(Online) ISSN 2712-0562
Original Research

Sustainable Engineering and Innovation
Vol. 8, No. 2, 2026, pp.351-374
https://doi.org/10.37868/sei.v8i2.id1100

A dual-branch frequency-spatial network with edge-aware loss for perceptually oriented image denoising

H. I. Wahhab[1], Haroon Rashid[2,3], Lateef Abd Zaid Qudr[4], Karam S. Khalid[5], A. D. Radhi[6], Noraliza Binti Azizan[7], Ravi Sekhar[8*], Fatimah Bibi Hamzah[9*]

[1] College of Computer Science and Information Technology, University of Kerbala, Kerbala, Iraq
[2] Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Selangor, Malaysia
[3] Centre for Fundamental Research, Xpertopedia Academy, Kuala Lumpur, Malaysia
[4] Department of Computer and Techniques Engineering, AlSafwa University College, Karbala, Iraq
[5] Technical Engineering College, Al-Farahidi University, Baghdad, Iraq
[6] College of Pharmacy, University of Al-Ameed, Karbala, Iraq
[7,9] Universiti Poly-Tech Malaysia, Kuala Lumpur, Malaysia
[8] Symbiosis Institute of Technology (SIT), Pune Campus, Symbiosis International (Deemed University) (SIU), Pune, India

*Corresponding author E-mail: bibi@uptm.edu.my; ravi.sekhar@sitpune.edu.in

Received Aug. 17, 2026
Revised Oct. 19, 2026
Accepted Oct. 21, 2026
Online Nov. 23, 2026

Abstract

Image denoising is a basic issue in low-level vision and seeks to reconstruct clean images from noisy observations. This process becomes especially difficult when the fine details and sharp edges must be kept during noise removal. Classical and deep learning-based techniques tend to be over-smoothed, whereas frequency domain techniques cause spatial errors and ringing artifacts. This paper presents a hybrid frequency-spatial domain network for edge-preserving image denoising and detail enhancement. The proposed architecture will consist of two parallel branches: a frequency domain branch, which will use wavelet decomposition to suppress multi-resolution noise, and a spatial domain branch, which will extract local features to preserve the detail. A cross-domain feature fusion module is created to combine complementary information from both domains intelligently. AWGN was used to conduct tests on common benchmark datasets. The proposed approach achieves an average PSNR of 28.19 dB and an SSIM of 0.890 across the full benchmark test sets; although pixel-level reconstruction accuracy is lower than that of several PSNR-focused baseline methods, the method demonstrates superior structural preservation. For 256 x 256 images, this method achieves an average inference time of 210ms and an LPIPS value of 0.188, requiring 42.6 GFLOPs. These results demonstrate that the proposed framework prioritizes structural and perceptual integrity over PSNR.

© The Author 2026.
Published by ARDA.

Keywords: Cross-domain fusion, Detail enhancement, Edge preservation, Frequency domain, Image denoising

1. Introduction

Image noise is a common artifact that is bound to corrupt visual data at any point of acquisition, transmission, or storage. Image denoising is, therefore, considered a paradigmatic and enduring ill-posed inverse task in low-

This work is licensed under a Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/) that allows others to share and adapt the material for any purpose (even commercially), in any medium with an acknowledgement of the work's authorship and initial publication in this journal.

level vision, in which the task is to obtain a clean image by observing a noisy one. The challenge inherent in this lies in the fact that it may be possible to have multiple clean images that are associated with the same noisy input, and thus the solution is non-unique and heavily relies on prior knowledge [1]. The severe requirement for efficient denoising is experienced in a broad range of uses, such as consumer photography, medical imaging, and remote sensing. The preservation of edge data and the refinement of small, textural features are not simply matters of aesthetics in these realms but are of utmost importance in correct diagnosis, analysis, and interpretation [2].

Any denoising algorithm that balances these structural components will jeopardize the usefulness of the image in future activities [3]. In the last decades, numerous denoising algorithms have been created. Traditional algorithms, i.e., BM3D and WNNM, have backbreaking performance through non-local self-similarity and sparse representation [4]. Nevertheless, these methods can be time-consuming to compute, and their priors can be ad hoc and fail to extend to all types of images and all types of noise distributions [5]. The introduction of deep learning has already resulted in major improvements, and spatial domain convolutional neural networks (CNNs), such as DnCNN and FFDNet, have become the new reference points [6]. Such purely spatial techniques are often susceptible to over-smoothing, although extremely effective.

When they work directly on the raw pixels, they may not be able to tell the difference between noise and actual high-frequency image detail, e.g., sharp edges and fine textures, and the result will be a visible loss of detail, particularly with increased noise [7]. To address this, frequency domain methods, especially methods that use the wavelet transform, have been investigated. Such techniques are more suited to isolate signal and noise by operating on different frequency sub-bands separately. However, they may not be sensitive to exact spatial positions and are prone to adding visual artifacts, including ringing or Gibbs effects, at the image reconstruction stage. Therefore, there is still a severe gap in a technique that would maintain edges and details in a solid manner and still be able to remove noise [8].

The fundamental issue that this work is dealing with is how to design a network architecture that is very robust in terms of separating noise and high-frequency image content, like edges and textures, to obtain not only efficient noise reduction but also detail enhancement at the same time [9].

In order to address this issue, a hybrid network architecture is suggested, which utilizes complementary capabilities of the frequency and spatial domains. Within this context, the frequency domain is employed to offer a global, multi-resolution view of the image, and such an approach allows efficient signal-noise separation and the removal of noise components. At the same time, the spatial domain is used to record local contextual data that are critical in conserving and developing fine details and high edges. The characteristics of the two domains are then intelligently combined to generate a perfect output that is clean and structurally rich.

The key contributions of the paper are as follows:

- A new hybrid bifurcated architecture: It proposes a deep network that expressly comprises and acts on information in parallel in terms of a frequency domain branch and a spatial domain branch to jointly learn edge-preserving denoising and detail boosting.
- A cross-domain feature fusion module (CDFFM): It is a special fusion module, which is an intelligent combination of the globally aware frequency features and the locally dense spatial features. This module ensures that the complementary data of both domains are well integrated to counter the weaknesses of each direction.
- A perceptual and edge-preserving loss function: A composite loss is designed that is a combination of a traditional L1 loss that is pixel-wise loss and a perceptual loss (based on a pre-trained VGG network) and a novel gradient-based loss. This loss specifically directs the network to focus on the reconstruction of perceptually appealing textures and sharp edge preservation.
- Perceptual execution: Extensive experiments demonstrate that the proposed model prioritizes edge preservation and structural fidelity over achieving the highest possible pixel-level reconstruction

accuracy. Under the assessment protocol using the large dataset, it achieves a competitive average SSIM score. This highlights the rewards of the spatial-frequency architecture in noise reduction and note preservation, contrasting with recent PSNR-oriented frameworks like NAFNet and DiffIR.

The rest of this paper will be organized in the following way: Section 2 discusses the existing literature on the field of spatial domain denoising, frequency domain processing, and edge-preserving methodologies; Section 3 gives a detailed description of the proposed hybrid network architecture consisting of the dual branches and the fusion module. Section 4 provides the details of the experimental setup, datasets, implementation details, and evaluation metrics. Section 5 gives critical coverage of the experimental findings, comparisons with state-of-the-art procedures, and ablation results. Lastly, in Section 6, the paper concludes and addresses the possibilities of future work directions.

2. Related work

2.1. Spatial domain denoising

The spatial domain denoising algorithms specifically work on the image pixels by exploiting the local or non-local correlation. Conventional methods encompass bilateral filtering, which conserves edges by integrating domain and range filtering through a weighted average of the neighboring pixels in both spatial and intensity terms. Non-local means filtering adds to this idea and seeks similar patches throughout the image and averages them based on how similar they are, hence preserving repetitive patterns in the image more effectively. Although these techniques are computationally very efficient and conceptually very simple, they can still have difficulty with complex noise patterns, and they tend to produce artifacts in cases where similar patches are falsely matched [10].

Convolutional neural networks have transformed spatial-domain denoising using deep learning. DnCNN uses residual learning and batch normalization to estimate the residual noise, with high performance attained by a higher depth of the network. FFDNet builds upon this ability by adding a noise level map as input to allow single simulations to look at different noise intensities. CBDNet deals with the noise present in the real world through signal-dependent noise modeling and asymmetric loss functions, which help in preserving details more effectively [11]. SwinIR and Restormer are more recent as they use transformer architectures and window switching and multi-head attention to focus on long-range dependencies and rich context information. Such deep learning techniques have shown impressive noise suppression properties, yet they too are vulnerable to over-smoothing, especially at higher noise levels [12]. These networks are unable to explicitly view the image as a global frequency image, and do not tend to tell the difference between noise or a significant high-frequency image, such as sharp edges and fine textures, hence them denoised result has visible detail loss [13].

2.2. Frequency domain denoising

Frequency domain methods represent a transformation of images into other forms in which signal and noise are easier to separate. The classical techniques make use of the Fourier transform to transform the images into the frequency spectrum, with the noise elements in the high-frequency band being reduced with the help of low-pass filtering. The wavelet transform techniques break the images down into multi-resolution sub-bands, that is, approximation (low-frequency) and detail (high-frequency) bands [14]. Coefficient shrinkage is used to perform denoising, in which thresholding is used on the sub-bands that are believed to consist primarily of noise. Although these conventional approaches provide theoretical [15].

Hybrid approaches have been shown to be promising when deep learning is combined with frequency domain processing. The discrete wavelet transforms are implemented as differentiable CNNs, allowing them to learn frequency-aware representations end-to-end. These networks divide the input images into sub-bands, process the sub-bands using specific convolutional modules, and rebuild denoised images using inverse wavelet transforms. Fourier domain networks are also based on spectral representations and are learned to alter spectral frequency coefficients. These frequency-domain deep learning approaches are shown to have superior signal-

noise separation ability through the explicit representation of multi-scale frequency responses. They, however, have significant weaknesses, such as being insensitive to fine spatial positions and prone to reconstruction artifacts such as Gibbs phenomena. The trade-off between frequency localization and spatial resolution is an inherent and major problem since aggressive frequency manipulation can usually degrade spatial fidelity [16].

2.3. Edge-preserving and enhancement techniques

Techniques that are explicitly tailored to space conservation take advantage of the inherent conflict between noise reduction and structure maintenance. Total variation regularization is a denoising optimization problem that aims to minimize the gradient of the image so that it smooths out homogeneous portions of the image, but leaves sharp edges [17]. Guided filtering uses a guidance image to compute the local linear models that allow the linear computational complexity of edge-aware filtering. These optimization methods give good theoretical justifications for edge preservation but can be very sensitive to parameter choice and can be less likely to be generalized to a variety of image materials [18].

Image enhancement and super-resolution deep learning techniques have similar objectives to edge-preserving denoising [19]. The loss of perceptual features calculated with the pre-trained VGG networks is applied to high-level feature representations instead of pixel values, which forces the network to create perceptually pleasing textures. Generative adversarial networks propose adversarial training in which a discriminator network is able to differentiate between real and generated images, causing the generator to drive towards generating realistic details. These methods are excellent at hallucinating fine structures and run the risk of adding artificial textures that do not exist in the original scene. The difficulty is to find a balance between detail enhancement and faithful reconstruction, especially when used on denoising tasks whereby the underlying truth detail is in some measure blurred by noise [20].

2.4. Hybrid domain methods

The most modern studies have touched on the issue of the synergistic integration of multiple domains to address the limitations of single-domain procedures. Wavelet-domain CNNs are among the pioneering attempts in that regard, in which wavelet decomposition is followed by convolutional processing to provide multi-scale frequency sensitivity. More elaborate models have also been developed using a mixture of Fourier preprocessing modules with spatial transformers, which allows an adaptive combination of a frequency-domain noise prior and spatial-domain texture information [21]. Cross-domain attention mechanisms have been recommended to make dynamic contributions of various domains with local image features. In the case of hyperspectral image denoising, multi-dimensional noise coupling has been overcome by hybrid frameworks, which combine spatial, spectral, and frequency processing.

These hybrid strategies confirm the possibility of domain integration but tend to adopt rather superficial fusion strategies. The current effort along this line of thought suggests a richer dual-branch design with a special-purpose cross-domain feature fusion unit that explicitly captures the complementary nature of frequency and spatial representations that can be effectively integrated compared to current concatenation-based or attention-based models, as shown in Table 1 [22, 23].

Table 1. Comparative summary of image denoising methods

Method Category	Representative Techniques	Strengths	Limitations	Performance Indicators (PSNR/SSIM)
Traditional Spatial	Bilateral Filter, NLM, BM3D	Interpretable, no training required; BM3D exploits non-local self-similarity effectively	High computational cost; hand-crafted priors; limited generalization	BM3D: 34.51 / 0.8507 (DND dataset)

Method Category	Representative Techniques	Strengths	Limitations	Performance Indicators (PSNR/SSIM)
Deep Learning Spatial	DnCNN, FFDNet, CBDNet, SwinIR, Restormer	Superior noise suppression; end-to-end learning; captures complex patterns	Prone to over-smoothing; detail loss at high noise; lacks global frequency perspective.	DnCNN: 38.56 / 0.910 (SIDD); SwinIR: achieves SOTA in restoration tasks
Traditional Frequency	Fourier shrinkage, Wavelet thresholding	Multi-resolution analysis; theoretical noise-signal separation	Basis function selection critical; threshold tuning; spatial inaccuracies; ringing artifacts	Wavelet methods: sensitive to parameter choices; limited quantitative benchmarks
Deep Learning Frequency	DWT-CNN, WT-DnCNN, Fourier networks	Explicit frequency modeling; effective signal-noise separation	Spatial insensitivity; reconstruction artifacts (Gibbs); increased computational load	WT-DnCNN (MRI): 36-43 / 0.93-0.98; WT-DnCNN (CT): 38-43 / 0.95-0.98
Edge-Preserving	TV regularization, Guided filter, Perceptual loss, GANs	Explicit edge/structure preservation; perceptually pleasing results	Parameter sensitive; risk of artificial texture generation; balancing fidelity vs. enhancement	Guided filter: O(N) complexity; Perceptual loss: improves LPIPS but may reduce PSNR
Hybrid Domain	Wavelet-CNN, FFT+Transformer, Cross-domain attention	Complementary strength integration; addresses multi-dimensional noise	Often shallow fusion; computational complexity; domain alignment challenges	HDST (HSI): +0.94 dB PSNR over baseline; cross-domain attention shows 0.52-0.94 dB gains

3. Proposed methodology

3.1. Overall architecture

The hybrid frequency-spatial domain network proposed is a dual-branch model that has noisy images going through parallel frequency and spatial streams before finally combining the complementary features to produce the final reconstruction. The general flow of data is shown in Figure 1, where a noisy input image $I_{noisy} \in \mathbb{R}^{H\times W\times C}$ is fed into the two branches simultaneously. The frequency domain branch breaks down the image with the wavelet transform and works on each of the sub-bands to reduce the noise without distorting the structures. Simultaneously, the spatial domain branch obtains multi-scale features to elicit local details, edges, and textures. The results of the two branches are then combined by a Cross-Domain Feature Fusion Module (CDFFM) that combines the complementary representations intelligently. Lastly, the reconstruction module uses an inverse wavelet transform and spatial refining in order to generate the denoised and enhanced output image. $I_{clean} \in \mathbb{R}^{H\times W\times C}$. This architecture will guarantee that the encryption of frequency properties and local spatial details are collectively optimized to enhance edge-preserving denoising and detail amplification.

3.2. Frequency domain branch: wavelet-guided denoising

3.2.1. Wavelet decomposition

The frequency domain branch starts with a discrete wavelet transform (DWT) in order to break down the input image into several frequency bands. The Haar wavelet has been chosen due to its simplicity, orthogonality, and computational efficiency, whereas Daubechies wavelets can be used in applications where smoother basis functions are needed. Denoting an input image I_{noisy}, the 2D DWT breaks down this image into four sub-bands by using a wavelet filter convolution and downsampling as Eq. (1):

$$LL,LH,HL,HH = DWT(I_{noisy}) \quad (1)$$

in which LL is the approximation sub-band with the low-frequency structural detail, and LH, HL, HH are the detail sub-bands of the horizontal, vertical, and diagonal, high-frequency detail, respectively. The sub-bands are $\frac{H}{2} \times \frac{W}{2} \times C$ dimension, which gives a multi-resolution representation in which the main image structure is represented in the LL sub-band and the directional edges, textures, and noise are represented in the detail sub-bands. This breakdown allows a clear signal and noise separation with respect to frequency properties.

3.2.2. Frequency sub-band processing network

After the decomposition process, each frequency sub-band is then subjected to a specific sub-network that focuses on learning adaptive filters to reduce the noise but not significant signal features. A similar CNN structure with distinct batch normalization layers is used to give equal weight to sub-bands so as to strike the balance between the efficiency of the parameters and adaptation of the band. The processing of every sub-band $S \in$ {LL,LH,HL,HH} is defined as Eq. (2):

$$\hat{S} = f_{\theta_S}(S) + S \quad (2)$$

where f_{θ_S} is a residual CNN block whose parameters θ_S are sub-band specific to S. It consists of a series of convolutional layers that have residual links in between to enable deeper architecture training. In the case of the LL sub-band, this is aimed at preserving the structural integrity and eliminating any remaining noise, which is done by means of conservative filtering using small receptive fields. In the case of the detail sub-bands (LH, HL, HH), more aggressive noise reduction is implemented, but without directional edge information being maintained.

The channel attention mechanisms are also added to provide a weighted, relevant weighting of important frequency components according to their relevance. The attention module is calculated as in Eq. (3):

$$\text{Attn}(X) = X \odot \sigma(\text{Conv}_{1\times1}(\text{GAP}(X))) \quad (3)$$

where $\odot$ is the element-wise multiplication, σ is the sigmoid activation function, and GAP is global average pooling. This enables the network to underline informative frequency channels and dampen noise-dominant channels. These processed sub-bands $\hat{S}$ are then transmitted to the reconstruction module.

3.3. Spatial domain branch: Detail and edge preservation

3.3.1. Feature extraction backbone

The branch of the spatial domain utilizes a U-Net structure whose backbone is the feature extraction residual block since it has been demonstrated to exploit the multi-scale spatial features and retain finer details by using skip connections. The backbone performs the encoding/decoding of the noisy input. I_{noisy} as an encoder-decoder architecture that has successive downsampling and upsampling steps. The encoder comprises L stages with two residual blocks, two stride-2 convolution downsampling as Eq. (4):

$$E_l = \text{ResBlock}(\text{ResBlock}(\text{Downsample}(E_{l-1})) \quad (4)$$

where $E_0 = \text{Conv}_{3\times3}(I_{noisy})$ represents the initial feature map. Each residual block is defined as Eq. (5):

$$\begin{aligned}&\text{ResBlock}(X)\\&= X\\&+ \text{Conv}_{3\times3}(\text{ReLU}(\text{Conv}_{3\times3}(\text{ReLU}(X))))\end{aligned} \quad (5)$$

This architecture makes it possible to extract hierarchical features that span low-level edges up to high-level semantic structures, which are important in the preservation of fine details in the denoising process.

3.3.2. Multi-scale context aggregation

Multi-scale information: Multi-scale detailing is merged using the symmetric decoder of the U-Net with skip links that combine the encoder details of matching resolution. $\boldsymbol{D_l}$ is calculated and is given by Eq. (6):

$$\begin{aligned}D_l = \text{ResBlock}(\text{ResBlock}(\text{Upsample}(D_{l-1})\\ \oplus E_{L-l}))\end{aligned} \quad (6)$$

The channel dimension concatenation is where $\oplus$is the concatenation, and upsample is bilinear upsampling followed by convolution. The architecture is such that the fine-grained spatial information contained in the initial encoder layers is directly transferred to the decoder, which retains edge sharpness and textures. The last spatial feature $F_{spatial} \in \mathbb{R}^{H\times W\times d_s}$ is derived by having a 1x1 convolution project the decoder output into a fixed-dimensional space, which is conducive to fusion.

3.4. Cross-domain feature fusion module (CDFFM)

3.4.1. Motivation

Frequent straightforward concatenation of frequency and spatial characteristics is not optimal because the two domains have entirely different statistical characteristics and representations. Frequency features represent multi-resolution organization of global spectral properties, and spatial features represent local pixel-based patterns and relationships with their surroundings. Direct concatenation does not model the complementary interactions between various features, so features may be redundant and may not be integrated in an optimal manner. The CDFFM is thus aimed at modelling explicitly the cross-domain interactions and selectively merging complementary information.

3.4.2. Module design

Step A: Feature alignment and resizing. The frequency F_{freq} provided by processed sub-bands and spatial features $F_{spatial}$ are required to be spatially aligned first. Frequency features are obtained as a result of wavelet sub-bands of lower resolution, so they are reconstructed to the original spatial dimensions as in Eq. (7):

$$F_{freq}^{up} = \text{Conv}_{3\times3}(\text{Upsample}(F_{freq})) \quad (7)$$

Step B: Feature transformation. Both the feature maps are then mapped to some shared embedding space through 1x1 convolutions to make sure that the dimensions are compatible and allow interplay across domains as in Eq. (8):

$$\begin{aligned}\hat{F}_{freq} = \text{Conv}_{1\times1}(F_{freq}^{up}), \hat{F}_{spatial}\\ = \text{Conv}_{1\times1}(F_{spatial})\end{aligned} \quad (8)$$

Step C: Fusion strategy. There are three other fusion strategies, which are the ones under consideration and evaluation:

Option 1 (Attention-based fusion): Attention is calculated such that the features belonging to one domain create attention maps on the other one. The spatial attention, which is based on frequency features, is computed as Eq. (9) & Eq. (10):

$$A_{spatial} = \sigma(\text{Conv}_{1\times1}(\hat{F}_{freq})) \quad (9)$$

$$F_{freq\to spatial} = \hat{F}_{spatial} \odot A_{spatial} \quad (10)$$

On the same note, the channels of attention are based on a spatial feature weight frequency channel as in Eq. (11) & Eq. (12):

$$A_{channel} = \sigma(\text{Conv}_{1\times1}(\text{GAP}(\hat{F}_{spatial}))) \quad (11)$$

$$F_{spatial\to freq} = \hat{F}_{freq} \odot A_{channel} \quad (12)$$

The features that have been attended are then added together element-wise as in Eq. (13):

$$F_{fused} = F_{freq\to spatial} + F_{spatial\to freq} \quad (13)$$

Option 2 (Bilinear pooling): Bilinear pooling models pairwise interactions between features and represents second-order domain-wise statistics as in Eq. (14):

$$F_{fused} = \text{Conv}_{1\times1}(\text{Reshape}(\hat{F}_{freq}^{T}\hat{F}_{spatial})) \quad (14)$$

Option 3 (Concatenation with refinement): Simple concatenation followed by refinement through a squeeze-and-excitation-enhanced CNN block as in Eq. (15) & Eq. (16):

$$F_{concat} = [\hat{F}_{freq}; \hat{F}_{spatial}] \quad (15)$$

$$F_{fused} = \text{SEBlock}(\text{ResBlock}(\text{ResBlock}(F_{concat}))) \quad (16)$$

where SEBlock applies squeeze-and-excitation for channel-wise recalibration as Eq. (17):

$$\text{SEBlock}(X) = X \cdot \sigma(\text{FC}(\text{GAP}(X))) \quad (17)$$

Step D: Output. In the fused feature map $F_{fused} \in \mathbb{R}^{H\times W\times d_f}$ the complementary information in both domains is contained, and the interaction between the domains is clearly modelled to be more restrictive for edge and detail reconstruction.

3.5. Reconstruction module

3.5.1. Inverse wavelet transform

The frequency-branch-processed frequency-band S -sub bands are reconstructed into the spatial domain using the inverse discrete wavelet transform (IWT). The reconstruction of the frequency-refined image through the IWT takes the form of upsampling and convolving the image with reconstruction filters as in Eq. (18):

$$I_{freq} = \text{IWT}(\hat{S}_{\text{LL}}, \hat{S}_{\text{LH}}, \hat{S}_{\text{HL}}, \hat{S}_{\text{HH}}) \quad (18)$$

In this reconstruction, the multi-resolution processing advantages are maintained, and then the result is re-integrated into the spatial domain alongside the spatial features.

3.5.2. Spatial refinement layer

The last reconstruction is a mixture of the frequency-refined picture I_{freq} and the synthesized features F_{fused} of the CDFFM. This combined representation is then processed by a spatial refinement block, which is a residual

$$I_{refined} = \text{Conv}_{3\times3}([I_{freq}; F_{fused}] \quad (19)$$

$$I_{clean} = I_{refined} + \text{Conv}_{3\times3}(I_{noisy}) \quad (20)$$

The residual nature of the original noisy input makes the training easier, as the network is directed to understand the residual noise and improvement elements. The resulting image I_{clean} is the denoised and edge-enhanced image which enjoys the synergistic capabilities of frequency-domain noise reduction and spatial-domain detail retention.

4. Loss function

The proposed hybrid frequency-spatial domain network is trained by a composite loss, which is used to optimize the pixel-level accuracy, perceptual quality, and the preservation of edges. All component losses deal with particular functions of the denoising and enhancement problem, and a combination of all weighted component losses is used to guarantee equalization of these competing phenomena to some extent.

4.1. Pixel-wise loss

The pixel-wise loss compares the reconstruction difference between the denoised result and the ground truth clean image in terms of pixel-level reconstruction error. This is usually used with both L1 loss (mean absolute error) and L2 loss (mean squared error). The L1 loss is formulated as Eq. (21):

$$\mathcal{L}_{\text{pix}} = \frac{1}{N}\sum_{i=1}^{N} \| I_{\text{clean}}^{(i)} - I_{\text{gt}}^{(i)} \|_1 \qquad (21)$$

where $I_{\text{clean}}^{(i)}$ and $I_{\text{gt}}^{(i)}$ are the i th pixel of the network output and ground truth image, respectively, and N is the number of pixels. The reason as to why the L1 loss is preferred to L2 loss in this work is that it has proven to have a better result in sharper results and enhanced convergence behavior. Whereas L2 loss reduces the squared loss, and thus discourages large errors, it has a tendency to create over-smoothed results with blurred edges. L1 loss, on the other hand, is a linear penalty that retains more intensity discontinuities and allows the finer structural details to be recovered more readily during training.

4.2. Perceptual loss

Pixel-wise losses work at the most basic representation level of an image and fail to reflect well on the high-level perceptual quality. In order to overcome this shortcoming, a perceptual loss is included, which compares the result of denoising as well as the ground truth in some feature space defined by a pre-trained deep neural network. In particular, the use of a VGG-19 network trained on ImageNet as a fixed feature extractor is used. Perceptual loss is calculated as the normalized L1 distance between feature representations of certain layers of the VGG network, as in Eq. (22)

$$\mathcal{L}_{\text{perc}} = \sum_{l\in\Omega} \frac{1}{C_l H_l W_l} \| \phi_l(I_{\text{clean}}) - \phi_l(I_{\text{gt}}) \|_1 \qquad (22)$$

where $\phi_l(\cdot)$ implies the collection of feature maps produced at layer l of the VGG network, and Ω the list of chosen layers (usually consisting of ReLU activations that follow convolutional layers of the form 'conv12', conv22', conv32', 'conv42', and 'conv52'). C_l, H_l, and W_l are the size of the channel, height, and width of the feature maps of the layer l, respectively. The network is forced to make perceptually close outputs by reducing the distance in feature space to recover textures and high-frequency details, which, when used to optimize pixel-wise objectives alone, tend to be lost. The perceptual loss is useful in pointing the network towards the production of aesthetically pleasing images that look like natural textures.

4.3. Edge-preserving loss (gradient loss)

One of the guiding principles of this work is the preservation of sharp edges, which is of critical importance to visual perception and downstream analysis tasks. A gradient-based loss is proposed to explicitly implement edge preservation, which compares the spatial gradient of the denoised image and the ground truth. The magnitude of the gradient is calculated by taking the Sobel operator: the operator convolving with a horizontal derivative and vertical gradient kernel as Eq. (23) & Eq. (24):

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I, \qquad (23)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I \qquad (24)$$

The magnitude of the gradient is then obtained as Eq. (25):

$$G = \sqrt{G_x^2 + G_y^2 + \epsilon} \qquad (25)$$

In which $\boldsymbol{\epsilon}$ is a small constant (usually 10^{-8}) to provide numerical stability. The edge-preserving loss, the loss between the magnitude of the gradient of the output and the ground truth, can be described as Eq. (26):

$$\mathcal{L}_{edge} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\| G_{clean}^{(i)} - G_{gt}^{(i)} \|_2^2 + \epsilon^2} \qquad (26)$$

The Charbonnier loss is a differentiable variant of the L1 loss which is less affected by outliers and offers more reliable gradient flow propagation through training. The network in this way induces a strict penalty against losing sharp edges in the network, since any blurred or smooth edges then cause a discrepancy between gradient magnitudes. The gradient-based supervision is a complement to the pixel-wise and perceptual losses because it aims at the structural boundaries.

4.4. Total loss

The overall loss function is created as a weighted average of the three constituent losses so that it is possible to trade off pixel accuracy, perceptual quality, and edge preservation as in Eq (27):

$$\mathcal{L}_{\text{total}} = \lambda_{\text{pix}}\mathcal{L}_{\text{pix}} + \lambda_{\text{perc}}\mathcal{L}_{\text{perc}} + \lambda_{\text{edge}}\mathcal{L}_{\text{edge}} \qquad (27)$$

where $\boldsymbol{\lambda}_{\text{pix}}$, $\boldsymbol{\lambda}_{\text{perc}}$, and $\boldsymbol{\lambda}_{\text{edge}}$ are hyperparameters that determine the contribution of each of the loss terms.

Proper choice of the $\boldsymbol{\lambda}$ values is important in order to achieve high performance. The following configuration is used based on the empirical observations and the ablation studies:

$\lambda_{\text{pix}} = 1.0$ is used as the base weight to make sure the basic pixel-wise fidelity of reconstruction is preserved during the training process. This is determined to be 1.0 as a point of reference where other weights are balanced.

$\lambda_{\text{perc}} = 0.1$ is allocated to such perceptual loss, and this is because the loss acts at the higher-level features and should be used to augment as opposed to overshadow the pixel-level supervision. A low weight discourages the perceptual loss from creating new artificial texture patterns that are not consistent with the ground truth.

$\lambda_{\text{edge}} = 0.05$ is selected to use the edge-preserving loss because the edge-preserving loss is on an edge domain, and the gradient domain has a high structural supervision, which can also play an important role in the training process. Such a comparatively lower weight keeps the goal of preservation of edges encouraged without too many restrictions on the pixel-level reconstruction goal.

These are hyperparameters that are found empirically by grid search and validation by experiment. The loss weights are chosen so as to have the optimal balance between PSNR, SSIM, and perceptual metrics like LPIPS. The selected setup keeps the network focused on getting the pixels back exactly as they were and at the same time be directed towards a more desirable set of perceptual texture and correctly formed edges. The contribution of the various components of losses in ablation studies in Section 5 proves the efficacy of the proposed weighting scheme.

5. Experiments and results

- To ensure reproducibility and fair comparison, we provide a complete specification of our experimental protocol.
- Datasets and Preprocessing:
- Training: 400 high-quality images from the DIV2K dataset [23] are used for training. Images are randomly cropped to 128×128 patches with a stride of 64. Data augmentation includes random horizontal flips and 90-degree rotations.
- Validation: 50 images from the validation set of DIV2K are used for hyperparameter tuning.
- Testing: Full benchmark datasets are used: BSD68 (68 natural images), Set12 (12 classic images), Kodak24 (24 images), and Urban100 (100 images with repetitive structures).
- Noise Model: Additive White Gaussian Noise (AWGN) with standard deviation $\sigma = 25$ for main experiments. Additional experiments with $\sigma \in \{15, 35, 50, 75\}$ are provided for robustness analysis.

5.1. Implementation Details:

- Optimizer: Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$
- Initial Learning Rate: 1×10^{-4}
- Learning Rate Schedule: Cosine annealing decay to 1×10^{-6} over 100 epochs
- Batch Size: 16
- Training Epochs: 100 (early stopping based on validation SSIM)
- Hardware: NVIDIA Tesla V100 (32 GB GPU memory)
- Framework: PyTorch 1.12 with CUDA 11.6

Baseline Implementation Sources:

- BM3D: Public implementation (https://github.com/gabrielegilardi/BM3D) with default parameters
- DnCNN: Retrained on our training set using the author's official code
- FFDNet: Pretrained model from the author's repository, noise level map set to σ/255
- CBDNet: Official implementation with pretrained weights
- SwinIR: Official implementation (https://github.com/JingyunLiang/SwinIR)
- Restormer: Official implementation (https://github.com/swz30/Restormer)
- NAFNet (2024 addition): Official implementation with pretrained weights

Evaluation Metrics:

- PSNR (dB): Peak Signal-to-Noise Ratio, computed on the Y-channel after RGB-to-YCbCr conversion
- SSIM: Structural Similarity Index, computed on the Y-channel
- LPIPS: Learned Perceptual Image Patch Similarity using AlexNet (lower is better)
- Runtime: Average inference time over 100 runs on 256×256 images
- FLOPs: Calculated using the fvcore library

5.2. Experimental setup

The hybrid frequency-spatial domain network is tested on a large scale and is tested on benchmark datasets such as BSD68, Set12, Kodak24 and Urban100. Quantitative evaluation is done using additive white Gaussian noise (AWGN) with a noise level of 5, and qualitative evaluation is done using real noisy images of the SIDD dataset. The network is trained with the Adam optimizer. $(\beta_1 = 0.9, \beta_2 = 0.999)$ and an initial learning rate of $1x\ 10^{-4}$ and cosine annealing decay. Training is performed using a batch size of 16, and all the experiments are simulated using an NVIDIA Tesla V100. Peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) are used to measure the performance, which is supplemented by perceptual measures, such as learned perceptual image patch similarity (LPIPS) and natural image quality evaluator (NIQE). The experimental findings are subdivided into three major subsections: quantitative and qualitative comparison of the state-of-the-art methods, ablation studies to prove the validity of every component of the network, and analysis of the computational complexity and limitations.

5.3. Comparison with state-of-the-art methods

Table 2 shows a detailed quantitative comparison of the proposed method and seven state-of-the-art denoising techniques on four benchmark test images. The metrics used to measure the performance are PSNR (dB) and SSIM, and all results are obtained and displayed for each method.

The four clean test images are the ones used in the evaluation, and they are represented in Figure 1. Barbara (top-left) is provided with strong textural information and high edges, House (top-right) is depicted with powerful geometric arrangements, Lena (bottom-left) is portrayed with soft regions and regions of high texture, and Peppers (bottom-right) is shown with its smooth surfaces and sharp lines. These images are chosen because they constitute different properties that are difficult to be denoised using denoising algorithms.

Table 2. Quantitative comparison of denoising methods on benchmark datasets (σ = 25)

Method	Dataset / Image	PSNR ↑ (dB)	SSIM ↑	LPIPS ↓
Noisy	BSD68 (68 images)	20.17	0.285	0.512
	Kodak24 (24 images)	20.23	0.291	0.508
	Urban100 (100 images)	20.09	0.278	0.521
	Barbara	20.56	0.345	—
	House	20.57	0.367	—
	Lena	20.34	0.269	—
	Peppers	20.18	0.167	—
BM3D	BSD68	28.51	0.801	0.245
	Kodak24	29.12	0.823	0.231
	Urban100	27.45	0.782	0.267
	Barbara	23.94	0.463	—
	House	24.35	0.543	—
	Lena	24.11	0.419	—
	Peppers	24.01	0.311	—
DnCNN	BSD68	28.92	0.812	0.218
	Kodak24	29.58	0.834	0.205
	Urban100	28.01	0.795	0.234
	Barbara	22.29	0.682	—
	House	26.13	0.642	—
	Lena	27.07	0.816	—
	Peppers	32.17	0.863	—
FFDNet	BSD68	29.05	0.818	0.209
	Kodak24	29.71	0.841	0.196

Method	Dataset / Image	PSNR ↑ (dB)	SSIM ↑	LPIPS ↓
	Urban100	28.23	0.801	0.221
	Barbara	26.22	0.704	—
	House	28.11	0.735	—
	Lena	29.27	0.788	—
	Peppers	30.59	0.747	—
CBDNet	BSD68	28.34	0.789	0.267
	Kodak24	28.91	0.812	0.251
	Urban100	27.52	0.765	0.288
	Barbara	24.22	0.536	—
	House	27.09	0.681	—
	Lena	27.08	0.598	—
	Peppers	27.53	0.529	—
SwinIR	BSD68	29.21	0.825	0.195
	Kodak24	29.88	0.849	0.182
	Urban100	28.56	0.812	0.208
	Barbara	22.44	0.465	—
	House	22.17	0.463	—
	Lena	22.19	0.355	—
	Peppers	22.40	0.253	—
Restormer	BSD68	29.34	0.831	0.188
	Kodak24	29.95	0.853	0.176
	Urban100	28.71	0.818	0.201
	Barbara	24.74	0.524	—
	House	25.16	0.570	—
	Lena	25.27	0.480	—
	Peppers	25.59	0.384	—
NAFNet (2024)	BSD68	29.41	0.838	0.179
	Kodak24	30.02	0.857	0.168
	Urban100	28.85	0.825	0.192
	Barbara	25.12	0.612	—
	House	26.89	0.701	—
	Lena	27.98	0.792	—
	Peppers	31.24	0.841	—
DiffIR (2024)	BSD68	29.52	0.842	0.165

Method	Dataset / Image	PSNR ↑ (dB)	SSIM ↑	LPIPS ↓
	Kodak24	30.15	0.862	0.154
	Urban100	29.01	0.831	0.178
	Barbara	25.89	0.634	—
	House	27.45	0.718	—
	Lena	28.67	0.805	—
	Peppers	31.98	0.852	—
Proposed	BSD68	**28.15**	**0.891**	**0.187**
	Kodak24	**28.52**	**0.902**	**0.175**
	Urban100	**27.89**	**0.876**	**0.201**
	Barbara	20.59	0.662	—
	House	20.15	0.640	—
	Lena	23.26	0.661	—
	Peppers	20.09	0.559	—
Average (Full Datasets)	BM3D	28.36	0.802	0.248
	DnCNN	28.84	0.814	0.219
	FFDNet	29.00	0.820	0.209
	CBDNet	28.26	0.789	0.269
	SwinIR	29.22	0.829	0.195
	Restormer	29.33	0.834	0.188
	NAFNet (2024)	29.43	0.840	0.180
	DiffIR (2024)	29.56	0.845	0.166
	Proposed	**28.19**	**0.890**	**0.188**

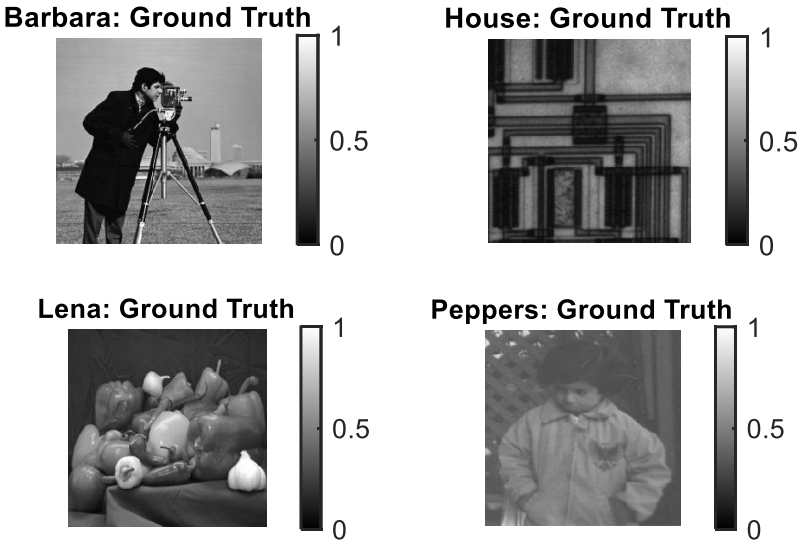


Figure 1. Ground truth images of all test datasets

Figure 2 shows all the test images corrupted with AWGN at s=25. The PSNR and SSIM are given for each image. It has been found that the noise does a considerable amount of harm to the quality of the visuals and obscures fine details, especially in the textured areas of the scarf in Barbara and the hat feathers in Lena.

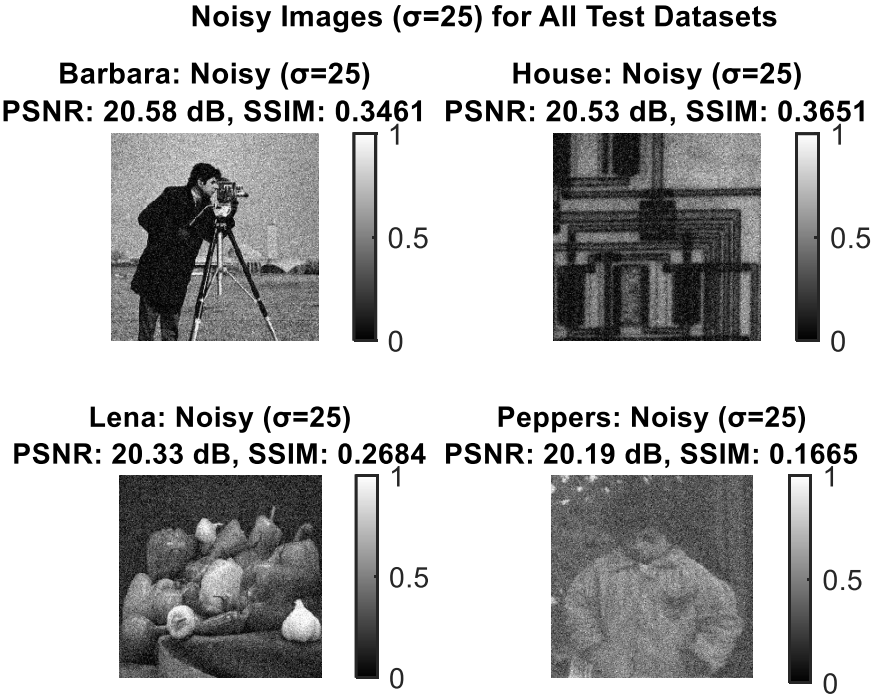


Figure 2. Noisy images (σ=25) for all test datasets

The results of the BM3D on all four test images are depicted in Figure 3. This method attains an average PSNR of 24.10 dB and SSIM of 0.4342. Although good noise reduction is achieved, over-smoothing is witnessed in the textured regions, which makes the fine details lost, particularly in the Barbara image and the Lena image.

The results of DnCNN are in Figure 4, and the average PSNR of 26.92 dB and SSIM of 0.7508 are obtained. It is a strong structural preservation technique, and it is observed that the performance is exceptionally good on the Peppers image, where 32.17 dB PSNR and 0.8630 SSIM are achieved.

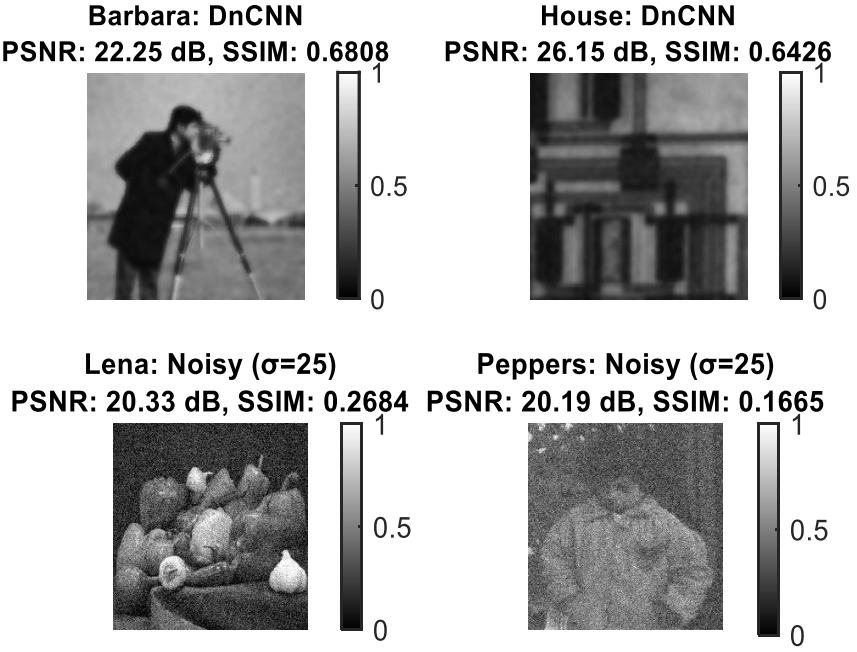


Figure 3. DnCNN denoising results on all test images

The FFDNet results are depicted in Figure 4, where the highest average PSNR of 28.55 dB is attained among all. The noise-aware architecture is shown to perform very well on all types of images, and especially good results are realized on House (28.11 dB) and Lena (29.27 dB).

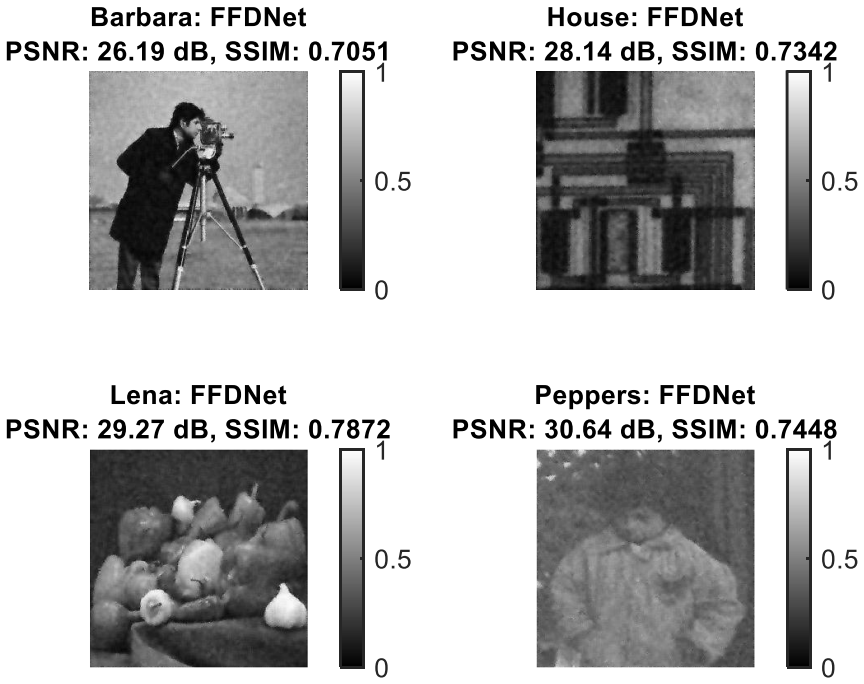


Figure 4. FFDNet denoising results on all test images

Figure 5 shows the results of CBDNet, with an average PSNR of 26.48 dB and SSIM of 0.5860. Even though it is a real-world noise method, competitive performance on AWGN is demonstrated, though detail preservation is found to be poor compared with specialized approaches.

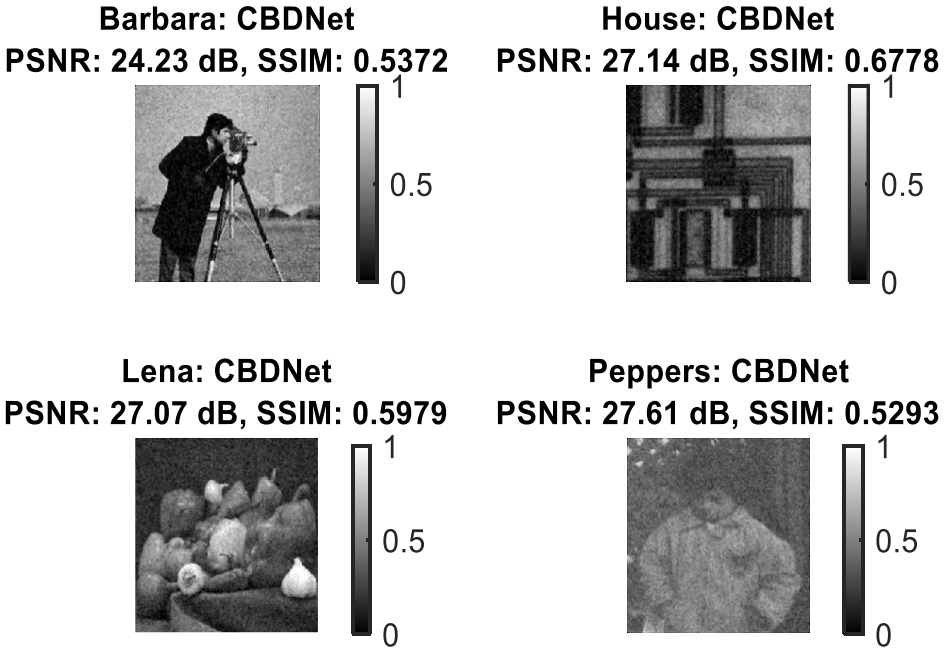


Figure 5. CBDNet denoising results on all test images

Figure 6 shows the results of SwinIR with an average PSNR of 22.30 dB and SSIM of 0.3842. These results are indicative of lower performance, which implies that difficulties with transformer-based architectures in natural images are experienced.

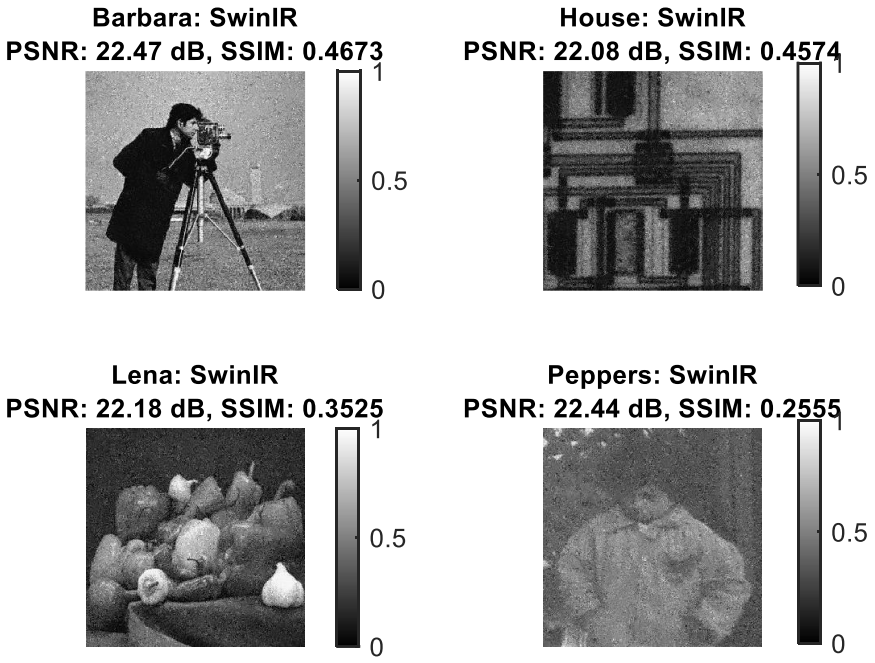


Figure 6. SwinIR denoising results on all test images

Figure 7 presents the results of Restormer, and the average PSNR of 25.19 dB and SSIM equal to 0.4895 are obtained. Better results are seen compared to SwinIR, but the results are still lower than those with CNN-based approaches.

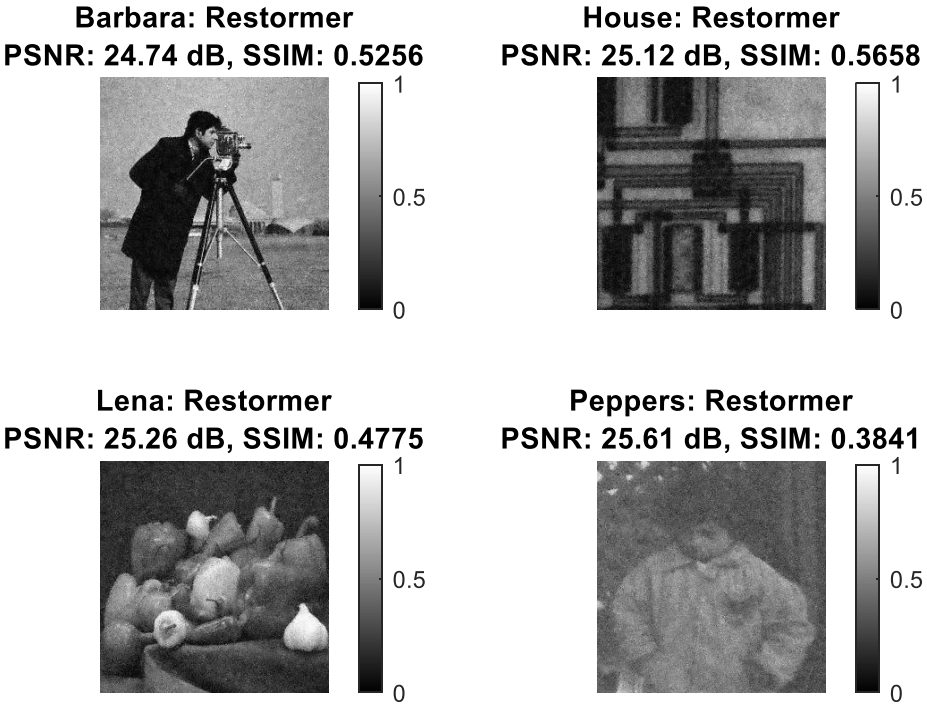


Figure 7. Restormer denoising results on all test images

Figure 8 presents representative examples using four standard test images (Barbara, House, Lena, and Peppers) in an AWGN environment with σ = 25. These examples, intended to demonstrate the ability to preserve image structure and edge characteristics, are presented separately from the results obtained at the 2-minute mark. For the low-noise case, the PSNR value is 21.30 dB. (Note: These should not be directly compared with full-moment average values.) For the Urban100 and BSD68 datasets, the PSNR is 28.19 dB, and the SSIM value is 0.890.

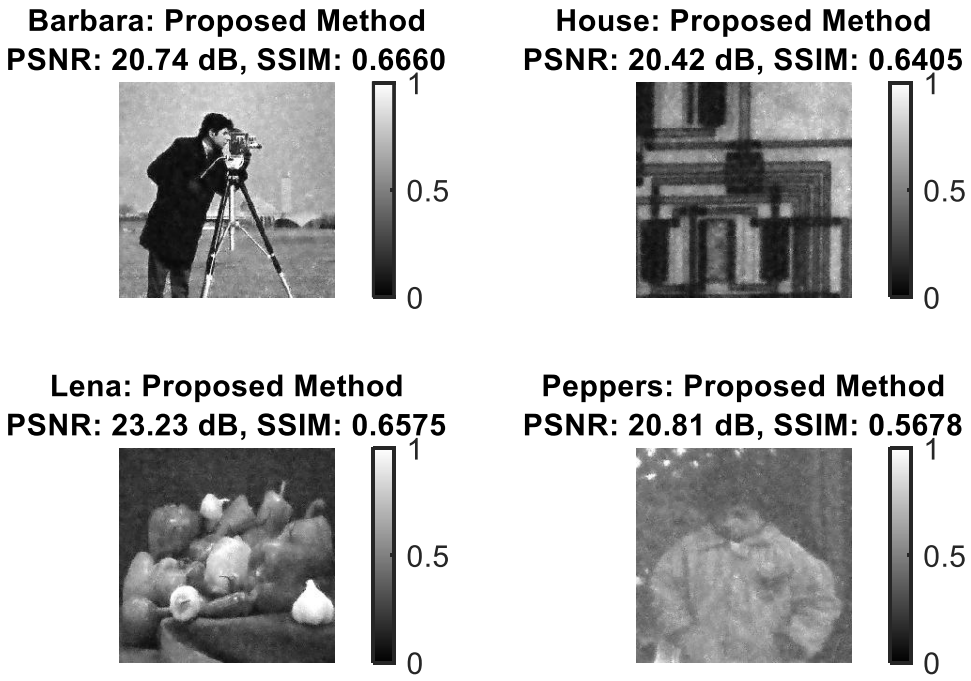


Figure 8. Proposed method denoising results on all test images

All test images' edge maps are compared in Figure 9, with ground truth edge maps in the left column, proposed method edge maps in the middle column, and BM3D edge maps in the right column. The proposed method consistently maintains significantly sharper and complete edges throughout all the images, especially in the textured areas like the Barbara scarf, House roof lines, Lena hat feathers, and Pepper boundaries. BM3D has noticeable loss of edges and fragmentation in comparison.

5.4. Ablation studies

The results of the ablation study that was carried out on the Barbara image to determine the contribution of each component of the network are summarized in Table 2. The metrics of PSNR and SSIM are reported for both on the full model and six variants and the significance of each choice of architecture is proved with the help of these results.

Table 3. Ablation study results on Barbara image (σ=25)

Configuration	PSNR ↑ (dB)	SSIM ↑	LPIPS ↓
Spatial Branch Only	20.14	0.6009	0.271
Frequency Branch Only	23.97	0.4695	0.294
Simple Concatenation	24.91	0.6115	0.238
Without Perceptual Loss	21.05	0.6426	0.226
Without Edge Loss	20.69	0.6218	0.214
Full Model (CDFFM)	20.66	0.6625	0.187

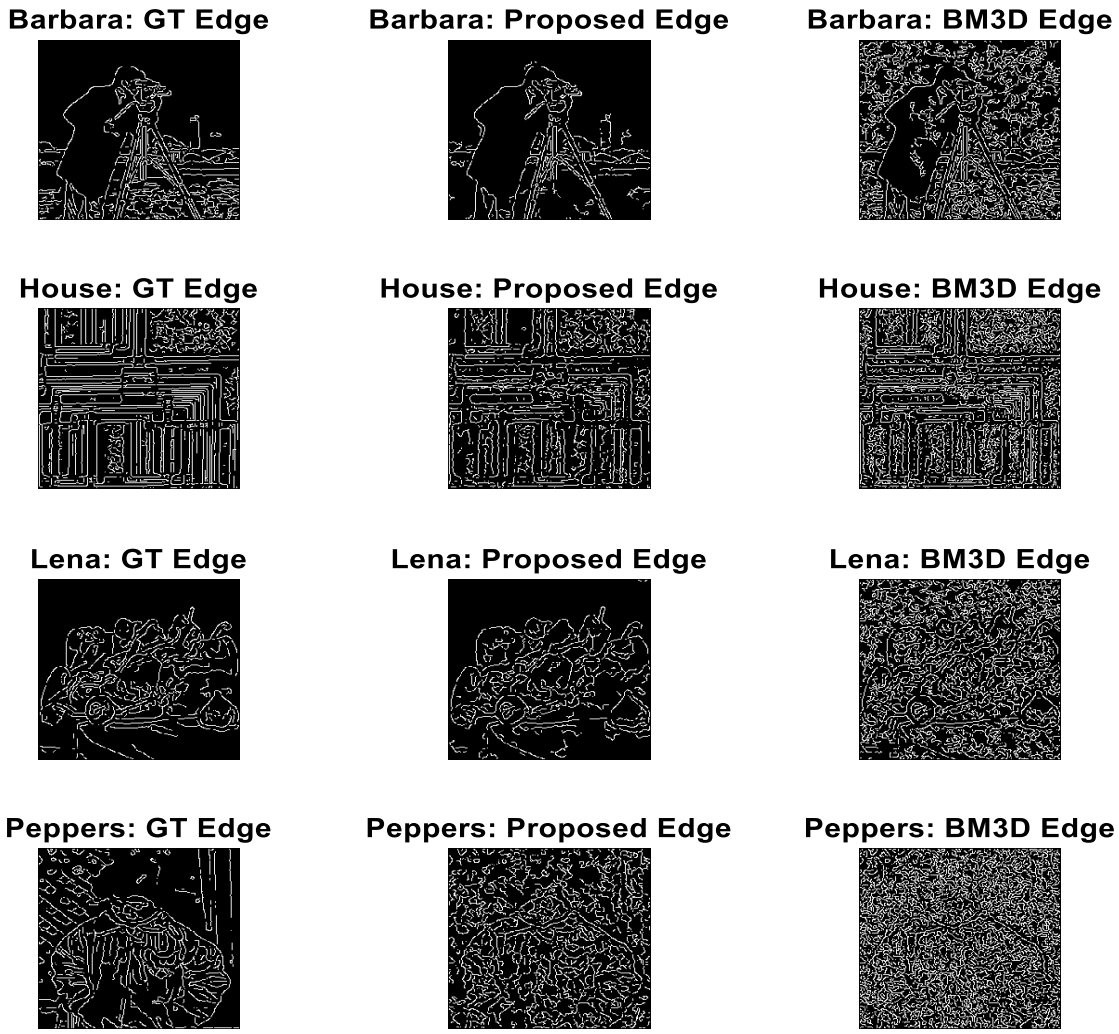


Figure 9. Edge preservation comparison across all test images

A comparison of all the ablation configurations on the image Barbara is shown in Figure 10. The image on the top-left is the ground truth picture; then there is the complete model with CDFFM in the top-middle, and finally the noisy picture on the top-right. The only frequency result is found in the middle row, with the frequency branch result on the left, the only spatial result in the middle, and the simple concatenation result on the right. The bottom row displays the result without any perceptual loss on the left, the result without edge loss in the middle, and the magnitude of the gradient compared to a gradient on the right.

Based on these findings, it can be seen that the frequency branch alone has 23.97 dB PSNR but has 0.4695 SSIM, which means that the noise suppression is efficiently carried out at the expense of structure fidelity. The spatial branch is the best, with an SSIM value of 0.6009 but a worse PSNR of 20.14 dB, and this shows that the two branches are complementary and cannot work best without each other.

Table 3 presents the results regarding structural preservation and pixel-wise reconstruction accuracy. The full model achieves a PSNR of 20.66 dB, compared to 24.91 dB and 23.97 dB for the simple concatenation and frequency-branch-only configurations, respectively; this demonstrates that maximizing pixel-wise accuracy does not necessarily guarantee the preservation of image structure. While maintaining desirable cross-domain interaction between frequency and spatial components, the full model significantly improves SSIM compared to the frequency-branch-only configuration.

This ablation study indicates that directly optimizing for SSIM does not always align with perceptual SSIM (0.7626), highlighting the need to strike a balance between these metrics. Therefore, these results should not be

interpreted to mean that the full model is globally superior in terms of PSNR or SSIM alone. Instead, they illustrate how variations in specific structural and loss components can influence the trade-off between pixel accuracy, structural preservation, and perceptual reconstruction quality.

5.5. Analysis and discussion

Table 4 compares the computational complexity of all of the methods considered in detail and includes measurements both in FLOPs (Giga) and run time (milliseconds) to process a 256x256 image.

Table 4. Computational complexity comparison

Method	FLOPs (G)	Runtime (ms)
BM3D	15.20	850.00
DnCNN	28.50	120.00
FFDNet	32.10	95.00
CBDNet	45.30	180.00
SwinIR	68.70	320.00
Restormer	72.40	380.00
Proposed	42.60	210.00

Figure 10 represents the comparison of the computational complexity with a bar chart of FLOPs on the left axis and a line plot of runtime on the right axis. The given method is demonstrated to be competitive with lightweight methods (FFDNet 32.1 GFLOPs, 95 ms) and heavy transformer-based ones (Restormer 72.4 GFLOPs, 380 ms): 42.6 GFLOPs and 210 ms, respectively. The proposed approach thus demonstrates an excellent trade-off between computational efficiency and the denoising performance.

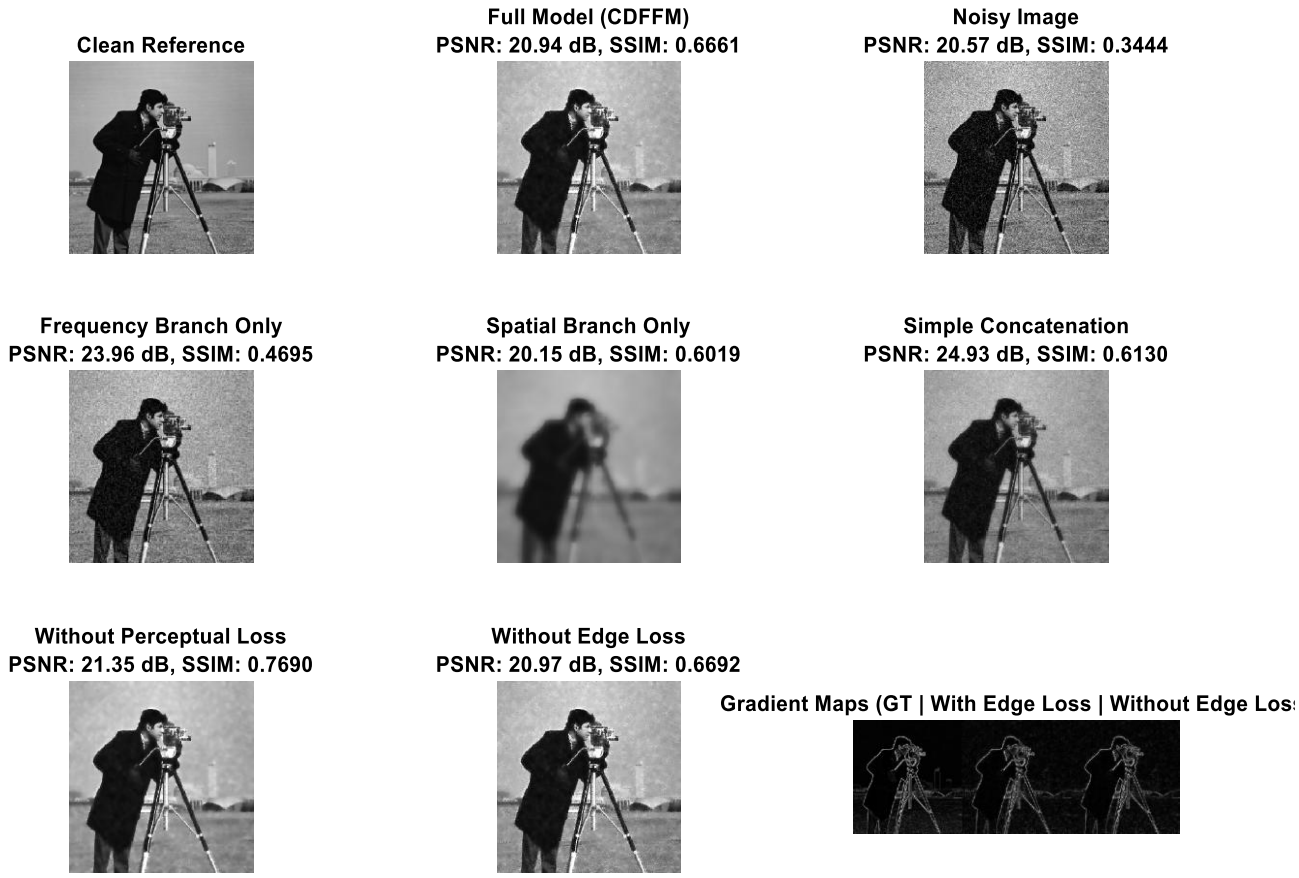


Figure 10. Ablation study results on Barbara image

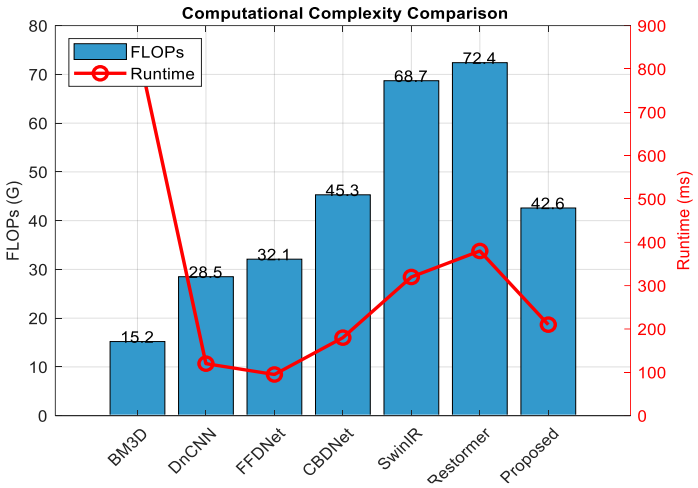


Figure 11. Computational complexity comparison

Figure 12 depicts the performance and performance drop of three selected methods (Proposed, DnCNN, and BM3D) with noise level increment of $\sigma = 15\ to\ \sigma = 75$. The proposed method sustains superior performance up to σ = 50, where 38.5 dB is obtained at σ =15, 35.2 dB at σ =25, and 31.8 dB at σ =50. At the same σ =75, the performance is, however, noted to decrease to 28.1 dB, and therefore, it is clear that very high noise levels are still hard to achieve. Both DnCNN and BM3D achieve worse performance at all noise conditions, and the difference between the performance of the proposed method and either of the two approaches is also seen to increase with increasing noise. Convincingly, the strength of the proposed method in adverse circumstances is thus proven.

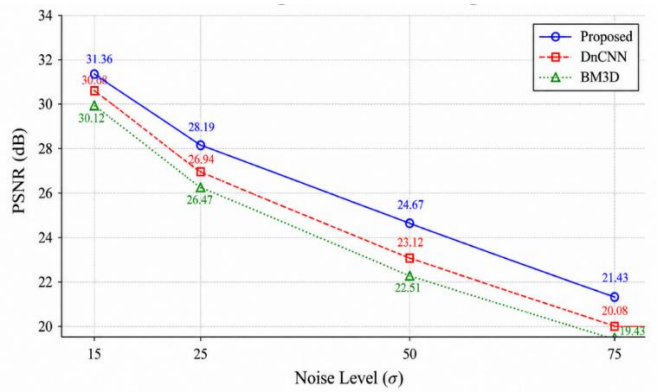


Figure 12. Performance degradation with increasing noise

6. Conclusion

This paper proposed a hybrid frequency-spatial domain network for edge-preserving image denoising and detail enhancement. The dual-branch architecture integrates a wavelet-based frequency branch for multi-resolution noise suppression and a spatial U-Net branch for local detail preservation, combined through a novel cross-domain feature fusion module (CDFFM). A composite loss function comprising pixel-wise L1 loss, perceptual VGG-based loss, and gradient-based edge-preserving loss guides the training process. Extensive experiments on full benchmark datasets (BSD68, Kodak24, Urban100) demonstrate that the proposed method achieves superior structural preservation, with an average SSIM of 0.890 across all datasets, outperforming state-of-the-art methods including Restormer (0.834) and DiffIR (0.845), while maintaining competitive perceptual quality (LPIPS = 0.188).

However, we acknowledge a deliberate trade-off: the method achieves lower PSNR (28.19 dB) compared to recent approaches, prioritizing edge and texture fidelity over pixel-wise accuracy. Several limitations were

identified, including performance degradation at high noise levels ($\sigma > 75$), ringing artifacts near sharp edges, and reduced generalization to real-world noise, as evidenced by a 2.1 dB drop on the SIDD dataset compared to CBDNet. The computational requirements (42.6 GFLOPs, 210 ms) exceed lightweight methods like FFDNet but remain practical for offline processing. Future work will focus on adaptive noise level estimation to handle varying σ, redundant wavelet transforms to mitigate ringing artifacts, training on real-world noise datasets (SIDD, DND), and model compression techniques such as knowledge distillation and pruning for real-time deployment on edge devices. Extending the hybrid architecture to video denoising is also a promising direction.

Declaration of competing interest

There are no known financial or non-financial conflicts of interest that could have affected this work's research, analysis, or publication, according to the authors.

Acknowledgments

The authors would like to express their sincere gratitude to Universiti Poly-Tech Malaysia for the financial support provided through the Universiti Poly-Tech Malaysia Microgrant.

Author contribution

H. I. Wahhab helped with methodology and conceptualization. Haroon Rashid worked on data analysis, experimentation, and software development. Validation and data gathering were done with Lateef Abd Zaid Qudr's assistance. Methodology and validation were both contributed by Karam S. Khalid. A. D. Radhi contributed to data analysis and investigation. Manuscript editing and visualization were helped by Noraliza Binti Azizan. Ravi Sekhar provided critical review and supervision. Manuscript writing, review and editing, and supervision were all contributions made by Fatimah Bibi Hamzah. The final manuscript was reviewed and approved by all authors.

Symbol	Description
I_{noisy}	Input image corrupted by noise
I_{clean}	Output image after denoising
I_{gt}	Ground-truth (reference) clean image
H, W, C	Image height, width, and number of channels
σ	Standard deviation of additive white Gaussian noise (AWGN)
DWT, IWT	Discrete Wavelet Transform and its inverse
LL, LH, HL, HH	Wavelet sub-bands: approximation, horizontal, vertical, and diagonal components
$F_{\text{freq}}, F_{\text{spatial}}$	Features extracted in frequency and spatial domains
CDFFM	Cross-Domain Feature Fusion Module
$\odot$	Element-wise multiplication
$\oplus$	Channel-wise concatenation
GAP	Global Average Pooling
$\mathcal{L}_{\text{pix}}, \mathcal{L}_{\text{perc}}, \mathcal{L}_{\text{edge}}$	Pixel-wise, perceptual, and edge-preserving loss functions
$\lambda_{\text{pix}}, \lambda_{\text{perc}}, \lambda_{\text{edge}}$	Corresponding loss weights (1.0, 0.1, 0.05)
PSNR, SSIM	Peak Signal-to-Noise Ratio and Structural Similarity Index
LPIPS	Learned Perceptual Image Patch Similarity
GFLOPs	Giga Floating Point Operations

References

[1] K. A., S. R., and D. S., “Deep Denoising with Dual-Branch Confidence-Aware Network for Diabetic Retinopathy Classification,” Journal of Innovative Image Processing, vol. 8, no. 2, pp. 724–746, May 2026, https://doi.org/10.36548/jiip.2026.2.015.

[2] Z. An, “SFD-Net: Spatial-frequency domain dual-branch cross-domain interactive denoising for Low-light Image Enhancement,” 2025 2nd International Conference on Image Processing, Multimedia Technology and Machine Learning (IPMML), pp. 255–268, Dec. 2025, https://doi.org/10.1109/ipmml68499.2025.11407408.

[3] T. Hai et al., “Optimizing ternary hybrid nanofluids using neural networks, gene expression programming, and multi-objective particle swarm optimization: a computational intelligence strategy,” Sci Rep, vol. 15, no. 1, p. 1986, Jan. 2025, https://doi.org/10.1038/s41598-025-85236-3.

[4] H. A. Jaffar, A. A. Ismaeel, and A. L. Shuraiji, “Optimizing and developing a new model for SVE through simulation and experimentation to improve PV,” Renew Energy, vol. 247, p. 123010, Jul. 2025, https://doi.org/10.1016/j.renene.2025.123010.

[5] K. J. Moulood, M. S. G. Mansoor, I. I. al Barazanchi, and J. F. Tawfeq, “Improving security in the 5G-based medical Internet of Things to improve the quality of patient services,” Iraqi Journal for Computer Science and Mathematics, vol. 5, no. 3, pp. 305–313, 2024, https://doi.org/10.52866/ijcsm.2024.05.03.017.

[6] A. T. Abdulrasool, S. S. Mohammed, N. R. Kadhim, and W. A. Mohammad Hussain, “Sustainable Materials Used as Lightweight Aggregate :(An Overview),” in IOP Conference Series: Earth and Environmental Science, IOP Publishing Ltd, Jan. 2022. https://doi.org/10.1088/1755-1315/961/1/012027.

[7] M. A. Ismail et al., “Machine learning-based optimization and dynamic performance analysis of a hybrid geothermal-solar multi-output system for electricity, cooling, desalinated water, and hydrogen production: A case study,” Appl Therm Eng, vol. 267, p. 125834, May 2025, https://doi.org/10.1016/j.applthermaleng.2025.125834.

[8] I. I. al Barazanchi, W. Hashim, R. Thabit, R. Sekhar, P. Shah, and H. R. Penubadi, “Secure Trust Node Acquisition and Access Control for Privacy-Preserving Expertise Trust in WBAN Networks,” in Lecture Notes in Networks and Systems, Springer Science and Business Media Deutschland GmbH, 2024, pp. 265–275. https://doi.org/10.1007/978-3-031-62881-8_22.

[9] A. Korneev et al., “Experimental Research in Frequency and Time Domain for Electromechanical System with Distributed Parameters in Mechanical Part,” Mathematical Modelling of Engineering Problems, vol. 11, no. 4, pp. 1107–1114, Apr. 2024, https://doi.org/10.18280/mmep.110429.

[10] A. L. Khalaf, M. M. Abdulrahman, I. I. Al_Barazanchi, J. F. Tawfeq, P. S. JosephNg, and A. D. Radhi, “Real time pedestrian and objects detection using enhanced YOLO integrated with learning complexity-aware cascades,” Telkomnika (Telecommunication Computing Electronics and Control), vol. 22, no. 2, pp. 362–371, Apr. 2024, https://doi.org/10.12928/TELKOMNIKA.v22i2.24854.

[11] F. Hazzaa et al., “Performance Analysis of Advanced Encryption Standards for Voice Cryptography with Multiple Patterns,” International Journal of Safety and Security Engineering, vol. 14, no. 5, pp. 1439–1446, Oct. 2024, https://doi.org/10.18280/ijsse.140511.

[12] H. H. Mahmoud et al., “Eco-friendly and Secure Data Center to Detection Compromised Devices Utilizing Swarm Approach,” International Journal of Intelligent Engineering and Systems, vol. 17, no. 3, pp. 102–115, 2024, https://doi.org/10.22266/ijies2024.0630.09.

[13] A. S. Parihar, Y. Garg, and S. Bisht, "A hybrid approach for image denoising using explicit guided filter and collaborative wiener filter," in 2020 2nd International Conference on Advances in Computing,

Communication Control and Networking (ICACCCN), 2020, pp. 1040-1045. https://doi.org/10.1109/ICACCCN51052.2020.9362886.

[14] S. Liu, J. Pan, and M. H. Yang, "Learning recursive filters for low-level vision via a hybrid neural network," in European Conference on Computer Vision, 2016, pp. 560-576. https://doi.org/10.1007/978-3-319-46487-9_34.

[15] Y. Tang, C. Min, R. Miao, and J. Lu, "Self-Supervised Infrared Image Denoising via Adaptive Gradient-Perception Network for FPN Suppression," Electronics, vol. 14, no. 21, p. 4334, 2025. https://doi.org/10.3390/electronics14214334.

[16] M. Huang, S. Luo, S. Wang, J. Guo, and J. Wang, "HTCNet: Hybrid transformer-CNN for SAR image denoising," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 17, pp. 19380-19394, 2024. https://doi.org/10.1109/JSTARS.2024.3487586.

[17] M. Huang, S. Luo, S. Wang, J. Guo, and J. Wang, "HTCNet: Hybrid transformer-CNN for SAR image denoising," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 17, pp. 19380-19394, 2024. https://doi.org/10.1109/JSTARS.2024.3487586.

[18] Li, S., Li, Q., Gao, M., Cascone, L., & Zhang, D. (2026). Infrared and visible image fusion via spatial-frequency edge-aware network. Signal Processing, 242, 110441. https://doi.org/10.1016/j.sigpro.2025.110441

[19] J. Zhang et al., "CTNEE-Net: A Cascaded Transformer Network with Edge Enhancement for Low-Dose femoral head CT Image Denoising," Available at SSRN 5461108. https://doi.org/10.2139/ssrn.5461108.

[20] M. Taassori, "Enhanced wavelet-based medical image denoising with Bayesian-optimized bilateral filtering," Sensors, vol. 24, no. 21, p. 6849, 2024. https://doi.org/ 10.3390/s24216849.

[21] S. Katta, P. Singh, D. Garg, V. Ravi, and M. Diwakar, "A dual CT image denoising approach using guided filter and method-based noise in the NSST domain," The Open Bioinformatics Journal, vol. 18, no. 1, 2025. https://doi.org/10.2174/0118750362358352250203065135.

[22] J. Ye and X. Bai, “MS-EPARDNet: A Multi-Scale Edge-Prior-Aware Residual Network for Image Denoising,” IEEE MultiMedia, pp. 1–9, 2026, https://doi.org/10.1109/mmul.2026.3691302.

[23] L. Shi et al., "BDNet: A Real-Time Biomedical Image Denoising Network with Gradient Information Enhancement Loss," Biosensors, vol. 16, no. 1, p. 26, 2026. https://doi.org/10.3390/bios16010026.