developing a brand new deep learning classifier from start, it is recommended to train the proposed deep learning approach on a pretrained classifier. We started with GoogLeNet because it was the best model in the 2014 ILSVRC ImageNet competition. There are a total of 144 layers in GoogLeNet, 22 of which are able to be

trained. These layers include: 9 inception layer modules, 2 convolution layers, 4 normalisation layers, and 1 fully connected layer. Six convolutional layers and a max-pooling layer were added in each inception module. The input layer of GoogLeNet has been upgraded to version 224 224 1. The GoogLeNet method. On the other hand, the ReLU activation function always used zero and disregarded any negative input. Leaky ReLU, on the other hand, is an improved variant of ReLU that converts all negative values to their positive counterparts.

The proposed FCNN-IDOA classifier lost its last five layers from GoogLeNet and gained fifteen new ones. In addition, the Leaky ReLU activation purpose was used in place of the ReLU activation purpose in the feature map layer to upsurge the suggested model's expressiveness and solve the dying ReLU problematic without modifying the fundamental convolution design. The final tally of layers augmented from 144 to 154 as a result of these changes.

The image size was instantly reduced because to the inclusion of a 7x7 pixel filter (patch) in the first convolution layer. The 2nd convolution layer was also 11, although it was shallower. A11 convolution block is the end result reduction. Moreover, the GoogLeNet inception module employs a wide range of convolution kernels, such as 11, 33, and 55, to extract features at varying scales, from the most fine-grained to the most fundamental. When using a wider convolution kernel, the features are computed over a more extensive region. The 11 convolution kernel is another example, providing more data with less processing time. One of the new features is the addition of four layers, each with a filter size of just one.

Additionally, the precision of the network's output was improved by the global average pooling layer. The suggested model's expressiveness was problem was overcome by replacing the features, the suggested hybrid model beat state-of-the-art pretrained models in terms of classification accuracy.

The epsilon value, size, number of filters, and filter names are described. The layers that make up the proposed hybrid FCNN-IDOA model are described in depth in Table 1.

**Table 1.** Characteristic of additional Layers in the projected hybrid perfect

S.No	Layer Name	Type	No of Filter	Filter Size	Epsilon
1	block_16_expand	Conv	960	1×1	
2	Block_16_expand_BN	Batch Norm			0.001
3	Block_16_enlarge_relu	Clipped ReLU Layer			
4	Block_16_depthwise	Grouped Convo	960	3×3	
5	Block_16_depthwise_BN	Batch Norm			0.001
6	Block_16_depthwise_relu	Clipped ReLU Layer			0.001
7	Block_16_project	Conv	320	1×1	
8	Block_16_scheme_BN	Batch Norm			0.001
9	Conv_1	Conv	1280	1×1	
10	Conv_1_bn	Batch Norm			0.001
11	Out_relu	Clipped ReLU Layer			
12	Global_regular_pooling2d_1	Global Average Pooling			
13	Logits	Fully Connected			
14	Logits_softmax	Softmax			
15	ClassificationLayer_Logits	Classification Layer			

#### 3.2.1 Image input layer

The suggested FCNN-IDOA model began with an image layer that contained the model's input, which in our case set the picture input size as 224 224 1. This value denotes the input image's size (1 for grayscale, 3 for colour). The input layer was read from to begin processing the photos.

#### 3.2.2 The convolution layers

To construct feature maps from input photos and recover

deep learning features, convolutional layers were utilised. In mathematics, two arguments are used. The dimensions of the filter in terms of height and breadth, as applied to the image matrix. Filter sizes of 77, 55, and 11 are used in the convolutional layers of our hybrid model, whereas the filter size of 33 is used in the max pooling layer. The input feature map is padded with padding name-value pairs by the convolutional layer. In Eq. (6), we see how to fold in discrete increments of time:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \quad (6)$$

where, W represents the kernel filter, x represents the input to the procedure, t represents the processing time, and s represents the output. When gathering input data in two dimensions, use Eq. (7).

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i, j) * K(i - m, j - n) \quad (7)$$

The i and j terms denote the regions of the objective matrix obligatory by the deep learning convolutional technique. The recommended technique for this step is to set the center of the filter to the primary position.

If cross-entropy is achieved with the proposed approach, Eq. (8) is used.

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) * K(m, n) \quad (8)$$

### 3.2.3 Activation function

Nonlinear transformation processes are frequently modelled using DL, and activation functions are a common tool for doing so. The Sigmoid, Tanh, and ReLU activation functions have proven to be the most popular and effective over the course of computer history. Since ReLU returns zero for all negative inputs, this effectively disables all negative inputs and leads to the dying ReLU problematic. A neuron is considered "dead" if it is permanently stranded on the other side of the network and produces zero as an output. To remedy the declining performance of ReLU, we replaced it in the feature map with leaky ReLU, an improved activation function for ReLU. The result of feeding a negative number into a leaky ReLU is not zero, but rather a little linear component of x. In the final 15 layers, we utilised a clipped ReLU activation function to execute a value below 0 were set to 0 and those above the ceiling were set to the ceiling we selected. In Eqs. (9)–(12), we can see the formulas for the activation functions:

ReLU:

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}, f'(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (9)$$

Sigmoid:

$$f(x) = \frac{1}{1+e^{-x}}, f'(x) = f(x)(1-f(x)) \quad (10)$$

Tanh:

$$\tanh(x) = \frac{2}{1+e^{-2x}} - 1, f'(x) = 1-f(x)^2 \quad (11)$$

Clipped ReLU:

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & 0 \leq x < ceiling \\ ceiling, & x \geq ceiling \end{cases} \quad (12)$$

Leaky ReLU:

$$f(x) = \begin{cases} x, & x \geq 0 \\ scale * x, & x < 0 \end{cases} \quad (13)$$

When given positive input, the leaky ReLU function returns x, while when given negative input, it returns a value equal to 0.01 times x, which is essentially meaningless. Therefore, no

neuron is inhibited, and we won't find any dead neurons.

### 3.2.4 Batch normalization layer

The outputs produced by the layers were normalised using the batch normalisation layer. The proposed FCNN-IDOA model's training time is shortened through normalisation, resulting in a quicker and more effective learning process. In Equations, the batch normalisation procedure is described (13)–(15):

$$Y_i = \frac{X_i - \mu\beta}{\sqrt{\sigma^2\beta + \epsilon}} \quad (14)$$

$$\sigma\beta = \frac{1}{M}(X_i - \mu\beta)^2 \quad (15)$$

$$\mu\beta = \frac{1}{M} \sum_{i=1}^M X_i \quad (16)$$

where, M is the total sum of input data,  $X_i = 1, \dots, M$ ,  $\mu\beta$  is the stack's regular value,  $\sigma\beta$  is the stack's  $Y_i$  represents the adjusted values after normalisation.

### 3.2.5 Pooling layer

After the convolution technique to reduce the size of the This could involve defining the term explicitly or providing additional information, such as the motivation behind using the term or how it relates to existing concepts in the field Authors should carefully review their explanations and seek feedback from peers or experts in the field. This external perspective can help identify areas that may be unclear or require further elaboration.

Feature map and remove unnecessary data) was used to streamline the info from the convolution layer. The two most popular pooling methods are average and maximal pooling. In the last 15 layers, we used global regular pooling. The network makes no learning during pooling. Three filters of size 3 x 3 were used for the pooling procedure. In Eq. (17), the pooling procedure is described.

$$S = w2 \times h2 \times d2 \quad (17)$$

$$w2 = \frac{(w1-f)}{A+1} \quad (18)$$

$$h2 = \frac{(h1-f)}{A+1} \quad (19)$$

$$d2 = d1 \quad (20)$$

where, w1 is the width of the MRI images, h1 is the height of image, d1 is the input MRI image size, f is the size of the filter, A is the amount of steps used, and S is the size of the created image.

### 3.2.6 Fully connected layer

The convolutional layers in the suggested model are layer. This is done by combining all of the features that the earlier layers had learned across various images. In order to classify the images, this layer chooses the most important patterns. Because there are three classes (meningioma, glioma, and pituitary) in the study, the output in the final completely linked layer is 3. Due to this, the projected FC layer's output value, which was obtained, is 3. For this, Eqs. (21) and (22) are employed:

$$U_i^l = \sum_j w_{ji}^{l-1} y_j^{l-1} \quad (21)$$

$$y_i^l = f(u_i^l) + b^{(l)} \quad (22)$$

where,  $l$  denotes the overall sum of layers,  $i$  and  $j$  denote the total sum of neurons,  $y_i^l$  denotes the value generated in the projected layer,  $w_{l-1ji}$  denotes the weight value of the hidden layer,  $y_{l-1i}$  denotes the value of neurons,  $u_i^l$  denotes the value of the layer, and  $b^{(l)}$  denotes the deviation value.

### 3.2.7 Softmax layer

The output of the fully linked layer is more uniformly produced thanks to the activation function. The network's probabilistic calculation is carried out by Softmax, which also generates work for each class in positive numbers. The given in Eq. (23):

$$P(y = j | x_i, W, b) = \frac{\exp^{x_i^T W_j}}{\sum_{j=1}^n \exp^{x_i^T W_j}} \quad (23)$$

where,  $A$ ,  $s$ ,  $W$ , and  $b$  are heaviness vectors.

### 3.2.8 Classification layer

The last layer of the projected designs is layer, which is applied to generate the each input. A probability distribution was returned by the Softmax activation function.

### 3.2.9 Training parameters

With the parameters listed in Table 2, we conducted experiments using IDOA (Section 3.2.10) methodology. In order to determine the best convergence for each CNN, we continuously tracked the development of training testing accuracy and deviation. There was an automated cutoff to training if there was no improvement in accuracy or a rise in error. The proposed FCNN-IDOA model was trained using stochastic gradient descent (SGD) with images. The suggested FCNN-IDOA model for brain tumour classification was trained on 120 epochs for optimal results.

**Table 2.** Limit values used in training systems

NAME	SGDM
MiniBatchSize	10
Sum of Epochs	120
Early Learning Rate	0.01
Shuffle	Every epoch
Validation Frequency	50

### 3.2.10 Feature optimization

For discriminative feature selection, it is recommended to employ the FCNN model, and its hyper-parameter tuning is optimised with the IDOA, a metaheuristic optimisation technique that static behaviour of dragonflies. In order to achieve a consistent degree of classification accuracy while decreasing the amount of features and redundant data, feature selection is viewed as a global combinatorial optimisation problem. This study yielded an enhanced optimisation technique called the Improved Dragonfly Optimisation Technique (IDOA). In the discrete search space, the chosen attributes of the dataset are ordered in every conceivable way. With such limited information, it may be possible to catalogue every combination of attributes. The enhanced dragonfly makes greater use of collective wisdom when making decisions, promoting diversity in the group and a healthy equilibrium between the exploratory and exploitative phases. This improves the algorithm's search efficiency. To select a subset of relevant features and leverage the strength of the

IDOA to improve classification results, hyper parameter adjustment is often more efficient, reduces overfitting, and eliminates redundant and noisy data. Depending on whether the player is actively attempting to evade an adversary or obtain food, the IDOA's two primary stages, exploitation and exploration, are modelled statically or dynamically, respectively. Cohesion, alignment, and separation are the three most common swarm behaviours. The IDOA expands on the original three behaviours by adding avoidance of danger and foraging for sustenance. These two actions are part of the IDOA to help the swarm live for a longer period of time. This approach takes into account two vectors: the initial position of dragonflies in a search space and the update step used to move them around. It is believed that the step vector also impacts the speed at which dragonflies fly. The position vector is revised after the step vector has been computed.

Both exploitative and exploratory behaviours are enabled by the IDOA's coefficient, adversary factor, and iteration number). Exploitation is characterised by high cohesiveness and low alignment, while exploration is characterised by low cohesion and high alignment. In order to improve randomization, probabilistic behaviour, and the identification of manufactured dragonflies, the standard DOA takes use of the Levy flying mechanism. Therefore, the DOA effectiveness is enhanced to a little degree by the Levy flight mechanism. However, the Levy flight mechanism cannot be used with the step size regulator. Agents must leave the search area if a significant distance is to be traversed. To get around these problems, the IDOA considers Brownian motion ( $P_g$ ) as a way to improve randomness, probabilistic behaviour, and the discovery of dragonflies. Eqs. (24) and (25) provide a mathematical determination of the Brownian motion ( $P_g$ )

$$P_g = \frac{1}{s\sqrt{2\pi}} \exp\left(-\frac{(\text{dimension}-\text{agents})^2}{2s^2}\right) \quad (24)$$

$$s = \sqrt{\frac{m_t}{m_s}}, \text{ and } m_s = 100 \times m_t \quad (25)$$

where,  $m=0.01$  denotes an agent's motion time and  $m$  the quantity of abrupt motions. The IDOA's parameter settings are as follows: The search domain is [0-1], there are five search agents, the extracted feature vectors are the dimension, and there are 20 iterations. 3476 feature vectors are chosen by the proposed IDOA and used as input values for classification by the DBN.

## 4. RESULTS AND DISCUSSIONS

### 4.1 Datasets

The experimental data was collected using the INbreast and CBIS-DDSM datasets [14]. INbreast data set: There were 120 cases (412 photos) in INbreast, 91 of which came from women with both breasts (four shots each) and 30 from those who had undergone a mastectomy (two photos each). Deformities were brought on by a variety of inflammatory lesions. The professional additionally provided us with the comprehensive plans in XML style. The CBIS-DDSM data show: One of the most widely used and comprehensive data sets, the breast data set, is organised into four distinct folders. The breast dataset is a sizable one that has been subdivided into normal, benign, malignant, and call-back benign subsets. One sample of various breast examinations is included in each folder. A total

of 1000 ultrasound breast images are used, 800 of which are used to train a logical-pool analysis. This paper proposes a process for efficiently removing local speckle noise from these images.

The RIBM-NLM procedure was tested on a personal computer (HP 15-dw, Hewlett-Packard company, Palo Alto, CA, USA) with a Core (TM) i3 processor and MATLAB (MATLAB 2017a, MathWorks, Natick, MA, USA) software.

To do this, we use a distance metric based on Euclidean geometry to evaluate how unlike two feature vectors are to one another. It's a 15-inch square patch. Clusters are formed, each one illuminating a different facet of the whole, as K (the number of clusters) increases. These tendencies also characterise the evolution of PSNR and MSE. If K is too large, however, the reconstructed image will be of worse quality since certain clusters will have insufficient candidates. As a result, both the PSNR and the MSE fall following the climax. Therefore, the best value of K may be selected according to the size of the input noisy image and the importance placed on simplicity. On the basis of this hypothesis, initial experiments to establish the optimum value of K for our method. The rate of change in both PSNR and MSE reaches a maximum at a specific value of K and then declines when K is increased further.

Reconstructed picture quality diminishes as K increases because of the emergence of clusters with inadequate numbers of candidates. As a result, the PSNR score drops rapidly after reaching a maximum. We can't pick the optimal value for K since PSNR is more crucial. However, K=675 is not optimum, while K=800 is the greatest attainable PSNR. K was calculated using an estimated number of subjects per cluster and an image size of 225 225 pixels. It takes 1.8 times as long to process the photographs on a computer when K is set to 800 as it does when K is set to 675. We'd like to have this done as quickly as possible, as that's one of our goals.

In this test, we employ a Gaussian blur with 10, 20, and 50 standard deviations. In order to facilitate meaningful comparisons between methods, we have chosen to keep the smoothing value h constant at 12. Using Eq. (1), we get the following when the block size is set to m=4.

In order to evaluate how well the projected strategy for reducing speckle compares to the alternatives, three quantitative measures are used. Mean squared error, ratio, and the index are three of them.

The SSIM formula is given by Eq. (26).

$$SSIM(x, y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (26)$$

In this case, x and y are two non-negative pictures that represent the original noisy image and the cleaned-up version, correspondingly. In the photos x and y, the average brightness is  $\mu_x$  and  $\mu_y$ . The covariance  $\sigma_{xy}$  is computed by taking the square root of the difference between the standard deviations of the x and y intensities. The variances of these intensities are denoted by  $\sigma_x^2$  and  $\sigma_y^2$ , respectively. Constants C1 and C2 were added to Eq. (29) to prevent the instability of division by zero in the factors. when  $\mu_x^2 + \mu_y^2 + \sigma_x^2 + \sigma_y^2$  are excessively near zero. A higher value of the SSIM standards, which range from zero to one, indicates a better de-noising effect.

The PSNR is distinct as in Eq. (27):

$$PSNR = 10 \log_{10} \left( \frac{L_D^2}{MSE} \right) \quad (27)$$

For this context, we will refer to MSE as the among the original and reconstructed pictures and LD as the magnitude of the intensity range's maximum and lowest values. The PSNR quantifies the range of signal-to-noise ratios present in a given picture. A greater PSNR indicates more effective noise cancellation. The definition of the (MSE) is given by Eq. (28). As MSE is reduced, image quality is enhanced.

$$MSE = \frac{\sum_{j=1}^{row} \sum_{i=1}^{column} (x(i, j) - y(i, j))^2}{Row \times Column} \quad (28)$$

For each image, the squared error, is calculated. To compare the suggested technique's computation speed to the other three approaches, processing time in seconds, or t(s), is also used. Furthermore, the statistical significance of the findings produced using the suggested technique associated to state-of-the-art methodologies is evaluated for the aforementioned three metrics using the t-test p-value pair-wise contrast approach. The results of the suggested model's validation are shown in Table 3.

**Table 3.** Investigation of proposed model on various noise ranges

	$\sigma=10$			
	PSNR	MSE	RMSE	t(s)
CI1	71.3869	0.003314	0.057567	81.003529
CI2	72.8369	0.00339	0.058223	80.160006
CI3	71.594	0.003678	0.060646	82.375192
CI4	72.7233	0.003046	0.055190	82.34266
CI5	72.177	0.003978	0.063071	80.703136
CI6	72.1992	0.003072	0.055425	82.167858
Av.	72.15288	0.003413	0.058354	81.4587301
$\sigma=20$				
CI1	66.9903	0.016732	0.129352	82.858238
CI2	66.0016	0.011358	0.106573	82.474946
CI3	65.2058	0.015101	0.122886	82.564148
CI4	66.9023	0.019146	0.138369	83.244738
CI5	65.7659	0.016782	0.129545	82.174985
CI6	66.7179	0.014479	0.120328	83.648587
Av.	66.263966	0.0155996	0.124509	82.827607
$\sigma = 50$				
CI1	55.146	0.282528	0.531533	83.405858
CI2	55.8126	0.269671	0.519298	84.678944
CI3	55.0384	0.236662	0.486479	83.601501
CI4	54.2856	0.228034	0.477529	84.899632
CI5	54.7928	0.20081	0.448118	81.204046
CI6	54.4104	0.237003	0.486829	84.764421
Av.	54.9143	0.2424513	0.491631	83.759067

In the above Table 3, analysis of proposed model on various noise ranges. In the noise range of  $\sigma=10$ , the CI1 reached the PSNR value of 71.3869 and the MSE value as 0.003314 and the RMSE value of 0.057567 and finally the t(s) value rate as 81.003529. Another, CI2 reached the PSNR value of 72.8369 and the MSE value as 0.00339, 0.058223 and finally the t(s) value rate as 80.160006. Then CI3 reached the PSNR value of 71.594, 0.003678 and the MSE value as 0.060646 and finally the t(s) value rate as 82.375192. Additionally, CI4 reached the PSNR value of 72.7233 and the MSE value as 0.003046 and the RMSE value of 0.055190 and finally the t(s) value rate as 82.34266. Then CI5 reached the PSNR value of 72.177 and the MSE value as 0.003978 and the MSE value as 0.063071 and finally the t(s) value rate as 80. 703136. After CI6 reached the PSNR value of 72.1992 and the MSE value as 0.003072 and the RMSE value of 0.055425 and finally the t(s) value rate

as 82.167858 respectively. And Av. reached the PSNR value of 72.15288 and the MSE value as 0.003413 and the RMSE value of 0.058354 and finally the t(s) value rate as 81.4587301 respectively.

After that the  $\sigma=10$ , the CI1 reached the PSNR value of 66.9903 and the RMSE value of 0.016732 and the RMSE value of 0.129352 finally the t(s) value rate as 82.858238 respectively. CI2 reached the PSNR value of 66.0016 and the MSE value as 0.011358 and the RMSE value of 0.106573 and finally the t(s) value rate as 82.474946. Additionally, CI3 reached the PSNR value of 65.2058 and the MSE value as 0.015101 and the RMSE value of 0.122886 and finally the t(s) value rate as 82.564148. After CI4 reached the PSNR value of 66.9023 and the MSE value as 0.019146 and the RMSE value of 0.138369 and finally the t(s) value rate as 83.244738. Then CI5 reached the PSNR value of 65.7659 and the MSE value as 0.016782 and the RMSE value of 0.129545 and finally the t(s) value rate as 82.174985 respectively. CI6 reached the PSNR value of 66.7179 and the MSE value as 0.014479 and the RMSE value of 0.120328 and finally the t(s) value rate as 83.648587. Another, Av. reached the PSNR value of 66.263966 and the MSE value as 0.0155996 and the RMSE value of 0.124509 and finally the t(s) value rate as 82.827607 respectively.

And, finally the  $\sigma=50$ , the CI1 reached the PSNR value of 66 55.146 and the RMSE value of 0.282528 and the RMSE value of 0.531533 and finally the t(s) value rate as 83.405858. After CI2 reached the PSNR value of 6655.8126 and the RMSE value of 0.269671 and the RMSE value of 0.519298 and finally the t(s) value rate as 84.678944. Then CI3 reached the PSNR value of 6655.0384 and the RMSE value of 0.236662, 0.486479 and finally the t(s) value rate as 83.601501. correspondingly. CI4 reached the PSNR value of 6654.2856 and the RMSE value of 0.228034 and the RMSE value of 0.477529 and finally the t(s) value rate as 84.899632 respectively. CI5 reached the PSNR value of 6654.7928 and the RMSE value of 0.20081 and the RMSE value of 0.448118 and finally the t(s) value rate as 81.204046 respectively. CI6 reached the PSNR value of 6654.4104 and the RMSE value of 0.237003 and the RMSE value as 0.486829 and finally the t(s) value rate as 84.764421. Another Av. reached the PSNR value of 66 and the MSE value of 54.9143 and the t(s) value of 0.2424513 and the RMSE value of 0.491631 and finally the t(s) value rate as 83.759067 respectively. We can discuss the significance of the obtained p-values and what they indicate about the differences or relationships between the compared conditions or groups. Authors can also relate the t-test results to their research objectives or hypotheses and discuss any unexpected or interesting findings.

## 5. CONCLUSION AND FUTURE WORK

The purpose of this research was to investigate the efficacy of employing novel hybrid models and a variety of convolution neural networks to eliminate noise in ultrasound pictures of BC. The GoogleNet framework served as the basis for the proposed FCNN-IDOA framework. Fifteen new, deeply nested layers were added to GoogleNet in place of the final five layers that were lost. Convolution neural networks with the function were modified to use the leaky function without compromising the original architecture. After the adjustments, there were 154 layers instead of 144. The PSNR of 77% was attained by the proposed hybrid model, which was

a record. The experimental results validated the superior classification ability of the proposed hybrid model for brain tumours. The suggested method also calculated additional descriptive and accurate features for noise removal, leading to high precision. Av. reached the PSNR value of 66 and the MSE value of 54.9143 and the t(s) value of 0.2424513 and the RMSE value of 0.491631 and finally the t(s) value rate as 83.759067 respectively.

Furthermore, testing shows that the greatest results were obtained from the FCNN model that made use of optimisation techniques. However, the hybrid framework achieved the highest accuracy when associated to the other models. Experimental results show that the suggested method is effective in reducing local speckle noise in ultrasound breast pictures while also preserving edge information and highlighting image characteristics, laying the framework for processing and use. The planned research might not look at the various forms of noise or other ultrasonic image elements. We promise that in our future studies, we will take a closer look at a wide range of photographs.

## REFERENCES

- [1] Wang, X., Ahmad, I., Javeed, D., Zaidi, S.A., Alotaibi, F.M., Ghoneim, M.E., Eldin, E.T. (2022). Intelligent hybrid deep learning model for breast cancer detection. *Electronics*, 11(17): 2767. <https://doi.org/10.3390/electronics11172767>
- [2] Macherla, H., Kotapati, G., Sunitha, M.T., Chittipireddy, K.R., Attuluri, B., Vatambeti, R. (2023). Deep learning framework-based chaotic hunger games search optimization algorithm for prediction of air quality index. *Ingénierie des Systèmes d'Information*, 28(2): 433-441. <https://doi.org/10.18280/isi.280219>
- [3] Bai, X., Zhang, D., Shi, S., Yao, W., Guo, Z., Sun, J. (2023). A fractional-order telegraph diffusion model for restoring texture images with multiplicative noise. *Fractal and Fractional*, 7(1): 64. <https://doi.org/10.3390/fractalfract7010064>
- [4] Xie, S., Song, J., Hu, Y., Zhang, C., Zhang, S. (2023). Using CNN with multi-level information fusion for image denoising. *Electronics*, 12(9): 2146. <https://doi.org/10.3390/electronics12092146>
- [5] Sai Lakshmi Inamanamelluri, H.V., Pulipati, V.R., Pradhan, N.C., Chintamaneni, P., Manur, M., Vatambeti, R. (2023). Classification of a New-born infant's jaundice symptoms using a binary spring search algorithm with machine learning. *Revue d'Intelligence Artificielle*, 37(2): 257-265. <https://doi.org/10.18280/ria.370202>
- [6] Kotapati, G., Ali, M.A., Vatambeti, R. (2023). Deep learning-enhanced hybrid fruit fly optimization for intelligent traffic control in smart urban communities. *Mechatronics and Intelligent Transportation Systems*, 2(2): 89-101. <https://doi.org/10.56578/mits020204>
- [7] Wang, S., Huang, T.Z., Zhao, X.L., Mei, J.J., Huang, J. (2018). Speckle noise removal in ultrasound images by first-and second-order total variation. *Numerical Algorithms*, 78: 513-533. <https://doi.org/10.1007/s11075-017-0386-x>
- [8] Dass, R. (2018). Speckle noise reduction of ultrasound images using BFO cascaded with wiener filter and discrete wavelet transform in homomorphic region. *Procedia Computer Science*, 132: 1543-1551.



- <https://doi.org/10.1016/j.procs.2018.05.118>
- [9] Feng, X., Huang, Q., Li, X. (2020). Ultrasound image de-speckling by a hybrid deep network with transferred filtering and structural prior. *Neurocomputing*, 414: 346-355. <https://doi.org/10.1016/j.neucom.2020.09.002>
- [10] Ayana, G., Dese, K., Raj, H., Krishnamoorthy, J., Kwa, T. (2022). De-speckling breast cancer ultrasound images using a rotationally invariant block matching based non-local means (RIBM-NLM) method. *Diagnostics*, 12(4): 862. <https://doi.org/10.3390/diagnostics12040862>
- [11] Chen, S., Hou, J., Zhang, H., Da, B. (2014). De-speckling method based on non-local means and coefficient variation of SAR image. *Electronics Letters*, 50(18): 1314-1316. <https://doi.org/10.1049/el.2014.0630>
- [12] Shahrezaei, I.H., Kim, H.C. (2019). Resolutional analysis of multiplicative high-frequency speckle noise based on SAR spatial de-speckling filter implementation and selection. *Remote Sensing*, 11(9): 1041. <https://doi.org/10.3390/rs11091041>
- [13] Jain, A., Rajpal, N., Mehta, R. (2022). De-speckling techniques for T1 weighted brain MRI images-a statistical comparison. *International Journal of Performability Engineering*, 18(2): 108-116. <https://doi.org/10.23940/ijpe.22.02.p5.108116>
- [14] Vimala, B.B., Srinivasan, S., Mathivanan, S.K., Muthukumaran, V., Babu, J.C., Herencsar, N., Vilcekova, L. (2023). Image noise removal in ultrasound breast images based on hybrid deep learning technique. *Sensors*, 23(3): 1167. <https://doi.org/10.3390/s23031167>
- [15] Huang, Z., Zhu, Z., Wang, Z., Li, X., Xu, B., Zhang, Y., Fang, H. (2023). D3CNNs: Dual denoiser driven convolutional neural networks for mixed noise removal in remotely sensed images. *Remote Sensing*, 15(2): 443. <https://doi.org/10.3390/rs15020443>
- [16] Li, Z., Liu, H., Cheng, L., Jia, X. (2023). Image denoising algorithm based on gradient domain guided filtering and NSST. *IEEE Access*, 11: 11923-11933. <https://doi.org/10.1109/ACCESS.2023.3242050>
- [17] Zhang, D., Zhou, F. (2023). Self-supervised image denoising for real-world images with context-aware transformer. *IEEE Access*, 11: 14340-14349. <https://doi.org/10.1109/ACCESS.2023.3243829>
- [18] Hu, Y., Xu, S., Cheng, X., Zhou, C., Hu, Y. (2023). A triple deep image prior model for image denoising based on mixed priors and noise learning. *Applied Sciences*, 13(9): 5265. <https://doi.org/10.3390/app13095265>
- [19] Xu, J., Yuan, M., Yan, D.M., Wu, T. (2023). Deep unfolding multi-scale regularizer network for image denoising. *Computational Visual Media*, 9(2): 335-350. <https://doi.org/10.1007/s41095-022-0277-5>
- [20] Yan, R., Liu, Y., Liu, Y., Wang, L., Zhao, R., Bai, Y., Gui, Z. (2023). Image denoising for low-dose CT via convolutional dictionary learning and neural network. *IEEE Transactions on Computational Imaging*, 9: 83-93. <https://doi.org/10.1109/TCI.2023.3241546>
- [21] Vatambeti, R., Mamidiseti, G. (2023). Routing attack detection using ensemble deep learning model for IIoT. *Information Dynamics and Applications*, 2(1): 31-41. <https://doi.org/10.56578/ida020104>