applied sciences

MDPI

Article

A Nested UNet Based on Multi-Scale Feature Extraction for Mixed Gaussian-Impulse Removal

Jielin Jiang [1,2], Li Liu [1], Yan Cui [3,*] and Yingnan Zhao [4]

[1] School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China; jiangjielin2008@163.com (J.J.); liuli9798@163.com (L.L.)
[2] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, Nanjing 210044, China
[3] College of Mathematics and Information Science, Nanjing Normal University of Special Education, Nanjing 210038, China
[4] School of Computer Science, Nanjing University of Information Science and Technology, Nanjing 210044, China; zh_yingnan@126.com
* Correspondence: csyanncui@njts.edu.cn

Abstract: Eliminating mixed noise from images is a challenging task because accurately describing the attenuation of noise distribution is difficult. However, most existing algorithms for mixed noise removal solely rely on the local information of the image and neglect the global information, resulting in suboptimal denoising performance when dealing with complex mixed noise. In this paper, we propose a nested UNet based on multi-scale feature extraction (MSNUNet) for mixed noise removal. In MSNUNet, we introduce a U-shaped subnetwork called MSU-Subnet for multi-scale feature extraction. These multi-scale features contain abundant local and global features, aiding the model in estimating noise more accurately and improving its robustness. Furthermore, we introduce a multi-scale feature fusion channel attention module (MSCAM) to effectively aggregate feature information from different scales while preserving intricate image texture details. Our experimental results demonstrate that MSNUNet achieves leading performance in terms of quality metrics and the visual appearance of images.

Keywords: multi-scale feature extraction; nested UNet; channel attention; mixed noise removal

Citation: Jiang, J.; Liu, L.; Cui, Y.; Zhao, Y. A Nested UNet Based on Multi-Scale Feature Extraction for Mixed Gaussian-Impulse Removal. *Appl. Sci.* **2023**, *13*, 9520. https://doi.org/10.3390/app13179520

Academic Editors: Yong Yang and Hyo Jong Lee

Received: 30 July 2023
Revised: 18 August 2023
Accepted: 21 August 2023
Published: 22 August 2023

1. Introduction

Digital images have found extensive applications in various fields, including medical imaging, remote sensing, and semantic segmentation [1]. However, images captured by cameras often suffer from mixed noise, which degrades image quality and affects subsequent computer vision tasks [2]. For example, hyperspectral images commonly exhibit a combination of additive white Gaussian noise (AWGN) and Poisson noise, while computed tomography images and complementary DNA microarray images experience a mixture of AWGN and impulse noise (IN) [3]. As a result, the removal of mixed noise has become a critical and challenging problem that requires further investigation.

In recent years, researchers have proposed many effective methods for removing combined AWGN and IN. Since IN only affects some of an image's pixel values [4], early methods for mixed noise removal typically adopted a two-stage approach, where IN is first suppressed and then AWGN is removed. Garnett et al. [5] introduced the rank-ordered absolute differences method for IN detection and integrated it into the bilateral filter framework [6], which performs adaptive image denoising for both AWGN and IN. Cai et al. [7] employed a variational framework with a l_1 data fidelity term and the Mumford–Shah regularization term to remove mixed noise from images. However, this method, while preserving some edge properties, may lead to image oversmoothing as it only considers local image information. To address this, Xiao et al. [8] proposed a $l_1 - l_0$

double-sparsity regularization-based method. They used the l_1 term to remove IN and introduced the l_0 term in an improved K-SVD algorithm to suppress residual noise after IN detection. Liu et al. [9] developed a generalized weighted $l_2 - l_0$ method for AWGN-IN removal based on maximum likelihood estimation and sparse representation. Jiang et al. [10] integrated the image sparse prior and non-local similar prior into a non-local sparse regularization term and proposed the weighted encoding with sparse nonlocal regularization (WESNR) method. Furthermore, Huang et al. [11] considered both the non-local self-similarity and low-rank properties of natural images and proposed a Laplace scale mixture combined with a non-local low-rank regularization (LSLR) model. In the domain of hyperspectral images, Zhuang et al. [12] proposed a method called FastHyMix, which estimates mixed noise by exploiting its high spectral correlation. This method enhances the accuracy of mixed noise estimation by utilizing a neural denoising network to learn image priors. Liu et al. [13] proposed a mixed noise removal method that combines a deterministic low-rank prior with an implicit regularization scheme. This method approximates the low-rank prior of the image using the matrix logarithm norm and utilizes an implicit regularizer to preserve image details.

Deep learning has emerged as a promising approach for removing mixed noise due to its ability to adapt to complex data and establish relationships between noisy and clean images. Compared to traditional denoising methods, deep learning relies on large amounts of training data to learn stable nonlinear mappings, making it a data-driven approach. Several deep learning-based methods have been proposed for mixed noise removal. Islam et al. [14] proposed a transfer learning approach called TL-CNN, which uses a convolutional neural network (CNN) to learn an end-to-end mapping from noisy to clean images. Abiko et al. [15] introduced a blind denoising method called BDCNN, which is entirely based on a CNN. The network structure of BDCNN consists of 50 convolution (Conv) blocks, with the first 25 blocks used for IN removal and the last 25 blocks used for AWGN removal. Wang et al. [16] incorporated a CNN regularizer into a traditional model-based variational approach, resulting in VA-CNN. VA-CNN utilizes the CNN-learned natural image prior to improve the variational method's accuracy in estimating noise parameters. Jiang et al. [17] proposed a non-local mean-based CNN (NM-CNN) method. This approach first detects the locations of outlier pixels using a median filter and replaces them with non-local mean values; subsequently, AWGN is removed using a CNN. Lyu et al. [18] introduced a generative adversarial network (GAN) that employs generators and discriminators for feature extraction. This method incorporates a joint loss function based on image prior and visual perceptual metrics, further enhancing image denoising performance. Mafi et al. [19] proposed a CNN architecture incorporating Conv, batch normalization (BN), and rectified linear units (ReLU) as basic components for mixed noise removal. In [20], a serial attention module-based CNN method (SACNN) is proposed, which employs a serial attention module to better preserve texture details. Overall, these deep learning-based methods have demonstrated significant improvements compared to traditional denoising methods.

In summary, most existing methods for removing mixed noise do not fully utilize the local and global information of the image, resulting in the inaccurate modeling of complex noise during the denoising process. This leads to the deformation and distortion of the image structure, causing the loss of fine details. To address this issue, this paper proposes MSNUNet for mixed noise removal.

The key contributions of MSNUNet can be summarized as follows:

1. We propose a nested UNet architecture based on multi-scale feature extraction for mixed noise removal. In MSNUNet, we introduce MSU-Subnet for multi-scale feature extraction. These multi-scale features contain rich local and global features, which help the model estimate noise more accurately and improve its robustness.
2. We introduce MSCAM into the MSNUNet model to effectively aggregate multi-scale features. Additionally, MSCAM utilizes channel attention (CA) to enhance the

extraction of important features, enabling the network to better preserve intricate textural details in images.

3. Our experimental results demonstrate that MSNUNet achieves superior performance in terms of quality metrics compared to state-of-the-art methods and generates visually satisfying denoised images.

The remainder of this paper is organized as follows: Section 2 describes the related work. Section 3 presents the mixed noise model, while Section 4 introduces the proposed MSNUNet model. In Section 5, extensive experiments are conducted to evaluate the performance of MSNUNet, and the conclusions of the study are presented in Section 6.

2. Related Work

2.1. Multi-Scale Feature Extraction

As CNNs typically utilize small kernels of sizes such as 1×1 and 3×3 for feature extraction, CNNs can only extract local features within a small range of perceptions and cannot extract multi-scale features. Multi-scale features represent samples of the signal at different granularities, and different features can be observed at different granularities to perform different tasks. Smaller or denser sampling can reveal more details, whereas larger or sparser sampling can reveal overall trends. Multi-scale features contain overall global information and local detailed information that is useful for image restoration tasks [21].

Multi-scale feature extraction can be achieved through two main approaches. The first method involves employing parallel Conv with different kernel sizes, followed by merging the features obtained from each parallel branch across channels. This structure, illustrated in Figure 1a as the separation–transformation–fusion structure of Inception [22], utilizes 1×1 Conv, 3×3 Conv, 5×5 Conv, and 3×3 maximum pooling operations to obtain features at different scales. The second method utilizes sampling to acquire feature maps at different scales. In [23], the authors proposed a feature pyramid network (FPN) that samples features from various layers at different scales for prediction, as depicted in Figure 1b. This approach effectively handles scale variations while maintaining a balance between expressive power, speed, and resource consumption. Unlike traditional image pyramid methods, the FPN approach eliminates the need to generate images at different scales before feature extraction. In the widely used U-Net method [24], pooling operations are employed to increase feature map channels during downsampling, and feature maps of different scales are fused during upsampling. While this technique reduces parameter counts and enhances network inference speed, it also leads to a loss of feature information, which can hinder image denoising tasks. In contrast, the MWCNN approach [25] replaces pooling operations with wavelet transforms to reduce the feature map size. Since wavelet transforms are reversible, this network can extract multi-scale features while preserving as many image texture details as possible.

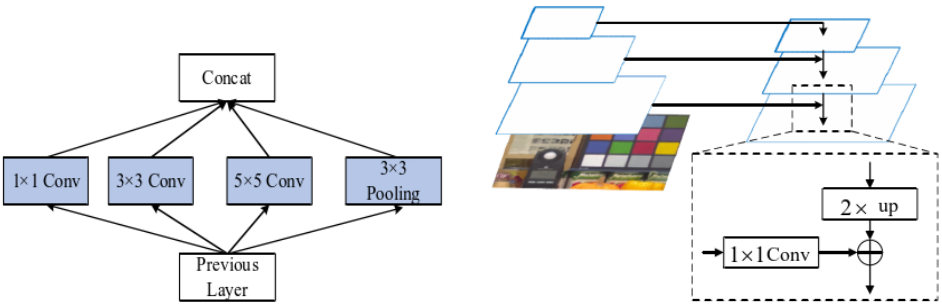


Figure 1. Illustration of multi-scale feature extraction. (**a**) Inception [22]; (**b**) FPN [23].

2.2. Attention Mechanism

Attention mechanisms (AM) [26] are based on studies of the human eye's vision. When observing a given object, the human eye first focuses on its global information, followed by local details. Introducing an AM to neural networks helps the network pay attention to and exploit important features. AMs can take different forms, including hard attention [27], soft attention, and self-attention. Soft attention mechanisms are preferred because they assign weights based on the relevance of different parts of the input image rather than making decisions based only on a subset of input image pixels as performed in hard attention mechanisms. Spatial attention (SA) [28] and CA [29] are two types of soft AMs. SA assigns varying weights to different spatial locations on the input feature map, while CA uses correlations and dependencies between channels to compress and reconstruct feature maps, thereby improving the network's ability to learn and represent image features.

The AM approach is particularly well-suited for denoising networks for two key reasons. Firstly, the network can automatically learn to use the AM without requiring any additional training steps, and secondly, the AM provides a clearer modeling ability for neural networks, making it easier to understand how the network solves the problem of interest.

3. Noise Model

We assume that $\boldsymbol{x}$ is a noise-free image, $\boldsymbol{y}$ is a corresponding noisy image, and $x_{i,j}$ is the pixel value at location (i, j). For the AWGN model, the relationship between these elements can be written as

$$y_{i,j} = x_{i,j} + G, \tag{1}$$

where G is an independent and identically distributed zero-mean AWGN.

There are two types of IN: salt–pepper impulse noise (SPIN) and random value impulse noise (RVIN). SPIN uses two fixed extreme values of $n_{min} = 0$ for pepper noise and $n_{max} = 255$ for salt noise to corrupt the image, whereas RVIN uses any value in the range $[n_{min}, n_{max}]$ to corrupt the image. The SPIN and RVIN models are as follows.

SPIN model:

$$y_{i,j} = \begin{cases} x_{i,j}, & \text{the probability } 1 - p_{sp} \\ n_{min}, & \text{the probability } p_{sp}/2 \\ n_{max}, & \text{the probability } p_{sp}/2 \end{cases}. \tag{2}$$

where p_{sp} denotes the probability of SPIN.

RVIN model:

$$y_{i,j} = \begin{cases} x_{i,j}, & \text{the probability } 1 - p_r \\ n_{i,j}, & \text{the probability } p_r \end{cases}. \tag{3}$$

where $n_{i,j}$ is a random pixel value in the range $[n_{min}, n_{max}]$ at the location (i, j) and p_r denotes the probability of RVIN.

In this paper, three types of mixed noise models are considered [20]:

(1) AWGN mixed with SPIN:

$$y_{i,j} = \begin{cases} x_{i,j} + G, & \text{the probability } 1 - p_{sp} \\ n_{min}, & \text{the probability } p_{sp}/2 \\ n_{max}, & \text{the probability } p_{sp}/2 \end{cases}. \tag{4}$$

(2) AWGN mixed with RVIN:

$$y_{i,j} = \begin{cases} x_{i,j} + G, & \text{the probability } 1 - p_r \\ n_{i,j}, & \text{the probability } p_r \end{cases}. \tag{5}$$

(3) AWGN mixed with RVIN plus SPIN:

$$y_{i,j} = \begin{cases} x_{i,j} + G, & \text{the probability } (1 - p_{sp})(1 - p_r) \\ n_{min}, & \text{the probability } p_{sp}/2 \\ n_{max}, & \text{the probability } p_{sp}/2 \\ n_{i,j}, & \text{the probability } p_r(1 - p_{sp}) \end{cases}. \quad (6)$$

4. MSNUNet

Most current mixed noise removal algorithms achieve good denoising results by utilizing local information or prior knowledge of the image. However, when the noise ratio increases or more complex noise, such as combined AWGN, SPIN, and RVIN, is encountered, the accuracy of these methods in estimating the noise distribution significantly decreases. This limitation hinders the model's ability to accurately model the noise and ultimately leads to the loss of image texture details. Additionally, although existing deep learning-based denoising models have shown promising results, the majority of these models perform denoising on low-resolution image blocks without considering the global information of the image. This limitation disrupts the overall structure and consistency of the image, thereby restricting the denoising performance of the model. In this paper, we propose MSNUNet for mixed noise removal. Firstly, the MSU-Subnet is introduced to enable the network to process high-resolution images more deeply and generate diverse receptive fields, capturing rich local and global features. Then, by integrating the MSCAM into the nested UNet architecture, MSNUNet is able to aggregate local and global features, which assist the model in preserving the overall structure and consistency of the image, resulting in improved denoising performance. Lastly, by introducing a channel attention block (CAB) in the MSCAM, the model enhances its ability to learn and extract important features, thereby preserving more image details.

4.1. Overall Pipeline

Figure 2 shows the architecture of MSNUNet for denoising images corrupted by mixed noise. Given an image $I \in \mathbb{R}^{H \times W \times 1}$, where $H \times W$ represents the spatial dimensions of the image, MSNUNet first applies a 3×3 Conv to extract low-level image features $F_0 \in \mathbb{R}^{H \times W \times C}$, where C denotes the number of channels. These features are then processed by a nine-block symmetric encoder–decoder structure to obtain the depth features $F_d \in \mathbb{R}^{H \times W \times C}$. The encoder blocks, including encoder1 and encoder2, and the decoder blocks, including decoder1 and decoder2, are filled with MSU-Subnet, a well-configured U-shaped subnetwork. By utilizing MSU-Subnet, MSNUNet effectively captures local and global features at multiple scales from high-resolution images. In the subsequent blocks with lower-resolution feature maps, downsampling further would result in the loss of valuable image information. To address this issue, we employ MSCAM, which combines local feature extraction with a CAB, allowing the network to extract local features while capturing channel correlations. This approach enables MSNUNet to efficiently extract multi-level features within blocks and aggregate multi-level features between blocks. Specifically, starting from the low-level feature F_0, the encoder gradually reduces the spatial size of the feature map while increasing the channel capacity. The decoder takes the potential feature $F_1 \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times 16C}$ as input and learns the noise distribution while gradually recovering the image resolution. Pixel-unshuffling and pixel-shuffling operations are applied during feature downsampling and upsampling, respectively [30]. The encoder is connected to the decoder through a skip connection to facilitate the image recovery process [24]. Before this connection, the output of the decoder is upsampled, and a summation operation is used to ensure a consistent number of channels. These design choices have resulted in improved quality, as described in the experimental section (Section 5). Finally, a 3×3 Conv is applied to the output features of the final decoder to generate the residual image $R \in \mathbb{R}^{H \times W \times 1}$. This residual image is then added to the noisy image I to obtain the restored image, $\hat{I} = I + R$. In the following section, we will describe the core components of MSNUNet, including MSCAM and MSU-Subnet, which are used to extract multi-scale features from the image.

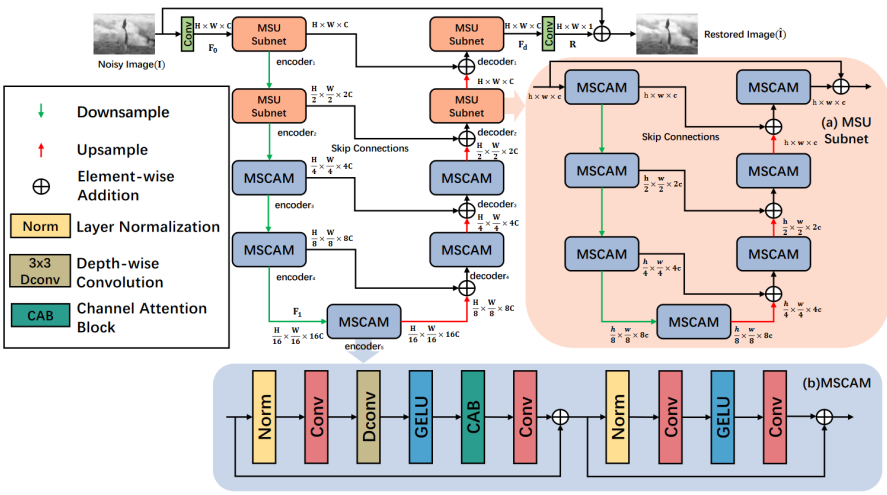


Figure 2. MSNUNet architecture for mixed noise removal.

4.2. MSCAM

In recent years, CNNs have gained wide adoption in image processing due to their remarkable performance. Conv serves as a fundamental building block of CNNs and typically includes Conv, BN, and ReLU functions. However, the standard Conv has inherent limitations in extracting exclusively local features [31], which poses a significant disadvantage for image processing. To overcome this limitation, MSNUNet utilizes MSU-Subnet to extract multi-scale features that are more diverse than those obtained through a standard Conv. These features consist of high-resolution feature maps with precise spatial information and low-resolution feature maps with reliable semantic information. To effectively integrate these rich features, we follow the component arrangement described in [32] and propose a new module called MSCAM, depicted in Figure 2b. MSCAM consists of two steps.

In step 1 of MSCAM, we apply layer normalization (LN) to normalize the input feature maps. The normalized feature maps are then processed using a 1×1 Conv, followed by a 3×3 depth-wise Conv (DConv). The DConv offers greater efficiency compared to traditional Conv as it has fewer parameters and lower computational costs. The Gaussian error linear unit (GELU) activation function is employed to implement the non-linear mapping relationship. The resulting feature maps are fed into the CAB, which captures the correlation between global feature channels. The internal structure of the CAB is illustrated in Figure 3. Finally, a 1×1 Conv is utilized to aggregate these features and output them to the second step of the module.

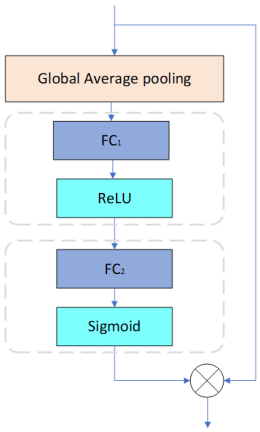


Figure 3. CAB internal structure.

The CAB implementation can be represented by the following equations [29], where the input is denoted as c and the output is denoted as c_3. These calculations are shown in Equations (7)–(9),

$$c_1 = FC_1[g(c)] \tag{7}$$

$$c_2 = FC_2[ReLU(c_1)] \tag{8}$$

$$c_3 = c * Sigmoid(c_2) \tag{9}$$

where the function g in Equation (7) represents the global average pooling. The input feature map, c, undergoes global average pooling followed by a fully connected layer, FC_1, to obtain c_1. Subsequently, c_1 is passed through the ReLU activation function and another fully connected layer, FC_2 , to obtain the compressed image features, c_2. The Sigmoid function is then applied to c_2, yielding weight coefficients between channels. Finally, these weight coefficients are multiplied with the input c to obtain the final result.

In Step 2, we employ a 1×1 Conv to enable the interaction of information among the diverse features acquired from the previous stage.

The incorporation of the CAB in MSCAM enhances the network's focus on important features [33], leading to the improved utilization and preservation of image texture details. Furthermore, residual connections are introduced to enhance the network's performance by aiding in the reconstruction of neglected high-frequency feature information.

4.3. MSU-Subnet

In current CNN designs, such as VGG [34], ResNet [35], and DenseNet [36], small kernels of sizes of 1×1 or 3×3 are commonly used for feature extraction. However, these small kernels have limited receptive fields, resulting in shallow output feature maps that only capture local features and fail to capture global features. To overcome this limitation, we propose a novel multi-scale feature extraction subnetwork, namely MSU-Subnet, to capture multi-scale features in high-resolution feature maps. Figure 2a showcases the structure of MSU-Subnet, which consists of three main components:

(i) **A U-shaped encoder-decoder structure:** The subnetwork takes intermediate feature maps (with a size of $h \times w$ and a number of channels equal to c) as inputs and employs a U-shaped architecture with seven blocks to extract multi-scale features. By progressively downsampling the feature maps and encoding them into a high-resolution feature map (with a size of $h \times w$ and a number of channels equal to c) through progressive upsampling, skip connections, and Convs, this structure effectively avoids the loss of fine details encountered with direct upsampling at larger scales. Additionally, by extracting features from deeper levels, the network can capture more diverse receptive fields and richer local and global features. The extracted multi-scale features can represent noise detail features of various granularity, enabling the network to capture a more accurate noise distribution and enhance the robustness of the model.

(ii) **MSCAM:** Serving as the base module for both MSU-Subnet and the entire network, MSCAM aggregates multi-scale features within the network. Not only does MSCAM aggregate multi-scale features within the network; it also utilizes CA to extract the correlation between feature channels. This allows the network to selectively attend to relevant features, thereby enhancing the effectiveness of feature extraction.

(iii) **Residual connection:** Experimental results indicate that the denoising quality of a network tends to decrease beyond a certain number of layers, potentially causing image degradation during network training. To mitigate this problem, residual connections are utilized to learn the residual mapping of the stacked layers, enabling the easier training of deeper networks.

By allowing the network to extract features at multiple scales, MSU-Subnet enhances the capabilities of feature extraction. In addition, the U-structure of MSU-Subnet offers a low computational overhead, as most operations and manipulations are applied to the downsampled feature maps.

4.4. Loss Function

In order to ensure a fairer and more robust denoising model, we employ the peak signal-to-noise ratio (PSNR) as a loss function for updating parameters [37]. The loss function can be represented as

$$MSE(y,x) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(y_{i,j} - x_{i,j})^2 \tag{10}$$

$$L(\theta) = 10 \cdot \log_{10}(255^2/MSE(y,x)) \tag{11}$$

where N and M are the image dimensions. Additionally, y and x stand for a noise-free image and the corresponding noisy image, respectively. These metrics are appealing for several reasons, including because they are easy to calculate, possess clear physical interpretations, and are mathematically convenient for optimization purposes.

5. Experiment and Analysis

To evaluate the denoising performance of the proposed MSNUNet, we applied the model to three benchmark datasets: BSD100 [38], Set12 [39], and Urban100 [40]. These datasets are widely used for image denoising tasks. The BSD100 [38] dataset consists of 100 real-world images capturing diverse scenes. Similarly, the Set12 [39] consists of 12 grayscale images that are widely used for image denoising tests. The Urban100 [40] dataset contains 100 urban scene images with complex textures. These datasets provide valuable resources for evaluating and comparing the efficacy of various image denoising algorithms. Thus, we conducted experiments on these three datasets to ensure fairness and reliability in the results. In Section 5.1, we discuss the experimental setup and provide details about the datasets utilized. The experimental results demonstrating the performance of the proposed MSNUNet are presented in Section 5.2. In Section 5.3, we discusses the proposed algorithm in comparison with other recent algorithms. Finally, in Section 5.4, we present extensive ablation experiments to evaluate the key components of MSNUNet.

5.1. Experiment Setup and Datasets

Implementation details. The proposed MSNUNet is an end-to-end trainable model with no pre-trained network, implemented using PyTorch 1.8.0 and a single NVIDIA RTX 3090 GPU. Specifically, we trained the model for a total of 300,000 iterations using the Adam [41] optimizer. Each 10,000 iterations corresponded to one epoch, and the entire training process required 30 epochs in total. The exponential decay rate parameters β_1, β_2, and weight decay were set as 0.9, 0.9, and 0, respectively. The initial learning rate was set to 1×10^3, gradually decreasing to 1×10^6 using the cosine annealing schedule [42]. Patch training and full image testing lead to performance degradation and denoised images with patch artifacts, which we addressed using a test-time local converter [43].

Noise ratio. We considered three types of mixed noise: AWGN+SPIN, AWGN+RVIN, and AWGN+RVIN+SPIN. For the first type, the standard deviation σ of the AWGN ranged from 20 to 30 in steps of 5, and the SPIN ratios were set to 15%, 30%, and 40%. For the second type, the σ of the AWGN ranged from 15 to 25 in steps of 5, and the RVIN ratio varied from p_r = 5% to 15% in steps of 5%. For the third type, the σ of the AWGN ranged from 5 to 15 in increments of 5, the RVIN ratio varied from p_r = 5% to 15% in increments of 5%, and the SPIN ratio varied from p_{sp} = 50% to 30% in decrements of 10%.

Training datasets. We trained the proposed MSNUNet on the DIV2K [44] dataset, which consists of 800 high-quality images for the training set and 100 images for the vali-

dation set. These images have an average resolution of around 1920 × 1080. Additionally, these datasets contain abundant details and intricate textures, making them well-suited for evaluating and comparing the performance of diverse image processing algorithms. The patch size and batch size were set to 256 × 256 and 8, respectively. We added three types of mixed noise to the patches for each of the 800 training images. In addition, the same three types of mixed noise used in the training set were applied to the 100 validation images.

5.2. Results

This section describes the extensive experiments performed to evaluate MSNUNet. For all three noise types, we compared MSNUNet with six competing algorithms, including two traditional methods (WESNR [10] and LSLR [11]) and four CNN-based methods (TL-CNN [14], VA-CNN [16], DeGAN [18], and SACNN [20]). After applying the competing methods, we calculated the PSNR and structural similarity index (SSIM) metrics of each method's processing results to measure the effectiveness of the diverse mixed noise removal algorithms and evaluate the quality of the denoising results. In Tables 1–3, for each σ of a given test set, the first line shows the PSNR, and the second line shows the SSIM.

As shown in Table 1, for the mixed AWGN+SPIN case, the denoising performance of MSNUNet outperforms all competing methods, which demonstrates the superiority of MSNUNet. Figure 4 illustrates the visual appearance of "Barbara" from the Set12 dataset. Figure 4b shows an image of a parrot corrupted by AWGN+SPIN (σ = 25, p_{sp} = 30%), while Figure 4c–i shows the processing results of the six compared algorithms and the proposed MSNUNet. Compared to the other methods, MSNUNet preserves more of the fine texture of the eye region, resulting in a significantly improved visualization.

Figure 4. Denoising results on the image "Barbara". (**a**) Original image. (**b**) Image corrupted with AWGN+SPIN (σ = 25, p_{sp}=30%). (**c**) WESNR [10]. (**d**) LSLR [11]. (**e**) TL-CNN [14]. (**f**) VA-CNN [16]. (**g**) DeGAN [18]. (**h**) SACNN [20]. (**i**) MSNUNet.

Similarly, for the mixed AWGN+RVIN case, MSNUNet achieves the best denoising performance (Table 2). With increasing RVIN proportions and decreasing AWGN proportions, the denoising performance of the competing algorithms, except for LSLR [11], steadily improves, with the most significant improvements beinobserved for MSNUNet. The denoising results for image 24077 of the BSD100 dataset are shown in Figure 5, where Figure 5b is corrupted by AWGN+RVIN (σ = 20, p_r = 10%). In Figure 5, the visual effect obtained by MSNUNet is more pleasing than all the other results.

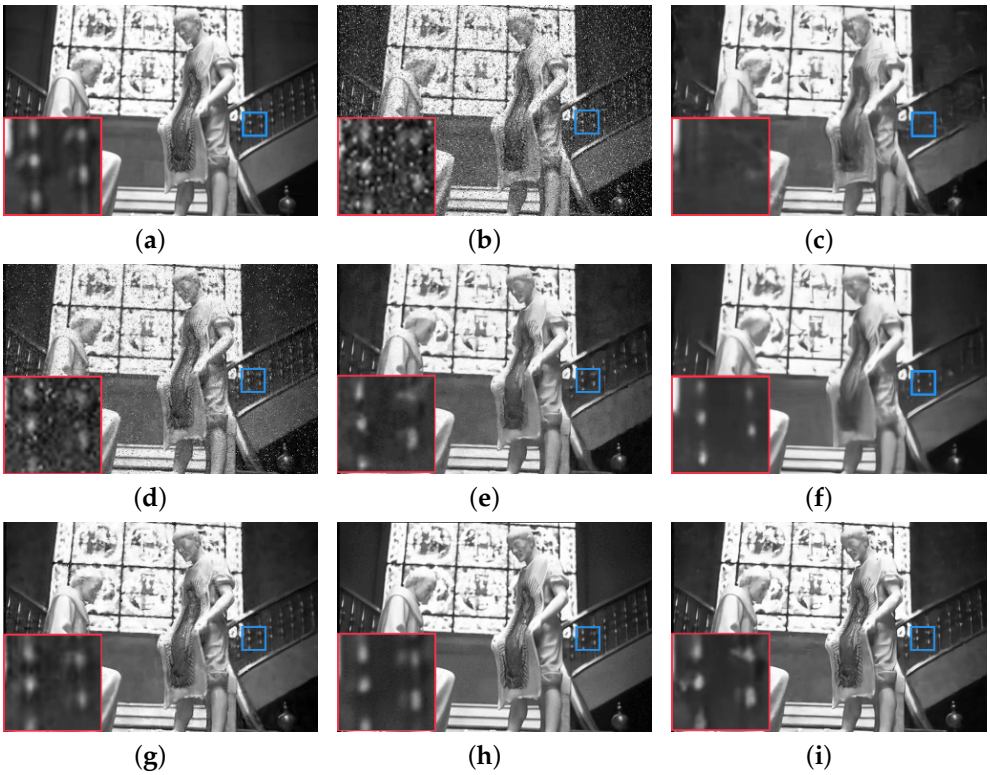


Figure 5. Denoising results on image 24077 of the BSD100 dataset. (**a**) Original image. (**b**) Image corrupted with AWGN+RVIN (σ = 20, p_r = 10%). (**c**) WESNR [10]. (**d**) LSLR [11]. (**e**) TL-CNN [14]. (**f**) VA-CNN [16]. (**g**) DeGAN [18]. (**h**) SACNN [20]. (**i**) MSNUNet.

For the mixed AWGN+SPIN+RVIN case, Table 3 reveals that our proposed MSNUNet delivers superior performance compared to nearly all the competing models at various mixed noise levels. Although the PSNR metric of SACNN [20] exceeds that of MSNUNet at σ = 5, p_r = 5%, and p_{sp} = 50%, our model exhibits exceptional denoising capability as the levels of AWGN and RVIN increase, highlighting its remarkable robustness. The denoising results for image 119082 of the BSD100 dataset are shown in Figure 6. Figure 6b shows images corrupted by AWGN+SPIN+RVIN (σ = 10, p_r = 10%, p_{sp} = 40%). As shown by the denoised images in Figure 6, DeGAN [18] and SACNN [20] effectively eliminate mixed noise and reconstruct clear image structures. WESNR [10] and LSLR [11] cannot clearly reconstruct the image's content, while TL-CNN [14] and VA-CNN [16] blur the texture and structure of the images; however, MSNUNet reconstructs fine textures more effectively than all other competing algorithms.

To evaluate the generality of MSNUNet, we tested the performance of the model on the color versions of the BSD100 [38] and Urban100 [40] datasets under six different mixed noise ratios. The results are shown in Table 4. The denoising results of the proposed model are presented in Figures 7 and 8. It can be observed that MSNUNet achieves impressive performance in denoising color images. This demonstrates the versatility and effectiveness of our approach in handling color image denoising tasks.

Table 1. The PSNR (dB) and SSIM (%) results of mixed noise removal (AWGN + SPIN). The **best results** are marked in **bold**.

Dataset		$p_{sp} = 15\%$							$p_{sp} = 30\%$							$p_{sp} = 40\%$						
		WESNR	LSLR	TL-CNN	VA-CNN	DeGAN	SACNN	MSNUNet	WESNR	LSLR	TL-CNN	VA-CNN	DeGAN	SACNN	MSNUNet	WESNR	LSLR	TL-CNN	VA-CNN	DeGAN	SACNN	MSNUNet
BSD100	$\sigma = 20$	27.16	28.70	28.62	28.79	28.47	29.07	**29.76**	26.70	28.05	27.93	27.77	27.87	28.63	**29.17**	25.78	27.65	27.42	26.98	27.47	28.12	**28.81**
		74.36	76.31	76.84	77.14	77.62	78.30	**84.30**	72.22	75.43	76.14	76.48	76.44	77.58	**82.68**	71.26	74.23	73.94	75.99	74.25	76.15	**81.69**
	$\sigma = 25$	26.20	27.71	27.62	27.85	27.58	27.95	**28.77**	25.63	27.17	26.99	27.15	26.94	27.54	**28.32**	24.81	26.77	26.50	26.69	26.43	27.11	**27.97**
		72.55	74.63	74.85	75.31	74.73	76.36	**81.04**	70.64	72.23	72.44	73.04	74.35	75.74	**79.76**	70.43	71.02	71.13	72.71	72.65	74.12	**78.53**
	$\sigma = 30$	25.31	26.89	26.77	27.15	26.85	27.21	**28.03**	24.91	26.51	26.02	26.55	26.43	27.04	**27.61**	24.46	26.13	25.49	26.11	26.10	26.83	**27.32**
		70.11	70.92	71.56	72.37	72.14	74.81	**78.42**	69.12	70.43	70.86	71.14	72.61	74.46	**77.04**	69.11	69.38	70.25	70.47	70.64	72.77	**76.06**
Set12	$\sigma = 20$	28.84	30.41	30.28	30.46	30.15	30.75	**31.45**	28.46	29.79	29.70	29.53	29.78	30.35	**30.91**	27.55	29.43	29.15	28.75	29.20	29.88	**30.57**
		78.36	80.34	80.86	81.19	81.63	82.35	**88.33**	77.06	80.29	80.98	81.37	81.28	82.41	**87.52**	76.57	79.50	79.23	81.3	79.5	81.41	**86.96**
	$\sigma = 25$	27.65	29.35	29.25	29.49	29.22	29.60	**30.39**	27.32	28.85	28.64	28.83	28.62	29.21	**29.99**	26.47	28.44	28.19	28.41	28.13	28.79	**29.66**
		77.84	79.92	80.15	80.66	80.08	81.70	**86.34**	76.53	78.21	78.42	79.01	80.24	81.69	**85.70**	76.94	77.57	77.67	79.24	79.20	80.66	**85.05**
	$\sigma = 30$	26.91	28.45	28.36	28.73	28.42	28.83	**29.62**	26.49	28.09	27.61	28.17	28.06	28.62	**29.21**	26.06	27.70	27.08	27.71	27.67	28.41	**28.92**
		76.42	77.21	77.88	78.65	78.44	81.09	**84.75**	76.03	77.35	77.83	78.12	79.56	81.40	**83.99**	76.44	76.68	77.57	77.86	77.94	80.11	**83.40**
Urban100	$\sigma = 20$	28.87	30.44	30.38	30.52	30.21	30.82	**31.49**	28.40	29.70	29.59	29.43	29.71	30.32	**30.84**	27.37	29.22	29.04	28.55	29.08	29.72	**30.41**
		81.52	83.51	84.05	84.38	84.83	85.46	**91.49**	80.21	83.44	84.06	84.44	84.41	85.53	**90.64**	79.60	82.55	82.26	84.39	82.57	84.55	**90.04**
	$\sigma = 25$	27.74	29.22	29.15	29.38	29.07	29.45	**30.29**	27.17	28.71	28.50	28.68	28.48	29.06	**29.86**	26.20	28.18	27.90	28.12	27.87	28.52	**29.39**
		81.11	83.24	83.37	83.88	83.34	84.97	**89.60**	79.86	81.44	81.64	82.31	83.58	84.94	**88.98**	80.11	80.72	80.77	82.36	82.31	83.79	**88.19**
	$\sigma = 30$	26.76	28.34	28.21	28.58	28.31	28.67	**29.47**	26.33	27.95	27.45	27.95	27.87	28.43	**29.02**	25.76	27.42	26.81	27.40	27.43	28.15	**28.63**
		79.83	80.61	81.30	82.03	81.81	84.56	**88.13**	79.39	80.74	81.12	81.38	82.89	84.69	**87.30**	79.63	79.92	80.76	80.98	81.10	83.32	**86.56**

Table 2. The PSNR (dB) and SSIM (%) results of mixed noise removal (AWGN + RVIN). The **best results** are marked in **bold**.

Dataset	$\sigma = 25, r = 5\%$							$\sigma = 20, r = 10\%$							$\sigma = 15, r = 15\%$						
	WESNR	LSLR	TL-CNN	VA-CNN	DeGAN	SACNN	MSNUNet	WESNR	LSLR	TL-CNN	VA-CNN	DeGAN	SACNN	MSNUNet	WESNR	LSLR	TL-CNN	VA-CNN	DeGAN	SACNN	MSNUNet
BSD100	24.17	23.14	25.97	24.78	27.13	27.73	**28.74**	24.56	19.12	26.61	24.91	27.73	28.11	**29.22**	25.14	16.70	27.23	25.93	28.04	28.77	**29.92**
	73.02	72.56	73.64	74.25	77.56	78.14	**80.89**	73.38	71.58	78.63	74.14	77.62	79.36	**82.74**	74.07	67.32	77.84	74.98	78.11	80.74	**85.26**
Set12	25.87	24.85	27.64	26.49	28.85	29.44	**30.44**	26.37	20.91	28.42	26.70	29.55	29.93	**31.03**	27.04	18.62	29.14	27.85	29.95	30.63	**31.81**
	78.69	78.17	79.30	79.84	81.23	83.75	**86.51**	78.39	76.58	83.69	79.18	83.6	84.41	**87.75**	78.15	71.43	81.95	79.09	83.17	84.85	**89.34**
Urban100	25.87	24.86	27.67	26.50	28.84	29.45	**30.47**	26.36	20.86	28.38	26.66	29.49	29.88	**30.99**	26.94	18.45	28.99	27.68	29.79	30.51	**31.69**
	82.09	81.65	82.72	83.30	85.64	87.22	**89.93**	81.60	79.81	85.87	82.35	86.79	87.51	**90.93**	80.95	74.25	84.73	81.81	85.96	87.65	**92.14**

Table 3. The PSNR (dB) and SSIM (%) results of mixed noise removal (AWGN + RVIN + SPIN). The **best results** are marked in **bold**.

Dataset	$\sigma = 5, p_r = 5\%, p_s = 50\%$							$\sigma = 10, p_r = 10\%, p_s = 40\%$							$\sigma = 15, p_r = 15\%, p_s = 30\%$						
	WESNR	LSLR	TL-CNN	VA-CNN	DeGAN	SACNN	MSNUNet	WESNR	LSLR	TL-CNN	VA-CNN	DeGAN	SACNN	MSNUNet	WESNR	LSLR	TL-CNN	VA-CNN	DeGAN	SACNN	MSNUNet
BSD100	21.53	19.36	24.01	22.84	28.46	**32.66**	31.52	21.24	17.30	23.77	22.63	28.15	29.20	**29.93**	20.84	16.04	23.80	22.18	27.76	27.46	**28.74**
	70.29	63.52	72.65	73.21	77.58	87.69	**90.77**	69.53	59.24	70.48	71.35	74.03	76.75	**86.37**	67.32	55.31	69.82	69.76	75.54	75.98	**82.35**
Set12	23.49	21.31	26.00	24.80	30.41	**34.62**	33.48	23.04	19.07	25.55	24.41	29.96	30.96	**31.71**	22.49	17.68	25.46	23.86	29.43	29.12	**30.40**
	72.69	65.85	75.08	75.58	78.92	90.06	**93.15**	73.08	62.77	74.06	74.94	78.56	80.26	**89.92**	72.32	60.26	74.74	74.72	78.52	79.98	**87.32**
Urban100	22.86	20.69	25.40	24.19	29.80	**34.03**	32.88	22.46	18.49	25.01	23.82	29.34	30.45	**31.15**	22.12	17.36	25.12	23.47	29.09	28.74	**30.04**
	74.12	67.36	76.39	76.97	78.37	91.43	**94.56**	75.29	64.94	76.21	77.11	79.69	82.41	**92.08**	74.92	62.92	77.44	77.41	81.20	81.66	**89.99**

Table 4. The PSNR (dB) and SSIM (%) results for color image denoising.

Dataset	$\sigma = 20, p_s = 15\%$	$\sigma = 25, p_s = 30\%$	$\sigma = 30, p_s = 40\%$	$\sigma = 25, p_r = 5\%$	$\sigma = 20, p_r = 10\%$	$\sigma = 15, p_r = 15\%$
BSD100	31.69 88.46	30.28 85.64	29.15 83.97	30.59 86.43	31.10 87.54	31.82 89.44
Urban100	32.96 87.73	31.32 87.16	30.08 85.41	31.92 87.53	32.47 87.66	33.13 89.52

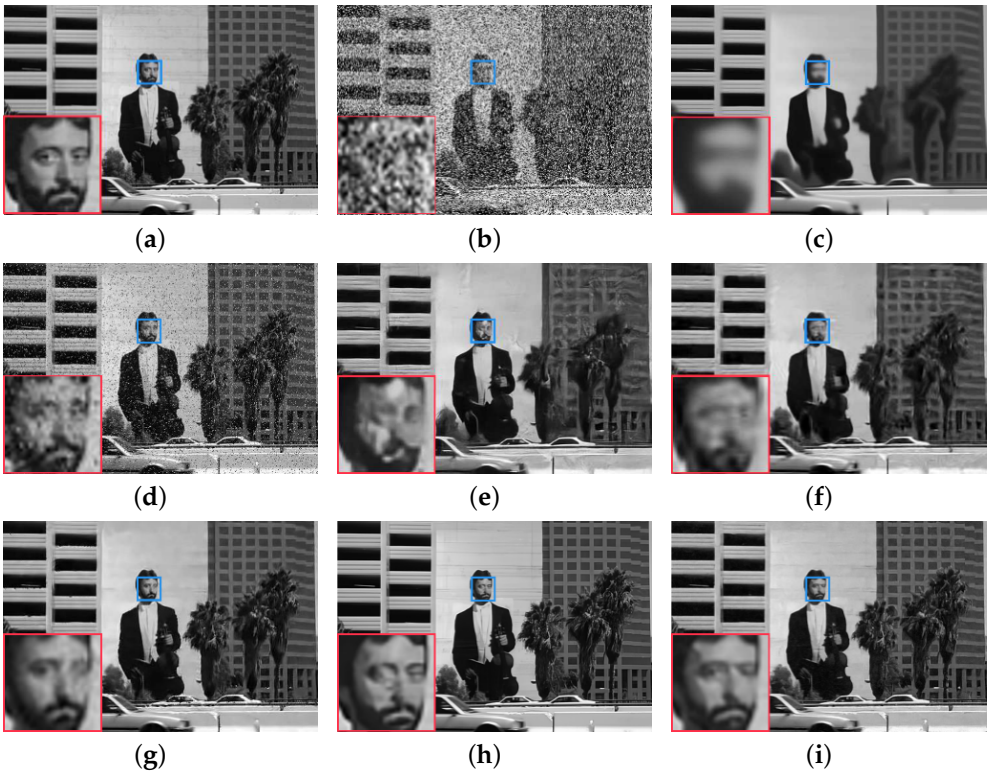


Figure 6. Denoising results on image 119082 of the BSD100 dataset. (**a**) Original image. (**b**) Image corrupted with AWGN + SPIN + RVIN (σ = 10, p_r = 10%, p_{sp} = 40%). (**c**) WESNR [10]. (**d**) LSLR [11]. (**e**) TL-CNN [14]. (**f**) VA-CNN [16]. (**g**) DeGAN [18]. (**h**) SACNN [20]. (**i**) MSNUNet.

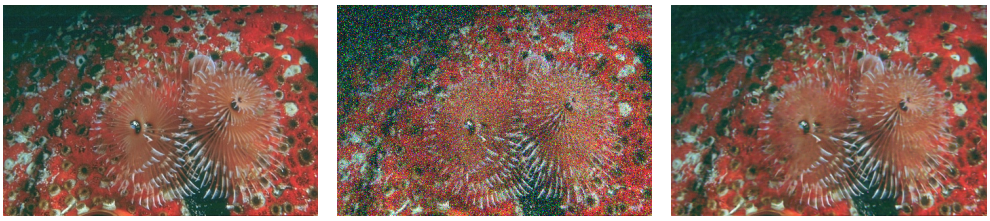


Figure 7. Denoising results on image 12084 of the BSD100 dataset. (**a**) Original image. (**b**) Image corrupted with AWGN + SPIN (σ = 20, p_{sp} = 15%). (**c**) MSNUNet.

(**a**) (**b**) (**c**)

Figure 8. Denoising results on image 105025 of the BSD100 dataset. (**a**) Original image. (**b**) Image corrupted with AWGN + RVIN (σ = 15, p_r = 15%). (**c**) MSNUNet.

5.3. Discussion

As mentioned in Section 5.2, the proposed MSNUNet algorithm demonstrates powerful performance in removing mixed noise, specifically AWGN+RVIN and AWGN + RVIN + SPIN. It surpasses state-of-the-art methods in terms of quality metrics and the visual appearance of the images. The effectiveness of MSNUNet in denoising can be attributed to its efficient extraction of multi-scale features. MSU-Subnet generates multiple receptive fields, providing rich local and global features for extraction. Given that mixed noise has a more complex distribution than a single source of noise, it is crucial to incorporate both local and global image information in the process of removing mixed noise. On the other hand, MSCAM utilizes a CAB to dynamically adjust the weights of individual channels in the global feature space. This helps in aggregating multi-scale features and allocating weights to relevant features based on their channel-wise correlations, thereby preserving more image details.

In the task of removing mixed noise, the network's primary objective is to learn the mapping relationship between clean and noisy images. DEGAN [18] used a generative adversarial network for this purpose. DeGAN [18] utilizes a generator to learn the direct mappings between clean and noisy images, generating new images with a distribution similar to clean images. The generated images are then evaluated by a discriminator, providing feedback to the generator for generating more realistic clean images. However, the training process of GANs is unstable, leading to convergence difficulties for the generator and discriminator. Consequently, DeGAN [18] performs poorly when image details are severely corrupted. In our approach, the symmetric UNet structure effectively extracts image features, and the residual connections stabilize the training process, preventing gradient vanishing.

In SACNN [20], local image features are extracted using Conv, and a hybrid attention mechanism is employed to learn weights for the image, incorporating SA and CA. With the powerful feature extraction capability of Conv, SACNN [20] effectively extracts local features and assigns weights through the hybrid attention mechanism in both the channel and spatial dimensions. This empowers the network to fully utilize valuable features and restore the corrupted details of the image. In our approach, we also employ CA to learn weights for feature information in the channel dimension. However, our method goes beyond SACNN [20] by extracting multi-scale features from high-resolution images using MSU-Subnet, which includes both local and global features. Throughout the denoising process of MSNUNet, MSCAM is utilized to effectively aggregate multi-scale features. As a result, MSNUNet outperforms SACNN [20] in denoising performance.

5.4. Ablation Studies

In our ablation experiments, the denoising model was trained for only 150,000 iterations. The tests were performed on BSD100 [38] and analyzed for a challenging AWGN + SPIN case (σ = 25, p_{sp}=30%). Tables 5–7 indicate the quality of improvements in performance achieved for various configurations. We then describe the impact of each component.

Table 5. Influence of image size. The **best results** are marked in bold.

Image Size	FLOPs (B)	PSNR (dB)	SSIM (%)
128 × 128	**29.02**	27.84	78.41
192 × 192	63.26	28.08	79.25
256 × 256	116.11	**28.16**	**79.66**

Table 6. Influence of MSCAM. The **best results** are marked in bold.

Network	Component	FLOPs (B)	PSNR (dB)	SSIM (%)
MSNUNet	(a) Resblock	**95.83**	27.78	77.94
	(b) MSCAM-C	110.32	28.03	79.12
	(c) MSCAM-D	105.93	27.92	78.44
	(d) MSCAM	116.11	**28.15**	**79.67**

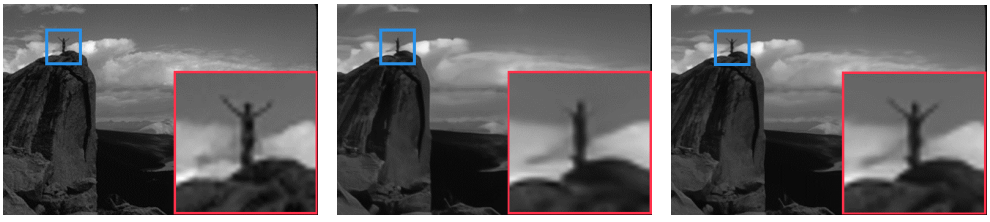
Table 7. Influence of MSU-Subnet. The **best results** are marked in bold.

Network	Component	FLOPs (B)	PSNR (dB)	SSIM (%)
UNet	(a) UNet with Resblock	**92.84**	27.64	77.82
	(b) UNet with MSCAM	108.46	28.01	79.10
MSNUNet	(c) U-block with Resblock	100.85	27.79	78.36
	(d) U-block with MSCAM	116.11	**28.15**	**79.66**

Effect of input size. We computed the FLOPs, PSNR, and SSIM for image sizes of 128 × 128, 192 × 192, and 256 × 256, respectively. As shown in Table 5, the PSNR gain becomes larger as the image size increases, and the FLOPs also increases. In this paper, we used patches with an image size of 256 × 256 as network inputs.

Effect of MSCAM. To fully validate MSCAM, we replaced all MSCAM blocks in MSNUNet with Conv-based Resblocks [35], while the others were left unchanged. In addition, we removed DConv and CAB from MSCAM, resulting in two MSCAM variants, MSCAM-C and MSCAM-D.

Table 6(d) demonstrates the exceptional progress achieved by our MSCAM approach, surpassing the standard Resblock (Table 6(a)) by an impressive 0.37 dB. Furthermore, the localization introduced by DConv enhances the robustness of MSCAM since its removal causes a decrease in PSNR (see Table 6(b)). In addition, including CAB produces a noteworthy enhancement of 0.23 dB, as revealed in Table 6(c). To further illustrate the effectiveness of MSCAM, Figure 9b,c shows the two cases of MSNUNet equipped with Resblock and MSCAM, respectively. It is evident that MSNUNet equipped with MSCAM restores additional details of the human within the image, which confirms that our method is able to retain more texture details.



(**a**) (**b**) (**c**)

Figure 9. Denoising results on image 14037 of the BSD100 dataset. (**a**) Original image. (**b**) MSNUNet equipped with Resblock. (**c**) MSNUNet equipped with MSCAM.

Effect of MSU-Subnet. Table 7(d) demonstrates that including a U-block design improves the denoising performance by 0.14 dB compared to a conventional U-network (see Table 7(b)). Furthermore, replacing the standard Resblock with MSCAM in the U-block

enhances the aggregation of multi-scale features, as shown by a decrease in PSNR upon its removal. Overall, MSNUNet contributes a substantial gain of 0.51 dB over the baseline (see Table 7(a)).

6. Conclusions

In this paper, we propose MSNUNet, a mixed noise removal method based on nested UNet and multi-scale feature extraction. In MSNUNet, the two-layer nested UNet structure can deeply extract multi-scale features from high-resolution images and aggregate these features more efficiently using the proposed MSCAM. This approach not only accurately models mixed noise by leveraging the richer local and global information in the original image but also effectively extracts important features of the image through CA, thus preserving more image texture details. The experimental results clearly demonstrate that MSNUNet can achieve leading quality measures and fine textures that outperform all other contemporary competing methods.

In the future, we will further develop our work in two aspects. On the one hand, we will focus on developing a lightweight solution for image denoising models. Due to the limited cost of hardware devices in real-world applications, most of the current deep learning-based denoising models cannot be deployed on hardware. We will explore two alternatives to address and alleviate this issue. Firstly, we will design more efficient neural network components, such as DConv, to reduce computational complexity. Secondly, we will optimize existing models using currently available lightweight techniques. On the other hand, we plan to design a versatile model for low-level computer vision tasks.

Author Contributions: Conceptualization, J.J. and Y.C.; methodology, J.J. and L.L.; software, L.L.; validation, L.L.; formal analysis, Y.Z. and Y.C.; investigation, J.J. and L.L.; resources, Y.C.; data curation, Y.Z.; writing—original draft preparation, L.L.; writing—review and editing, J.J. and L.L.; visualization, Y.C.; supervision, Y.Z.; project administration, Y.Z.; funding acquisition, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported in part by the National Natural Science Foundation of China under Grant 62001236, in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 20KJA520003, in part by the China Postdoctoral Science Foundation under Grant 2018M640441, and in part by the Six Talent Peaks Project of Jiangsu Province under Grant JY-051.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: https://paperswithcode.com/datasets (accessed on 21 May 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cheng, Z.; Qu, A.; He, X. Contour-Aware Semantic Segmentation Network with Spatial Attention Mechanism for Medical Image. *Vis. Comput.* **2022**, *38*, 749–762. [CrossRef]
2. Qi, G.; Hu, G.; Mazur, N.; Liang, H.; Haner, M. A Novel Multi-Modality Image Simultaneous Denoising and Fusion Method Based on Sparse Representation. *Computers* **2021**, *10*, 129. [CrossRef]
3. Geng, M.; Meng, X.; Yu, J.; Zhu, L.; Jin, L.; Jiang, Z.; Qiu, B.; Li, H.; Kong, H.; Yuan, J.; et al. Content-Noise Complementary Learning for Medical Image Denoising. *IEEE Trans. Med. Imaging* **2022**, *41*, 407–419. [CrossRef]
4. Ananthi, V. p.; Balasubramaniam, P.; Raveendran, P. Impulse Noise Detection Technique Based on Fuzzy Set. *IET Signal Process.* **2018**, *12*, 12–21. [CrossRef]
5. Garnett, R.; Huegerich, T.; Chui, C.; He, W. A Universal Noise Removal Algorithm with an Impulse Detector. *IEEE Trans. Image Process.* **2005**, *14*, 1747–1754. [CrossRef]
6. Tomasi, C.; Manduchi, R. Bilateral Filtering for Gray and Color Images. In Proceedings of the Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), Bombay, India, 7 January 1998; pp. 839–846.
7. Cai, J.-F.; Chan, R.H.; Nikolova, M. Two-Phase Approach for Deblurring Images Corrupted by Impulse plus Gaussian Noise. *Inverse Probl. Imaging* **2008**, *2*, 187–204. [CrossRef]

8. Xiao, Y.; Zeng, T.; Yu, J.; Ng, M.K. Restoration of Images Corrupted by Mixed Gaussian-Impulse Noise via L1–L0 Minimization. *Pattern Recognit.* **2011**, *44*, 1708–1720. [CrossRef]
9. Liu, J.; Tai, X.-C.; Huang, H.; Huan, Z. A Weighted Dictionary Learning Model for Denoising Images Corrupted by Mixed Noise. *IEEE Trans. Image Process.* **2013**, *22*, 1108–1120. [CrossRef]
10. Jiang, J.; Zhang, L.; Yang, J. Mixed Noise Removal by Weighted Encoding with Sparse Nonlocal Regularization. *IEEE Trans. Image Process.* **2014**, *23*, 2651–2662. [CrossRef]
11. Huang, T.; Dong, W.; Xie, X.; Shi, G.; Bai, X. Mixed Noise Removal via Laplacian Scale Mixture Modeling and Nonlocal Low-Rank Approximation. *IEEE Trans. Image Process.* **2017**, *26*, 3171–3186. [CrossRef]
12. Zhuang, L.; Ng, M.K. FastHyMix: Fast and Parameter-Free Hyperspectral Image Mixed Noise Removal. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 4702–4716. [CrossRef]
13. Liu, J.; Wu, J.; Xu, M.; Huang, Y. Plug-and-Play-Based Algorithm for Mixed Noise Removal with the Logarithm Norm Approximation Model. *Mathematics* **2022**, *10*, 3810. http://doi.org10.3390/math10203810. [CrossRef]
14. Islam, M.T.; Mahbubur Rahman, S.M.; Omair Ahmad, M.; Swamy, M.N.S. Mixed Gaussian-Impulse Noise Reduction from Images Using Convolutional Neural Network. *Signal Process. Image Commun.* **2018**, *68*, 26–41. [CrossRef]
15. Abiko, R.; Ikehara, M. Blind Denoising of Mixed Gaussian-Impulse Noise by Single CNN. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 1717–1721.
16. Wang, F.; Huang, H.; Liu, J. Variational-Based Mixed Noise Removal with CNN Deep Learning Regularization. *IEEE Trans. Image Process.* **2020**, *29*, 1246–1258. [CrossRef]
17. Jiang, J.; Yang, K.; Yang, J.; Yang, Z.-X.; Chen, Y.; Luo, L. A New Nonlocal Means Based Framework for Mixed Noise Removal. *Neurocomputing* **2021**, *431*, 57–68. [CrossRef]
18. Lyu, Q.; Guo, M.; Pei, Z. DeGAN: Mixed Noise Removal via Generative Adversarial Networks. *Appl. Soft Comput.* **2020**, *95*, 106478. [CrossRef]
19. Mafi, M.; Izquierdo, W.; Martin, H.; Cabrerizo, M.; Adjouadi, M. Deep Convolutional Neural Network for Mixed Random Impulse and Gaussian Noise Reduction in Digital Images. *IET Image Process.* **2020**, *14*, 3791–3801. [CrossRef]
20. Jiang, J.; Yang, K.; Xu, X.; Cui, Y. A Serial Attention Module-Based Deep Convolutional Neural Network for Mixed Gaussian-Impulse Removal. *IET Image Process.* **2023**, *17*, 1837–1851. [CrossRef]
21. Li, D.; Shi, G.; Wu, Y.; Yang, Y.; Zhao, M. Multi-Scale Neighborhood Feature Extraction and Aggregation for Point Cloud Segmentation. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 2175–2191. [CrossRef]
22. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
23. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
24. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, Proceedings of the 18th International Conference, Munich, Germany, 5–9 October 2015*; Springer International Publishing: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
25. Liu, P.; Zhang, H.; Zhang, K.; Lin, L.; Zuo, W. Multi-Level Wavelet-CNN for Image Restoration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–23 June 2018; pp. 773–782.
26. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.
27. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 6–11 July 2015; pp. 2048–2057.
28. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp.3–19.
29. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
30. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.
31. Bhaumik, G.; Verma, M.; Govil, M.C.; Vipparthi, S.K. ExtriDeNet: An Intensive Feature Extrication Deep Network for Hand Gesture Recognition. *Vis. Comput.* **2022**, *38*, 3853–3866. [CrossRef]
32. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
33. Li, Y.; Meng, J.; Zhu, Z.; Huang, X.; Qi, G.; Luo, Y. Context Convolution Dehazing Network with Channel Attention. In Proceedings of the 2021 5th Asian Conference on Artificial Intelligence Technology (ACAIT), Haikou, China, 29–31 October 2021; pp. 259–265.

34. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
36. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
37. Horé, A.; Ziou, D. Image Quality Metrics: PSNR vs. SSIM. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2366–2369.
38. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In Proceedings of the Proceedings Eighth IEEE International Conference on Computer Vision, ICCV 2001, Vancouver, BC, Canada, 7–14 July 2001; Volume 2, pp. 416–423.
39. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [CrossRef] [PubMed]
40. Huang, J.-B.; Singh, A.; Ahuja, N. Single Image Super-Resolution from Transformed Self-Exemplars. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5197–5206.
41. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
42. Loshchilov, I.; Hutter, F. SGDR: Stochastic Gradient Descent with Warm Restarts. *arXiv* **2016**, arXiv:1608.03983.
43. Chu, X.; Chen, L.; Chen, C.; Lu, X. Improving Image Restoration by Revisiting Global Information Aggregation. In *European Conference on Computer Vision;* Springer Nature: Cham, Switzerland, 2022; pp. 53–71.
44. Agustsson, E.; Timofte, R. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1122–1131.