



Multi-scale information fusion generative adversarial network for real-world noisy image denoising

Xuegang Hu¹ · Wei Zhao¹

Received: 29 December 2023 / Revised: 28 May 2024 / Accepted: 7 June 2024 / Published online: 18 June 2024
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

Image denoising is crucial for enhancing image quality, improving visual effects, and boosting the accuracy of image analysis and recognition. Most of the current image denoising methods perform superior on synthetic noise images, but their performance is limited on real-world noisy images since the types and distributions of real noise are often uncertain. To address this challenge, a multi-scale information fusion generative adversarial network method is proposed in this paper. Specifically, In this method, the generator is an end-to-end denoising network that consists of a novel encoder–decoder network branch and an improved residual network branch. The encoder–decoder branch extracts rich detailed and contextual information from images at different scales and utilizes a feature fusion method to aggregate multi-scale information, enhancing the feature representation performance of the network. The residual network further compensates for the compressed and lost information in the encoder stage. Additionally, to effectively aid the generator in accomplishing the denoising task, convolution kernels of various sizes are added to the discriminator to improve its image evaluation ability. Furthermore, the dual denoising loss function is presented to enhance the model’s capability in performing noise removal and image restoration. Experimental results show that the proposed method exhibits superior objective performance and visual quality than some state-of-the-art methods on three real-world datasets.

Keywords Image denoising · Multi-scale information fusion · Generative adversarial network · Dual denoising loss

1 Introduction

Image denoising not only enhances visual perception, but also ensure the integrity and authenticity of the image, which is convenient for subsequent target detection [1], image classification [2] and medical image processing [3]. Especially, with the rapid development of computer vision technology, image denoising becomes more important.

Image noise can be broadly classified into two types: synthetic noise and real-world noise (or real noise). Synthetic noise refers to noise that follows a certain probability distribution and whose noise level can be set independently, such as gaussian noise, salt-and-pepper noise, and gamma

noise. However, the type and distribution of real-world noise is often uncertain. It may originate from internal components of the system or device as internal noise, or it may be external noise caused by environmental factors. This type of noise exhibits a diverse structure that is difficult to describe with simple parameters. In order to solve the problem of image denoising, researchers have proposed many methods. Generally, these methods are divided into traditional methods and deep learning methods. Traditional image denoising can be roughly categorized into two types: filter-based methods and model-based methods. Filter-based methods mainly perform noise suppression in spatial domain and transform domain, such as non-local means (NLM) [4], block matching and 3D filtering (BM3D) [5]. Model-based methods mainly design regularization with prior information for image denoising, including k-means singular-value decomposition (KSVD) [6], weighted nuclear norm minimization (WNNM) [7]. However, traditional methods still have some drawbacks, either involving cumbersome feature extraction processes, or having high computational requirements, or facing difficulties in directly handling complex real noise. Therefore, these

✉ Wei Zhao
s220603020@stu.cqupt.edu.cn

Xuegang Hu
huxg@cqupt.edu.cn

¹ School of Science, Chongqing University of Posts and Telecommunications, Chongqing 400065, People’s Republic of China

methods are difficult to meet the current practical requirements.

In recent years, due to the success of deep learning, image denoising methods based on deep learning have achieved superior performance compared to traditional methods and have become the mainstream methods [8]. Among them, the deep learning-based convolutional neural network (CNN) model [9] has been widely used in image denoising. For example, DNCNN [10] was the first method to use CNN for blind denoising. It employed a residual learning strategy, allowing the network to directly learn the difference between the noisy input image and the clean image. This enabled the network to focus more on learning the characteristics of noise, rather than the entire image content. Additionally, it applied batch normalization after each convolutional layer to accelerate the training process and improve model performance. However, DnCNN may have limited generalization capabilities for certain specific texture or detail patterns. This could result in the smoothing of these textures or details while removing noise. FFDNet [11] used adjustable noise level maps as inputs to the model to achieve non-blind denoising. Nevertheless, its noise level maps needed to be manually set based on empirical knowledge. CBDNet [12] was an extension of FFDNet, which designed a network for estimating noise and thus becoming a blind denoising model. RIDNet [13] introduced feature attention modules to enhance the information interaction of the network on channel features, so as to achieve significant denoising performance. To enhance the denoising performance of the model, MIRNet [14] designed multiple novel modules with attention mechanisms to extract multi-scale feature information. MPRNet [15] built a multi-stage architecture to exchange information across different stages, reducing the loss of detailed information and thus better balancing the competing goals of spatial details and high-level contextual information during the image restoration phase. Recently, some new denoising methods have been proposed. For example, APD-Nets [16] first attempted to introduce adaptive regularization and complementary prior information into denoising networks, thereby improving the generalization ability and image quality restoration of denoising networks. MSIDNet [17] utilized a fusion mechanism to fully utilize multi-scale features, enhancing the network's information perception and thus improving the image's visual effects. MDRN [18] introduced a multi-scale feature extraction module alongside a dilated residual module, which are designed to extract multi-scale features, thereby enhancing the performance of image restoration. TSIDNet [19] employed a data sub-network for denoising and a feature extraction network for global feature extraction, then aggregated the information from both networks to enhance the model's robustness and denoising performance. To address the non-blind denoising and noise estimation issues of many CNN methods for real image

denoising, CFNet [20] designed a new conditional filter to adaptively adjust the denoising manner and an affine transformation block for noise prediction. Although CNN-based methods could accomplish achieve noise removal, these methods were purely end-to-end, without utilizing the neural network to provide a more detailed evaluation of the denoised images during the training process. From this analysis, these methods not only led to the underutilization of the learning capacity of CNN and the parallel computing capability of GPU but also potentially resulted in inconsistent quality of the denoised images.

The generative adversarial network (GAN) model [21] based on deep learning was proposed by Goodfellow. It utilized neural networks to offer detailed assessments of the processed data during training, leading to superior performance compared to CNN in specific image processing tasks, including image translation [22] and image inpainting [23–25]. GAN model consists of two parts: a generator and a discriminator. The role of a generator is to generate data that is as consistent as possible with the original data distribution. The discriminator is more of an auxiliary generator in the entire model, and its main task is to judge whether the given data is the original real sample data or the fake sample data forged by the generator model. In the training phase, the two models compete with each other through the max-min game. The objective function of GAN model [21] is as follows:

$$\min_G \max_D V(D, G) = E_{x \sim p_{\text{data}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (1)$$

where G is the generator, D is the discriminator. $p_{\text{data}}(x)$ represents the distribution of real data, and $p_z(z)$ represents the distribution of random noise. $E_{x \sim p_{\text{data}}(x)} [\log D(x)]$ stands for the expected loss of the discriminator on real data, and $E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$ represents the expected loss of the discriminator on generated data. From Eq. 1, we can see that the objective function ensures that the discriminator performs well on real data (minimizing misclassification of real data), while the generator aims to produce fake data that can “fool” the discriminator (maximizing the discriminator's misclassification of fake data). This constitutes a min-max problem, as the discriminator attempts to maximize its ability to distinguish between real and fake data, while the generator seeks to minimize the discriminator's ability to distinguish its generated data. WGAN [26] improved the stability of the GAN model by replacing JS divergence or KL divergence with wasserstein distance to measure the difference between samples. WGAN-GP [27] added an additional gradient penalty term to make the gradient of the model more stable in training, thus solving the problem of pattern collapse that both GAN and WGAN were prone to. At the same

time, the gradient penalty term introduced by WGAN-GP makes the discriminator have the ability to accurately distinguish the true and false samples by maintaining the smooth gradient, making the generated samples more realistic.

Based on the above theoretical research of GAN, many scholars have successfully applied GAN to image denoising. For example, GCBD [28] was one of the earliest methods to utilize GAN to model real-world noise for constructing a noise dataset. Its approach involved the generator randomly generating noise patches, which, when combined with original images, could produce corresponding noisy images. This augmentation of the model's training dataset effectively addressed the challenge of finding paired datasets in practical applications. Similarly, ADGAN [29] utilized GAN to generate noisy samples for dataset augmentation, and it introduced a feature loss function to extract image features, thus improving the restoration performance of image details. To enhance the blind denoising performance, BDGAN [30] utilized the technology that improved the stability of training and obtained improved image quality to modify the architecture of the generator network. It also designed multiple discriminators with different receptive fields to conduct multi-scale evaluation on images, so as to obtain high-quality images. GRDN [31] proposed a new GAN-based method for modeling real noise, addressing the challenge of obtaining paired datasets. It also enhanced the denoising performance of the model through extensive and hierarchical use of residual connections. Subsequently, DANet [32] introduced an innovative bayesian framework that simultaneously completes the tasks of noise elimination and noise generation using dual adversarial learning. DeGAN [33] harnessed the mutual game between the generator network and the feature extractor network, along with additional training from the feature extractor network, to enable the generator network to accomplish a direct mapping from the noisy image domain to the noise-free image domain. Although this method effectively removed mixed noise and restored damaged images, it was not effective for more complex and realistic noisy images. HI-GAN [34] build a deeper network structure through dense residuals to improve the denoising effect. Nevertheless, this approach also resulted in the accumulation and repetition of information, leading to a decrease in feature propagation efficiency and making the model training more challenging. The recent DGCL [35] used two independent GANs to learn from denoised images and image datasets separately, partially addressing the issues of complex network structures and training difficulties associated with single GAN-based methods. However, DGCL did not perform adversarial training on the final fused results, leading to suboptimal image quality. While these GAN-based methods have brought more possibilities for image denoising, the practicability of many of these denoising models still needs to be further improved, especially on real noise images.

To solve above problems, in this paper, we redesign the generator, discriminator and loss function in the GAN model. First, the design inspiration of the generator mainly comes from references [36–41]. Specifically, literature [36, 37] proposed a multi-scale network model based on convolutional neural network, which verified that more information could be learned by using multi-scale for feature extraction. MSGAN [38] designed a new multi-scale module and added this module to the skip connections, so that the operation could improve the model performance. Literature [39] extracts more high-level information by increasing the number of convolutional layers to the network model, and utilized the residual bottleneck proposed by He et al [40] in the denoising network to solve the problem of gradient calculation caused by too deep network. DCANet [41] built a dual CNN containing two different branches to learn complementary features to obtain noisy estimated images. According to these studies, In our generator, we not only extracted multi-scale features, but also fused the extracted features before passing them through the skip connection. Our network also used two branches, one of which employed the optimized residual module that we improved, and two branches directly dealt with the noise without estimating the noise. Then, the design of discriminator was primarily inspired by PatchGAN [42]. One of the key contributions of PatchGAN was designing a discriminator that could focus on multiple regions to evaluate the images. Nevertheless, the original PatchGAN discriminator utilized downsampling for multi-scale feature extraction, which resulted in the loss of certain information and thereby limited the discriminator's ability to capture and differentiate subtle details in the images. So we made improvements to the discriminator to address this issue. Finally, there are some other works that can help us to design the loss function. For example, references [29, 30] introduced the perceptual loss proposed in image super-resolution [43] to enhance the detailed information of denoised images and improve the visual effect. The total variation algorithm [44] and literature [45] directly provided regularization loss function with denoising performance to achieve image denoising. These loss functions were adopted by us to constrain the training of our network, and inspired by them, we proposed a new function to improve the performance of the model.

In summary, to address the limited performance of most image denoising algorithms on real noise, this paper proposes a multi-scale information fusion generative adversarial network (MIFGAN) algorithm. The algorithm can be implemented in machine vision software in the future to effectively eliminate real noise, thereby enhancing image quality. This enhancement facilitates the machine vision system's ability to process and analyze image data more efficiently, ultimately leading to improved performance and accuracy of the machine system. For this paper, the main contributions are as follows:

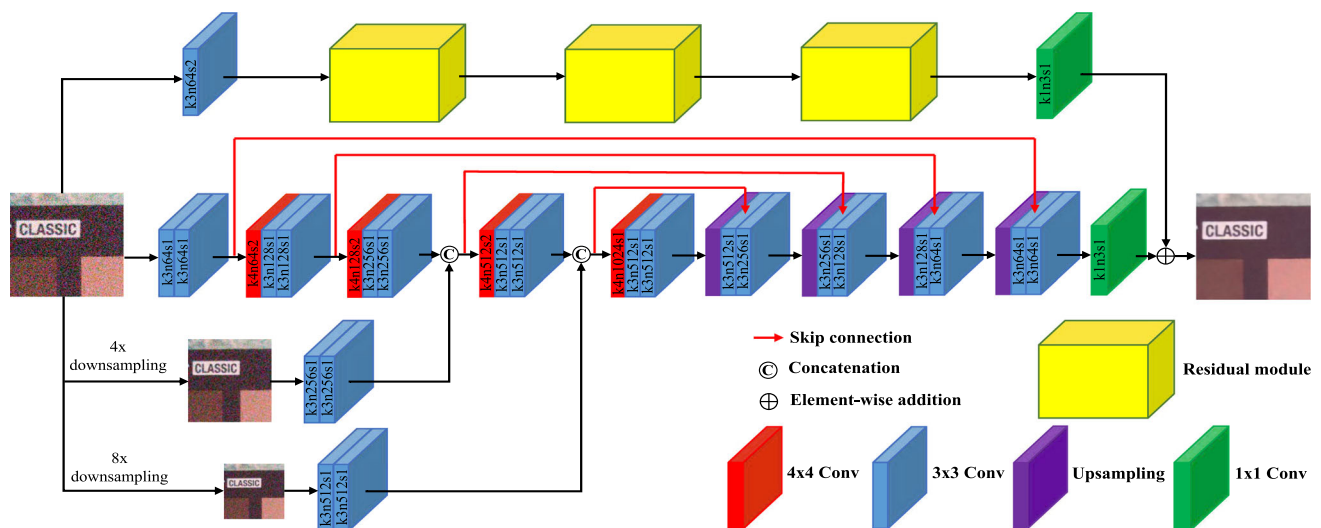


Fig. 1 The architecture of the generator in MIFGAN, where k , n and s represent the kernel size, channel number and stride in the convolutional layer respectively

- (1) Utilizing the generative adversarial network model and the concept of multi-scale information fusion, a novel denoising algorithm is proposed that can significantly enhance the quality of real image denoising.
- (2) A novel encoder–decoder network branch is designed, which extracts and fuses multi-scale features of images with different resolutions in the encoder stage, enabling noise reduction while preserving crucial image details. Moreover, to address feature compression and gradient calculation issues, an improved residual network branch is introduced.
- (3) To further improve the model’s denoising performance, we design a discriminator with richer receptive field that aims to effectively capture the global features and context information of the image, thereby the denoised image can be evaluated in more detail.
- (4) The dual denoising loss function is presented. It can be combined with other loss functions in the training phase to further optimize the performance of the model.

2 Proposed method

2.1 Generator network architecture

The generator of MIFGAN is the core of the whole network framework, which is an end-to-end denoising network. Its input is a noisy image, and its output is a clean image. The specific network architecture is shown in Fig. 1. By integrating multi-scale contextual information, more feature information can be supplemented, thereby enhancing the denoising performance of the model. The so-called multi-scale feature means that the feature maps with differ-

ent resolution sizes contain different information. Typically, higher-resolution images can provide richer details and more precisely capture edge features, while lower-resolution images can provide overall structure and global features. Therefore, We process the noisy images at three different scales: original resolution, four times downsampling, and eight times downsampling. We extract features from these images at respective scales and then combine the extracted information at appropriate positions. The specific realization process is as follows:

Due to the excellent image reconstruction capability, scalability, and adaptability of the U-net [46] network, it has shown good performance in many image processing tasks such as image super-resolution [47] and image inpainting [48]. So we modify U-net to be the backbone of the encoder–decoder branch of the generator. The encoder performs four downsampling operations for feature extraction, reducing the size of the feature maps by factors of two, four, eight, and sixteen. Meanwhile, the decoder conducts four upsampling operations for information recovery, successively increasing the reduced feature maps by factors of two, four, eight, and sixteen. Additionally, skip connections are used to transfer information extracted from the downsampling to the upsampling stage, enabling feature reuse. In order to obtain richer information such as image structure and content in the downsampling stage, we use a 4×4 convolution with a step size of 2 to replace the pooling operation. The checkerboard effect occurs during upsampling due to the use of deconvolution [49]. Therefore, we use bilinear upsampling for image reconstruction. After each downsampling and upsampling, two 3×3 convolutions with step size 1 are used to extract information on the feature map. In addition, we downsample the noisy image by a factor of four and eight, respectively, result-

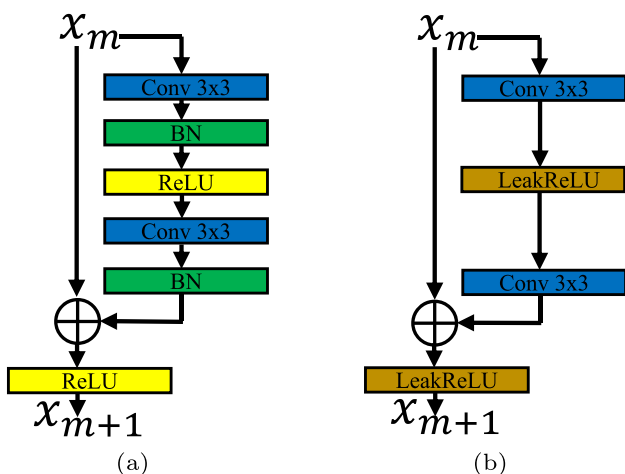


Fig. 2 **a** The architecture of residual bottleneck in ResNet. **b** The architecture of residual module in MIFGAN

ing in images with a corresponding reduction in size. Then extract shallow-level information using two 3×3 convolutions. Then, the feature map information of different scales is spliced and fused in the corresponding downsampling stage of the encoder, which can improve the feature representation ability in the subsampling, and can also transfer more information to the decoder through skip connections.

Additionally, the residual bottleneck proposed by ResNet [50] can extract feature information by stacking and can well solve the problem of gradient disappearance and gradient explosion caused by the deepening of network layers in the stacking process. Therefore, we design a new residual module by combining the advantages of the residual bottleneck. The specific architecture of the residual module designed by us and the residual bottleneck proposed by ResNet is shown in Fig. 2. In our residual module, we remove the batch normalization (BN) layer used in the original residual network. This modification improves the training speed of the model and prevents the degradation of information in denoised images caused by normalization. At the same time, we utilize the leaky rectified linear unit (LeakReLU) as the activation function in the encoder–decoder branch, because the model can still be trained stably when the activation function is negative. So, we also modify the activation function in the original residual module from rectified linear unit (ReLU) to LeakReLU. A new branch of the residual network is constructed by cascading the residual modules designed by us. It achieves image denoising and restoration by learning the residual transformation between the output denoised image and the input noisy image. Unlike the feature extraction methods of downsampling and upsampling, our residual network extracts shallow features from deep features of context directly on the original size image, avoiding the loss of detailed information caused by sampling operations. Its processing flow is that the noise image is first passed

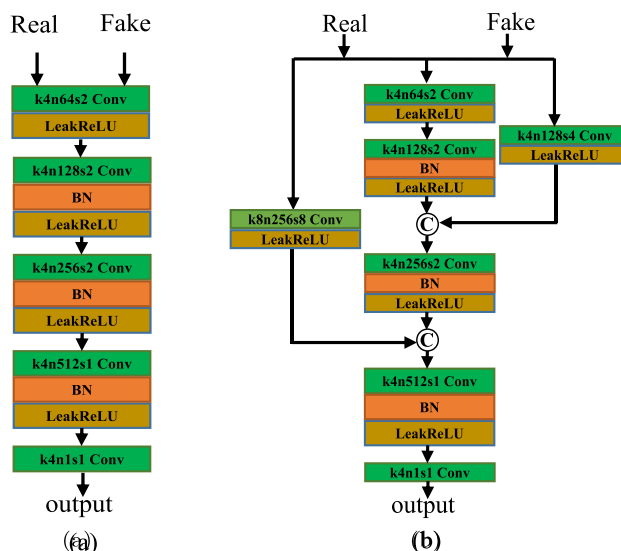


Fig. 3 **a** The discriminator architecture in PatchGAN. **b** The discriminator architecture in MIFGAN. The kernel size, channel number and stride in the convolutional layer are denoted by k , n and s respectively

through a 3×3 convolution with a step size of 1 to change the number of channels and obtain a feature map with rich information, and then the feature map is input into the residual module for information extraction. At this stage, the size of the feature map is always consistent with the input image. The ablation study results in Sect. 3.5.1 show that we can obtain improved results when using 3 residual modules in the residual branch.

We restore the feature maps obtained from the two branches to a clean image using a 1×1 convolution. We use 1×1 convolutions because they have fewer parameters and computations, which can improve training time and reduce memory usage. Finally, we perform pixel-wise addition to fuse the images outputted by the two branches, resulting in the final clean image. We will verify the validity of each part of our generator in the ablation study in Sect. 3.5.2.

2.2 Discriminator network architecture

Our discriminator is improved based on PatchGAN [42]. Figure 3 shows the network architecture of the two discriminators. It was confirmed in the literature [42] that using PatchGAN allowed for capturing more high-frequency information and improve the image processing capability of the generator to a certain extent. The original PatchGAN can evaluate image features at multiple scales through downsampling, but important information will be lost in the downsampling process. So, we add a 4×4 convolution and an 8×8 convolution on the basis of PatchGAN to extract features directly from the image and resulting in feature maps that are four times and eighth times smaller than the input image

size, respectively. These extracted features are then aggregate the extracted information at the corresponding positions in the discriminator downsampling stage. This operation can enhance the receptive field of the discriminator, so that the discriminator can capture more global information and local information, so as to make up for the loss of information in the downsampling stage, and improve the discriminator's ability to judge each area of the image. Our discriminator enhances the generator's generation capability through the adversarial relationship in GAN. Specifically, the new discriminator allows the generator to achieve superior denoising effects while also improving the restoration of the structure and details in the denoised image. We will validate the effectiveness of our discriminator in the ablation study Sect. 3.5.3.

2.3 Loss function

Our total objective function is composed of four loss functions: adversarial loss, content loss, total variation loss and dual denoising loss. In this section, we will provide a detailed explanation for each loss function. We will show the effects of each function in Sect. 3.5.4.

Adversarial loss: In order to make the model training stable, we use the objective function in WGAN-GP [27] as the loss function of adversarial training in our model. The mathematical formula is shown in Eq. 2.

$$\min_G \max_D L(G, D) = E_{G(x) \sim P_g} [D(G(x))] - E_{x \sim P_r} [D(x)] + \lambda_{gp} \times L_{gp} \quad (2)$$

where $E_{G(x) \sim P_g} [D(G(x))]$ represents the mathematical expectation of the scores given by the discriminator D to the generated images $G(x)$ when the generated images follow the generated distribution P_g . $E_{x \sim P_r} [D(x)]$ represents the mathematical expectation of the scores given by the discriminator D to the real images x when the real images follow the real data distribution P_r . L_{gp} represents the gradient penalty, λ_{gp} is the weight of the gradient penalty. Compared to the adversarial loss in Eq. 1, the main difference in the adversarial loss in Eq. 2 utilizes the wasserstein distance to measure the difference between two distributions and adds a regularization term as a gradient penalty, in order to ensure that the gradient changes smoothly between the real data distribution and the generated data distribution. The specific mathematical formula for the gradient penalty L_{gp} is shown in Eq. 3.

$$L_{gp} = E_{\hat{x} \sim P_{\hat{x}}} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \quad (3)$$

In Eq. 3, \hat{x} represents a linearly interpolated sample between the real data distribution P_r and the generated data distribution P_g .

According to the game idea, Eq. 2 can be divided into the adversarial loss of the generator and the adversarial loss of the discriminator. Specifically, the generator's adversarial loss is shown in Eq. 4 and the discriminator's adversarial loss is shown in Eq. 5.

$$L_{adv} = -E_{G(x) \sim P_g} [D(G(x))] \quad (4)$$

Eq. 4 demonstrates the adversarial loss of the generator. The generator's goal is to produce samples that are as close as possible to the real data distribution.

$$L_D = E_{G(x) \sim P_g} [D(G(x))] - E_{x \sim P_r} [D(x)] + \lambda_{gp} \times L_{gp} \quad (5)$$

Eq. 5 demonstrates the adversarial loss of the discriminator. The discriminator's goal is to correctly distinguish between real samples and generated samples. It desires a high score $D(x)$ for real samples x (drawn from the distribution P_r) and a low score $D(G(x))$ for the generated samples $G(x)$ from the generator G . One of the key differences between this discriminator and a traditional GAN discriminator is the introduction of a gradient penalty term in the equation, which helps stabilize the training process and prevents mode collapse.

Content loss: Our model cannot generate denoised images with rich content by relying only on adversarial loss. So, to constrain the MIFGAN denoised image can reach the same standard as the ground-truth image, we construct the content loss function. As Eq. 6.

$$L_{\text{content}} = \lambda_{\text{pixel}} \times L_{\text{pixel}} + \lambda_{\text{edge}} \times L_{\text{edge}} + \lambda_{\text{vgg}} \times L_{\text{vgg}} \quad (6)$$

In Eq. 8, L_{pixel} represents pixel loss, L_{edge} represents edge loss and L_{vgg} represents perceptual loss. In addition, the coefficients λ_{pixel} , λ_{edge} and λ_{vgg} represent the weights of these three loss functions in the whole content loss respectively.

The use of L1 loss in image denoising will lack the protection of relevant details, resulting in image detail texture loss and edge sharpening after denoising. The use of L2 loss is only the calculation of the sum of squares of all pixels of the image, which will cause the image to become more blurred and smooth. So L1 loss and L2 loss are not conducive to image reconstruction in our denoising task. Therefore, we use the charbonnier loss proposed in the literature [51] as our pixel loss. This loss function addresses certain issues inherent in L1 and L2 losses, resulting in an improved image reconstruction performance. The specific form of this loss is

given in Eq. 7:

$$L_{pixel} = \frac{1}{N} \sum_{i=1}^N \sqrt{(y_i - G(x)_i)^2 + \ell^2} \tag{7}$$

Where N denotes the total number of image pixels and i denotes the image pixel index. y represent the ground-truth image, and $G(x)$ represent the denoised image produced by the generator. In the equation, ℓ is a small constant that controls the smoothness of the loss function. When the pixel difference is small, the behavior of the loss function resembles L2 loss; whereas, when the pixel difference is large, it behaves more like L1 loss. This way, the charbonnier loss is able to better preserve image edges and details during the denoising process while avoiding overly blurred results.

More high-frequency texture information in the image content reconstruction process is what we need to focus on preserving. So, we introduce edge loss [52, 53] to improve the detail representation of our denoised images. The mathematical formula is expressed as follows:

$$L_{edge} = \sqrt{(\Delta(y) - \Delta(G(x)))^2 + \ell^2} \tag{8}$$

In Eq. 8, the Laplacian operator [54] represented by Δ is first used to calculate the gradient information of the ground-truth image y and the denoised image $G(x)$ to extract the edge information. The difference between the tow images is then calculated using the square root of the square including the penalty coefficient.

In the denoising task, the result of more favorable quantitative evaluation criteria is the goal we pursue. However, sometimes the denoised image can not meet our visual good feeling when it has a high peak signal-to-noise ratio. Therefore, in order to make the image more refined after noise removal, the content is more clear, and meets the human aesthetic in a more pleasing manner. We introduce perceptual loss [43] to enhance visual perception.

$$L_{vgg} = \frac{1}{C_i W_i H_i} \|V_{gg_i}(y) - V_{gg_i}(G(x))\|_2^2 \tag{9}$$

Eq. 9 is the mathematical symbolic form of perceptual loss. We use the pretrained model of VGG19 [55] to extract the features of the denoised image $G(x)$ and the ground-truth image y . These extracted feature differences are then used to compute the perceptual loss. In the above equation V_{gg_i} is represented as the i th feature layer in the VGG19 model. C_i , W_i and H_i are the number of channels, width and height of the i th layer, respectively.

Total variation loss: Total variation loss [44] was successful in previous denoising work. This function can effectively remove noise and also promote the preservation of texture

details. The loss function is defined as follows:

$$L_{tv} = \|\nabla_h(G(x))\|_2^2 + \|\nabla_v(G(x))\|_2^2 \tag{10}$$

where $\nabla_h(\nabla_v)$ is the gradient operator along the horizontal (vertical) direction. The loss function makes full use of the context information of the denoised image $G(x)$, measures the change of pixel value in the vertical and horizontal directions in the form of gradient, and smooths the noise of the detected abnormal noise points. We introduce this loss to promote the spatial smoothness of the output image and avoid over-pixelation.

Dual denoising loss: To enhance the denoising capability of the generator, we conduct a thorough study of the loss function and design a novel loss function called the dual denoising loss, which aims to further constrain and guide the training process of the generator. The form of the function is as follows:

$$L_{dual} = \frac{1}{N} \sum_{i=1}^N S(y_i - G(G(x))_i) \tag{11}$$

where N denotes the total number of image pixels and i denotes the image pixel index. $S(\cdot)$ represents the Smooth L1 Loss, whose specific calculation form is shown in Eq. 12.

$$S(m) = \begin{cases} 0.5m^2 & \text{if } |m| < 1 \\ |m| - 0.5 & \text{otherwise} \end{cases} \tag{12}$$

The design idea of the dual denoising loss function is as follows. Ideally, our denoising model only processes noisy images. When a clean image passes through the denoising network, the output image should remain consistent with the original input. If the denoised image still contains noise, we re-input the denoised image into our network, and the network will process the remaining noise again. By passing the noisy image through the network twice, we can further reduce any residual noise left after the first pass. Therefore, based on the above theoretical analysis, we propose the function in Eq. 11, which constrains the denoised image to be closer to the standard clean image. In Eq. 11 we use the Smoothing L1 loss [56] to measure the difference between the image $G(G(x))$ and the ground-truth image y . Smooth L1 loss combines the advantages of L1 loss and L2 loss, enabling it to address issues like gradient vanishing and gradient explosion that arise in some special cases. Specifically, when the difference between the predicted value and the true value is small, the Smooth L1 Loss uses the L2 loss (squared error), while for larger differences, it employs the L1 loss (absolute error). The above theoretical analysis demonstrates that employing this design in our dual denoising function helps stabilize gradients during training and makes the model more robust to

outliers. To further prove that our proposed dual denoising loss can constrain the network training to be more stable and enhance denoising performance, we conduct ablation experiments on this function in Sect. 3.5.4 to verify its effectiveness.

In summary, the total loss function constraining our generator is given by Eq. 13.

$$L_G = \lambda_{adv} \times L_{adv} + \lambda_{conten} \times L_{content} + \lambda_{tv} \times L_{tv} + \lambda_{dual} \times L_{dual} \quad (13)$$

where λ_{adv} , λ_{conten} , λ_{tv} and λ_{dual} represent the weights of adversarial loss, content loss, total variation loss and dual denoising loss respectively. The weight setting for the loss function will be discussed in Sect. 3.2. Our discriminator is only for evaluation. Therefore, the discriminator only needs to be constrained by the adversarial loss, and the overall loss function of the discriminator is the function in Eq. 5.

3 Experiments

3.1 Datasets

Three datasets are used for the experiment, namely the SIDD dataset [57], the DND dataset [58] and the PolyU dataset [59]. We use these three datasets for the following reasons: First, these three datasets are publicly available and have official evaluation criteria. Second, these datasets are the collection of complex noise in the world scene, which conforms to the distribution standard of real noise. Third, the noise types in these datasets are complex, so the performance of denoising methods can be accurately gauged.

SIDD: Smartphone image denoising dataset (SIDD) provides 320 pairs of high-resolution color images for training data. To ensure stable and efficient training of our model, we crop each pair of training sets into 256×256 patches for training our model. The dataset also provides 40 pairs of images for validation, and each pair of validation images is evaluated on 32 image blocks of size 256×256 . Therefore, there are a total of 1280 image pairs of size 256×256 in the 40 validation datasets for validation.

DND: Darmstadt noise dataset (DND) provides 50 noisy images for testing. Each test image does not provide a clean image. Test results can only be obtained through the official online test website. The official test benchmark is to divide each piece of test data into 20 512×512 image boxes for evaluation. So, the DND ended up evaluating 1000 images of 512×512 size.

PolyU: The PolyU dataset is a collection of 40 indoor scenes. The dataset provides 100 noise images in the size of 512×512 for the denoising test, and provides a ground standard image for each test data.

Algorithm 1 Training Procedure of MIFGAN

Input: Noisy image sets $N = \{n_1, n_2, \dots, n_m\}$ and corresponding clean image sets $C = \{c_1, c_2, \dots, c_m\}$

Initialization: Generator parameters θ_G , discriminator parameters θ_D , optimizer parameters, learning rate, batch size B , number of iterations T

Loss Functions: Define generator loss L_G and discriminator loss L_D

- 1: **for** $t = 1$ to T **do**
- 2: Randomly select a batch of noisy images $N_B = \{n_1, n_2, \dots, n_B\}$ and corresponding clean images $C_B = \{c_1, c_2, \dots, c_B\}$
- 3: Generate denoised images $G(N_B) = \{g_1, g_2, \dots, g_B\}$ using generator G with parameters θ_G
- 4: Calculate the discriminator loss L_D
- 5: Update the discriminator parameters θ_D using L_D and Adam optimizer
- 6: Calculate the generator loss L_G
- 7: L_G captures the difference between $G(N_B)$ and C_B , driving the generator to recover lost image information
- 8: Update the generator parameters θ_G using L_G and Adam optimizer
- 9: **end for**

Output: The trained denoising generator network G

3.2 Experimental settings

The training and testing of the experiment are based on the pytorch1.7.0 deep learning framework on the Nvidia GeForce GTX 3090 GPU. Both generator and discriminator adopt the Adam optimizer. We set the momentum parameters to $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is uniformly set to $2e-4$ during the whole training process. The batch size is 16 and a total of 200 epochs are trained.

The parameters in the loss function are set as follows: First, the value of penalty coefficient λ_{gp} in Eqs. 2 and 5 is set to 10 according to the research results in WGAN-GP [27]. Then, the value of parameter ℓ^2 in Eqs. 7 and 8 is set as 0.001 following the default setting in literature [51, 52]. Finally, we learn from the experience of weight setting for pixel loss, edge loss, and perceptual loss in the literature [54–56], and our work focuses on reconstruction of pixels. Before fully training the model, we first take a small subset of the dataset and conduct multiple training sessions. Before each training session, we adjust the weights of our loss function. Through multiple experiments, the results show that when $\lambda_{pixel} = 10$, $\lambda_{edge} = 0.5$ and $\lambda_{vgg} = 1$, the model has superior performance. When we set the parameters to $\lambda_{adv} = 0.01$, $\lambda_{conten} = 1$, $\lambda_{tv} = 0.01$ and $\lambda_{dual} = 0.1$, the model can be trained stably and effectively. The specific training procedure of our proposed model is shown in Algorithm 1.

3.3 Evaluation metrics

The evaluation of image denoising mainly includes quantitative evaluation and qualitative evaluation. The qualitative evaluation is mainly based on human visual intuition to

Table 1 Quantitative comparison results of different denoising methods on the SIDD dataset

Method	BM3D	MSIDNet	APD-Nets	CBDNet	FFDNet	MDRN	RIDNet
PSNR	25.65	39.45	39.56	30.78	33.42	39.01	38.71
SSIM	0.685	0.911	0.955	0.754	0.874	0.953	0.914
Method	CFNet	MSGAN	DANet	DCANet	TSIDNet	MIRNet	MIFGAN
PSNR	39.34	39.11	39.43	39.27	39.43	39.72	40.27
SSIM	0.955	0.945	0.956	0.956	0.910	0.959	0.960

The best results are highlighted in bold

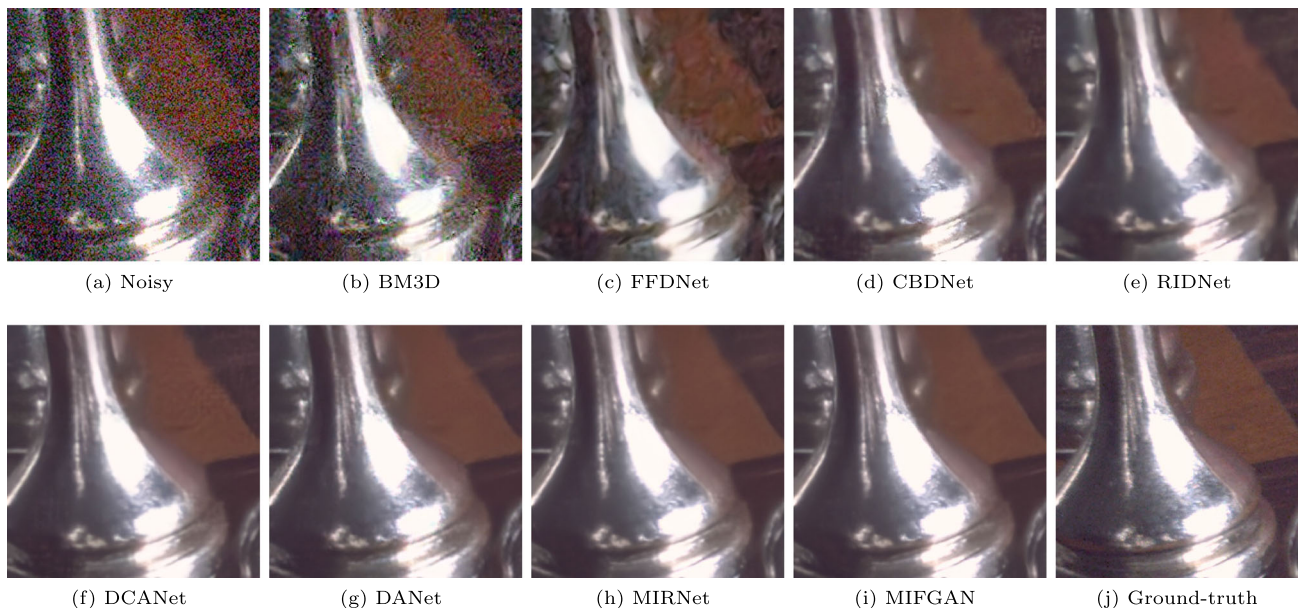


Fig. 4 Visual comparison results of different denoising methods on SIDD dataset

judge whether the image quality is in line with people’s aesthetic. Quantitative evaluation uses Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) [60] is the most popular and recognized index at present.

PSNR is usually used to measure the noise removal effect of denoising algorithm. The specific mathematical formula is as follows:

$$PSNR = 20 \times \log_{10} \left(\frac{MAX_{pixel}}{\sqrt{MSE(G(x), y)}} \right) \tag{14}$$

where MAX_{pixel} represents the maximum pixel value of our image, usually 255. $MSE(G(x), y)$ uses the mean square error to calculate the difference between a denoised image $G(x)$ and a clean image y , which should be as small as possible. Therefore, the larger the PNSR value, the higher the image quality after denoising.

This evaluation index SSIM is more consistent with the human visual perception of images. SSIM compares the difference between denoised image and clean image mainly from three aspects: structure, brightness and contrast. The

formula for calculating SSIM is as follows:

$$SSIM = \frac{(2\mu_{G(x)}\mu_y + M_1)(2\sigma_{G(x)y} + M_2)}{(\mu_{G(x)}^2 + \mu_y^2 + M_1)(\sigma_{G(x)}^2 + \sigma_y^2 + M_2)} \tag{15}$$

where $\mu_{G(x)}$ and μ_y represent the mean values of the denoised image $G(x)$ and the clean image y , respectively. $\sigma_{G(x)}^2$ and σ_y^2 represent the variances of the pixel values of the denoised image $G(x)$ and the clean image y , respectively. $\sigma_{G(x)y}$ represents the covariance between the denoised image $G(x)$ and the clean image y . M_1 and M_2 are constants to prevent the denominator from being 0 in the calculation. The SSIM is calculated between 0 and 1. The closer the evaluation result is to 1, the higher the similarity between the denoised image and the clean image, and the more favorable the image quality.

3.4 Comparison with other methods

In order to evaluate the effectiveness and competitiveness of our denoising algorithm. Our algorithm is evaluated quanti-

tatively and qualitatively with some state-of-the-art methods on SIDD, DND and PolyU datasets.

Evaluation results on the SIDD dataset: As shown in the results in Table 1, MIFGAN has superior quantitative evaluation metrics on this dataset compared to other methods. Specifically, compared with CFNet and RIDNet, our PSNR and SSIM increased by 0.93, 0.005 and 1.56, 0.046 respectively. The quantitative evaluation results of MIFGAN are significantly superior than BM3D, CBDNet and FFDNet. Figure 4 shows the results of the visual comparison. BM3D and FFDNet are less effective in removing real noise. The denoised images of RIDNet and DCANet produced artifacts. DANet and MIRNet are more blurred than MIFGAN in floor texture details. In contrast, our method is more effective for noise removal and superior for detail recovery.

Evaluation results on the DND dataset: Table 2 presents the results of quantitative evaluation. Compared with DCANet, FFDNet and MDRN methods, the PSNR and SSIM of MIFGAN are increased by 0.25 and 0.001, 5.42 and 0.107, 0.39 and 0.002, respectively. Although our method is 0.004 lower than ADGAN and 0.003 lower than MSGAN in SSIM, we are far higher than them in PSNR. Figure 5 shows the visual results of the different methods. BM3D, CDNCNN-B, and FFDNet still have a large amount of noise residual. FFDNet and CBDNet generate blurred structures due to the excessive smoothing operation during denoising, which makes the texture details disappear. In the gray texture in the black area, the fine texture structure retained by MIFGAN is clearer than that of DANet and DCANet. Experimental results show that MIFGAN algorithm is more competitive.

Evaluation results on the PolyU dataset: The results of quantitative evaluation are shown in Table 3. According to the data results, the MIFGAN demonstrates the best performance, the PSNR and SSIM of the MIFGAN compared to the TSIDNet are improved by 1.3 and 0.027, respectively. Additionally, when compared to the DCANet, which exhibits a higher performance, the MIFGAN still manages to improve the PSNR and SSIM by 0.9 and 0.009, respectively. Fig. 6 shows the comparison of visualization. For easy observation, we enlarge the text part in the upper right corner of the image without destroying the image. From the figures, it can be observed that the texts in DANet and MIRNet appear blurred. CBDNet still contains residual noise. The texts in DCANet and MPRNet are not as clear as those in MIFGAN, and the texture structure of the red brick in MIFGAN is more detailed and clear. The experimental results demonstrate that our algorithm possesses superior generalization ability.

3.5 Ablation study

To demonstrate the effectiveness of our algorithm, we conduct ablation studies on the SIDD dataset.

3.5.1 Ablation study of the residual modules

In the generator architecture, we use multiple residual modules on the original resolution of the image for denoising. In order to determine the number of residual modules when the effect is best. We performed ablation experiments for the number of residual modules. The detailed results are shown in Table 4. According to the data in the table, it can be concluded that when the residual module is added on the basis of the three residual modules, the values of PSNR and SSIM do not increase, and even the index data becomes worse. Although the PSNR is increased by 0.01 when using 7 residual modules compared to 3 residual modules, the number of parameters of the model is more than twice as large, which results in more time for our model to train and test. So, in the end, we set the number of residual modules to 3, which allows us to achieve good results in terms of PSNR and SSIM with only 0.22 million parameters.

3.5.2 Ablation study for generator

In order to verify the effectiveness of the generator in MIFGAN, we perform ablation experiments for four structures in the generator that deal with different scales. The results of the ablation study are shown in Table 5. According to the data results, the denoising effect is poor when we only use the U-net architecture. After sequentially adding 4 times and 8 times downsampling architecture, both PSNR and SSIM are gradually improved. When we add our residual network structure, both objective metrics have the best performance. Therefore, the results of the ablation study of the generator architecture show that each part of the denoising network in MIFGAN has an irreplaceable role.

3.5.3 Ablation study for discriminator

To verify the effectiveness of the discriminators in MIFGAN, and we perform ablation study on the discriminator. Our specific operation is to replace the discriminators in MIFGAN with those in DANet, PatchGAN and BDGAN respectively to train our network. The specific results of the experiments are shown in Table 6.

We can conclude from the data in Table 6 that MIFGAN improves PSNR and SSIM by 2.68 and 0.009, 1.25 and 0.005, 1.04 and 0.004, respectively, compared with the use of a simple fully-connected layer discriminator in DANet, the use of an ordinary PatchGAN discriminator in the literature [34], and the use of multiple different scale discriminators in BDGAN. Therefore, the experimental results show that our proposed discriminator can effectively assist the model for denoising.

Table 2 Quantitative comparison results of different denoising methods on the DND dataset

Method	APD-Nets	BM3D	CDNCNN-B	FFDNet	CBDNet	TSIDNet	MSIDNet
PSNR	39.69	34.51	37.90	34.40	38.06	39.74	39.72
SSIM	0.952	0.851	0.943	0.847	0.942	0.954	0.954
Method	ADGAN	DANet	MDRN	MSGAN	CFNet	DCANet	MIFGAN
PSNR	38.13	39.55	39.43	39.59	39.65	39.57	39.82
SSIM	0.958	0.953	0.952	0.957	0.952	0.953	0.954

The best results are highlighted in bold

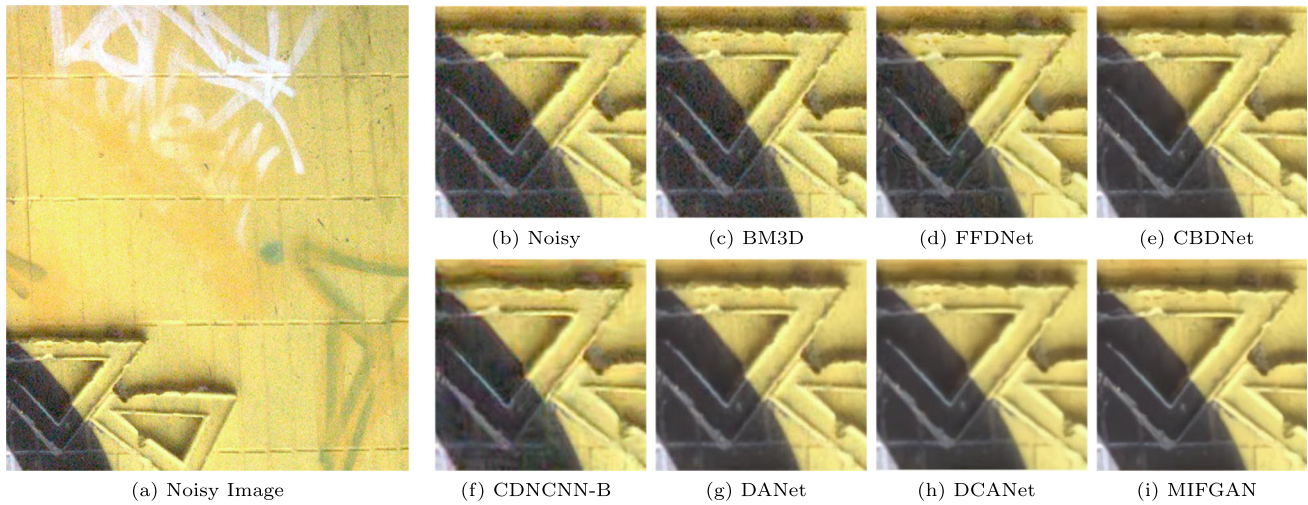


Fig. 5 Visual comparison results of different denoising methods on the DND dataset

Table 3 Quantitative comparison results of different denoising methods on the PolyU dataset

Method	DGCL	MSIDNet	TSIDNet	DANet	DCANet
PSNR	38.11	37.23	37.44	37.17	37.84
SSIM	0.961	0.947	0.955	0.923	0.973
Method	RIDNet	MIRNet	MPRNet	CBDNet	MIFGAN
PSNR	38.52	37.35	37.50	37.86	38.74
SSIM	0.980	0.956	0.971	0.971	0.982

The best results are highlighted in bold

Table 4 The effect of the number of residual modules on PSNR, SSIM and Params(M)

Number	1	2	3	4	5	6	7
PSNR	38.82	39.52	40.27	40.25	40.11	40.27	40.28
SSIM	0.949	0.953	0.960	0.960	0.956	0.958	0.960
Params(M)	0.076	0.149	0.224	0.297	0.371	0.445	0.519

“M” is short for million

Table 5 Ablation study of network architecture in generator for denoising effect, where down indicates downsampling, ✓ indicates use and ✗ indicates non-use

Generator	U-net	4× down	8× down	Residual network	PSNR	PSNR
MIFGAN	✓	✗	✗	✗	34.72	0.930
MIFGAN	✓	✓	✗	✗	36.69	0.937
MIFGAN	✓	✓	✓	✗	36.94	0.939
MIFGAN	✓	✓	✓	✓	40.27	0.960

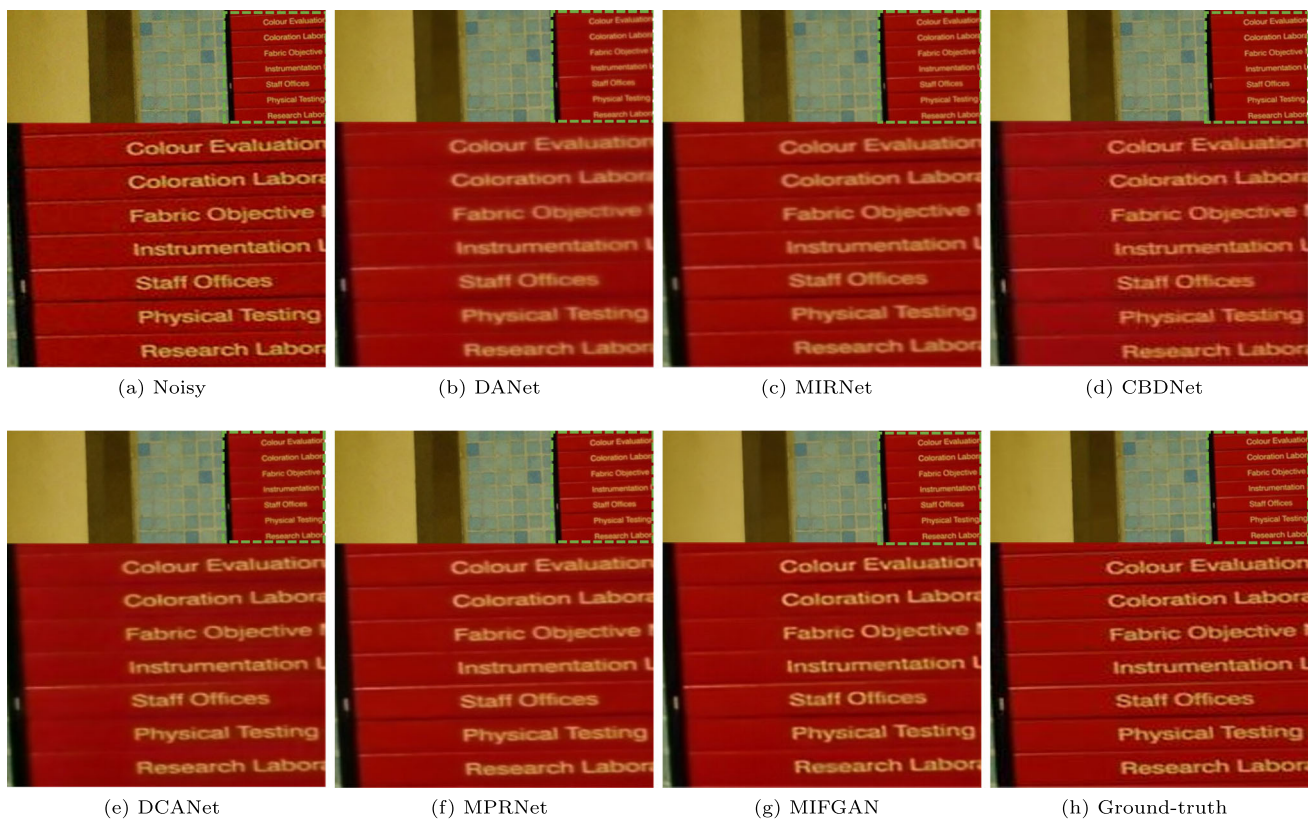


Fig. 6 Visual comparison results of different denoising methods on the PolyU dataset

Table 6 Ablation study of the effect of discriminator on model performance, where DANet, PatchGAN, BDGAN and MIFGAN refer to the discriminator in the corresponding methods

Discriminator	DANet	PatchGAN	BDGAN	MIFGAN
PSNR	37.59	39.02	39.23	40.27
SSIM	0.951	0.955	0.956	0.960

3.5.4 Ablation study for loss functions

There are a total of 6 loss functions used in MIFGAN to train the generator. In order to prove the effectiveness of each loss function, we conduct ablation studies on these 6 loss functions. The results of the quantitative evaluation are shown in Table 7. Both PSNR and SSIM values are low when we use only adversarial loss. After we successively introduce pixel loss, total variation loss, and edge loss, the quantitative evaluation results steadily improve. After introducing our proposed double denoising loss, PSNR and SSIM reach 39.80 and 0.956, respectively. Finally, after we add the perceptual loss, our SSIM and PSNR are further improved, breaking through to 40.27 and 0.960 respectively. Figure 7 shows the visual effects after denoising using different loss functions. It is straightforward to see that our network relying

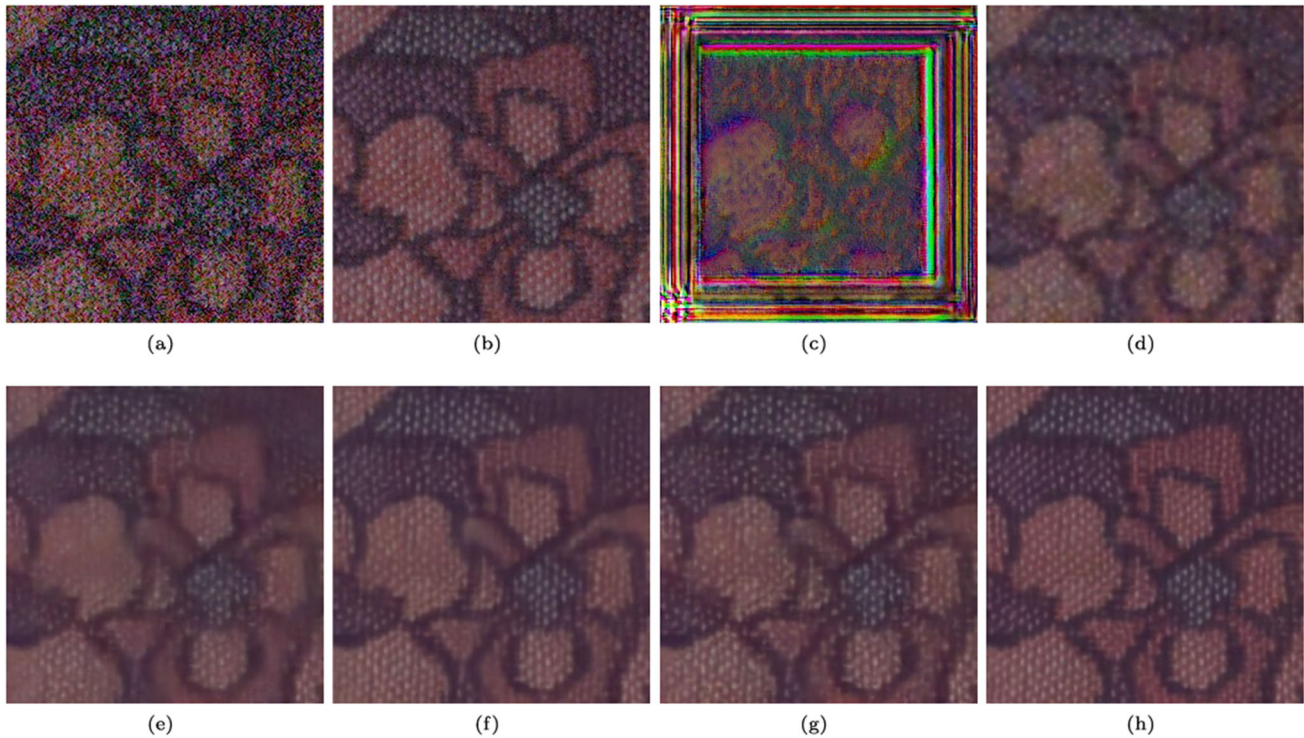
only on adversarial loss leads to unstable denoising training and makes it difficult to generate image structures. With the introduction of pixel loss, the image can generate a rough structure, but there is still noise and atomization phenomenon. After adding the total variation loss, the image achieves enhanced denoising quality. After adding the edge loss, the image texture details are clearer. After the introduction of double denoising loss, it can strengthen the recovery of some important detailed textures in the image. Finally, the addition of perceptual loss makes the image more realistic and bright. So, the experimental results show that each function in MIFGAN has an indispensable role.

4 Conclusion

In this paper, we propose a multi-scale information fusion generative adversarial network (MIFGAN) for real-world noisy image denoising. Our encoder–decoder network employs a multi-scale information fusion strategy to enhance the model’s capabilities for image denoising and restoration, while our customized residual network further mitigates noise and preserves image information. Additionally, By incorporating convolutional kernels of various sizes into the discriminator, we can increase its receptive field, enabling

Table 7 Ablation study of the effect of loss function on denoising performance

Loss	L_{adv}	L_{pixel}	L_{tv}	L_{edge}	L_{dual}	L_{vgg}	PSNR	SSIM
MIFGAN	✓	✗	✗	✗	✗	✗	12.77	0.410
MIFGAN	✓	✓	✗	✗	✗	✗	32.87	0.915
MIFGAN	✓	✓	✓	✗	✗	✗	35.08	0.922
MIFGAN	✓	✓	✓	✓	✗	✗	37.65	0.941
MIFGAN	✓	✓	✓	✓	✓	✗	39.80	0.956
MIFGAN	✓	✓	✓	✓	✓	✓	40.27	0.960

**Fig. 7** The visual effect of the proposed network with different loss functions. **a** noisy. **b** clean. **c** L_{adv} loss. **d** $L_{adv} + L_{pixel}$ loss. **e** $L_{adv} + L_{pixel} + L_{tv}$ loss. **f** $L_{adv} + L_{pixel} + L_{tv} + L_{edge}$ loss. **g** $L_{adv} + L_{pixel} + L_{tv} + L_{edge} + L_{dual}$ loss. **h** $L_{adv} + L_{pixel} + L_{tv} + L_{edge} + L_{dual} + L_{vgg}$ loss

it to gather more comprehensive contextual information and more effectively assist the generator in completing the denoising task. Our proposed dual denoising loss combined with other loss functions together constitutes a set of multi-scale objective functions, which improves the image denoising and restoration performance of the model. The experimental results show that our method can effectively remove complex noise in real images and the denoised images are more clear and realistic. Compared with other methods, our method exhibits superior practicability when applied to real noisy images.

Our algorithm is not only meaningful for the image denoising task, but can also be widely applied to other image enhancement tasks such as image super-resolution and image inpainting by modifying the relevant parameters in the model. However, the number of parameters in our model is relatively large, which can lead to limitations on some micro-embedded

devices. Conducting further research into developing more lightweight denoising models is another research direction.

Acknowledgements The authors would like to thank the reviewers for all of their careful, constructive and insightful comments in relation to this work.

Funding No Financial support.

Data availability The data and materials are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

References

1. Zeng, N., Wu, P., Wang, Z., Li, H., Liu, W., Liu, X.: A small-sized object detection oriented multi-scale feature fusion approach with application to defect detection. *IEEE Trans. Instrum. Meas.* **71**, 1–14 (2022)
2. Ning, X., Tian, W., Yu, Z., Li, W., Bai, X., Wang, Y.: Hcfnn: high-order coverage function neural network for image classification. *Pattern Recognit.* **131**, 108873 (2022)
3. Cheng, Z., Qu, A., He, X.: Contour-aware semantic segmentation network with spatial attention mechanism for medical image. *The Vis. Comput.* **38**, 749–762 (2022)
4. Buades, A., Coll, B., Morel, J.-M.: A non-local algorithm for image denoising. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, pp. 60–65 (2005)
5. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007)
6. Aharon, M., Elad, M., Bruckstein, A.: K-svd: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.* **54**(11), 4311–4322 (2006)
7. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2862–2869 (2014)
8. Pan, Y., Ren, C., Wu, X., Huang, J., He, X.: Real image denoising via guided residual estimation and noise correction. *IEEE Trans. Circuits Syst. Video Technol.* **33**(4), 1994–2000 (2022)
9. Jain, V., Seung, S.: Natural image denoising with convolutional networks. *Adv. Neural Inf. Process. Syst.* **21** (2008)
10. 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.* **26**(7), 3142–3155 (2017)
11. Zhang, K., Zuo, W., Zhang, L.: Ffdnet: toward a fast and flexible solution for cnn-based image denoising. *IEEE Trans. Image Process.* **27**(9), 4608–4622 (2018)
12. Guo, S., Yan, Z., Zhang, K., Zuo, W., Zhang, L.: Toward convolutional blind denoising of real photographs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1712–1722 (2019)
13. Anwar, S., Barnes, N.: Real image denoising with feature attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3155–3164 (2019)
14. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.-H., Shao, L.: Learning enriched features for real image restoration and enhancement. In: Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16, pp. 492–511 (2020)
15. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.-H., Shao, L.: Multi-stage progressive image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14821–14831 (2021)
16. Jiang, B., Lu, Y., Wang, J., Lu, G., Zhang, D.: Deep image denoising with adaptive priors. *IEEE Trans. Circuits Syst. Video Technol.* **32**(8), 5124–5136 (2022)
17. Zhou, L., Zhou, D., Yang, H., Yang, S.: Multi-scale network toward real-world image denoising. *Int. J. Mach. Learn. Cybern.* **14**(4), 1205–1216 (2023)
18. Jia, X., Peng, Y., Ge, B., Li, J., Liu, S., Wang, W.: A multi-scale dilated residual convolution network for image denoising. *Neural Process. Lett.* **55**(2), 1231–1246 (2023)
19. Zhou, L., Zhou, D., Yang, H., Yang, S.: Two-subnet network for real-world image denoising. *Multimed. Tools Appl.*, 1–17 (2023)
20. Zuo, Y., Yao, W., Zeng, Y., Xie, J., Fang, Y., Huang, Y., Jiang, W.: Cfnnet: conditional filter learning with dynamic noise estimation for real image denoising. *Knowl.-Based Syst.* **284**, 111320 (2024)
21. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **27** (2014)
22. Zhao, J., Lee, F., Hu, C., Yu, H., Chen, Q.: Lda-gan: lightweight domain-attention gan for unpaired image-to-image translation. *Neurocomputing* **506**, 355–368 (2022)
23. Chen, Y., Xia, R., Yang, K., Zou, K.: Gcam: lightweight image inpainting via group convolution and attention mechanism. *Int. J. Mach. Learn. Cybern.* 1–11 (2023)
24. Chen, Y., Xia, R., Yang, K., Zou, K.: Dargs: image inpainting algorithm via deep attention residuals group and semantics. *J. King Saud Univ.-Comput. Inf. Sci.* **35**(6), 101567 (2023)
25. Chen, Y., Xia, R., Yang, K., Zou, K.: Mfmam: image inpainting via multi-scale feature module with attention module. *Comput. Vis. Image Underst.* **238**, 103883 (2024)
26. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning, pp. 214–223 (2017)
27. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. *Adv. Neural Inf. Process. Syst.* **30** (2017)
28. Chen, J., Chen, J., Chao, H., Yang, M.: Image blind denoising with generative adversarial network based noise modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3155–3164 (2018)
29. Lin, K., Li, T.H., Liu, S., Li, G.: Real photographs denoising with noise domain adaptation and attentive generative adversarial network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 1717–1721 (2019)
30. Zhu, S., Xu, G., Cheng, Y., Han, X., Wang, Z.: Bdgan: Image blind denoising using generative adversarial networks. In: Pattern Recognition and Computer Vision: Second Chinese Conference, PRCV 2019, Xi'an, China, November 8–11, 2019, Proceedings, Part II 2, pp. 241–252 (2019)
31. Kim, D.-W., Ryun Chung, J., Jung, S.-W.: Grdn: Grouped residual dense network for real image denoising and gan-based real-world noise modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 2086–2094 (2019)
32. Yue, Z., Zhao, Q., Zhang, L., Meng, D.: Dual adversarial network: Toward real-world noise removal and noise generation. In: Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16, pp. 41–58 (2020)
33. Lyu, Q., Guo, M., Pei, Z.: Degan: mixed noise removal via generative adversarial networks. *Appl. Soft Comput.* **95**, 106478 (2020)
34. Vo, D.M., Nguyen, D.M., Le, T.P., Lee, S.-W.: Hi-gan: a hierarchical generative adversarial network for blind denoising of real photographs. *Inf. Sci.* **570**, 225–240 (2021)
35. Zhao, S., Lin, S., Cheng, X., Zhou, K., Zhang, M., Wang, H.: Dual-gan complementary learning for real-world image denoising. *IEEE Sens. J.* **24**(1), 355–366 (2024)
36. Song, Y., Zhu, Y., Du, X.: Grouped multi-scale network for real-world image denoising. *IEEE Signal Process. Lett.* **27**, 2124–2128 (2020)
37. Wang, Y., Wang, G., Chen, C., Pan, Z.: Multi-scale dilated convolution of convolutional neural network for image denoising. *Multimed. Tools Appl.* **78**, 19945–19960 (2019)
38. Yu, X., Fu, Z., Ge, C.: A multi-scale generative adversarial network for real-world image denoising. *Signal Image Video Process.* **16**, 257–264 (2022)
39. Wang, Z., Wang, L., Duan, S., Li, Y.: An image denoising method based on deep residual gan. In: Journal of Physics: Conference Series, vol. 1550, p. 032127 (2020)

40. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
41. Wu, W., Lv, G., Duan, Y., Liang, P., Zhang, Y., Xia, Y.: Dcanet: Dual convolutional neural network with attention for image blind denoising. arXiv preprint [arXiv:2304.01498](https://arxiv.org/abs/2304.01498) (2023)
42. Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
43. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14, pp. 694–711 (2016)
44. Rubin, L.: Nonlinear total variation based noise removal algorithms. *Phys. D: Nonlinear Phenom.* **60**, 259–265 (1992)
45. Li, R., Pan, J., Li, Z., Tang, J.: Single image dehazing via conditional generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8202–8211 (2018)
46. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18, pp. 234–241 (2015)
47. Chen, Y., Xia, R., Yang, K., Zou, K.: Micu: image super-resolution via multi-level information compensation and u-net. *Expert Syst. Appl.* **245**, 123111 (2024)
48. Chen, Y., Xia, R., Yang, K., Zou, K.: Dnnam: image inpainting algorithm via deep neural networks and attention mechanism. *Appl. Soft Comput.* **154**, 111392 (2024)
49. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. *Distill* **1**(10), 3 (2016)
50. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14, pp. 630–645 (2016)
51. Lai, W.-S., Huang, J.-B., Ahuja, N., Yang, M.-H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 624–632 (2017)
52. Seif, G., Androutsos, D.: Edge-based loss function for single image super-resolution. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1468–1472 (2018)
53. Jiang, K., Wang, Z., Yi, P., Chen, C., Huang, B., Luo, Y., Ma, J., Jiang, J.: Multi-scale progressive fusion network for single image deraining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8346–8355 (2020)
54. Kamgar-Parsi, B., Rosenfeld, A.: Optimally isotropic laplacian operator. *IEEE Trans. Image Process.* **8**(10), 1467–1472 (1999)
55. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
56. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
57. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1692–1700 (2018)
58. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1586–1595 (2017)
59. Xu, J., Li, H., Liang, Z., Zhang, D., Zhang, L.: Real-world noisy image denoising: A new benchmark. arXiv preprint [arXiv:1804.02603](https://arxiv.org/abs/1804.02603) (2018)
60. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Xuegang Hu was born in Chongqing, China in 1965. He received the B.S. degree in applied mathematics from China West Normal University, China, in 1988, the M.S. and the Ph.D. degrees in applied mathematics from Sichuan University, China, in 1995 and 2006, respectively. From 2002 to 2008, he was an associate professor with Chongqing University of Posts and Telecommunications, China. Since then, he has been a professor with the University. He is the author of 3 books, more than 80 articles and 3 inventions. His research interests include digital image processing and analysis and partial differential equations and their applications.



Wei Zhao was born in Sichuan, China in 1999. He received the B.S. degree in information and computing science from Sichuan University of Science & Engineering, in 2021. He is currently pursuing the M.S. degree in mathematics at Chongqing University of Posts and Telecommunications. His research interests include computer vision, deep learning, and image denoising.