mathematics

Article

NSNet: An N-Shaped Convolutional Neural Network with Multi-Scale Information for Image Denoising

Yifen Li [1] and Yuanyang Chen [2,*]

[1] School of Economics and Management, Changsha University, Changsha 410022, China
[2] School of Automation, Central South University, Changsha 410083, China
* Correspondence: 204612129@csu.edu.cn

Abstract: Deep learning models with convolutional operators have received widespread attention for their good image denoising performance. However, since the convolutional operation prefers to extract local features, the extracted features may lose some global information, such as texture, structure, and color characteristics, when the object in the image is large. To address this issue, this paper proposes an N-shaped convolutional neural network with the ability to extract multi-scale features to capture more useful information and alleviate the problem of global information loss. The proposed network has two main parts: a multi-scale input layer and a multi-scale feature extraction layer. The former uses a two-dimensional Haar wavelet to create an image pyramid, which contains the corrupted image's high- and low-frequency components at different scales. The latter uses a U-shaped convolutional network to extract features at different scales from this image pyramid. The method sets the mean-squared error as the loss function and uses the residual learning strategy to learn the image noise directly. Compared with some existing image denoising methods, the proposed method shows good performance in gray and color image denoising, especially in textures and contours.

Keywords: image denoising; wavelet transform; Unet; image pyramid; multi-scale features

MSC: 68U10

check for updates

Citation: Li, Y.; Chen, Y. NSNet: An N-Shaped Convolutional Neural Network with Multi-Scale Information for Image Denoising. *Mathematics* **2023**, *11*, 2772. https://doi.org/10.3390/math11122772

Academic Editor: Jakub Nalepa

Received: 4 June 2023
Revised: 15 June 2023
Accepted: 18 June 2023
Published: 19 June 2023

1. Introduction

Image denoising is one of the basic tasks of computer vision and is of wide interest to academia and industry, as it can effectively improve image quality. The purpose of image denoising is to remove noise from a corrupted image and restore its original content as much as possible. In many computer vision tasks, image denoising is often used as a preprocessing method to improve the practical performance of advanced computer vision tasks [1]. Over the past few decades, many outstanding image denoising methods, as shown in Figure 1, have been proposed, including filtering-based [2,3], sparse-representation-based [4–8], external-prior-based [9–12], low-rank-representation-based [13,14], and deep-learning-based methods [15–18].

Filtering-based methods were the first techniques to be applied to image denoising and rely on the self-similarity of images. Well-known approaches include Gaussian filtering, mean filtering, and median filtering. These three methods assume that the pixels in an image do not exist in isolation and have connections to other pixels. However, Buades et al. [2] found that similar pixels are not limited to local areas, and making full use of the redundant information in an image can improve the image denoising performance. Hence, they proposed a nonlocal mean filtering method (NLM) based on existing smoothing filtering methods. Although NLM can achieve good denoising performance, it needs to find a sufficient number of similar blocks when computing each pixel, which gives it high computational complexity. To solve this problem, Kostadin et al. [3] proposed a block-matching and 3D filtering (BM3D) method, which has a good denoising performance and fast computational speed.

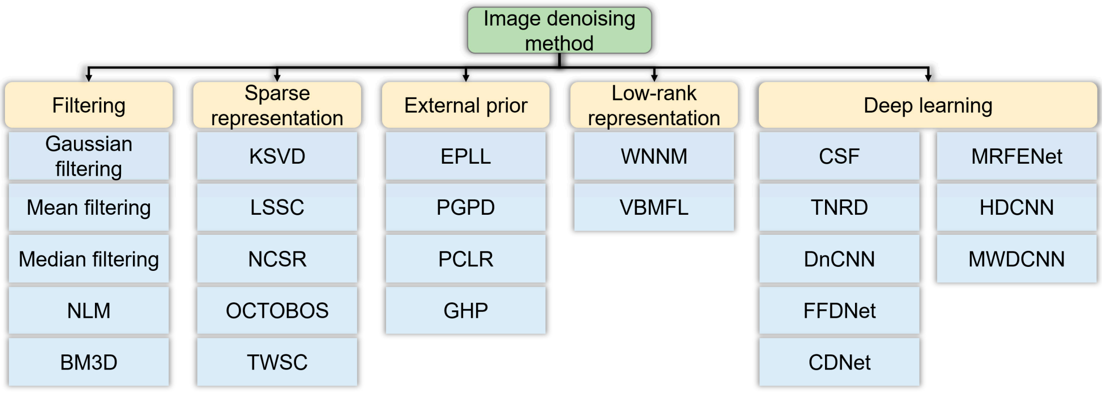


Figure 1. Classification of image denoising methods.

Sparse-representation-based methods are based on image sparsity and achieve image denoising by training an over-complete dictionary. A more representative case is the K-singular value decomposition (KSVD) method using sparse representation [4]. Inspired by KSVD, Mairal et al. [5] combined image self-similarity with sparse coding to decompose similar patches using similar sparse patterns, thus forming a Learned Simultaneous Sparse Coding (LSSC) method. Although sparse representation models have shown good results in image denoising, the sparse representation of traditional models may not be accurate enough due to the degradation of the observed images. To further improve the performance of image denoising based on sparse representation, Dong et al. [6] proposed a nonlocally centralized sparse representation (NCSR), which transformed the denoising problem into a problem of suppressing sparse coding noise. In addition, because the sparse coding of images using a single transform can limit performance, Wen et al. [7] proposed a structured over-complete sparsifying transform model with block cosparsity (OCTOBOS). These methods [4–7] have exhibited good results in denoising additive Gaussian white noise (AWGN); however, it is difficult to obtain good performance in real image denoising. To achieve better denoising of real images, Xu et al. [8] proposed a trilateral weighted sparse coding scheme (TWSC).

External-prior-based methods realize image denoising by using the statistical properties of natural images. A representative method is the denoising method based on expected patch log likelihood (EPLL) proposed by Zoran and Weiss [9]. This method applies the Gaussian mixture model to learn prior knowledge from a large number of natural image blocks and applies it to the denoising of other natural images. Similar to EPLL, Xu et al. [10] proposed a patch group prior-based denoising method (PGPD) to learn the self-similar features of natural images from groups of similar patches using the Gaussian mixture distribution. Inspired by EPLL, Chen et al. [11] proposed an external patch prior-guided internal clustering approach by combining an image external prior and an internal self-similarity prior, which is named PCLR. To improve the texture restoration capability of the image denoising method, Zou et al. [12] proposed a gradient histogram preservation method (GHP) based on texture enhancement. GHP improves texture recovery by preserving the gradient distribution of the corrupted image.

Low-rank representation-based methods exploit the low-rank properties of natural images and achieve denoising by extracting their low-rank components. A typical case is the Weighted Nuclear Norm Minimization (WNNM) proposed by Gu et al. [13]. Low-rank matrix factorization is also a method used to extract low-rank components from a dataset and is often applied in cases where the image size is large and its rank is much smaller than the length and width of the dataset. The most well-known method is the low-rank matrix factorization based on variational Bayesian (VBMFL), which was proposed by Zhao et al. [14]. This method improves the robustness of the model to outliers by using a Laplace distribution to establish a noise model.

Deep-learning-based denoising methods are currently the most popular. They usually learn the direct mapping from the corrupted image to the clean image or the noise. Since deep-learning-based denoising methods do not rely on image priori (e.g., self-similarity, sparsity, gradient, statistical properties, and low-rank properties), they do not have to spend much time finding and processing similar blocks in the images. Thus, they not only achieve a good denoising performance but also have a fast inference speed. Schmidt et al. [15] proposed a method based on a cascade of shrinkage fields (CSF) to improve the denoising performance while considering computational efficiency. Chen et al. [16] extended conventional nonlinear reaction–diffusion models with several parametrized linear filters as well as several parametrized influence functions and proposed a trainable nonlinear reaction–diffusion method (TNRD). Although CSF and TNRD show good denoising performance, they can only provide the best denoising results at known noise levels. To solve the problem of blind image denoising, Zhang et al. [17] proposed a deep learning method using a denoising convolutional neural network (DnCNN), which was the first application of residual learning to general image denoising. The application of residual learning to image denoising has greatly improved the denoising performance of networks and inspired many outstanding denoising methods based on deep learning [17–22]. In addition, Zhang et al. [18] further improved the DnCNN and proposed a fast and flexible denoising convolutional neural network (FFDNet), which achieves a good trade-off between the inference speed and denoising performance by downsampling and manually inputting a noise estimation map. Binh et al. [23] combined DnCNN with ResNet and proposed a convolutional denoising neural network called FlashLight CNN. A complex-valued deep convolutional neural network called CDNet was proposed by Quan et al. [24], and it effectively improved the denoising performance of the network. Guan et al. [25] proposed an image denoising method for remote sensing images called MRFENet. It demonstrated good denoising performance and preserved the edge details of the images. Zhang et al. [26] utilized dilated convolutions to capture more contextual information and then proposed a hybrid denoising neural network called HDCNN to enhance the denoising performance of CNN networks in complex application scenarios. Tian et al. [27] combined dynamic convolution, wavelet transform, and discriminative learning to propose a convolutional neural network based on the wavelet transform called Multi-stage Image Denoising CNN with the Wavelet Transform (MWDCNN). To reduce the parameter size and training burden of deep denoising networks, Tang et al. [28] employed a cascaded residual network and proposed a lightweight, multi-scale, efficient convolutional neural network.

The results of most denoising methods are obtained directly from the fusion of high-level features, while low-level features containing texture and contour information are ignored, resulting in the loss of some important information. Furthermore, since the convolutional operation prefers to extract local features, it is difficult to extract global information such as textures and contours when the objects in the image are relatively large. To solve these problems, an N-shaped convolutional neural network, named NSNet, using multi-scale features is proposed in this paper. In this model, a 2D Haar wavelet is used to construct an image pyramid that contains high- and low-frequency components of the corrupted image at different scales. The multi-scale features are extracted from the image pyramid by a U-shaped convolutional network [29], and the low- and high-level features are fused by skip connections in the U-shaped network. The 2D Haar wavelet is widely used in image denoising, and many scholars have achieved excellent denoising performance with it [30–33]. To verify the denoising performance of NSNet, the denoising of gray and color images was carried out at different noise levels and compared with existing denoising methods. The contributions of this work are summarized as follows:

(1) An N-shaped convolutional neural network for extracting multi-scale information is proposed. The network exploits multi-scale information to compensate for the

drawbacks of convolutional operations in extracting global features, which effectively improves the network's ability to recover textures and contours.

(2) A scheme for constructing image pyramids using a 2D Haar wavelet is proposed. The image pyramid is obtained by using a multi-scale 2D Haar wavelet, and each layer of the pyramid contains one low-frequency component and three high-frequency components. In image denoising, the high-frequency components can be used as an estimate of the noise level to facilitate denoising.

(3) NSNet shows good denoising performance for AWGN at a noise level range of (0, 55) and good recovery of textures and contours. It provides a solution for applications that need not only denoising but also texture and contour recovery.

The rest of this paper is organized as follows. Section 2 presents the techniques involved in the proposed model. Section 3 describes the proposed NSNet and the construction of the image pyramid in detail. Section 4 presents the results of experiments, and Section 5 concludes the paper.

2. Theoretical Aspects

2.1. The 2D Haar Wavelet

The process of decomposing an image using the 2D Haar wavelet is shown in Figure 2. The blocks with five-pointed stars in the figure are the finite impulse response filter. $h_\varphi = \left\{1/\sqrt{2}, 1/\sqrt{2}\right\}$ denotes a low-pass filter, and $h_\psi = \left\{-1/\sqrt{2}, 1/\sqrt{2}\right\}$ denotes a high-pass filter. The down arrow (↓) indicates downsampling, which means adding two adjacent pixels in the column or row direction.

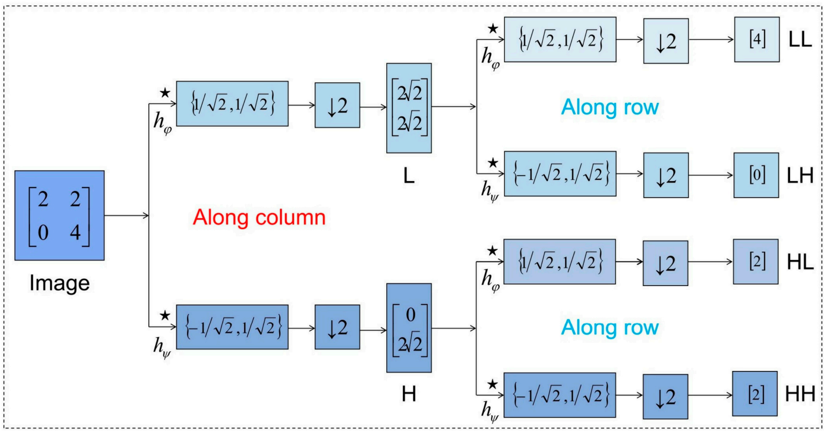


Figure 2. The 2D Haar wavelet transform.

The image is first processed with the two filters separately and is compressed in the column direction to obtain a low-frequency component L and a high-frequency component H. The two components are then processed with low- and high-pass filters in turn and compressed in the row direction to obtain the low-frequency component LL, the high-frequency component LH in the vertical direction, the high-frequency component HL in the horizontal direction, and the high-frequency component HH in the diagonal direction. According to the principle of 2D Haar wavelets, assuming that the corrupted image is $X \in R^{m\times n}$, the formulas for the components of the image can be simplified as

$$A_{ij} = X(2i, 2j), \;\; B_{ij} = X(2i+1, 2j) \tag{1}$$

$$C_{ij} = X(2i, 2j+1), \;\; D_{ij} = X(2i+1, 2j+1) \tag{2}$$

$$LL_{ij} = \frac{1}{2}\left(A_{ij} + B_{ij} + C_{ij} + D_{ij}\right), \;\; HL_{ij} = \frac{1}{2}\left(B_{ij} + D_{ij} - A_{ij} - C_{ij}\right) \tag{3}$$

$$LH_{ij} = \frac{1}{2}\left(C_{ij} + D_{ij} - A_{ij} - B_{ij}\right), \; HH_{ij} = \frac{1}{2}\left(A_{ij} + D_{ij} - B_{ij} - C_{ij}\right) \tag{4}$$

where $\{i|0 \leq i \leq m/2\}$ and $\{j|0 \leq j \leq n\}$. The two-scale 2D Haar wavelet applied to the image "Monarch" is shown in Figure 3. Through the 2D Haar wavelet, the image is decomposed into three high-frequency components (LH, HL, HH) and a low-frequency component (LL).

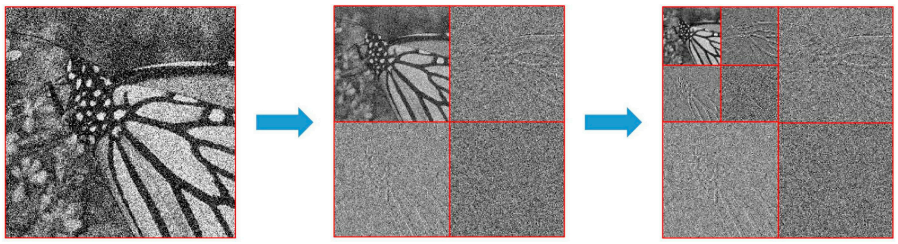


Figure 3. Two-scale 2D Haar wavelet processing results on the image "Monarch".

2.2. U-Shaped Convolutional Network

The U-shaped convolutional network was first proposed by Ronneberger et al. [29]. The network has been widely used in various fields for its powerful encoding and decoding capabilities [34,35]. The U-shaped convolutional network includes four downsampling operators and four upsampling operators and uses skip connections at the same stage, which not only gives the network the ability to extract multi-scale features but also ensures that the output integrates more low-level features [21]. The structure of the U-shaped convolutional network with batch normalization (BN) is shown in Figure 4.

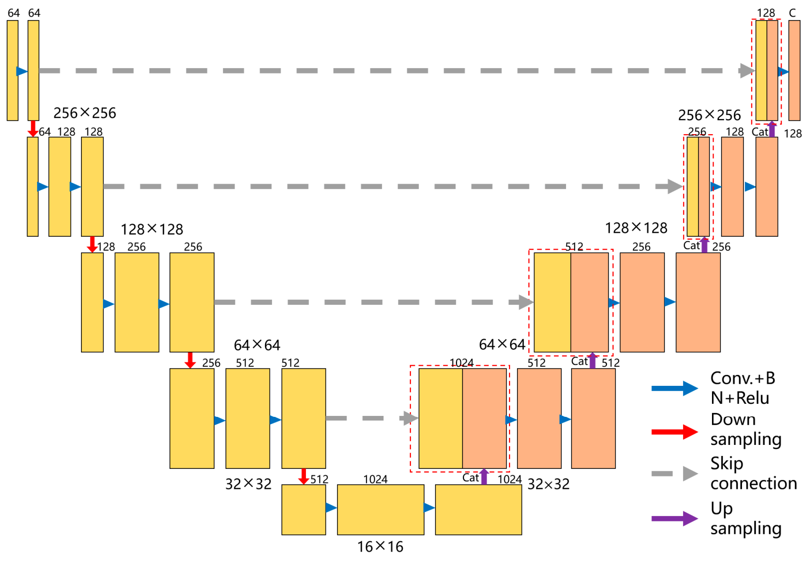


Figure 4. Network structure of the U-shaped network with BN.

When the neural network becomes very deep, an internal covariate offset may occur, which can lead to two problems: (1) it affects the learning efficiency and makes the learning process unstable, and (2) it makes the input data of the later layers too large or small, thus falling into the saturation area of the activation function and causing the learning process to stop prematurely. To solve the problem of the internal covariate offset, a general approach is to add BN, as proposed by Ioffe and Szegedy [36], to the U-shaped convolutional network.

The BN layer is usually placed between the convolutional operation and the Rectified Linear Unit (ReLU), and the parameters of the BN layer are adjusted by training. Supposing that there is a mini-batch $B = \{X_{1\cdots m}\}$ of size m, and the parameters to be learned are γ and β, the BN process can be expressed as

$$\gamma = \sqrt{Var[x]}, \ \ \beta = E[x], \tag{5}$$

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i, \ \ \delta_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2, \tag{6}$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\delta_B^2 + \varepsilon}}, \ \ y_i = \gamma\hat{x}_i + \beta, \tag{7}$$

where ε is a constant added to the mini-batch variance for numerical stability.

2.3. Residual Learning

When the network becomes very deep, some convolutional layers may appear to have identity mapping, resulting in degradation problems and vanishing and exploding gradients. To solve this problem, He et al. [37] proposed a residual network with the residual block shown in Figure 5. It connects the input and output directly through a shortcut connection, allowing $F(x)$ to learn small changes. This not only allows the convolutional layer to maintain identity mapping but also avoids vanishing and exploding gradients. The relationship between the input x and output x^* of the residual block is

$$x^* = F(x) + x. \tag{8}$$

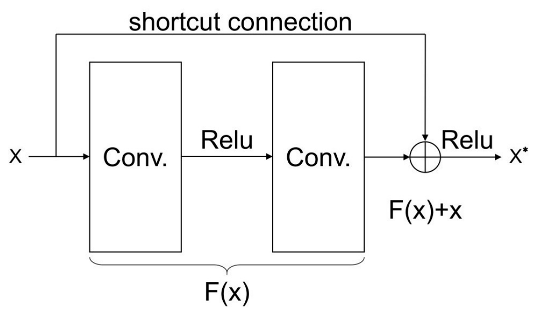


Figure 5. Basic structure of a residual block.

In image denoising, when the noise level is low, the mapping from the noisy image to the clean image is close to an identity mapping, which is not conducive to the training of the network. To solve this problem, Zhang et al. [17] first applied residual learning to image denoising. Assuming that the input image is $Input$ and the output image is $Output$, their relationship is

$$F(x) = Output - Input. \tag{9}$$

It can be seen from (9) that residual learning for image denoising uses noise as the training target, which is a valuable technique for improving the denoising performance of the model.

3. The Proposed NSNet Model

In this section, the proposed NSNet model is introduced in detail; its architecture is shown in Figure 6. It mainly consists of a multi-scale input layer and a multi-scale feature extraction layer. The multi-scale input layer uses a 2D Haar wavelet to create an image pyramid, which decomposes the corrupted image into high- and low-frequency components at different scales. The multi-scale feature extraction layer uses a U-shaped convolutional network to extract features at different scales from the image pyramid.

Additionally, NSNet sets the mean-squared error as the loss function and uses the residual learning strategy to learn the noise directly.

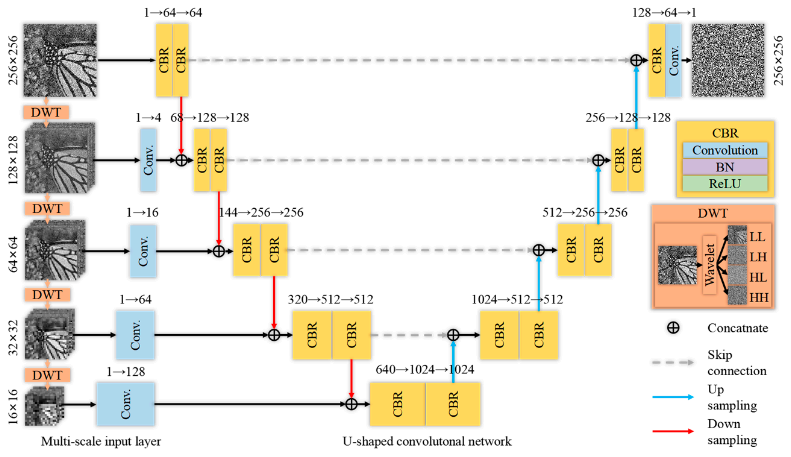


Figure 6. Architecture of NSNet.

The 2D Haar wavelet can decompose the image into four sub-images, each with a size half that of the original image. By using a 2D Haar wavelet to decompose the image, we can obtain

$$LL_1, (LH_1, HL_1, HH_1) = dwt(y), \tag{10}$$

where $dwt(\cdot)$ represents the 2D Haar wavelet, y is the corrupted image, LL_1 is the low-frequency component, and LH_1, HL_1, and HH_1 are the high-frequency components. Then, the 2D Haar wavelet is applied once again to the low-frequency component LL_1 to obtain

$$LL_2, (LH_2, HL_2, HH_2) = dwt(LL_1), \tag{11}$$

Finally, to obtain the image pyramid shown in Figure 7, the same operation is repeated twice, resulting in

$$LL_3, (LH_3, HL_3, HH_3) = dwt(LL_2), \tag{12}$$

$$LL_4, (LH_4, HL_4, HH_4) = dwt(LL_3). \tag{13}$$

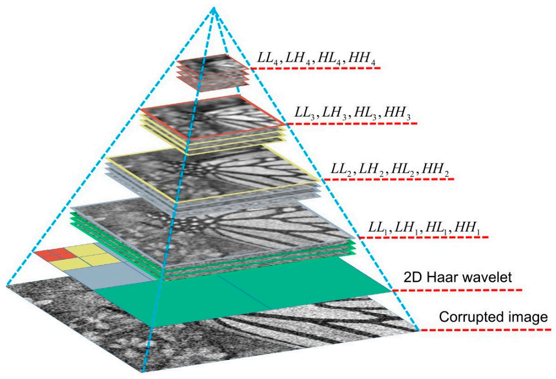


Figure 7. Image pyramid constructed by a 2D Haar wavelet.

The image pyramid contains images at five different scales, each of which corresponds to a different stage of the U-shaped convolutional network. In addition to the original scale, each scale contains a low-frequency component LL and three high-frequency components LH, HL, and HH. As shown in Figure 7, the low-frequency component is close to the corrupted input image, while the high-frequency components contain a lot of noise and some textures, which can be considered an estimate of the noise level.

The image degradation model is established as $y = x + v$, where y denotes the corrupted image, x denotes the clean image, and v denotes the noise. The proposed model inputs the corrupted image y into the network to predict the noise $v \approx F(y)$ and finally obtains the clean image $x = y - v$ through simple subtraction. The mean-squared error is used as the loss function:

$$L(\theta) = \frac{1}{2N}\sum_{i=1}^{N}||F(y_i;\theta) - (y_i - x_i)||_F^2, \tag{14}$$

where θ represents the parameter set of the model, N is the total number of images, and x_i and y_i represent the *ith* clean image and noisy image, respectively.

For convenience, the proposed model trained at a known noise level is named NSNet-S, and the model trained at an unknown noise level is named NSNet-B. The pseudo-code of the proposed method is shown in Algorithm 1.

Algorithm 1 The algorithm of NSNet

```
Input: All training images $D$ from the observed dataset, denoising mode (B or S), noise level $noiseL$, range of noise level $noiseR$, maximum epoch $Mepoch$.
Output: The trained network $f$.
 1:  Initialing model parameters $\theta$ and learning rate $\eta$;
 2:  Sampling $m$ patches $(x_1, \cdots, x_m)$ from $D$;
 3:  for epoch = 1 to $Mepoch$ do
 4:    if epoch > 30 then
 5:      $\eta \leftarrow \eta/10$;
 6:    end if
 7:    Set $\hat{g} = 0$;
 8:    for $i = 1$ to $m$ do
 9:      if mode == "B" then
10:        Setting $noiseL$ as an integer at the range $noiseR$ randomly;
11:      end if
12:      Adding Gaussian noise with the noise level of $noiseL$ to $x_i$:
             $y_i = x_i + noise_i$;
13:      Performing multi-scale wavelet transform on $y_i$ to obtain $y_i^1, y_i^2, y_i^3, y_i^4$:
             $y_i^1 = \{LL_1, LH_1, HL_1, HH_1\} = dwt(y_i), y_i^2 = \{LL_2, LH_2, HL_2, HH_2\} = dwt(LL_1),$
             $y_i^3 = \{LL_3, LH_3, HL_3, HH_3\} = dwt(LL_2), y_i^4 = \{LL_4, LH_4, HL_4, HH_4\} = dwt(LL_3)$;
14:      Predicting noise using the network $f$ with parameter $\theta$:
             $out_i \leftarrow f(y_i, y_i^1, y_i^2, y_i^3, y_i^4; \theta)$;
15:      Calculating the loss according to Equation (14);
16:      Computing the gradient: $\hat{g} \leftarrow \hat{g} + \frac{1}{m}\nabla_\theta L(\theta)$;
17:    end for
18:    Updating $\theta$: $\theta \leftarrow \theta - \eta \times \hat{g}$;
19:  end for
```

4. Experimental Results

Gray image denoising and color image denoising were carried out to compare the denoising performance of the proposed NSNet with those of existing models, including NLM [2], BM3D [3], KSVD [4], NCSR [6], OCTOBOS [7], TWSC [8], GHP [9], EPLL [10], PGPD [11], PCLR [12], WNNM [13], CSF [15], TNRD [16], and FFDNet [18]. Moreover, two different types of DnCNNs [17] were also selected as the compared models. They are DnCNN-S and DnCNN-B, which are trained at known and unknown noise levels, respectively.

4.1. Evaluation Metrics

The results of all denoising methods were analyzed quantitatively in terms of the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM).

(1) Supposing that the recovered image is $I \in R^{m\times n}$ and the corrupted image is $K \in R^{m\times n}$, the PSNR is calculated as

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2, \tag{15}$$

$$PSNR = 10\cdot\log_{10}\left(\frac{MAX_I^2}{MSE}\right), \tag{16}$$

where MSE is the mean-squared error, and MAX_I denotes the maximum value of the pixels in the image. In general, $MAX_I = 255$ if each pixel is represented in 8-bit binary form or 1 if it is represented in 1-bit binary.

(2) The SSIM is calculated as

$$SSIM(I,K) = \frac{(2\mu_I\mu_K + c_1)(\sigma_{IK} + c_2)}{\left(\mu_I^2 + \mu_K^2 + c_1\right)\left(\sigma_I^2 + \sigma_K^2 + c_2\right)} \tag{17}$$

where μ_I and μ_K denote the means of I and K, respectively, and σ_I and σ_K denote their standard deviations, while σ_{IK} denotes the covariance of I and K, and c_1 and c_2 are constants.

4.2. Experimental Setting

For the ablation experiment, NSNet, NSNet without BN, Unet, and Unet without BN were compared. All compared methods were trained using 400 images of size 180 × 180 pixels, as mentioned in [17]. The test sets were Set12, which is widely used in the evaluation of denoising methods, and BSD68 [38]. In training the model, the size of the patch was set to 48 × 48, and 128 × 618 patches were cropped from the 400 images. Four denoising methods were trained at noise levels of 15, 25, 35, 45, and 50. For a noise level of α, the noise was generated by a Gaussian distribution with a mean of zero and a variance of α.

For the denoising of gray images, the 400 images were still used as the training set, and 128 × 2934 patches of size 48 × 48 were cropped from them. Since most image denoising methods can only obtain the best denoising performance at a known noise level, to achieve a fair comparison, the proposed method was trained at an unknown noise level and at noise levels of 15, 25, and 50. The test sets were Set12 and BSD68, neither of which participated in model training.

For color image denoising, 432 images were selected from the color image dataset CBSD500 [39] as training samples, and the remaining 68 images (CBSD68) were used as the test set. The test set also included Kodak24 [40] and McMaster [41]. In this experiment, 96 × 3900 patches were cropped from the 432 images to train the color image denoising model. The other settings were largely consistent with the settings used for gray image denoising. The specific settings of NSNet are shown in Table 1.

Table 1. Specific settings of NSNet used in all experiments.

Experiment	Model	Noise Level	Patch Size	Number of Patches
Ablation experiment	NSNet-S	15, 25, 35, 45, 50	48 × 48	128 × 618
Gray image denoising	NSNet-S	15, 25, 50	48 × 48	128 × 2934
Gray image denoising	NSNet-B	(0, 55)	48 × 48	128 × 2934
Color image denoising	NSNet-B	(0, 55)	48 × 48	96 × 3900

When training NSNet-S, each clean image input to the model was corrupted by the same level of noise. When training NSNet-B, each clean image input to the model was corrupted by noise at a level drawn randomly from the range (0, 55). The Adam optimizer was used to tune the model with an initial learning rate of 0.001. The maximum training epoch was 50. After 30 epochs, the learning rate was adjusted to 0.0001. The size of each mini-batch was set to 128. The denoising network was trained in PyTorch, and all experiments were carried out in the pycharm environment running on a PC with an AMD Ryzen 9 5900HX with Radeon Graphics 3.30 GHz CPU and an NVIDIA GeForce RTX 3070 Laptop GPU.

4.3. Ablation Experiment

This section describes the ablation experiment that was carried out to demonstrate the effectiveness of the main components of the proposed model. The experiment tested the denoising performance of NSNet, Unet, NSNet without BN, and Unet without BN at noise levels of 15, 25, 35, 45, and 50. The denoising results on the Set12 dataset and the gray version of BSD68 are shown in Table 2, in which values with # and * represent the best and second-best denoising performance, respectively.

Table 2. Average PSNR/SSIM of four denoising methods in the ablation experiment.

Model	**Nosie Level**	**Method**			
		NSNet	**Unet**	**NSNet(-BN)**	**Unet(-BN)**
Set12	$\sigma = 15$	32.90 #/0.9040 #	32.75/0.9017	32.76 */0.9025 *	31.81/0.8835
	$\sigma = 25$	30.50 #/0.8643 #	30.48 */0.8640 *	30.33/0.8610	30.28/0.8602
	$\sigma = 35$	28.95 #/0.8319 #	28.89 */0.8300 *	28.65/0.8243	28.77/0.8270
	$\sigma = 45$	27.81 #/0.8041 #	27.80 */0.8029 *	27.66/0.7993	26.52/0.7945
	$\sigma = 50$	27.32 #/0.7910 #	27.32 */0.7901 *	26.93/0.7737	26.90/0.7724
BSD68	$\sigma = 15$	31.79 #/0.8927 #	31.71 */0.8912 *	31.70/0.8916 *	31.04/0.8748
	$\sigma = 25$	29.31 #/0.8322 #	29.29 */0.8321 *	29.20/0.8294	29.18/0.8288
	$\sigma = 35$	27.82 #/0.7849 #	27.76 */0.7810 *	27.65/0.7779	27.71/0.7788
	$\sigma = 45$	26.78 #/0.7446 #	26.77 */0.7442 *	26.70/0.7413	26.63/0.7364
	$\sigma = 50$	26.35 #/0.7267 #	26.34 */0.7260 *	26.10/0.7101	26.11/0.7106

Note: NSNet(-BN) means NSNet without BN, and Unet(-BN) means Unet without BN. # The best denoising performance. * The second-best denoising performance.

The denoising performance of NSNet is better than that of Unet at all noise levels, which shows that multi-scale input can improve the denoising performance. The results for NSNet and NSNet without BN show that the denoising performance of NSNet is greatly improved after adding the BN layer. As mentioned in [29], the BN layer can improve the denoising performance of the neural network. At all noise levels, NSNet without BN achieves better denoising than Unet without BN, and its denoising performance is close to that of Unet at low noise levels, which indicates that the multi-scale input greatly improved the denoising performance of the model at low noise levels. With increases in the noise level, the structure of the compressed image is increasingly corrupted; thus, it cannot provide accurate information for the network. In this case, the performance of NSNet is similar to that of Unet. For example, the denoising performance of NSNet is similar to that of Unet at a noise level of 50.

4.4. Gray Image Denoising

In this experiment, AWGN was added to Set12 and the gray version of BSD68, and the noise levels were set to 15, 25, and 50. The denoising results of all methods on Set12 are shown in Tables 3–5. Due to insufficient parameters, CSF cannot be tested at a noise level of 50.

Table 3 shows the denoising results of all methods at a noise level of 15. NSNet-S has a better denoising performance than other methods and obtained the first-ranked denoising performance on ten images. NSNet-S ranked a very close second in terms of denoising the image “House”. In this case, the denoising performance of NSNet-S is 1.86 dB higher than that of the worst method, NLM, on average, and 0.1 dB higher than that of DnCNN-S, on average, in terms of PSNR. The average results of all methods show that there is little difference in the denoising performance of most methods at a noise level of 15. At a low noise level, the self-similarity, sparsity, and low-rank properties of the image are still relatively complete, and the traditional denoising methods (e.g., BM3D, NCSR, TWSC, PCLR, WNNM) also achieve a good denoising performance.

Table 3. PSNRs of all methods on Set12 at a noise level of 15.

Model	C.man	House	Pappers	Starfish	Monar.	Airpl.	Parrot	Lena	Barbara	Boat	Man	Couple	Average
NLM	30.05	33.23	31.59	30.30	30.47	29.42	30.07	33.16	31.33	31.25	31.27	30.97	31.09
BM3D	31.92	34.95	32.70	31.15	31.86	31.08	31.38	34.27	33.11	32.14	31.93	32.11	32.38
KSVD	31.46	34.24	32.20	30.70	31.41	30.75	30.95	33.71	32.42	31.76	31.53	31.47	31.89
NCSR	32.01	35.04	32.66	31.50	32.25	31.19	31.37	34.12	33.06	32.08	31.98	32.00	32.44
OCTOBOS	31.91	34.32	32.49	31.04	31.72	30.98	31.29	33.91	32.59	31.87	31.74	31.77	32.14
TWSC	32.01	35.11 #	32.83	31.64	32.47	31.14	31.52	34.39	33.64 #	32.24	32.09	32.15	32.60
EPLL	31.82	34.14	32.58	31.08	32.03	31.16	31.40	33.87	31.34	31.91	31.97	31.91	32.10
GHP	31.48	34.07	32.40	31.09	31.63	30.77	31.16	33.54	32.01	31.72	31.62	31.54	31.92
PCLR	32.23	35.07	33.00	31.75	32.63	31.45	31.62	34.27	33.12	32.25	32.16	32.14	32.64
PGPD	31.83	34.79	32.61	31.25	32.15	31.19	31.32	34.04	32.74	32.03	31.99	32.07	32.33
WNNM	32.18	35.15	32.97	31.83	32.72	31.40	31.61	34.38	33.61 *	32.28	32.12	32.18	32.70
CSF	31.95	34.40	32.83	31.56	32.34	31.34	31.36	34.07	31.93	32.01	32.09	31.99	32.32
TNRD	32.19	34.55	33.03	31.76	32.57	31.47	31.63	34.25	32.14	32.15	32.24	32.11	32.51
DnCNN-S	32.59 *	34.99	33.24 *	32.13 *	33.25 *	31.67 *	31.88 *	34.58	32.61	32.42 *	32.43 *	32.43 *	32.85 *
DnCNN-B	32.14	34.96	33.09	31.92	33.08	31.54	31.64	34.52	32.03	32.36	32.37	32.38	32.67
FFDnet	32.37	35.05	33.01	31.95	32.92	31.55	31.79	34.60 *	32.48	32.36	32.37	32.43 *	32.74
NSNet-S	32.67 #	35.09 *	33.33 #	32.29 #	33.32 #	31.79 #	31.97 #	34.69 #	32.80	32.50 #	32.48 #	32.52 #	32.95 #
NSNet-B	32.02	34.63	32.74	32.08	32.87	31.31	31.60	34.42	30.59	32.29	32.18	32.20	32.41

Note: # The best denoising performance. * The second-best denoising performance.

Table 4. PSNRs of all methods on Set12 at a noise level of 25.

Model	C.man	House	Pappers	Starfish	Monar.	Airpl.	Parrot	Lena	Barbara	Boat	Man	Couple	Average
NLM	29.97	30.38	29.02	27.82	28.07	27.33	27.98	30.38	28.59	28.74	28.80	28.32	28.61
BM3D	29.45	32.86	30.16	28.56	29.25	28.43	28.93	32.08	30.72	29.91	29.62	29.72	29.98
KSVD	28.90	32.10	29.65	28.19	28.81	28.16	28.46	31.36	29.58	29.32	29.11	28.88	29.38
NCSR	29.43	32.89	30.05	28.77	29.43	28.45	28.86	31.92	30.62	29.77	29.58	29.49	29.94
OCTOBOS	29.25	32.08	29.78	28.24	28.78	28.28	28.67	31.56	29.88	29.51	29.26	29.23	29.54
TWSC	29.54	33.05	30.32	28.98	29.71	28.55	29.08	32.22	31.26 #	29.99	29.71	29.79	30.18
EPLL	29.24	32.04	30.07	28.43	29.30	28.56	28.91	31.62	28.55	29.69	29.63	29.48	29.63
GHP	29.28	32.50	30.04	28.66	29.02	28.28	28.87	31.69	30.29	29.71	29.49	29.37	29.77
PCLR	29.67	32.98	30.46	28.87	29.75	28.77	29.13	32.17	30.65	30.00	29.77	29.73	30.16
PGPD	29.26	32.79	30.07	28.49	29.29	28.54	28.80	31.93	30.28	29.82	29.66	29.68	29.88
WNNM	29.64	33.23	30.40	29.03	29.85	28.70	29.13	32.25	31.24 *	30.03	29.77	29.82	30.26
CSF	29.47	32.40	30.28	28.80	29.62	28.72	28.89	31.80	29.03	29.77	29.72	29.53	29.84
TNRD	29.71	32.54	30.55	29.02	29.86	28.89	29.18	32.01	29.41	29.92	29.88	29.71	30.06
DnCNN-S	30.21 #	33.10	30.82 *	29.36	30.41 *	29.08 *	29.44 *	32.41	30.01	30.20	30.08	30.08	30.43 *
DnCNN-B	30.03	33.04	30.73	29.24	30.37	29.06	29.35	32.40	29.67	30.19	30.06	30.05	30.35
FFDnet	30.05	33.26 *	30.72	29.28	30.29	29.01	29.42	32.57 *	29.98	30.23 *	30.07 *	30.15 *	30.42
NSNet-S	30.12 *	33.30 #	31.05 #	29.90 #	30.48 #	29.24 #	29.45 #	32.64 #	30.39	30.27 #	30.14 #	30.22 #	30.60 #
NSNet-B	30.05	33.05	30.62	29.62 *	30.33	28.97	29.39	32.57 *	27.54	30.23 *	29.99	30.05	30.20

Note: # The best denoising performance. * The second-best denoising performance.

Table 5. PSNRs of all methods on Set12 at a noise level of 50.

Model	C.man	House	Pappers	Starfish	Monar.	Airpl.	Parrot	Lena	Barbara	Boat	Man	Couple	Average
NLM	24.26	25.69	24.79	23.88	24.16	23.60	24.35	25.97	24.31	24.80	25.06	24.41	24.61
BM3D	26.13	29.69	26.68	25.04	25.82	25.10	25.90	29.05	27.23	26.78	26.81	26.46	26.73
KSVD	25.68	27.97	26.09	24.53	25.30	24.61	25.38	27.86	25.47	25.95	26.10	25.30	25.85
NCSR	26.15	29.62	26.53	25.09	25.77	24.93	25.71	28.90	26.99	26.67	26.67	26.19	26.60
OCTOBOS	25.62	28.59	26.15	24.57	25.04	24.86	25.39	28.37	26.17	26.30	26.27	25.82	26.10
TWSC	26.46	30.17	26.88	25.41	26.27	25.38	26.11	29.08	27.54 *	26.88	26.82	26.48	26.96
EPLL	26.03	28.77	26.63	25.04	25.78	25.24	25.84	28.42	24.82	26.65	26.72	26.24	26.35
GHP	25.91	28.51	26.38	24.39	25.53	24.76	25.71	27.43	25.44	25.99	25.92	25.46	25.95
PCLR	26.56	29.77	27.03	25.32	26.24	25.50	26.15	29.11	27.11	26.99	26.94	26.55	26.94
PGPD	26.40	29.73	26.69	25.10	25.89	25.34	25.84	29.00	26.84	26.82	26.84	26.47	26.75
WNNM	26.42	30.33	26.91	25.43	26.32	25.42	26.09	29.25	27.79 #	26.97	26.94	26.64	27.04
CSF	-	-	-	-	-	-	-	-	-	-	-	-	-
TNRD	26.61	29.49	27.08	25.42	26.32	25.59	26.16	28.94	25.70	26.94	26.99	26.50	26.81
DnCNN-S	27.26	29.96	27.35	25.64	26.83	25.83	26.42	29.34	26.15	27.19	27.19	26.86	27.17
DnCNN-B	27.26	29.91	27.35	25.60	26.84	25.82	26.48	29.34	26.32	27.18	27.17	26.87	27.18
FFDnet	27.24	30.36 *	27.41	25.68	26.92	25.79	26.57 #	29.63	26.41	27.30 *	27.26 *	27.04 *	27.30 *
NSNet-S	27.38 #	30.43 #	27.54 #	26.24 *	26.93 *	25.96 #	26.50	29.74 #	27.04	27.38 #	27.31 #	27.14 #	27.47 #
NSNet-B	27.35 *	30.24	27.47 *	26.17 #	26.94 #	25.89 *	26.54 *	29.70 *	25.46	27.30 *	27.19	27.01	27.27

Note: # The best denoising performance. * The second-best denoising performance.

Table 4 shows the denoising results obtained by all methods at a noise level of 25. With increases in the noise level, the best and second-best denoising performance was obtained with the deep-learning-based methods, which shows the superiority of deep learning in image denoising. As the noise level increases, the self-similarity and other features of the image are increasingly corrupted, which leads to the fact that the traditional image-prior-based denoising methods are no longer advantageous. The deep-learning-based denoising methods learn the potential noise directly from the corrupted image and rely less on the prior knowledge of the image. This allows them to achieve a good denoising performance, even at high noise levels. In this case, NSNet-S is ranked first in terms of denoising performance on ten images and second on one image. NSNet-B is ranked second in terms of denoising performance on three images. The denoising performance of NSNet-S is 1.99 dB higher than that of the worst method, NLM, and 0.17 dB higher than that of the second-best method, DnCNN-S, in terms of PSNR. Compared with the traditional denoising methods BM3D, NCSR, TWSC, and WNNM, NSNet-B is close to WNNM and outperforms BM3D, NCSR, and TWSC at a noise level of 25.

Table 5 shows the denoising results obtained by all methods at a noise level of 50. In this case, NSNet-S is ranked first in terms of denoising performance on eight images and second on two images. NSNet-B is ranked first in terms of denoising performance on two images and second on six images. Compared with other methods, NSNet-B provides a better denoising performance at a high noise level. The reason for this may be that an image with high noise will cause a greater deviation than one with low noise, and the network will pay more attention to the restoration of images with high noise when training NSNet-B. In this case, NSNet-S is 2.86 dB higher than the worst method, NLM, and 0.17 dB higher than the second-best method, FFDNet. NSNet-B also has an outstanding performance; its denoising performance surpasses that of DnCNN-B and WNNM and differs from that of FFDNet by only 0.03 dB.

In the above three experiments, NSNet shows a good denoising performance in most cases, although that of NSNet on the image "Barbara" is not as good as that of traditional methods (e.g., TWSC, WNNM). "Barbara" has similar rich textures, and the method based on image self-similarity can effectively use them to achieve better denoising. Both texture and noise are high-frequency information; therefore, image denoising methods that use residual learning tend to treat texture as noise, which makes the denoising performance of the proposed model poor. In addition, the same experiments were conducted on the dataset BSD68 to better demonstrate the denoising performance of NSNet. The average PSNRs and SSIMs of all methods on Set12 and BSD68 are shown in Table 6. Compared with other methods, the average denoising performance of NSNet-S on the two datasets is the best, and NSNet-B has a better denoising performance at a high noise level.

Table 6. Average PSNRs/SSIMs of all methods on datasets Set12 and BSD68.

Model	**Set 12**			**BSD68**		
	σ = 15	**σ = 25**	**σ = 50**	**σ = 15**	**σ = 25**	**σ = 50**
NLM	31.09/0.8594	28.62/0.7711	24.61/0.5695	29.82/0.8322	27.56/0.7296	24.01/0.5212
BM3D	32.38/0.8957	29.98/0.8510	26.73/0.7681	31.08/0.8722	28.57/0.8017	25.62/0.6869
KSVD	31.89/0.8847	29.38/0.8308	25.85/0.7260	30.86/0.8677	28.29/0.7889	25.18/0.6548
NCSR	32.44/0.8958	29.94/0.8501	26.60/0.7673	31.19/0.8770	28.61/0.8045	25.59/0.6864
OCTOBOS	32.14/0.8889	29.54/0.8378	26.10/0.7433	31.08/0.8744	28.46/0.7989	25.33/0.6705
TWSC	32.60/0.8989	30.18/0.8549	26.96/0.7731	31.28/0.8782	28.76/0.8077	25.77/0.6903
EPLL	32.10/0.8936	29.63/0.8444	26.35/0.7475	31.19/0.8825	28.68/0.8123	25.68/0.6877
GHP	31.92/0.8693	29.77/0.8415	25.95/0.7562	30.83/0.8513	28.49/0.8039	24.94/0.6809
PCLR	32.64/0.8979	30.16/0.8542	26.94/0.7763	31.38/0.8799	28.84/0.8106	25.88/0.6970
PGPD	32.33/0.8900	29.88/0.8447	26.75/0.7602	31.14/0.8705	28.64/0.8019	25.76/0.6877
WNNM	32.70/0.8982	30.26/0.8557	27.04/0.7775	31.32/0.8766	28.80/0.8029	24.43/0.6838
CSF	32.32/0.8923	29.84/0.8450	-/-	31.24/0.8746	28.73/0.8055	-/-
TNRD	32.51/0.8967	30.06/0.8520	26.81/0.7666	31.42/0.8825	28.91/0.8155	25.96/0.7024
DnCNN-S	32.85 */0.9025 *	30.43 */0.8616	27.17/0.7828	31.74 */0.8907 *	29.23 */0.8279	26.24/0.7189
DnCNN-B	32.67/0.9000	30.35/0.8599	27.18/0.7816	31.62/0.8868	29.16/0.8244	26.23/0.7164
FFDnet	32.74/0.9024	30.42/0.8631 *	27.30 */0.7900 *	31.64/0.8902	29.19/0.8828	26.29/0.7239
NSNet-S	32.95 #/0.9054 #	30.59 #/0.8662 #	27.47 #/0.7956 #	31.81 #/0.8936 #	29.33 #/0.8339 #	26.42 #/0.7316 #
NSNet-B	32.41/0.8943	30.20/0.8595	27.27/0.7895	31.46/0.8870	29.18/0.8304 *	26.32 */0.7242 *

Note: # The best denoising performance. * The second-best denoising performance.

Figures 8 and 9 show the denoising performance of all compared methods on gray images at noise levels of 25 and 50, respectively. In order to facilitate the comparison of the denoising performance, the results were transformed into pseudo-color images. The red box in Figure 8 shows the restoration effect of all compared methods on the "grass". It can be seen that NSNet's recovery of the "grass" is closer to the clean image than the other compared methods and results in sharper and clearer edges and textures. The red box in Figure 9 further demonstrates the advantages of NSNet in edge and texture restoration. As shown in Figure 9, compared to other methods, NSNet can not only make the edges and textures sharper but also restore more image details.

4.5. Color Image Denoising

The previous section compared the denoising performance of the methods on gray images, where BM3D, DnCNN, and FFDNet had the better denoising performance and computational speed. In the experiment described in this subsection, these methods were selected for a further comparative test of the denoising ability of the proposed model with color images. The datasets used here are the color versions of CBSD68, Kodak24, and McMaster.

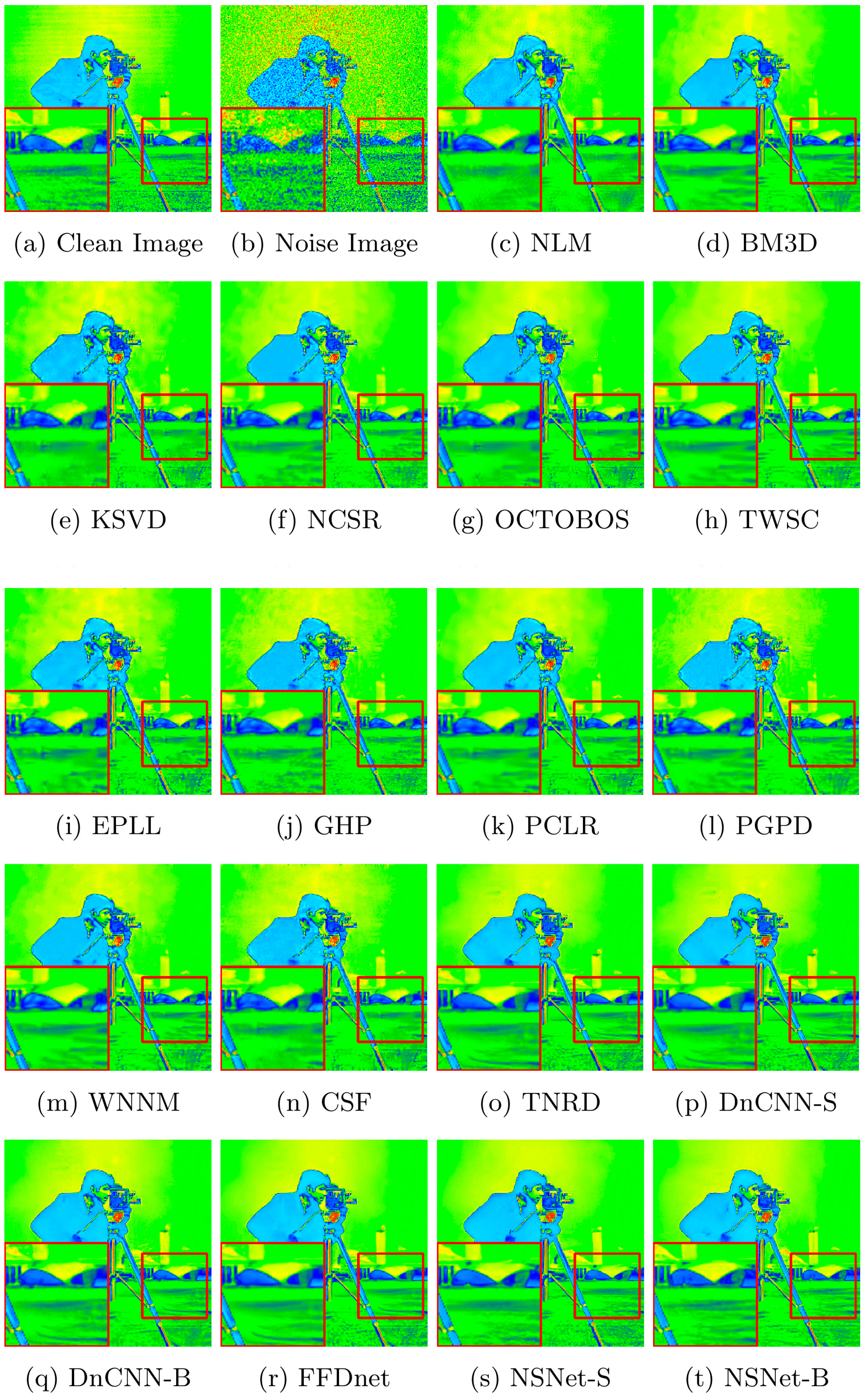
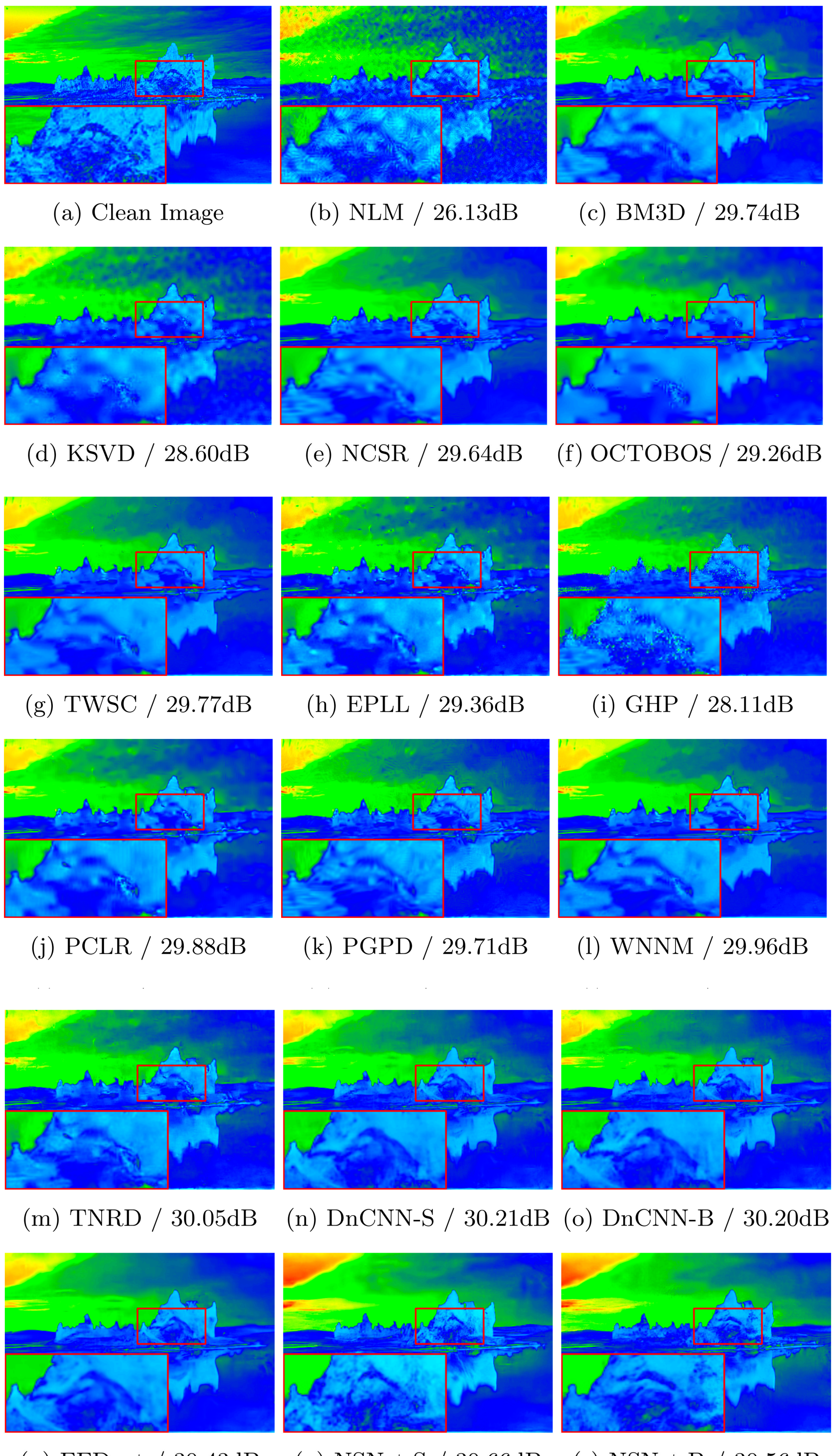


Figure 8. Results of some denoising methods at a noise level of 25.



(a) Clean Image (b) NLM / 26.13dB (c) BM3D / 29.74dB

(d) KSVD / 28.60dB (e) NCSR / 29.64dB (f) OCTOBOS / 29.26dB

(g) TWSC / 29.77dB (h) EPLL / 29.36dB (i) GHP / 28.11dB

(j) PCLR / 29.88dB (k) PGPD / 29.71dB (l) WNNM / 29.96dB

(m) TNRD / 30.05dB (n) DnCNN-S / 30.21dB (o) DnCNN-B / 30.20dB

(p) FFDnet / 30.43dB (q) NSNet-S / 30.66dB (r) NSNet-B / 30.56dB

Figure 9. Results of some denoising methods at a noise level of 50.

The denoising performance of four methods (BM3D, DnCNN, FFDNet, and NSNet) at noise levels of 35, 45, and 55 is shown in Table 7. The proposed NSNet is the best model in the denoising experiments with CBSD68 and Kodak24, with McMaster ranked second.

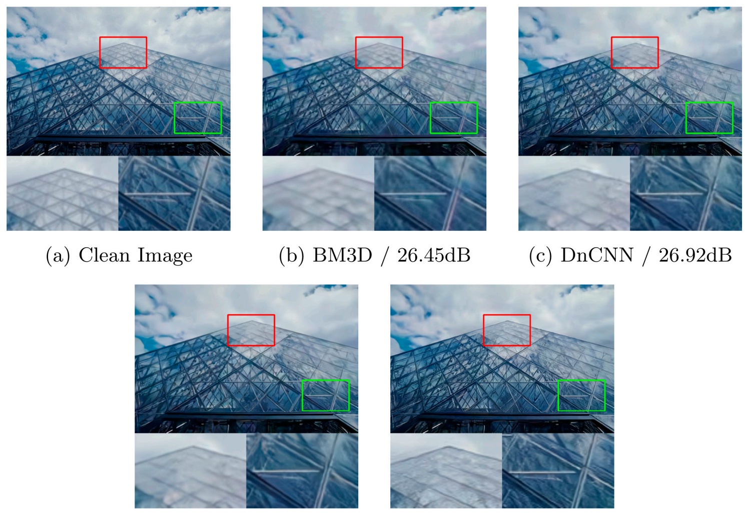
Table 7. Average PSNRs/SSIMs of all methods on color images.

Dataset	Noise Level	Method			
		BM3D	**DnCNN**	**FFDNet**	**NSNet**
CBSD68	$\sigma = 35$	28.88/0.8160	29.60 */0.8422 *	29.59/0.8408	29.69 #/0.8481 #
	$\sigma = 45$	27.84/0.7793	28.43/0.8060 *	28.44 */0.8048	28.58 #/0.8150 #
	$\sigma = 55$	26.97/0.7468	27.50/0.7736	27.56 */0.7738 *	27.70 #/0.7854 #
Kodak24	$\sigma = 35$	29.89/0.8208	30.45/0.8387	30.56 */0.8405 *	30.67 #/0.8478 #
	$\sigma = 45$	28.91/0.7906	29.31/0.8057	29.44 */0.8083 *	29.59 #/0.8187 #
	$\sigma = 55$	28.06/0.7629	28.38/0.7753	28.57 */0.7811 *	28.72 #/0.7929 #
McMaster	$\sigma = 35$	29.93/0.8237	30.15/0.8392	30.83 #/0.8550 #	30.60 */0.8527 *
	$\sigma = 45$	29.00/0.7988	29.08/0.8116	29.68 #/0.8275 *	29.55 */0.8281 #
	$\sigma = 55$	28.13/0.7712	28.17/0.7832	28.76 #/0.8032 *	28.73 */0.8049 #

Note: # The best denoising performance. * The second-best denoising performance.

In the denoising experiment with the dataset CBSD68, NSNet performed 0.81 dB better than BM3D, 0.09 dB better than DnCNN, and 0.1 dB better than FFDNet at a noise level of 35. It was 0.74 dB better than BM3D, 0.15 dB better than DnCNN, and 0.14 dB better than FFDNet at a noise level of 45, and 0.73 dB better than BM3D, 0.2 dB better than DnCNN, and 0.14 dB better than FFDNet at a noise level of 55. In the denoising experiment with the dataset Kodak24, NSNet was 0.78 dB better than BM3D, 0.22 dB better than DnCNN, and 0.11 dB better than FFDNet at a noise level of 35. It was 0.68 dB better than BM3D, 0.28 dB better than DnCNN, and 0.15dB better than FFDNet at a noise level of 45, and 0.66 dB better than BM3D, 0.34 dB better than DnCNN, and 0.15 dB better than FFDNet at a noise level of 55. In the denoising experiment using the McMaster dataset, although the denoising performance of NSNet was not as good as that of FFDNet, the comparison of all methods in terms of SSIM shows that NSNet is able to recover the structure of the color image better.

Figure 10 shows the denoising results of four methods (BM3D, DnCNN, FFDNet, and NSNet) on one image of the public dataset CBSD68 at a noise level of 45. To better demonstrate the denoising performance of the proposed method, two representative parts are highlighted. These show that NSNet repairs the texture better than other methods. Figure 11 shows the denoising results of four methods at a noise level of 40. NSNet has a more significant repair effect on “stone” in the image, and its recovery of textures is better than those of other methods.



(a) Clean Image (b) BM3D / 26.45dB (c) DnCNN / 26.92dB

(d) FFDnet / 27.05dB (e) NSNet / 27.12dB

Figure 10. Comparison of denoising results on one color image of the public dataset CBSD68 [39] at a noise level of 45.

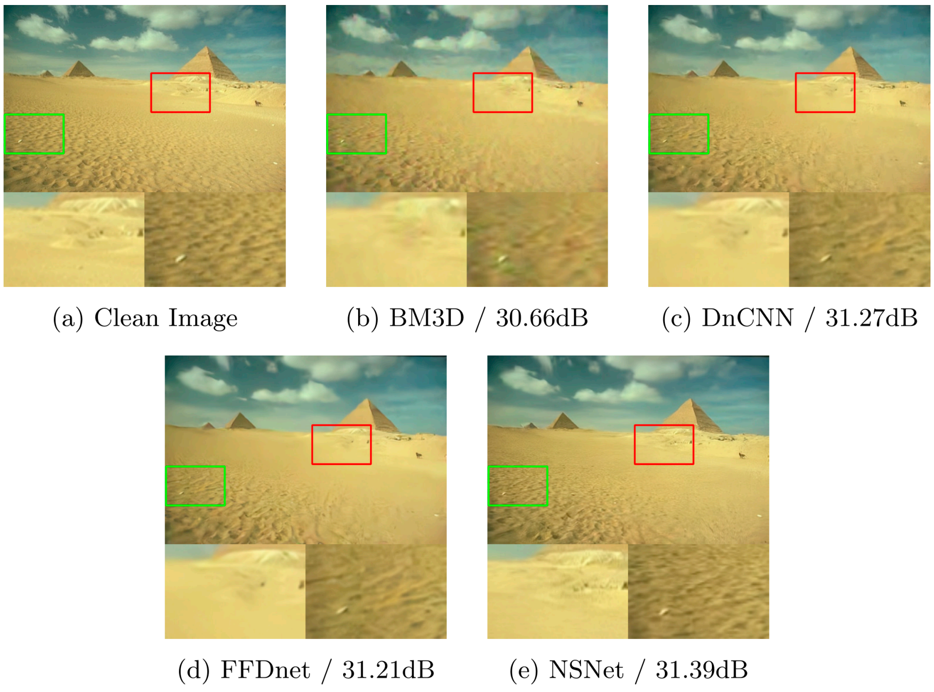


Figure 11. Comparison of denoising results on one color image of the public dataset CBSD68 [39] at a noise level of 40.

5. Conclusions

In this study, an N-shaped convolutional network that can extract multi-scale information is proposed. NSNet is able to extract multi-scale information from corrupted images and uses it to compensate for the drawbacks of convolutional operations in extracting global features, thus enhancing NSNet's ability to capture global information and reconstruct textures and contours. Ablation experiments demonstrate that extracting multi-scale information is beneficial to improving the denoising performance of the model at noise levels in the range of (0, 50). Gray and color image denoising experiments demonstrate that NSNet outperforms many existing image denoising methods at noise levels of 15, 25, and 50, especially at high noise levels. In addition, NSNet has a good blind denoising performance, where its performance at high noise levels is close to that at known noise levels.

Author Contributions: Conceptualization, writing—review and editing, visualization, supervision, and funding acquisition, Y.L. Methodology, software, validation, investigation, and writing—original draft preparation, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Changsha Municipal Natural Science Foundation under Grant kq2208432.

Data Availability Statement: Not applicable.

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

References

1. Chatterjee, P.; Milanfar, P. Is denoising dead? *IEEE Trans. Image Process.* **2009**, *19*, 895–911. [CrossRef] [PubMed]
2. Buades, A.; Coll, B.; Morel, J.-M. A non-local algorithm for image denoising. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; Volume 2, pp. 60–65.
3. Dabov, K.; Foi, A.; Katkovnik, V.; Egiazarian, K. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **2007**, *16*, 2080–2095. [CrossRef] [PubMed]
4. Elad, M.; Aharon, M. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.* **2006**, *15*, 3736–3745. [CrossRef] [PubMed]
5. Mairal, J.; Bach, F.; Ponce, J.; Sapiro, G.; Zisserman, A. Non-local sparse models for image restoration. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 2272–2279.

6. Dong, W.; Zhang, L.; Shi, G.; Li, X. Nonlocally centralized sparse representation for image restoration. *IEEE Trans. Image Process.* **2013**, *22*, 1620–1630. [CrossRef] [PubMed]
7. Wen, B.; Ravishankar, S.; Bresler, Y. Structured overcomplete sparsifying transform learning with convergence guarantees and applications. *Int. J. Comput. Vis.* **2015**, *114*, 137–167. [CrossRef]
8. Xu, J.; Zhang, L.; Zhang, D. A trilateral weighted sparse coding scheme for real-world image denoising. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 20–36.
9. Zoran, D.; Weiss, Y. From learning models of natural image patches to whole image restoration. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 479–486.
10. Xu, J.; Zhang, L.; Zuo, W.; Zhang, D.; Feng, X. Patch group based nonlocal self-similarity prior learning for image denoising. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 244–252.
11. Chen, F.; Zhang, L.; Yu, H. External patch prior guided internal clustering for image denoising. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 603–611.
12. Zuo, W.; Zhang, L.; Song, C.; Zhang, D. Texture enhanced image denoising via gradient histogram preservation. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1203–1210.
13. Gu, S.; Zhang, L.; Zuo, W.; Feng, X. Weighted nuclear norm minimization with application to image denoising. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2862–2869.
14. Zhao, Q.; Meng, D.; Xu, Z.; Zuo, W.; Yan, Y. L1-norm low-rank matrix factorization by variational bayesian method. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 825–839. [CrossRef] [PubMed]
15. Schmidt, U.; Roth, S. Shrinkage fields for effective image restoration. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2774–2781.
16. Chen, Y.; Yu, W.; Pock, T. On learning optimized reaction diffusion processes for effective image restoration. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5261–5269.
17. 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]
18. Zhang, K.; Zuo, W.; Zhang, L. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Trans. Image Process.* **2018**, *27*, 4608–4622. [CrossRef] [PubMed]
19. Guo, S.; Yan, Z.; Zhang, K.; Zuo, W.; Zhang, L. Toward convolutional blind denoising of real photographs. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1712–1722.
20. Ma, R.; Zhang, B.; Hu, H. Gaussian pyramid of conditional generative adversarial network for real-world noisy image denoising. *Neural Process. Lett.* **2020**, *51*, 2669–2684. [CrossRef]
21. Anwar, S.; Barnes, N. Real image denoising with feature attention. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3155–3164.
22. Li, D.; Chen, H.; Jin, G.; Jin, Y.; Chen, E. A multiscale dilated residual network for image denoising. *Multim. Tools Appl.* **2020**, *79*, 34443–34458. [CrossRef]
23. Binh, P.H.T.; Cruz, C.; Egiazarian, K. Flashlight CNN image denoising. In Proceedings of the IEEE European Signal Processing Conference, Amsterdam, The Netherlands, 18–21 January 2021; pp. 670–674.
24. Quan, Y.; Chen, Y.; Shao, Y.; Teng, H.; Xu, Y.; Ji, H. Image denoising using complex-valued deep CNN. *Pattern Recognit.* **2021**, *111*, 107–113. [CrossRef]
25. Guan, X.; Hu, W.; Fu, H. Remote sensing image denoising algorithm with multi-receptive field feature fusion and enhancement. *Acta Photonica Sin.* **2022**, *51*, 365–377.
26. Zheng, M.; Zhi, K.; Zeng, J.; Tian, C.; You, L. A hybrid CNN for image denoising. *J. Artif. Intell. Technol.* **2022**, *2*, 93–99. [CrossRef]
27. Tian, C.; Zheng, M.; Zuo, W.; Zhang, B.; Zhang, Y.; Zhang, D. Multi-stage image denoising with the wavelet transform. *Pattern Recognit.* **2023**, *134*, 109050. [CrossRef]
28. Tang, Y.; Wang, X.; Zhu, J.; Gao, Y.; Jiang, A. LMENet: A lightweight multiscale efficient convolutional neural network for image denoising. In Proceedings of the IEEE Region 10 Conference 2022, Hong Kong, China, 1–4 November 2022; pp. 1–6.
29. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention 2015, Munich, Germany, 5–9 October 2015; pp. 234–241.
30. Gungor, M.A.; Gencol, K. Developing a compression procedure based on the wavelet denoising and jpeg2000 compression. *Optik* **2020**, *218*, 164933. [CrossRef]
31. Liu, P.; Zhang, H.; Zhang, K.; Lin, L.; Zuo, W. Multi-level wavelet-CNN for image restoration. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 773–782.
32. Gungor, M.A. A comparative study on wavelet denoising for high noisy ct images of COVID-19 disease. *Optik* **2021**, *235*, 166652. [CrossRef] [PubMed]
33. Qian, Y.; Huang, Z.; Fang, H.; Zuo, Z. Wglfnets: Wavelet-based global-local filtering networks for image denoising with structure preservation. *Optik* **2022**, *261*, 169089. [CrossRef]

34. Fu, H.; Cheng, J.; Xu, Y.; Wong, D.W.K.; Liu, J.; Cao, X. Joint optic disc and cup segmentation based on multi-label deep network and polar transformation. *IEEE Trans. Med. Imaging* **2018**, *37*, 1597–1605. [CrossRef] [PubMed]
35. Liu, Y.; Cheng, M.-M.; Hu, X.; Wang, K.; Bai, X. Richer convolutional features for edge detection. In Proceedings of the 2017 Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5872–5881.
36. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
38. Roth, S.; Black, M.J. Fields of experts: A framework for learning image priors. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; Volume 2, pp. 860–867.
39. 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 Eighth IEEE International Conference on Computer Vision, Vancouver, BC, Canada, 7–14 July 2001; Volume 2, pp. 416–4232.
40. Franzen, R. Kodak Lossless True Color Image Suite. 1999. Available online: http://r0k.us/graphics/kodak (accessed on 1 January 2022).
41. Zhang, L.; Wu, X.; Buades, A.; Li, X. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *J. Electron. Imaging* **2011**, *20*, 23016.