Image Denoising Based On Deep Feature Fusion And U-Net Network

Yong Zhang*

Software College, Shenyang Normal University Shenyang 110034 China

** Corresponding author. E-mail: zhyongsfw@163.com*

Received: Nov. 12, 2024; Accepted: Dec. 21, 2024

Image noise hinders the understanding of images by advanced visual tasks, and removing image noise is a challenging task. The traditional denoising methods can not only destroy the texture of the image, but can not save the image texture after removing the noise. Therefore, we propose a novel image denoising method based on deep feature fusion and U-Net network. This new method uses a two-branch U-Net network to fuse features and preserve image texture. In this paper, two encoders with independent parameters are proposed to extract more useful information respectively, and a fusion module with series connection is proposed to make better use of the extracted information and remove redundant information. Finally, the decoder is used to reconstruct the image, and the U-Net peer connection is used on the symmetric convolutional layer in the network. A large number of experimental results show that the proposed algorithm can effectively remove synthetic noise and real noise, and the reconstructed image has a good effect on both subjective visual effect and objective evaluation index.

Keywords: Image denoising, deep feature fusion, U-Net network, symmetric convolutional layer

http://dx.doi.org/10.6180/jase.202510_28(10).0020

1. Introduction

In the process of image acquisition, storage and transmission, different noises are inevitably introduced to the image. The presence of noise makes it impossible to obtain complete image information, which will seriously affect advanced visual understanding tasks such as image saliency detection [1], image segmentation [2], and image classification [3]. Especially in the medical field, as long as the medical image is polluted by noise, it may affect the doctor's judgment and later treatment of the patient's condition. Therefore, it is of great practical significance to study the method of image denoising. The source of noise is complex, and the types of noise are diverse, once the image is infected with noise, it is difficult to recover its original content and texture. Therefore, the purpose of image denoising is to use an observable noise image to recover its original content and texture as much as possible. The research methods of image denoising can be roughly divided into two categories: 1) traditional methods; 2) Based on convolutional neural network. Traditional methods mainly include spatial pixel feature denoising algorithms [4, 5], transform domain denoising algorithms [6, 7] and spatial and other domain combined denoising algorithms, such as 3D block matching filtering algorithms (BM3D) [8], WNNM, CSF, etc. They usually use the prior information of noisy images such as the features between pixels in the image, self-similarity, sparsity and smoothness, in order to achieve the purpose of denoising. Although it can produce a certain de-noising effect, due to the complexity of manual acquisition of prior information, the obtained traditional de-noising methods usually have high time complexity and problems such as texture loss, so it is difficult to apply in real life.

In recent years, convolutional neural networks (CNN) have shown strong ability of automatic feature extraction and can be accelerated by graphics processors [9–11]. CNN-

based methods have received extensive attention and have been studied for the task of image denoising. In the early stage of CNN-based denoising methods, a three-layer CNN with performance comparable to BM3D was first proposed, which not only achieved the same denoising performance as BM3D, but also greatly improved the processing speed, attracting the attention of the industry and the research of the academic community. Mao et al. [12] proposed an end-to-end network, which effectively used encoding-decoding network to remove noise, and used skip connection to effectively prevent gradient disappearance. While effectively reducing image noise, this method still had the problem that the reconstructed image is too smooth. Shan et al. [13] proposed a network DnCNN that combined residual learning and batch normalization operation. The denoising performance of this method was improved, but the recovered image texture information was not well restored, especially when the noise amplitude was higher, it was more obvious. Therefore, Li et al. [14] proposed a new multi-scale image denoising method based on void convolution. This method extracted a variety of different scale feature maps from the input, increased the field of view of the network, and helped to recover the texture information of the image to a certain extent. After that, Lee et al. [15] proposed to use recoverable up and down sampling to downsample the noisy images into multiple images to achieve denoising on small images, thus expanding the field of view without increasing the network consumption. Wu et al. [16] also proposed a network RNAN based on the non-local attention mechanism, which effectively utilized the non-local prior information of images to further optimize the texture of the recovered images. Although the mentioned CNN-based denoising methods have advanced denoising performance, while removing noise, they will introduce a certain degree of damage to the image texture, resulting in the image can not retain the image texture after denoising.

In order to overcome the above shortcomings, this paper constructs an image fusion denoising network based on U-Net, which can effectively reconstruct more image textures. Considering that different input images contain different image information, this paper proposes to use two parameter independent encoders to extract more useful information respectively, and proposes a series connection mode fusion module to make better use of the extracted information and remove redundant information. Finally, the decoder is used to reconstruct the image, the U-Net peer connection is used on the symmetric convolutional layer in the network, and the mean square error is used to train the network. The proposed network is composed of convolutional layer, batch normalization operation (BN) and nonlinear function (Leaky ReLU), and the output feature map of each layer is used as the input of the next layer. The encoder and decoder are symmetric and consist of two convolutional layers respectively, while the fusion consists of one convolutional layer and several residual blocks. In this paper, the image fusion denoising method first selects two pre-trained models with different denoising parameters (one parameter is the same as the noise amplitude of the noise image, and the other is slightly less than the noise amplitude of the noise image) of a denoising method, and then puts the same noise image into the two pre-trained models to obtain two initial denoising results. In the two initial denoising results obtained by the above method, the denoising effect is better than the texture retention effect in one result, and the texture retention effect is better than the texture denoising effect in the other result. Then the two initial denoising results are placed in different encoders to extract the features of the image, and the feature map containing different information is obtained. Then the fusion device composed of residual blocks is used to fuse the two features, and finally the noise-free image is reconstructed by the decoder.

2. Materials and methods

The structure diagram of the image denoising method that fuses features and retains textures in this paper is shown in Fig. 1, which is divided into three parts: feature extraction (encoder), feature fusion (fusion module) and image reconstruction (decoder). The steps of the new method in this paper are as follows:

1. Selecting an existing denoising method and using a noise image to pre-train two pre-training models with different denoising parameters of the method, namely pre-training model A and pre-training model B in Fig. 1;

2. The initial denoising result 1 and initial denoising result 2 obtained in step 1 are put into two identical encoders with independent parameters to extract different image features;

3. Put the features obtained in step 2 into the fusion module for feature fusion, and remove redundant features to obtain more useful features;

4. The decoder is used to reconstruct the final fusion denoising image from the feature map obtained in step 3.

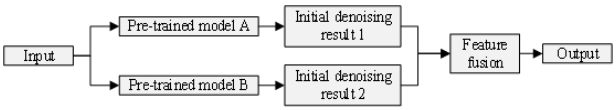

Fig. 1. Proposed image denoising method

2.1. Image fusion denoising network

The multi-branch convolutional neural network (MBCNN) proposed in this paper uses ResNet-50, which is pre-trained in ImageNet, as the backbone network. At present, ResNet is the most widely used CNN feature extraction network, which solves the problem of decreasing classification accuracy of deepening network layers. The feature of ResNet is to propose basic residual learning building blocks. The lowest level of a residual module can not only receive input from the middle layer, but also directly receive input from the top layer through short-circuiting. The advantage of this is that when the gradient is backpropagated, the lowest layer can also directly receive the uppermost gradient to avoid the situation of gradient dispersion explosion in the deep layer of the network. Through the design of this residual module, ResNet can well solve the model degradation after the deepening of the convolutional neural network, so that the network has enough depth to obtain high-level semantic information. The features in the image include low, middle and high semantic information, and the features extracted from the lower convolutional layer represent more obvious local information. With the deepening of the number of convolutional layers, the feature information extracted by convolutional layers becomes more and more abstract and can better represent the global information. ResNet-50 model is used to extract the feature maps of different levels of storage tanks. The extracted three levels of high, medium and low are fused, so that the fused features contain the feature information of the three levels, which can reflect the scene information more comprehensively. The deep CNN learns the feature maps of the three levels respectively, and then inputs them to the fusion layer. The fusion layer combines different hierarchical features into a higher-order feature set, enabling the learning of complementary relationships between different features.

The proposed multi-branch structure is a typical convolutional network style, and conv3-x, conv4-x and conv5-x are selected as low, medium and high features for the three branches respectively, as shown in Table 1. The branch consists of a series of Conv, Batch Normalization layers (BN) and linear commutation activation function (ReLu), namely: Conv-BN-Relu, the size of convolution kernel is 3×3, the number of convolution kernel is 128, after two convolution operations, the feature size of the three levels is matched, and the dimension is reduced to the unified size.

Low-level features are obtained from the $l-th$ layer, and the formula of the $l-th$ layer feature map y_l is:

$$y_l = \mathrm{ReLu}\left(\sum_{M^{l-1}} y_{l-1} \times k_l + b_l\right) \tag{1}$$

Where y_l represents the extracted low-level feature. $\times$ indicates a convolution operation. y_{l-1} is the feature map of $l-th$ layer. k_l represents the convolution kernel between y_{l-1} and y_l for concatenation. b_l represents the bias of y_l. ReLu is the activation function. M^{l-1} indicates the number of feature maps at $l-th$ layer.

Middle features are obtained from the m - th layer, and the m-th layer feature map y_m formula is:

$$y_m = \mathrm{ReLu}\left(\sum_{M^{m-1}} y_{m-1} \times k_m + b_m\right) \tag{2}$$

Where y_m represents the extracted middle-level feature. y_{m-1} is the feature map of $m-th$ layer. k_m represents the convolution kernel between y_{m-1} and y_m for concatenation. b_m represents the bias of y_m.ReLu is the activation function. M^{m-1} indicates the number of feature maps at $m-th$ layer.

High features are obtained from the $h-th$ layer, and the $h-th$ layer feature map y_h formula is:

$$y_h = \mathrm{ReLu}\left(\sum_{M^{h-1}} y_{h-1} \times k_h + b_h\right) \tag{3}$$

Where y_h represents the extracted high-level feature. y_{h-1} is the feature map of $h-$ th layer. k_h represents the convolution kernel between y_{h-1} and y_h for concatenation. b_h represents the bias of y_h. ReLu is the activation function. M^{h-1} indicates the number of feature maps at $h-th$ layer.

Due to the complexity of some images, it is necessary to carry out multi-level fusion of features extracted by CNN model. This paper puts forward a new fusion method for multi-level feature fusion, namely splitting-fusion-aggregating. The feature channels are divided into multiple groups to reduce the complexity and solve the fusion problem of each group separately. Existing methods usually operate on the overall feature channel, which has a very complex non-uniform distribution, which contains the concept of similarity between different objects, for example, appearance, shape, color, etc.

The main idea is to group the low, middle and high level features extracted based on CNN network into channel groups, and divide the feature channels into G(G takes 32) smaller groups to make more effective use of the feature information. In each group, the feature information is

Table 1. Branch network structure configuration

Feature level	Layer name	Input size	Output size
Low level	Conv3-x	$28 \times 28 \times 512$	$7 \times 7 \times 128$
Middle level	Conv4-x	$14 \times 14 \times 1024$	$7 \times 7 \times 128$
High level	Conv5-x	$7 \times 7 \times 2048$	$7 \times 7 \times 128$

further extracted, and the fusion is carried out within the group to obtain the fusion feature of the model. Then polymerization completes the integration process. The fusion process consists of five steps: input - disassembly - fusion - aggregation - output.

1. The input part is the low-level feature y_l, the intermediate feature y_m and the high-level feature y_h extracted by the network in this paper.

2. Partitioning is to divide the low 128 -dimensional feature channels into 32 groups, each group has 4 feature channels, expressed as $\{y_l^1, y_l^2, \cdots, y_l^{32}\}$. The middle-layer 128-dimensional feature channels are divided into 32 groups, which are represented as $\{y_m^1, y_m^2, \cdots, y_m^{32}\}$. The 128 -dimensional feature channels in the high layer are divided into 32 groups, which are represented as $\{y_h^1, y_h^2, \cdots, y_h^{32}\}$, as shown in Eq. (4).

$$\begin{cases} y_l \to \left\{y_l^1, y_l^2, \cdots, y_l^{32}\right\} \\ y_m \to \left\{y_m^1, y_m^2, \cdots, y_m^{32}\right\} \\ y_h \to \left\{y_h^1, y_h^2, \cdots, y_h^{32}\right\} \end{cases} \tag{4}$$

3. Fusion. The three split grouping features, high, medium and low, are connected by group and fused together accordingly. The size of the merged feature group is equal to the sum of the feature sizes of the three hierarchical groups, as shown in Eq. (5).

$$y_i = \sum_{i=1}^{G=32} \left(y_l^i + y_m^i + y_m^i\right) \tag{5}$$

4. Aggregation. The groups after fusion are further combined, and these groups are aggregated, as shown in Eq. (6).

$$y = \cup_{i=1}^{G=32} y_i \tag{6}$$

5. Output. The aggregated features fuse three different levels of features, and then perform global average pooling operations on the aggregated features to further integrate global information.

2.2. Modified loss function

In the process of deep learning sample training, one-hot label is used to calculate cross entropy loss, and cross entropy loss LCE is expressed as:

$$LCE = \sum_{i=1}^{m} p(x) \log q(x) \tag{7}$$

Where $q(x) = \frac{e^{w_{y_i}^T x_i + b_i}}{\sum_{j=1}^{n} e^{w_{y_i}^T x_i + b_i}}$, m is the sample number of each batch. n is the number of sample types. $x_i \in R^d$ represents the $i-th$ depth feature and belongs to class y_i. D is the feature size. $W_j \in R^n$ represents column j of weight $W \in R^{d \times n}$ in the last fully connected layer.$b \in R^n$ is the offset term. $p(x) = [x_{i1}, x_{i2}, \cdots, x_{in}]$ is the type label vector corresponding to the sample, where only one value is 1 and the rest is $\underline{0}$. Where $q(x)$ is the true probability distribution for the sample. $p(x)$ is the predicted probability distribution for the sample.

The prediction probability is evaluated by the forward propagation algorithm, and the distance between the two probability distributions $q(x), p(x)$ is minimized by the cross moisture loss LCE. However, the standard cross entropy loss function has some disadvantages. It assigns the same weight to the samples, and does not discriminate between the difficult and easy samples, resulting in an imbalance between the accuracy and the classification accuracy. Only the loss of the correct label position (one-hot label 1) in the training sample is considered, while the loss of the wrong label position (one-hot label 0) is ignored, so that the model can fit well on the training set. However, because the loss of other error label positions is not calculated, the probability of prediction errors increases when forecasting.

To solve the problem of cross moisture, a new loss function LNCE based on cross moisture is proposed in this paper, as shown in Eq. (8).

$$\begin{cases} LNCE = (1-\lambda)L_1 + \lambda L_2 \\ LNCE = -(1-\lambda)[1 - \log q(x)]^\gamma \sum_{i=1}^{m} p(x) \log q(x) \\ \quad -\lambda[1 - \log q(x)]^\gamma \sum_{i=1}^{m} p'(x) \log q(x) \end{cases} \tag{8}$$

There are two parts involved, which use hyperparameter input to balance L1 and L2. Inspired by reference

[17], modulation factors in the loss of L1 and L2 are proposed, as shown in Eq. (9). Adding the modulation factor $[1-\log q(x)]^{\gamma}$ to the standard cross entropy and setting $\gamma > 0$ can reduce the relative loss of well-classified samples and pay more attention to the hard-to-distinguish samples.It can be seen that when a sample is misclassified and $\log q(x)$ is very small, $[1-\log q(x)]^{\gamma}$ is close to 1 and the loss is not affected. When $\log q(x)$, $[1-\log q(x)]^{\gamma}$ tends to 0 , that is, the sample losses of well-classified samples are weighted and the sample weights are smoothed.

$$L1 = [1-\log q(x)]^{\gamma} \sum_{i=1}^{m} p(x) \log q(x) \tag{9}$$

In order to pay attention to both the loss of correct label position and the loss of wrong label position, the following loss function L2 based on label smoothing is proposed, as shown in Eq. (10). Introducing a fixed distribution $p(x)$, it is the type label vector corresponding to the sample, where the values are all 1. It can be seen that L2 loss takes both correct and incorrect label position losses into account, which leads to an increase in the final loss and an improvement in the learning ability of the model. That is, to fall back to the original loss, you have to learn better, that is, the $p(x)$ prediction probability results are closer to the correct classification, and equally as far as possible away from the wrong classification.

$$L2 = -[1-\log q(x)]^{\gamma} \sum_{i=1}^{m} p'(x) \log q(x) \tag{10}$$

The classifier in this paper is softmax, and the gradient descent algorithm is set to:

$$W^{n+1} = W^{n} - \delta \frac{\partial}{\partial W^{n}} \mathrm{LNCE}\left(W^{n}\right) \tag{11}$$

Where δ is the learning rate. W is the parameter vector of the softmax classifier. $\mathrm{LNCE}\left(W^{n}\right)$ is a loss function proposed in this paper. The gradient descent algorithm is used to feed back and optimize the classifier parameters. The backpropagation algorithm is then combined with the gradient descent algorithm to update each parameter.

3. Experimental results and analysis

To verify the effectiveness of the proposed method, a diverse 2K resolution high-quality image dataset with different noise levels (DIV2K, diverse 2K resolution high-quality images dataset) and Smart Phone Image Denoising Dataset with Real Noise (SIDD) are used to train the proposed method to remove synthetic noise and real noise from images respectively. DIV2K dataset consists of 800 high-definition images with a resolution of 2K, and SIDD dataset contains 10 scenes. In each training batch, 16 64×64 image blocks are input for the synthetic noise removal experiment. 16 128×128 image blocks are input for the real noise removal experiment. ADAM algorithm is used to optimize the network parameters. The initial learning rate is set to 10^{-4}. Proposed method is based on PyTorch framework and uses GeForceRTX 1080Ti GPU for training.

PSNR (peak signal to noise ratio) and SSIM (structural similarity) are used to evaluate the difference between the images.

3.1. Synthetic noise removal experiment

The test set of the synthetic noise removal experiment adopts the grayscale data set Kodak lossless true color image set (Kodak24) and the color dataset Berkeley segmentation dataset 68 (BSD68), adding Gaussian white noise with standard deviation σ of 10, 30, 50 and 70 to the sharp image to obtain a noisy image.

The comparison algorithm is the traditional BM3D algorithm, deep learning-based TNRD (trainable nonlinear reaction diffusion) algorithm [18], and the residual encoder-decoder algorithm (RED) [19], denoising convolutional neural networks (DnCNN) algorithm, image restoration convolutional neural networks (IRCNN) [20], fast and flexible de-noising convolutional neural network (FFDNet) algorithm [21], real image denoising network (RIDNet) [22], color block-matching and 3D filtering (CBM3D) algorithm [23] and variational depth image restoration (VDIR) algorithm [24].

The denoising performance of the proposed algorithm and the comparison algorithm for images with different noises in the two data sets is shown in Table 1 and Table 2 respectively.

It can be seen that on the gray data set Kodak24, the average PSNR of the proposed algorithm is optimal. When σ is 30,50 and 70 , the average PSNR of the proposed algorithm is 0.05 dB , 0.1 dB and 0.11 dB higher than that of RIDNet algorithm, respectively. On the color data set BSD68, the average PSNR value of the proposed algorithm is the best. When σ is 30,50 and 70 , the average PSNR value is 0.21 dB, 0.23 dB and 0.26 dB higher than RIDNet algorithm, respectively.

In a synthetic noise environment of $\sigma = 50$, the subjective denoising effects of the proposed algorithm and the comparison algorithm on the two datasets are shown in Figs. 2 and 3. As can be seen from Fig. 2, on the grayscale images of Kodak24 data set, the edges of flowers with the BM3D algorithm restored images are cloudy, and the texture inside the flowers is also very fuzzy. DnCNN and FFDNet algorithms can reconstruct clear flower texture, but the window partition is fuzzy. The reconstruction result of

Table 2. Comparative results of synthetic noise removal experiments (Grayscale images) on Kodak24 data set

Method	$\sigma = 10$	$\sigma = 30$	$\sigma = 50$	$\sigma = 70$
BM3D	34.50	29.24	27.10	25.84
TNRD	34.52	28.98	27.31	25.06
RED	35.13	29.88	27.77	26.50
DnCNN	35.01	29.73	27.62	26.19
IRCNN	34.87	29.64	27.56	26.21
FFDNet	34.92	29.81	27.74	26.45
RIDNet	34.93	30.01	27.90	26.62
Proposed	35.13	30.06	28.00	26.73

Table 3. Comparative results of synthetic noise removal experiments (Grayscale images) on BSD68 data set

Method	$\sigma = 10$	$\sigma = 30$	$\sigma = 50$	$\sigma = 70$
TNRD	33.47	27.75	26.07	23.94
RED	34.00	28.57	26.46	25.20
DnCNN	36.42	30.51	28.12	26.67
IRCNN	36.17	30.33	27.97	26.51
FFDNet	36.25	30.42	38.07	26.64
RIDNet	36.24	30.58	28.23	26.80
CBM3D	36.02	29.84	27.49	26.11
VDIR	36.45	30.75	28.44	27.05
Proposed	36.55	30.79	27.46	27.06

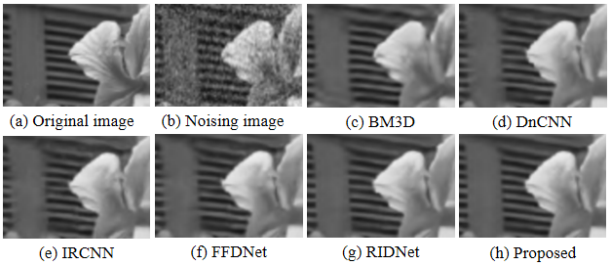

Fig. 2. Raw images, noisy images and denoised images with different algorithms in Kodak24 dataset

Fig. 3. Raw images, noisy images and denoised images with different algorithms in BSD68 dataset

the algorithm in this paper is closest to the original image, and we can see clear flower texture and window partition. As can be seen from Fig. 3, in the color images of the BSD68 data set, the images restored by BM3D algorithm are very fuzzy, and the phenomenon of excessive smoothing of the main body edges appears. The chicken down in the image restored by DnCNN and FFDNet algorithm is not realistic enough, and a large number of artifacts appear at the edge of the main body. In this paper, the algorithm reconstructed a very realistic chicken feather, no fuzzy phenomenon, the subjective feeling is the best.

3.2. Real noise removal experiment

The test set of real noise removal experiment adopts SIDD verification data set, which contains 1280 256×256 real noise images and corresponding clear images. The noise performance of the proposed algorithm and the comparison algorithm on this data set is shown in Table 4. It can be seen from the table that the PSNR value of the proposed algorithm is 0.27dB higher than that of the VDIR algorithm. The number of parameters of MIRNet algorithm is nearly 3 times that of the proposed method, but its PSNR value is only 0.12dB higher than that of the proposed algorithm. These results show that the proposed algorithm has a better ability to remove real noise than synthetic noise.

Table 5 compares the computational complexity of different algorithms. Where, FLOPs represents the number of floating-point operations per second required to compute a 256×256 color image. It can be seen from the table that the proposed algorithm has a smaller number of parameters than CBDNet and DANet algorithms and achieves better denoising performance. The performance of the MIRNet algorithm is slightly better than that of the proposed algorithm, but the number of parameters and FLOPs of the

Table 4. Comparative results of real noise removal experiments

Method	Parameter size/MB	PSNR	SSIM
CBM3D	6.3	25.76	0.686
DnCNN	0.8	23.77	0.584
RIDNet	1.6	38.82	0.915
VDN	7.9	39.39	0.909
DANet	64	39.58	0.958
MIRNet	31.9	39.83	0.959
VDIR	2.3	39.44	0.957
DUMRN	26.7	39.55	0.957
Proposed	12.2	39.71	0.958

model are 2.6 times and 1.7 times that of the proposed algorithm respectively, which consumes a lot of computing resources.

3.3. Ablation experiment

In order to evaluate the performance of the proposed method in different modules, BSD68 color image dataset is adopted as the test set. In the case of $\sigma = 50$, the proposed method is combined with ordinary differential equation network (ODENet) [25]. The comparison and ablation experiment results are shown in Table 6.

It can be seen that the PSNR value of D2 algorithm is 0.14dB higher than that of U-Net algorithm, which verifies the effectiveness of multi-scale feature extraction module. The PSNR value of D1 algorithm is 0.12dB higher than that of U-Net algorithm, and the PSNR value of this algorithm is 0.03dB higher than that of D2 algorithm, both of which verify the advantages of multi-scale feature fusion module in removing noise.

4. Conclusion

In this paper, the existing image denoising method based on convolutional neural network will introduce a certain degree of damage to the image texture while removing noise, resulting in the image can not retain the image texture after denoising, and a new image denoising method is proposed. The method first selects a denoising method, and uses the pre-trained model of two different denoising parameters of the denoising method to obtain different initial denoising results of the same noisy image respectively. Then, in order to increase the information that can be fused, Two encoders were used to extract the features of different denoising results, and the features of the image were fused into the fusion module at the same time. Finally, the noise-free image was reconstructed by the decoder. Experiments show that the proposed method has strong generalization ability on different data sets, and the texture information recovered from noiseless images is more consistent with the original texture. It also proves that the proposed fusion method can effectively use the prior information of images to fuse clearer image textures. Compared with the existing popular image denoising methods, the proposed method is superior to these two mainstream methods both in visual effect and objective index.

References

[1] Y. Li, G. Liu, D. P. Bavirisetti, X. Gu, and X. Zhou, (2023) *"Infrared-visible image fusion method based on sparse and prior joint saliency detection and LatLRR-FPDE"* **Digital Signal Processing *134***: 103910. DOI: 10.1016/j.dsp.2023.103910.

[2] L. Teng, Y. Qiao, M. Shafiq, G. Srivastava, A. R. Javed, T. R. Gadekallu, and S. Yin, (2023) *"FLPK-BiSeNet: Federated learning based on priori knowledge and bilateral segmentation network for image edge extraction"* **IEEE Transactions on Network and Service Management *20***(2): 1529–1542. DOI: 10.1109/TNSM.2023.3273991.

[3] X. Meng, X. Wang, S. Yin, and H. Li, (2023) *"Few-shot image classification algorithm based on attention mechanism and weight fusion"* **Journal of Engineering and Applied Science *70***(1): 14. DOI: 0.1186/s44147-023-00186-9.

[4] S. Wang, L. Li, X. Li, J. Zhang, L. Zhao, X. Su, and F. Chen, (2023) *"A denoising network based on frequency-spectral-spatial-feature for hyperspectral image"* **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing *16***: 6693–6710. DOI: 10.1109/JSTARS.2023.3285454.

[5] A. Ulu, G. Yildiz, and B. Dizdaroğlu, (2023) *"MLFAN: Multilevel Feature Attention Network With Texture Prior for Image Denoising"* **IEEE Access *11***: 34260–34273.

[6] Z. Li, H. Liu, L. Cheng, and X. Jia, (2023) *"Image denoising algorithm based on gradient domain guided filtering and NSST"* **IEEE Access *11***: 11923–11933. DOI: 10.1109/ACCESS.2023.3242050.

Table 5. Comparison of model complexity and PSNR values

Method	Parameter size/MB	FLOPs/G	PSNR/dB
CBDNet	4.4	80	39.57
DANet	63.1	65	39.58
MIRNet	31.9	1572	39.83
Proposed	12.3	923	39.71

Table 6. Results of ablation experiments on BSD68 data set with different denoising algorithms

Method	ODE	Single scale	multi-scale	PSNR/dB
ODENet	✓			27.77
U-Net		✓		28.29
D1		✓		28.41
D2			√	28.43
Proposed	✓		✓	28.46

[7] A. M. H. Abadi and M. R. H. Fatemi, (2023) *"Iterative based image and video denoising by fractional block matching and transform domain filtering"* **Authorea Preprints**:

[8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, (2007) *"Image denoising by sparse 3-D transform-domain collaborative filtering"* **IEEE Transactions on image processing *16***(8): 2080–2095. DOI: 10.1109/TIP.2007.901238.

[9] S. Yin, H. Li, Y. Sun, M. Ibrar, and L. Teng, (2024) *"Data Visualization Analysis Based on Explainable Artificial Intelligence: A Survey"* **IJLAI Transactions on Science and Engineering *2***(2): 13–20.

[10] S. Yin, H. Li, A. A. Laghari, T. R. Gadekallu, G. A. Sampedro, and A. Almadhor, (2024) *"An anomaly detection model based on deep auto-encoder and capsule graph convolution via sparrow search algorithm in 6G internet-of-everything"* **IEEE Internet of Things Journal *11***(18): 13–20. DOI: 10.1109/JIOT.2024.3353337.

[11] H. Li and C. Zhao, (2024) *"Fusion Cycle GAN: A Multiple Feature Fusion based Cycle-consistent Generative Adversarial Network for Person Re-Identification"* **Journal of Science and Engineering *1***(1): 7–12.

[12] X. Mao, C. Shen, and Y.-B. Yang, (2016) *"Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections"* **Advances in neural information processing systems *29***: 2810–2818.

[13] C. Shan, X. Guo, and J. Ou, (2019) *"Residual learning of deep convolutional neural networks for image denoising"* **Journal of Intelligent & Fuzzy Systems *37***(2): 2809–2818. DOI: 10.3233/JIFS-190017.

[14] Y. Li, Y. Wang, and N. Wu, (2021) *"Noise suppression method based on multi-scale Dilated Convolution Network in desert seismic data"* **Computers & Geosciences *156***: 104910. DOI: 10.1016/j.cageo.2021.104910.

[15] S. Lee, M. Negishi, H. Urakubo, H. Kasai, and S. Ishii, (2020) *"Mu-net: Multi-scale U-net for two-photon microscopy image denoising and restoration"* **Neural Networks *125***: 92–103. DOI: 10.1016/j.neunet.2020.01.026.

[16] X. Wu, K. Zhang, Y. Hu, X. He, and X. Gao, (2024) *"Multi-scale non-local attention network for image super-resolution"* **Signal Processing *218***: DOI: 10.1016/j.sigpro.2023.109362.

[17] S. Yin, L. Wang, M. Shafiq, L. Teng, A. A. Laghari, and M. F. Khan, (2023) *"G2Grad-CAMRL: an object detection and interpretation model based on gradient-weighted class activation mapping and reinforcement learning in remote sensing images"* **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing *16***: 3583–3598. DOI: 10.1109/JSTARS.2023.3241405.

[18] Y. Chen and T. Pock, (2016) *"Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration"* **IEEE transactions on pattern analysis and machine intelligence *39***(6): 1256–1272. DOI: 10.1109/TPAMI.2016.2596743.

[19] G. Cheng, A. Matsune, Q. Li, L. Zhu, H. Zang, and S. Zhan. "Encoder-decoder residual network for real super-resolution". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2019. DOI: 10.1109/CVPRW.2019.00270.

[20] X. Deng and P. L. Dragotti, (2020) *"Deep convolutional neural network for multi-modal image restoration*

and fusion" **IEEE transactions on pattern analysis and machine intelligence *43***(10): 3333–3348. DOI: 10.1109/TPAMI.2020.2984244.

[21] K. Zhang, W. Zuo, and L. Zhang, (2018) *"FFDNet: Toward a fast and flexible solution for CNN-based image denoising"* **IEEE Transactions on Image Processing *27***(9): 4608–4622. DOI: 10.1109/TIP.2018.2839891.

[22] R. Ma, S. Li, B. Zhang, and Z. Li, (2021) *"Towards fast and robust real image denoising with attentive neural network and PID controller"* **IEEE Transactions on Multimedia *24***: 2366–2377. DOI: 10.1109/TMM.2021.3079697.

[23] D. Honzátko and M. Kruliš, (2019) *"Accelerating block-matching and 3D filtering method for image denoising on GPUs"* **Journal of Real-Time Image Processing *16***(6): 2273–2287. DOI: 10.1007/s11554-017-0737-9.

[24] R. Kaftory, Y. Y. Schechner, and Y. Y. Zeevi. "Variational distance-dependent image restoration". In: *2007 IEEE conference on computer vision and pattern recognition*. IEEE. 2007, 1–8. DOI: 10.1109/CVPR.2007.383262.

[25] T. Ma, H. Zhou, Y. Tian, and N. Al-Nabhan, (2021) *"A novel rumor detection algorithm based on entity recognition, sentence reconfiguration, and ordinary differential equation network"* **Neurocomputing *447***: 224–234. DOI: 10.1016/j.neucom.2021.03.055.