Multilevel Edge Features Guided Network for Image Denoising

Faming Fang, Juncheng Li, Yiting Yuan, Tieyong Zeng, and Guixu Zhang

Abstract—**Image denoising is a challenging inverse problem due to complex scenes and information loss. Recently, various methods have been considered to solve this problem by building a well-designed convolutional neural network (CNN) or introducing some hand-designed image priors. Different from previous works, we investigate a new framework for image denoising, which integrates edge detection, edge guidance, and image denoising into an end-to-end CNN model. To achieve this goal, we propose a multilevel edge features guided network (MLEFGN). First, we build an edge reconstruction network (Edge-Net) to directly predict clear edges from the noisy image. Then, the Edge-Net is embedded as part of the model to provide edge priors, and a dual-path network is applied to extract the image and edge features, respectively. Finally, we introduce a multilevel edge features guidance mechanism for image denoising. To the best of our knowledge, the Edge-Net is the first CNN model specially designed to reconstruct image edges from the noisy image, which shows good accuracy and robustness on natural images. Extensive experiments clearly illustrate that our MLEFGN achieves favorable performance against other methods and plenty of ablation studies demonstrate the effectiveness of our proposed Edge-Net and MLEFGN. The code is available at https://github.com/MIVRC/MLEFGN-PyTorch.**

Index Terms—**Edge guidance, edge reconstruction network (Edge-Net), image denoising.**

I. Introduction

IMAGE denoising is an extremely hot topic in computer vision, which aims to reconstruct a clean image from the noisy one (see Fig. 1). It is a crucial step for many real applications since the quality of the denoised images will significantly influence the performance of downstream tasks, such as image classification [1], [2], image segmentation [3], [4], object detection [5], [6], and other high-level computer vision tasks. However, due to the complex scenes and information loss, it is still considered a challenging inverse problem. In order to elegantly solve the problem of image denoising, we aim to build a flexible approach with the following desirable properties: 1) it is able to perform image denoising in an end-to-end manner; 2) it is efficient and can achieve superior performance; and 3) it can integrate image priors to accelerate model convergence and improve the visual effects of the denoised images.

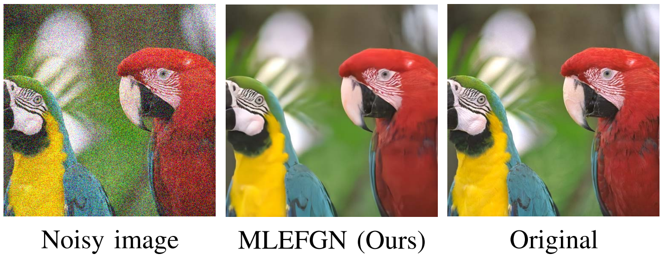


Fig. 1. Example of color image denoising (AWGN, noise level: $\sigma = 50$).

In recent years, various methods have been proposed for image denoising, which can be roughly divided into traditional methods [7]–[19] and learning-based methods [20]–[28]. Among them, the traditional methods can be further divided into three categories: 1) spatial filtering methods, such as bilateral filters [7], nonlocal means filters [8], and guided filters [9]; 2) model-based methods, such as total variation (TV) approaches [11], [12], nonlocal self-similarity (NSS) models [13], and sparse dictionary learning models [14]; and 3) wavelet transform-based methods [10]. At that time, the block matching 3-D filter (BM3D [15]) and the weighted nuclear norm minimization (WMMN [16]) achieved the best results. Nevertheless, these methods suffer from several drawbacks: 1) these methods are usually implemented in an iterative manner, which may lead to a huge amount of calculation and low efficiency and 2) these methods usually use hand-designed priors, such as sparsity [21] or NSS [14], which might be not suitable for arbitrary natural images.

The other category of image denoising method aims to solve the problem via learning the mapping function between the noisy and clean images, such as MLP [20], sparse coding [21], and deep learning-based methods [22]–[28]. Among them, deep learning-based methods achieve advanced

Manuscript received September 13, 2019; revised April 6, 2020 and July 13, 2020; accepted August 8, 2020. This work was supported in part by the Key Project of the National Natural Science Foundation of China under Grant 61731009, in part by the National Natural Science Foundation of China under Grant 61871185, and in part by the Chenguang Program supported by the Shanghai Education Development Foundation and Shanghai Municipal Education Commission under Grant 17CG25. The work of Tieyong Zeng was supported in part by the NSFC under Grant 11671002, in part by the CUHK Startup, and in part by the CUHK Direct Allocation Grant under Grant 4053342, Grant 4053405, Grant RGC 14300219, Grant RGC 14302920, and Grant NSFC/RGC N_CUHK 415/19. *(Corresponding authors: Tieyong Zeng; Guixu Zhang.)*

Faming Fang, Juncheng Li, Yiting Yuan, and Guixu Zhang are with the Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai 200062, China, and also with the School of Computer Science and Technology, East China Normal University, Shanghai 200062, China (e-mail: fmfang@cs.ecnu.edu.cn; cvjunchengli@gmail.com; gxzhang@cs.ecnu.edu.cn).

Tieyong Zeng is with the Department of Mathematics, The Chinese University of Hong Kong, Hong Kong (e-mail: zeng@math.cuhk.edu.hk).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.3016321

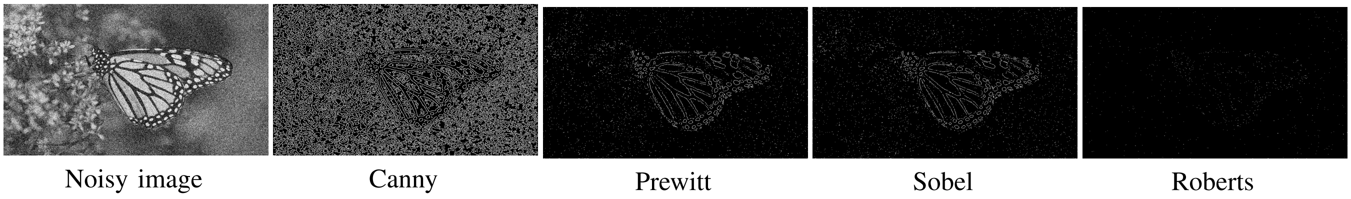


Fig. 2. Results of different edge detectors acting on a noisy image (noise level: $\sigma = 50$). It is obvious that these edge detectors are extremely sensitive to noise.

results with the blossom of deep convolutional neural networks (CNNs). The success of the CNN-based image denoising method is attributed to its powerful feature extraction capability and well-designed network structure. For instance, the DnCNN [26] was proposed for the Gaussian image denoising, which took advantage of batch normalization and residual learning to achieve competitive denoising results. The BM3D-Net [27] is a nonlocal-based network that introduced BM3D into CNN by wavelet shrinkage. The FFDNet [28] is a flexible denoising model that took the noise level map and the noisy image as the inputs for image denoising, which achieved the state-of-the-art results. Most of the aforementioned CNN-based methods aim to learn the mapping between clean and noisy images directly by minimizing the loss function. However, it is difficult to learn accurate mapping due to the information loss.

Reviewing previous methods, we find that plenty of works have pointed out that making full use of image priors can effectively improve the quality of the reconstructed images. Indeed, some image priors have been proposed for image restoration, such as TV prior, sparse prior, and edge prior. Among them, the edge prior is one of the most effective and accessible priors. As an important component of image features, it has been widely used in image reconstruction tasks [29]–[31]. Meanwhile, some studies introduced edges into the CNN model to guide image reconstruction or enhance the reconstructed image. For instance, Jiang *et al.* [32] proposed an edge-guided recurrent residual network (DEGRRN) for single-image super-resolution. Chupraphawan and Ratanamahatana [33] proposed a deep CNN with edge feature (DCEF) for image denoising. Both DEGRRN and DCEF first applied the off-the-shelf edge detector (e.g., Sobel or Canny) on the degraded image to obtain corresponding edge maps and then sent it to the network along with the input image to reconstruct the final image. However, this way seriously limits the model performance and may not be suitable for image denoising. As shown in Fig. 2, existing edge extractors are extremely sensitive to noise. This means that it is difficult to extract clear edges from the noisy image. Therefore, how to obtain accurate edges from the noisy image and how to make full use of them to reconstruct high-quality noise-free images are the focus of this work. To achieve this, we propose an efficient multilevel edge features guided network (MLEFGN) for image denoising. Especially, we first build an edge reconstruction network (Edge-Net) to directly predict clear edges from the noisy image. Then, the Edge-Net is embedded as a part of the model to provide edge priors, and a dual-path network is applied to extract the image and edge features. Finally, a multilevel edge features guidance mechanism is introduced to reconstruct the denoised image.

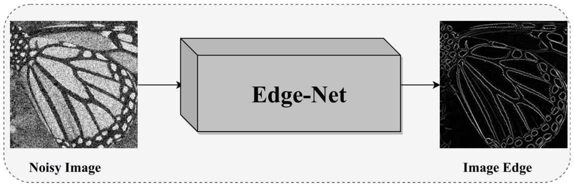


Fig. 3. Principle of edge detection. The Edge-Net is used to reconstruct clear image edges from the noisy image.

In summary, the main contributions of this article are as follows.

1) We propose a new edge guidance framework for image denoising, which integrates edge detection, edge guidance, and image denoising in an end-to-end model.
2) We propose an Edge-Net. Edge-Net is the first CNN model that can directly reconstruct clear edges from the noisy observation.
3) We propose an MLEFGN. MLEFGN is a well-designed model that can make full use of edges predicted by the Edge-Net to reconstruct high-quality noise-free images.

II. Related Works

Recently, plenty of image priors guidance methods have been proposed for image reconstruction. Among them, the edge-guided methods have achieved excellent performance and received widespread attention. In this work, we aim to propose a new edge guidance framework for image denoising. Therefore, how to extract and utilize image edges are the focus of this research.

A. Image Edge Detection

The points at which the brightness of an image changes drastically are usually organized into a set of curve segments called image edges. Image edges are the key components of high-frequency details, which are widely used in image recovery, analysis, and understanding. Therefore, extracting clear and accurate image edges is important for image reconstruction. Nowadays, many edge detectors have been designed, such as Canny [34], Prewitt [35], Sobel, and Roberts operators. However, all of these edge detectors are extremely sensitive to noise (see Fig. 2) since these operators are designed for clear images. Hence, it is difficult to obtain clear and accurate image edges from the noisy image by directly using the existing methods. To solve this problem, we intend to extract image edges via the powerful modeling capabilities of CNN. As shown in Fig. 3, we aim to build an Edge-Net to directly

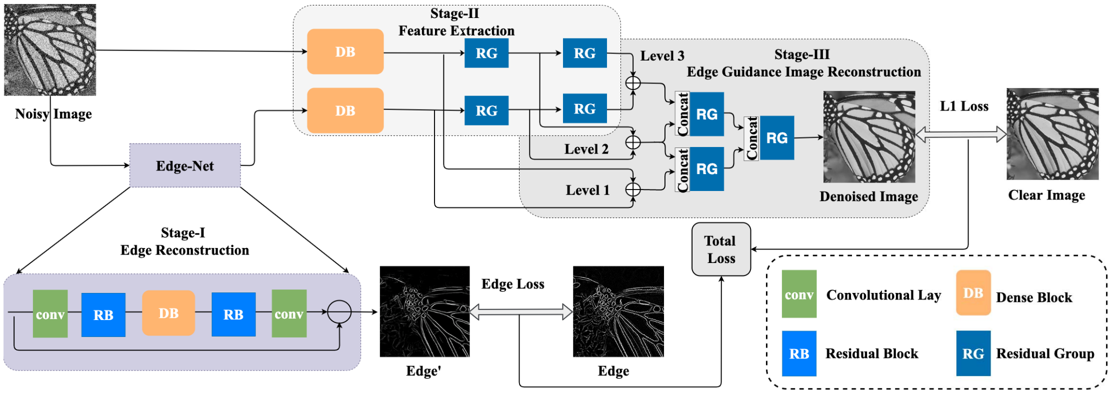


Fig. 4. Complete architecture of our proposed MLEFGN. The green, orange, sky blue, navy blue, and purple blocks represent the convolutional layer, DB, RB, RG, and EdgeNet, respectively.

predict the clear and accurate image edges from the noisy image.

B. Edge-Guided Image Reconstruction

In the past few decades, prior-guided image reconstruction methods have received great attention. For example, the TV prior [36], sparsity prior [14], and edge prior [37], [38] all achieved significant results. Among them, edge prior is one of the most effective and widely used priors since image edges are the key components of high-frequency features. Nowadays, many edge-guided methods have been proposed for image reconstruction. For instance, Lu *et al.* [31] proposed an edge-guided dual-modality image reconstruction method for CT and MRI, which aimed to establish a knowledge-based connection between these two different imaging modalities; Canh *et al.* [30] proposed an edge-preserving weighting scheme by utilizing the nonlocal structure and histogram of the natural image in the gradient domain for compressive sensing recovery; and Guo and Yin [29] proposed an edge-guided compressive sensing reconstruction approach, which can recover high-quality images from fewer measurements. Recently, some studies introduced image edges into the CNN model to further improve the quality of reconstructed images. For instance, Yang *et al.* [39] proposed a DEGRRN for single image superresolution (SISR); Liu *et al.* [68] presented a phase congruency edge map guided multiscale deep encoder–decoder network for SISR; and Chupraphawan and Ratanamahatana [33] designed a DCEF for image denoising. All these methods prove the importance and effectiveness of image edges for image restoration. However, these deep learning-based methods also show some disadvantages.

1) It is difficult for them to obtain correct edges by directly applying the off-the-shelf edge detector (e.g., Sobel, Canny, and Prewitt) on the degraded image. This is because existing edge detectors (whether binarized or nonbinarized edge detector) are extremely sensitive to other factors (e.g., noise and blur kernel), which makes them not robust enough to detect correct edges from the degraded images.
2) They directly add or concatenate the input image with the extracted image edges and then send them to the network for learning. This will greatly weaken the effect of edges, thus inhibiting the model performance. To solve these problems, we aim to explore a more robust method that can extract correct edges from the degraded image and fully use them.

III. Multilevel Edge Features Guided Network

The degradation model of image denoising can be formulated as $y = x + \upsilon$, where y is the noisy image obtained from the clear image x, and υ is the additive white Gaussian noise (AWGN). The goal of image denoising is to reconstruct a clear image x from the noisy one y. Nowadays, CNN-based image denoising methods convert this problem into a learning problem, which aims to learn the mapping function between y and x via a large number of training data sets. That is, given a training data set $\{x^i, y^i\}_{i=1}^{M}$, we need to solve

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{M}\sum_{i=1}^{M} \|F_\theta(y^i) - x^i\|_2^2 \tag{1}$$

where $F_\theta(\cdot)$ and $F_\theta(y^i)$ denote the model and the reconstructed clear image, respectively. Meanwhile, θ denotes the parameter set of the model, and M represents the number of images in the training data set.

Following previous works, we also consider this as a learning problem and propose an MLEFGN for image denoising. As shown in Fig. 4, the MLEFGN can be divided into three stages. In Stage-I, we build an efficient Edge-Net to directly reconstruct clear and accurate image edges from the noisy image. This is the core component of our model and the basic condition for the edge-guided image denoising strategy. In Stage-II, we apply a dual-path dense network to extract image features and edge features, respectively. Finally, we introduce a multilevel edge features guidance mechanism to achieve the final denoised image reconstruction. We define I_{noisy} and I_{clear} as the input and output of the MLEFGN, respectively. Therefore, the edge reconstruction in Stage-I can be expressed as

$$I'_{\text{edge}} = E(I_{\text{noisy}}) \tag{2}$$

where $E(\cdot)$ denotes the Edge-Net and I'_{edge} represents the predicted image edges. Then, we construct a dual-path network for feature extraction in Stage-II. In each path, we use a simplified version of dense block (DB) [40] and two residual groups (RGs) to extract the corresponding features

$$f_{\text{image}} = F_{\text{Top}}(I_{\text{noisy}}) \tag{3}$$

$$f_{\text{edge}} = F_{\text{Bottom}}(I'_{\text{edge}}) \tag{4}$$

where $F_{\text{Top}}(\cdot)$ and $F_{\text{Bottom}}(\cdot)$ denote the corresponding path network, and f_{image} and f_{edge} represent the extracted image low- and high-frequency (image edges) features, respectively. Meanwhile, these two branches are independent of each other. In order to make full use of image edges, we introduce an edge guidance module to reconstruct the final denoised image in Stage-III

$$I'_{\text{clear}} = F_{\text{EGM}}(f_{\text{image}}, f_{\text{edge}}) \tag{5}$$

where $F_{\text{EGM}}(\cdot)$ denotes the edge guidance module and I'_{clear} represents the reconstructed denoised image.

MLEFGN is an end-to-end image denoising model that embeds the Edge-Net as a part of the model to provide edge priors. For this, we propose an edge preservation loss to further improve the model performance. Different from previous works that only learn the mapping between noisy and clear images, we need to ensure the accuracy of the reconstructed edges. Hence, given a training data set $\{I^i_{\text{noisy}}, I^i_{\text{clear}}, I^i_{\text{edge}}\}^M_{i=1}$, we need to solve

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{M} \sum_{i=1}^{M} \|F(I^i_{\text{noisy}}, E(I^i_{\text{noisy}})) - I^i_{\text{clear}}\|_1 + \lambda \|E(I^i_{\text{noisy}}) - I^i_{\text{edge}}\|_1 \tag{6}$$

where θ denotes the parameter set of our MLEFGN, λ is a hyperparameter, and $E(\cdot)$ and $F(\cdot)$ denote the Edge-Net and MLEFGN, respectively.

MLEFGN is a well-designed edge-guided image denoising model, which is the first CNN model that implements edge detection, edge guidance, and image denoising in a single network. It is worth noting that the Edge-Net is served as a submodel of the MLEFGN, which is trained together with the whole model in an end-to-end manner. Each component of the model will be described in Sections III-A–III-C.

A. Stage-I: Edge Reconstruction

As aforementioned, using the off-the-shelf edge detectors is difficult to reconstruct clear and accurate edges from the noisy image since all of these edge detectors are sensitive to noise (see Fig. 2). To address this problem, we propose an Edge-Net to directly predict the clear and accurate image edges from the noisy image via the powerful modeling capabilities of CNN.

As shown in Fig. 4, the purple block represents the Edge-Net, which consists of two convolutional layers, two residual blocks (RBs), and one DB. Among them, the convolutional layers are used to change the spatial dimension of images, and the residual and DBs are used to extract image features. Fig. 5(a) and (c) shows the RB [41] and DB [40], respectively. Different from the original RB, we remove the batch normalization layers to speed up the model. As for DB, each layer in the block is connected to every other layer in a feedforward fashion. Consequently, the pth layer receives feature maps of all preceding layers $(X_0, X_1, \ldots, X_{p-1})$ as input

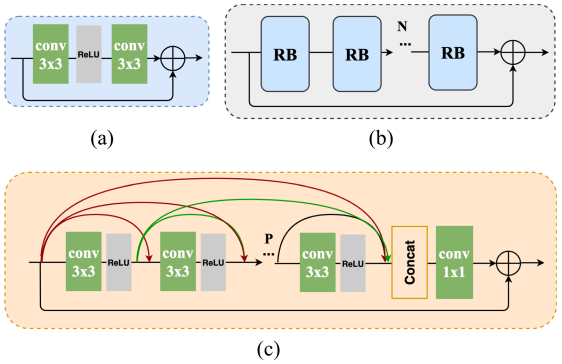


Fig. 5. Architecture of (a) RB, (b) RG, and (c) DB.

$$X_p = H_p([X_0, X_1, \ldots, X_{p-1}]) \tag{7}$$

where $[X_0, X_1, \ldots, X_{p-1}]$ denotes the concatenation of feature maps produced in layers $0, \ldots, p-1$. Here, $p \leq P$, and P represents the number of layers in the DB. Meanwhile, H_p indicates the current convolutional layer, and X_p represents the output of this layer. Note that DBs are known for their powerful feature extraction capabilities. However, this dense connection mechanism will bring a large number of parameters. To solve this issue, a 1×1 bottleneck layer is introduced at the tail of the block to reduce model parameters. Furthermore, we also introduce local residual learning to improve the information flow and, thus, improve model performance.

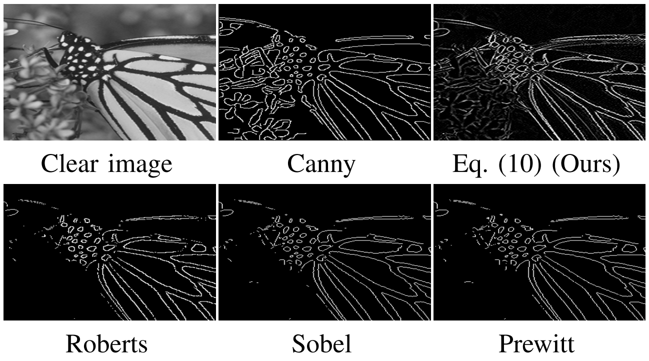
In this stage, we aim to build an Edge-Net to learn the mapping between the noisy image and its corresponding clear edges; thus, we can directly reconstruct accurate edges from the noisy image. However, we find it difficult to learn this mapping directly. Therefore, we transform this problem into a feature removal problem. Especially, we build a network to learn how to eliminate low-frequency features and image noise simultaneously so that the remaining features are the required image edges. Define all convolutional layers in the Edge-Net as $R(\cdot)$, and the operations of the Edge-Net can be rewritten as

$$I'_{\text{edge}} = E(I_{\text{noisy}}) = I_{\text{noisy}} - R(I_{\text{noisy}}) \tag{8}$$

where I'_{edge} denotes the predicted image edges. Meanwhile, the edge loss of the Edge-Net can be defined as

$$\mathcal{L}_{\text{edge}} = \left\| I'_{\text{edge}} - I_{\text{edge}} \right\|_1 \tag{9}$$

where I_{edge} is the clear edges detected on the clear image. The easiest way to obtain I_{edge} is to apply the existing edge detector (e.g., Canny, Sobel, or Roberts) on a clear image. However, these detectors use the binarization measurement to convert all edge values to 0 and 1, which causes a lot of information loss and the appearance of false edges. Meanwhile, we find that these detectors are not robust enough to fit all images, which hinders their widespread application. Therefore, we recommend retaining more edge information by eliminating the binarization strategy. To achieve this, we propose a new


Clear image Canny Eq. (10) (Ours)

Roberts Sobel Prewitt

Fig. 6. Results of different edge detectors acting on a clear image. It is obvious that our method can obtain more high-frequency details.

curvature formula for edge detection, which does not use the binarization strategy

$$I_{\text{edge}} = \text{div}(\text{u}_\text{a}, \text{u}_\text{b}) \tag{10}$$

where $u_i = (\nabla_i I_{\text{clear}}/(1 + |\nabla I_{\text{clear}}|^2)^{1/2}), i = \{a, b\}$, and a and b represent the horizontal and vertical directions, respectively. Meanwhile, ∇ and $\text{div}(\cdot)$ represent the gradient and divergence operation, respectively. Compared with existing edge extractors, our curvature formula can accurately describe the changes in the gradient regions and obtain more high-frequency details (see Fig. 6); thus, we can get more precise edges as our training labels. In this article, we use (10) to obtain the label for training because of its simplicity and feasibility. However, it is worth noting that (10) can be replaced by any better edge detector. This means that one can directly use other nonbinarized edge extraction methods or achieve edge extraction by suppressing the binarization operation of the off-the-shelf edge detector (e.g., Sobel, Canny, and Prewitt).

B. Stage-II: Feature Extraction

To fully extract and utilize image and edge features, we construct a dual-path network to extract the image and edge features separately. In this stage, each path of the network contains a DB and two RGs. Among them, the DB is used to extract features, and RGs are used to further refine the extracted features. As shown in Fig. 5, we use N RBs to form an RG. However, we observe that simply stacking RB will achieve suboptimal results. Therefore, we apply a long skip connection for global residual learning, such as RCAN [42]. This strategy can ease the flow of information across RBs and stabilize the model training. With the help of DB and RG, abundant image and edge features can be extracted in this stage. This is essential for the subsequent edge guidance and high-quality denoised image reconstruction.

C. Stage-III: Edge Guidance Image Denoising

In the previous stage, we have obtained plenty of image and edge features. However, these two types of features are independent of each other, which is not conducive to achieving good results. The most widely used method is to directly fuse or add the image and edge features to obtain the final image. However, this method will lead to feature loss or conflict. To solve this problem, we propose a multilevel guidance mechanism. This strategy can make full use of different levels of image and edge features to stimulate the potential of edge guidance, which is conducive to reconstruct high-quality denoised images.

In particular, image or edge features will gradually disappear as the depth of the network increases. Therefore, only using the final output from Stage-II to reconstruct the denoised image is not the optimal method. In order to increase the diversity of features, we also use the hierarchical image and edge features. As shown in Fig. 4, the output of each module in Stage-II is considered as a level; thus, all the extracted features can be divided into three levels, and the ith level image and edge features fusion can be defined as

$$f^i_{\text{fuse}} = f^i_{\text{image}} + f^i_{\text{edge}} \tag{11}$$

where f^i_{image} and f^i_{edge} denote the image and edge feature maps extracted in the ith level ($i = 1, 2, 3$), respectively. After that, we can obtain three outputs f^1_{fuse}, f^2_{fuse}, and f^3_{fuse}.

Since these outputs contain rich image and edge features, how to make full use of them is essential for reconstructing high-quality denoised images. The easiest method is directly concatenating these features and transferring them to a clear image. However, we find this method cannot achieve the best results and make the model difficult to converge. To solve this problem, we design a simple inverted pyramid module (IPM) for feature fusion and image reconstruction. As shown in Fig. 4, this module is located at the tail of MLEFGN and consists of three RGs. All these features (f^1_{fuse}, f^2_{fuse}, and f^3_{fuse}) will be sent to the module, and the RG will be used to concatenate and fuse the features in adjacent levels

$$I^1_{\text{IPM}} = F^1_{\text{RG}}\left(\left[f^1_{\text{fuse}}, f^2_{\text{fuse}}\right]\right) \tag{12}$$

$$I^2_{\text{IPM}} = F^2_{\text{RG}}\left(\left[f^2_{\text{fuse}}, f^3_{\text{fuse}}\right]\right) \tag{13}$$

$$I^3_{\text{IPM}} = F^3_{\text{RG}}\left(\left[I^1_{\text{IPM}}, I^2_{\text{IPM}}\right]\right) \tag{14}$$

where $[\cdot]$ is the concatenation operation, F^i_{RG} denotes the ith RG, and I^i_{IPM} represents the corresponding output. Finally, we use L_1 as the content loss to minimize the difference between the denoised and clear images

$$\mathcal{L}_{\text{content}} = \left\| I'_{\text{clear}} - I_{\text{clear}} \right\|_1 \tag{15}$$

where $I'_{\text{clear}} = I^3_{\text{IPM}}$, which represents the reconstructed denoised image. Therefore, the total loss of MLEFGN can be rewritten as

$$\mathcal{L}_{total} = \mathcal{L}_{\text{content}} + \lambda \mathcal{L}_{\text{edge}}. \tag{16}$$

Overall, we propose an MLEFGN for image denoising. Especially, MLEFGN contains three stages, and all these stages are closely combined and trained in an end-to-end manner. This is the first image denoising model that integrates edge detection, edge guidance, and image denoising at the same time.

TABLE I

PSNR (dB) RESULTS OF DIFFERENT IMAGE DENOISING METHODS ON **GRAY-SCALE** IMAGES (**SET12**) WITH NOISE LEVELS $\sigma = 15, 25, 35$, AND 50. "AVERAGE" REPRESENTS THE AVERAGE RESULT OF THE DATA SET, AND THE BEST RESULTS ARE HIGHLIGHTED IN **RED**

Images	C.ma	House	Peppers	Starfish	Monarch	Airplane	Parror	Lena	Barbara	Boat	Man	Couple	**Average**
Noise Level	σ**=15**												
BM3D [15]	31.91	34.93	32.69	31.14	31.85	31.07	31.37	34.26	33.10	32.13	31.92	32.10	32.37
TNRD [43]	32.19	34.53	33.04	31.75	32.56	31.46	31.63	34.24	32.13	32.14	32.23	32.11	32.50
NLED$^6_{7\times7}$ [44]	32.28	34.76	33.10	31.75	32.71	31.59	31.70	34.35	32.53	32.16	32.22	32.13	32.61
WNNM [16]	32.17	35.13	32.99	31.82	32.71	31.39	31.62	34.27	33.60	32.27	32.11	32.17	32.70
IRCNN [45]	32.55	34.89	33.31	32.02	32.82	31.70	31.84	34.53	32.43	32.34	32.40	32.40	32.77
DnCNN [26]	32.61	34.97	33.30	32.20	33.09	31.70	31.83	34.62	32.64	32.42	32.46	32.47	32.86
FFDNet [28]	32.42	35.01	33.10	32.02	32.77	31.58	31.77	34.63	32.50	32.35	32.40	32.45	32.75
ADNet [46]	**32.81**	35.22	**33.49**	32.17	33.17	**31.86**	**31.96**	34.71	32.80	32.57	32.47	32.58	32.98
MLEFGN (Ours)	32.56	**35.41**	33.42	**32.29**	**33.44**	31.82	31.90	**34.80**	**33.05**	**32.60**	**32.51**	**32.67**	**33.04**
Noise Level	σ**=25**												
BM3D [15]	29.45	32.85	30.16	28.56	29.25	28.42	28.93	32.07	30.71	29.90	29.61	29.71	29.97
TNRD [43]	29.72	32.53	30.57	29.02	29.85	28.88	29.18	32.00	29.41	29.91	29.87	29.71	30.06
NLED$^6_{7\times7}$ [44]	29.75	32.81	30.66	29.09	30.03	28.99	29.29	32.18	30.11	29.90	29.86	29.74	30.18
WNNM [16]	29.64	32.22	30.42	29.03	29.84	28.69	29.15	32.24	**31.24**	30.03	29.76	29.82	30.26
IRCNN [45]	30.08	33.06	30.88	29.27	30.09	29.12	29.47	32.43	29.92	30.17	30.04	30.08	30.38
DnCNN [26]	30.18	33.06	30.87	29.41	30.28	29.13	29.43	32.44	30.00	30.21	30.10	30.12	30.43
FFDNet [28]	30.06	33.27	30.79	29.33	30.14	29.05	29.43	32.59	29.98	30.23	30.10	30.18	30.43
ADNet [46]	**30.34**	33.41	**31.14**	29.41	30.39	29.17	**29.49**	32.61	30.25	30.37	30.08	30.24	30.58
MLEFGN (Ours)	30.29	**33.61**	30.98	**29.66**	**30.52**	**29.25**	29.46	**32.76**	30.57	**30.40**	**30.13**	**30.30**	**30.66**
Noise Level	σ**=35**												
BM3D [15]	27.92	31.36	28.51	26.86	27.58	26.83	27.40	30.56	28.98	28.43	28.22	28.15	28.40
MLP [20]	28.08	31.18	28.54	27.12	27.97	27.22	27.72	30.82	27.62	28.53	28.47	28.24	28.46
WNNM [16]	**28.80**	31.92	28.75	27.27	28.13	27.10	27.69	30.73	**29.48**	28.54	28.33	28.24	28.69
DnCNN [26]	28.61	31.61	29.14	27.53	28.51	27.52	27.94	30.91	28.09	28.72	28.66	28.52	28.82
FFDNet [28]	28.54	31.99	29.18	27.58	28.54	27.47	28.02	31.20	28.29	28.82	28.70	28.68	28.92
MLEFGN (Ours)	28.78	**32.47**	**29.37**	**27.77**	**28.70**	**27.62**	**28.03**	**31.36**	29.09	**29.00**	**28.79**	**28.88**	**29.15**
Noise Level	σ**=50**												
BM3D [15]	26.13	29.69	26.68	25.04	25.82	25.10	25.90	29.05	27.22	26.78	26.81	26.46	26.72
MLP [20]	26.37	29.64	26.68	25.43	26.26	25.56	26.12	29.32	25.24	27.03	27.06	26.67	26.78
TNRD [43]	26.62	29.48	27.10	25.42	26.31	25.59	26.16	28.93	25.70	26.94	26.98	26.50	26.81
WNNM [16]	26.45	30.33	26.95	25.44	26.32	25.42	26.14	29.25	27.79	26.97	26.94	26.64	27.05
IRCNN [45]	26.88	29.96	27.33	25.57	26.61	25.89	26.55	29.40	26.24	27.17	27.17	26.88	27.14
DnCNN [26]	27.03	30.00	27.32	25.70	26.78	25.87	26.48	29.39	26.22	27.20	27.24	26.90	27.18
FFDNet [28]	27.03	30.43	27.43	**25.77**	26.88	25.90	**26.58**	29.68	26.48	27.32	27.30	27.07	27.32
ADNet [46]	27.31	30.59	**27.69**	25.70	26.90	25.88	26.56	29.59	26.64	27.35	27.17	27.07	27.37
MLEFGN (Ours)	27.15	**31.00**	27.63	**25.77**	**27.01**	**26.05**	26.56	**29.85**	**27.37**	**27.40**	**27.32**	**27.35**	**27.54**

IV. EXPERIMENTS

A. Data Sets

In this article, we choose the AWGN as our research object due to its extensiveness and practicality. Therefore, we choose Set12 [47], BSD68 [48], Kodak24 [49], CBSD68 [50], and Urban100 [51] with different AWGN noise levels as our test data sets. Following the previous works [26], [28], we use Set12, BSD68, and Urban100 to evaluate the performance of MLEFGN on gray-scale images and CBSD68, Kodak24, and Urban100 to test its performance on color images. It is well-known that real noise images are more complicated than simulated images. In order to further verify the effectiveness and robustness of MLEFGN, we utilize RNI6 [52] and RNI15 [53] to test the ability of MLEFGN for real noise removing. Both of them have no ground truth; thus, we can use it to test our model performance with unknown noise mode and level.

B. Implementation Details

1) Training Setting: Before training, we first apply (10) on the training data set to obtain their corresponding clear image edges. Then, we generate noisy images by adding AWGN with different noise levels. To verify the effectiveness of the model, we set noise level $\sigma = 15, 25, 35, 50$, and 75 for gray-scale images and $\sigma = 10, 30, 50$, and 70 for color images. During training, we randomly choose 16 noisy patches with the size of $S \times S$ ($S = 96$ for gray-scale images and $S = 64$ for color images) as inputs, and the learning rate is initialized as 10^{-4}. In addition, MLEFGN is implemented with the PyTorch framework and updated with the Adam optimizer.

2) Model Setting: In this article, we propose an MLEFGN for image denoising. In order to achieve an end-to-end manner, the Edge-Net is embedded as a part of the MLEFGN to provide image edge priors. Meanwhile, we set $\lambda = 0.5$ to control the proportion of $\mathcal{L}_{\text{edge}}$ according to plenty of experiments. In the final model, each RG contains five ($N = 5$) RBs, each DB contains eight ($P = 8$) convolutional layers, and each layer contains 64 ($C = 64$) channels. Meanwhile, the kernel sizes of all the convolutional layers are all set as 3×3 except for the bottleneck layer, whose kernel size is 1×1.

C. Comparisons With State-of-the-Art Methods

In order to verify the effectiveness of MLEFGN, we compare MLEFGN with more than 14 image denoising methods, including EPLL [54], BM3D [15], MLP [20], RED [56], MemNet [57], TNRD [43], NLED$^6_{7\times7}$ [44], NN3D [58], WNNM [16], IRCNN [45], DnCNN [26], FFDNet [28], ADNet [46], and N^{3}Net [55]. All denoised images are evaluated with PSNR. Due to page limitations, we only provide a part of the results; more results can be found on our homepage.

1) PSNR Results on Gray-Scale Images: In Tables I, II, and III, we report the average PSNR results of different

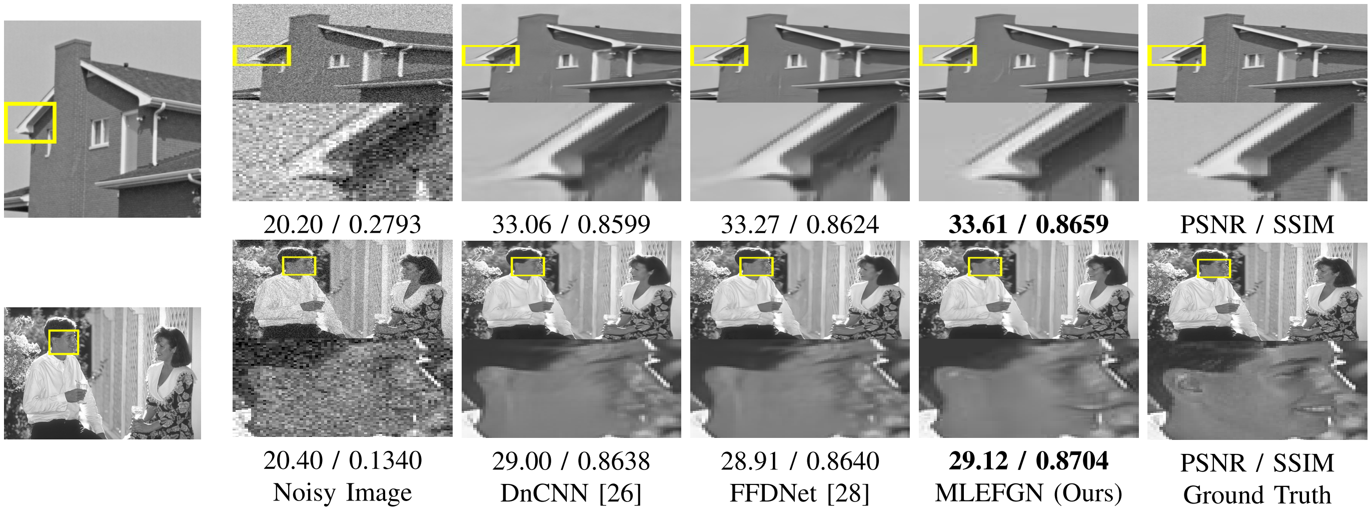


Fig. 7. Visual comparison of MLEFGN with DnCNN [26] and FFDNet [28] on gray-scale images (noise level: $\sigma = 25$).

TABLE II

AVERAGE PSNR (dB) RESULTS OF DIFFERENT IMAGE DENOISING METHODS ON **GRAY-SCALE** IMAGES (**BSD68**) WITH NOISE LEVELS $\sigma = 15, 25, 35$, AND 50

Method	σ=15	σ=25	σ=35	σ=50
BM3D [15]	31.08	28.57	27.08	25.62
EPLL [54]	31.21	28.68	27.16	25.67
WNNM [16]	31.37	28.83	27.30	25.87
TNRD [43]	31.42	28.92	N/A	25.97
$\text{NLED}_{7\times7}^{6}$ [44]	31.43	28.93	N/A	N/A
MLP [20]	31.50	28.96	27.50	26.03
DnCNN [26]	31.73	29.23	27.69	26.23
FFDNet [28]	31.63	29.19	27.73	26.29
ADNet [46]	31.74	29.25	N/A	26.29
N^{3}Net [55]	N/A	29.30	N/A	**26.39**
MLEFGN (Ours)	**31.81**	**29.34**	**27.85**	**26.39**

TABLE III

AVERAGE PSNR (dB) RESULTS OF DIFFERENT IMAGE DENOISING METHODS ON **GRAY-SCALE** IMAGES (**URBAN100**) WITH NOISE LEVELS $\sigma = 15, 25$, AND 50

Method	σ=15	σ=25	σ=50
TNRD [43]	31.98	29.29	25.71
BM3D [15]	32.34	29.70	25.94
IRCNN [45]	32.49	29.82	26.14
DnCNN [26]	32.68	29.97	26.28
NN3D [58]	N/A	30.09	26.47
FFDNet [28]	32.42	29.92	26.52
N^{3}Net [55]	N/A	30.19	26.82
WNNM [16]	32.97	30.39	26.83
MLEFGN (Ours)	**33.21**	**30.64**	**27.22**

image denoising methods on Set12 [47], BSD68 [48], and Urban100 [51] with noise level $\sigma = 15$, 25, 35, and 50, respectively. Meanwhile, we provide detailed results of each image in Set12 to compare these methods in Table I more intuitively. Among them, BM3D and WNNM are two representative model-based methods based on nonlocal self-similarity, DnCNN, FFDNet, and N^{3}Net are the most classic and famous CNN-based methods, and ADNet and $\text{NLED}_{7\times7}^{6}$ are the latest methods proposed in 2020. According to these tables, we can clearly observe that our MLEFGN achieves the best results on all data sets. Compared with the classic CNN-based denoising method (e.g., DnCNN and FFDNet), the performance of MLEFGN has been significantly improved. In addition, the results of our MLEFGN are slightly better than N^{3}Net on BSD68. It is worth noting that MLEFGN achieves excellent performance on Urban100, which is 0.4 dB higher than N^{3}Net. This is because Urban100 contains 100 images of natural buildings with obvious object edges. With the help of Edge-Net, our MLEFGN can better reconstruct them. All these results demonstrate the powerful denoising ability of MLEFGN on image gray-scale images.

2) PSNR Results on Color Images: As for color image denoising, we employ three data sets, namely, Kodak24 [49], CBSD68 [50], and Urban100 [51]. Among them, CBSD68 is the corresponding color version of the gray-scale BSD68 database. In Table IV, we provide the average results of different image denoise methods on these three data sets. Different from gray-scale images, we choose noise level $\sigma = 10$, 30, 50, and 70 for color images. The purpose is to increase the number of noise levels to better test the performance and generalization of MLEFGN. It is obvious that our MLEFGN achieves the best results on all data sets. This further demonstrates the effectiveness and robustness of MLEFGN, which can achieve excellent performance in different color spaces, noise levels, and data sets.

3) Visual Comparisons: In Figs. 7 and 8, we provide the visual results of the denoised images with noise levels 25 and 50. In this part, we choose two of the most representative CNN-based image denoising methods: DnCNN [26] and FFDNet [28] for comparison. DnCNN and FFDNet achieve competitive results in image denoising, which remains the two most widely used methods. However, we find that they are not perfect in processing image details, especially on textures and edges. As shown in Fig. 7, we can observe that the denoised images produced by DnCNN and FFDNet still contain noise and artifacts. For example, there are a lot of noises and artifacts under the eaves, and the face is blurred and unclear. In contrast, our MLEFGN can reconstruct high-quality

TABLE IV

AVERAGE PSNR (dB) RESULTS OF DIFFERENT IMAGE DENOISING METHODS ON COLOR IMAGES (KODAK24 [49], CBSD68 [50], AND URBAN100 [51]) WITH NOISE LEVELS σ = 10, 30, 50, AND 70. BEST RESULTS ARE HIGHLIGHTED IN RED COLOR

Method	Kodak24 [49]				CBSD68 [50]				Urban100 [51]			
Noise Level	σ=10	σ=30	σ=50	σ=70	σ=10	σ=30	σ=50	σ=70	σ=10	σ=30	σ=50	σ=70
TNRD [43]	34.33	28.83	27.17	24.94	33.36	27.64	25.96	23.83	33.60	27.40	25.52	22.63
RED [56]	34.91	29.71	27.62	26.36	33.89	28.46	26.35	25.08	34.59	29.02	26.40	24.74
MemNet [57]	N/A	29.67	27.65	26.40	N/A	28.39	26.33	25.08	N/A	28.93	26.53	24.93
CBM3D [15]	36.57	30.89	28.63	27.27	35.91	29.73	27.38	26.00	36.00	30.36	27.94	26.31
IRCNN [45]	36.70	31.24	28.93	N/A	36.06	30.22	27.86	N/A	26.53	30.28	27.69	N/A
DnCNN [26]	36.98	31.39	29.16	27.64	36.31	30.40	28.01	26.56	36.21	30.28	28.16	26.17
FFDNet [28]	36.81	31.39	29.10	27.68	36.14	30.31	27.96	26.53	35.77	30.53	28.05	26.39
MLEFGN (Ours)	**37.04**	**31.67**	**29.38**	**27.94**	**36.37**	**30.56**	**28.21**	**26.75**	**36.42**	**31.32**	**28.92**	**27.28**

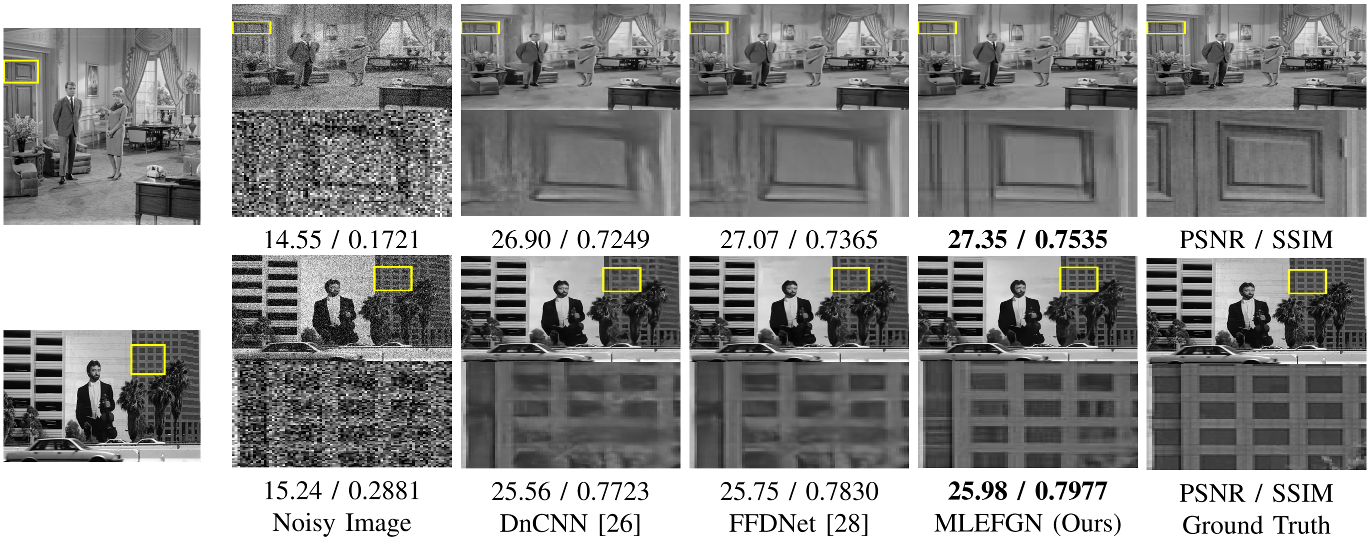


Fig. 8. Visual comparison of MLEFGN with DnCNN [26] and FFDNet [28] on gray-scale images (noise level: $\sigma = 50$).

images with clear textures and edges. In Fig. 8, we show the reconstruction images with noise level $\sigma = 50$. We can clearly see that the input images have been seriously polluted, and the edges reconstructed by DNCNN and FFDNet have been severely damaged. However, with the help of edge priors, our MLEFGN can reconstruct clear and accurate image edges.

All the aforementioned experiments fully demonstrate the excellent denoising ability of the MLEFGN in both gray-scale and color images. This improvement is mainly due to: 1) the introduced edge priors can effectively protect the image edges from damage, thereby rebuilding sharp and accurate image edges; 2) the well-designed dual-path network can extract the image and edge features effectively; and 3) the introduced multilevel edge features guidance mechanism can maximize the use of image edges to reconstruct the final denoised images.

V. INVESTIGATION

A. Compare With Previous Edge Guidance Methods

TABLE V

PSNR (dB) COMPARISON WITH DCEF ON SET12 AND BSD68

Datasets	Set12			BSD68		
Noisy Level	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50
DCEF	32.65	33.33	26.92	31.50	29.14	26.05
MLEFGN (Ours)	**33.04**	**33.66**	**27.54**	**31.81**	**29.34**	**26.39**

Image edges are the key components of high-frequency details. Therefore, many researchers introduced it to their methods for image reconstruction. Among them, we found a denoising method that introduces image edges into the CNN model, named DCEF [33]. DCEF was proposed by Chupraphawan *et al.* in 2019, and the method can be divided into three steps: 1) extracting image edges from the noisy input using the Canny detector; 2) adding the noisy image and the extracted edges to obtain the new input; and 3) building a CNN model to reconstruct the final denoised image. Different from this method, we aim to explore a new edge guidance framework that can extract accurate image edges and effectively use image edges. In Table V, we show the average PSNR comparison between these two methods. It is obvious that the performance of MLEFGN has been significantly improved. Furthermore, we can observe that as the noise level increases, the PSNR gap gradually increases. This is because as the noise level increases, the image edges are deteriorated, making the edge priors become more important. With the powerful modeling capabilities of CNN, the Edge-Net can directly reconstruct sharp edges from the noisy image. Therefore, we can reconstruct high-quality denoised images.

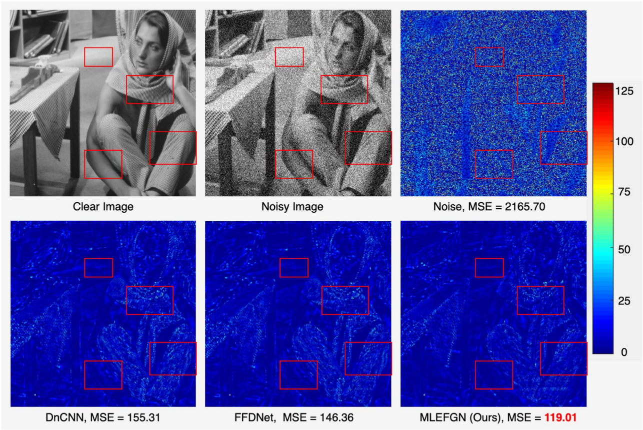


Fig. 9. Recovery performance comparison (noise level: $\sigma = 50$). Please zoom in to view details.

B. Image Region Exploration

The image region can be divided into smooth, edges, and texture [59], [60]. Among them, the smooth region is the easiest region to reconstruct, and edge guidance methods will bring better quality to edge region. However, its good preservation of texture is still challenging as small textural variation cannot be well differentiated from noise [59], [61]. In Figs. 7 and 8, we have provided a visual comparison of the denoised images. In this part, we will further explore the reconstruction effect of each region of the image.

1) In Fig. 9, we show the mean squared error (mse) map of the reconstructed image. In the mse map, the bluer the pixel, the smaller the mse error, and the better the reconstruct result. According to the figure, we can observe that DnCNN, FFDNet, and our MLEFGN can effectively protect the texture. However, after zooming in, we can see that there are fewer bright areas in MLEFGN's mse map, which means that MLEFGN can reconstruct higher quality denoised images.
2) In Fig. 10, we provide detailed visualization results of the image region, including smooth, edges, and texture. It is obvious that all regions of the noisy image were severely damaged. With the help of CNN's powerful modeling capabilities, DnCNN and FFDNet can recover images details to a certain extent. However, a closer inspection of both denoised results reveals that there is still a bit of noise in the smooth region, some flaw in the edge region, and a lot of artifacts in the texture region. In contrast, MLEFGN shows excellent performance in all of these regions.

C. Effectiveness on High Noise Level

High-noise-level image denoising is still a challenging task since noise will severely damage images. To further explore the denoising capabilities of MLEFGN at high noise levels, we provide PSNR and visual comparisons on gray-scale images with noise $\sigma = 75$. In Table VI, we provide the average PSNR on Set12 and BSD68. Combining with Tables I and II, we can find the following phenomena.

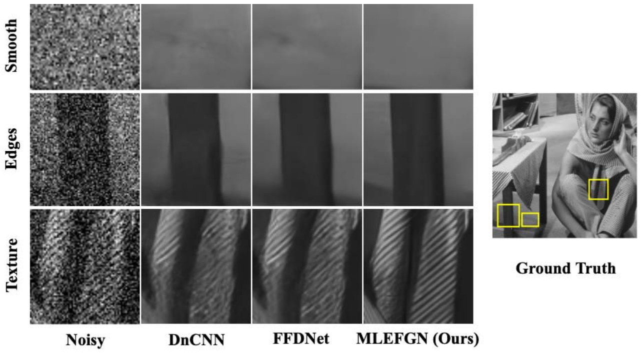


Fig. 10. Image region visualization (noise level: $\sigma = 50$).

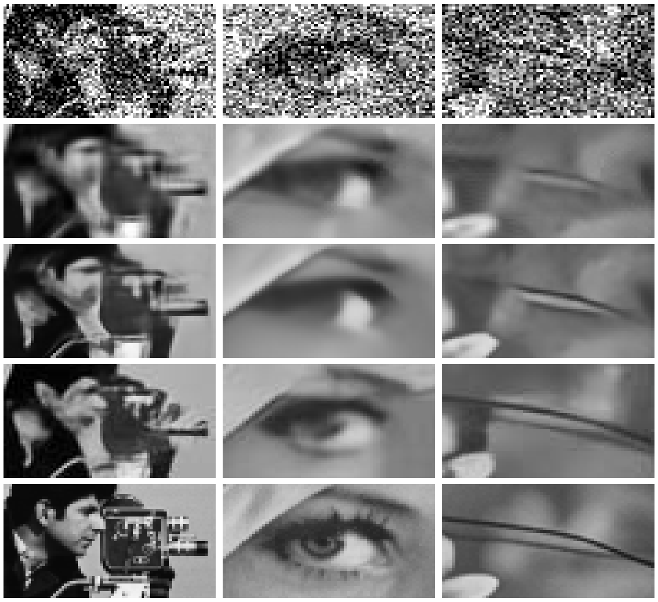


Fig. 11. Visual comparison on gray-scale images (noise level: $\sigma = 75$). Top to bottom: noisy image, denoised images by BM3D, denoised images by FFDNet, denoised images by MLEFGN (ours), and the ground truth.

TABLE VI
PSNR Comparison on Gray-Scale Images (Noise Level: $\sigma = 75$)

Noise Level σ=75				
	BM3D	DnCNN	FFDNet	MLEFGN (Ours)
Set12	24.91	25.20	25.49	**25.69**
BSD68	24.21	24.64	24.79	**24.85**

1) The PSNR gap between BM3D and CNN-based methods (DnCNN, FFDNet, and MLEFGN) further increases under a high noise level.
2) At relatively low noise levels, the performance of DnCNN is comparable to FFDNet or even better than it. However, as the noise level increases, the performance of FFDNet is better than DnCNN.
3) The performance of MLEFGN is better than FFDNet whether at low or high noise levels.

In Fig. 11, we show the visual comparison at high noise level $\sigma = 75$. It is obvious that the content in the input image has been severely damaged, making it difficult for even humans to recognize. Although BM3D and FFDNet can effectively remove noise, we found that the denoised

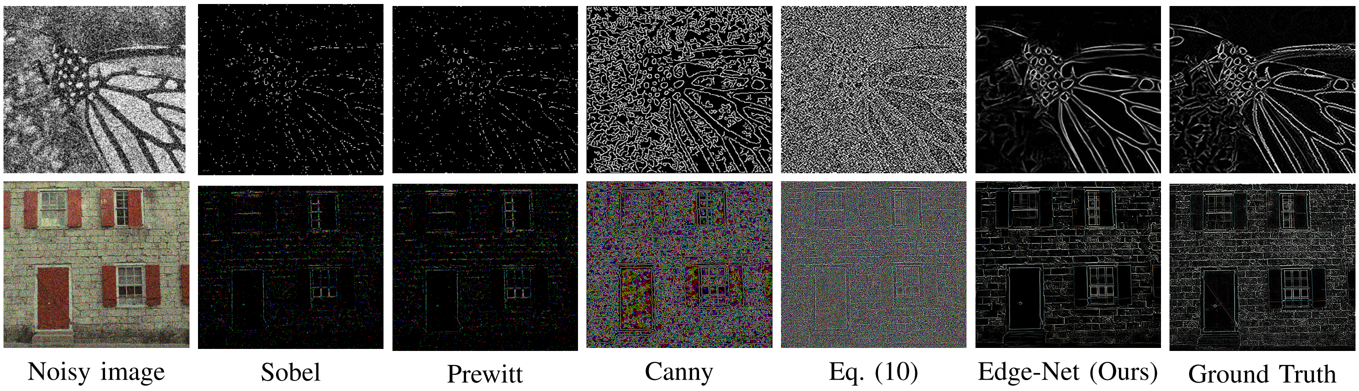


Fig. 12. Visual comparison of image edges extracted by different methods. Noise levels of gray-scale and color images are set to $\sigma = 75$ and 50, respectively.

image contains a lot of artifacts, and the details (edges and texture) have been destroyed. In contrast, the denoised images reconstructed by our MLEFGN achieve an excellent visual effect, which has clear and accurate edges. This is because high-noise image denoising has stricter requirements on model performance. With the help of edge priors, MLEFGN can reconstruct high-quality denoised images even under high noise levels. However, it is worth noting that the smoothing effect and some incorrect regions can also be observed in our results. This might be caused by the loss of information and incorrect edge guidance. Contaminated by high-level noise, a lot of information in the image will be lost. This is still a great challenge for Edge-Net to reconstruct the correct edges. Fortunately, our results are better than existing methods, both in PSNR and visual effects. In future work, we will pay more attention to the high-noise-level image denoising task and commit to proposing a more effective edge reconstruction model and image denoising model.

VI. Ablation Study

In this article, we propose an MLEFGN for image denoising. In order to study the effectiveness of the proposed method, we provide a series of ablation studies in this section.

A. Study of the Edge-Net

Edge-Net is the key component of MLEFGN, which is designed to extract clear and accurate edges directly from the noisy one. Due to its uniqueness and completeness, it can work independently or be embedded in any denoising model to provide image priors. To further explore its performance, we design the following experiments.

1) We train an independent Edge-Net to verify its ability to extract edges from the noisy image. As shown in Fig. 12, we provide some edge results extracted by classical edge detectors, including Sobel, Prewitt, Canny, curvature formula (10), and our Edge-Net. It is obvious that all these edge operators are sensitive to noise, so it is difficult to extract clear and accurate edges from the noisy image. In addition, our proposed curvature formula (10) is also sensitive to noise although it can obtain more high-frequency details from the clear image. Therefore, we build an Edge-Net instead of using them directly to extract image edges. According to this figure, we can clearly observe that with the powerful modeling capabilities of CNN, our Edge-Net can reconstruct clear and accurate image edges. Compared with other methods, image edges reconstructed by the Edge-Net are amazing. This fully illustrates the effectiveness of Edge-Net in edge extraction.

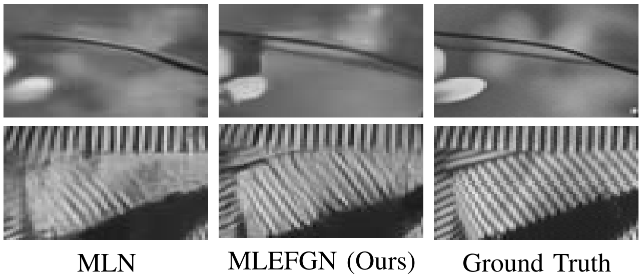


Fig. 13. Visual comparison of MLN and MLEFGN (noise level: $\sigma = 75$).

2) Edge-guided image restoration methods have received increasing attention in the past few decades. In this work, Edge-Net serves as a part of MLEFGN to provide edge priors to reconstruct high-quality denoised images. In order to verify the effect of the edge priors provided by the Edge-Net, we build a new model named the multilevel network (MLN). MLN uses the framework of MLEFGN as the backbone and removes the Edge-Net in the new network. Therefore, both paths of the network in MLN receive the same noise image as input. In Fig. 13, we show the visual comparison of MLN and MLEFGN at noise level $\sigma = 75$. It is obvious that the edges of the denoised image reconstructed by MLN have been damaged. This is because the edges in the input image have been severely damaged, so it is difficult for the network to recover them. Contrarily, the introduced Edge-Net can effectively predict and reconstruct clear edges, so our MLEFGN can rebuild more accurate images than MLN. This further demonstrates the importance and effectiveness of the edge priors provided by the Edge-Net.

3) All the aforementioned experiments demonstrate the effectiveness of the edge guidance mechanism. In this

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

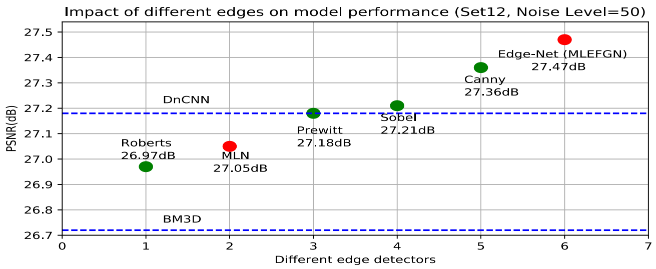


Fig. 14. Study of different edge detectors on Set12 (noise level: $\sigma = 50$).

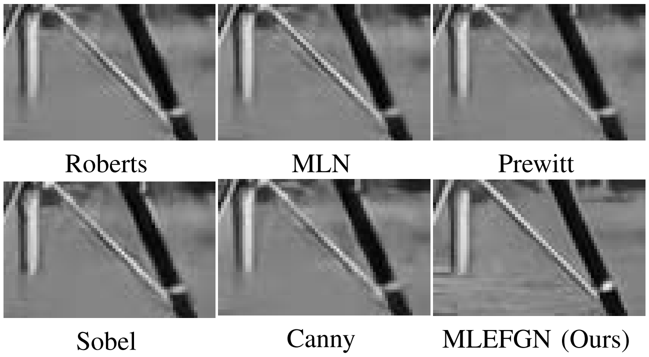


Fig. 15. Visual comparison of the denoised images with $\sigma = 50$.

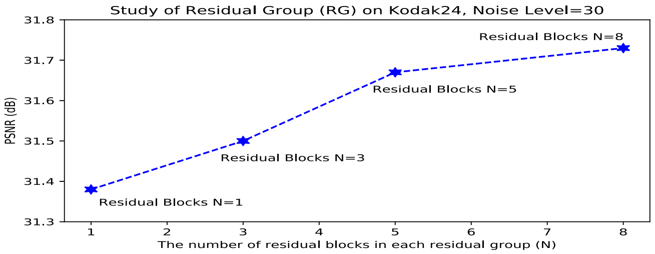


Fig. 16. Study of RG on Kodak24 (noise level: $\sigma = 30$). We set $N = 1, 3, 5$, and 8 in each RG, respectively.

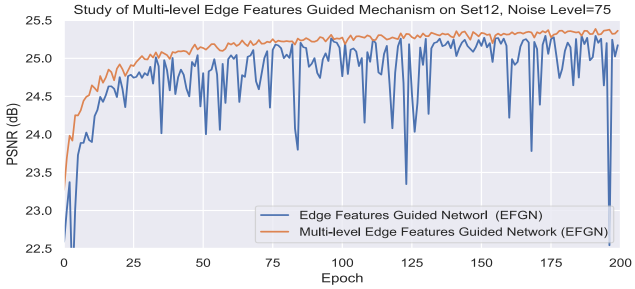


Fig. 17. Study of the multilevel guidance mechanism (noise level: $\sigma = 75$).

part, we will explore the impact of edges extracted by different edge detectors on model performance. To achieve this, we build a simplified version of MLEFGN. Different from the original MLEFGN, we set $N = 2$, $P = 4$, and $C = 32$ in this model in order to reduce training time. Then, we replace the Edge-Net with the edges extracted by different edge operators, e.g., Prewitt, Sobel, and Canny. This means that the Edge-Net will not exist in the model; thus, the DBs in the dual-path network will directly receive the noisy image and the extracted edges as input. Meanwhile, in order to show the effect of edge guidance, we also provide the results of MLN mentioned in Section VI-A [2)]. In Fig. 14, we provide the PSNR results of the reconstructed images guided by these edge detectors. According to the figure, we draw the following conclusions: 1) model performance will be affected by the quality of the extracted edges; 2) the higher the quality of the edges, the better the model performance; 3) effective edges can improve model performance, while incorrect edges will reduce model performance (Roberts) due to the remained noise and false edges; and 4) with the guidance of edges provided by the Edge-Net, the model achieves the best results. In addition, Fig. 15 shows the visual comparison of the denoised images reconstructed by these models. It is obvious that although other models also achieve good PSNR results, the reconstructed denoised images still contain a lot of artifacts and wrong edges. In contrast, our MLEFGN can reconstruct high-quality denoised images with sharp edges and accurate textures.

All the aforementioned experiments prove the effectiveness of the proposed Edge-Net. Meanwhile, these ablation studies indicate that the quality of edges and edge guidance mechanisms are extremely important in image denoising.

B. Study of the Residual Group

As shown in Fig. 4, in Stage-II and Stage-III, we use the RGs for feature extraction and fusion, respectively. RG is an efficient module, which contains N RBs. In order to explore the impact of different numbers of RBs on model performance, we design this ablation experiment. As shown in Fig. 16, we can clearly observe that as N increases, the model performance can be further improved. This means that the performance of MLEFGN can be further improved. However, the purpose of this research is not to purely pursue model performance but to provide a new edge guidance framework for image denoising. Therefore, we set $N = 5$ in the final model to achieve a good balance between model size and performance.

C. Study of the Multilevel Guidance Mechanism

In order to make full use of different levels of image and edge features, we propose a multilevel guidance mechanism in Stage-III. In this part, we aim to explore the effectiveness of this mechanism. To achieve this, we build a new model named edge features guided network (EFGN). EFGN is the modified version of MLEFGN, which makes the following changes: 1) only the last level of image and edge features is fused ($f^3_{\text{fuse}} = f^3_{\text{image}} + f^3_{\text{edge}}$); 2) the IPM is decomposed and moved to the end of the model to ensure that the parameters of the two models are consistent; and 3) the recombined IPM takes f^3_{fuse} as input to reconstruct the final denoised image. In Fig. 17, we provide the performance changes of these two models during training. It is obvious that the curve of our

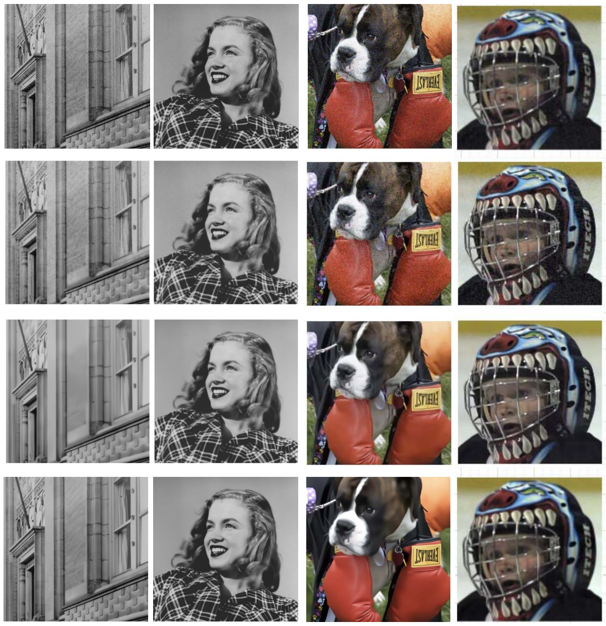


Fig. 18. Real image denoising. Top to bottom: noisy images, denoised images by DnCNN-B, denoised images by FFDNet, and denoised images by our MLEFGN. We set $\sigma = 15$ and $\sigma = 50$ for gray-scale and color images in MLEFGN, respectively. Please zoom in to view details.

MLEFGN (orange line) grows steadily as the training epoch increases. However, the curve of EFGN (blue line) fluctuates seriously, which is difficult to converge. This demonstrates that the introduced multilevel guidance mechanism and IPM can make the fusion of image and edge features more stable so that edge guidance can be fully realized. Therefore, we can draw the conclusion that the proposed multilevel guidance mechanism is effective.

VII. Discussion

A. Exploring on Real Images

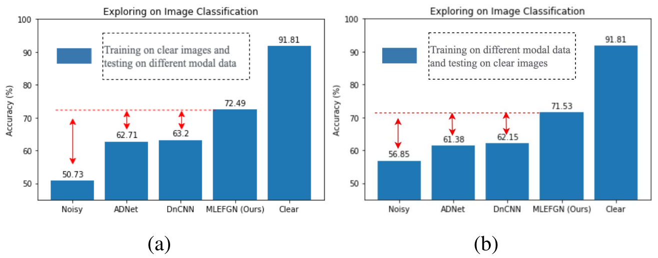
Real image noise comes from multiple sources, such as camera imaging pipelines (shot noise, amplifier noise, and quantization noise), scanning, and lossy compression. Meanwhile, all these noises are usually nonuniform and non-Gaussian. Therefore, the task of real image denoising is more difficult. In this part, real noisy images are used to further assess the practicability of our MLEFGN. Accordingly, we choose RN6 and RN15 as the test data sets, which all have no ground truth. Therefore, the performance only can be evaluated by visual comparison. We choose DnCNN-B and FFDNet for comparison because they are widely accepted as the benchmark for image denoising. In Fig. 18, we provide a visual comparison of these methods. It is obvious that the results of MLEFGN are better than DnCNN-B and close to FFDNet. It is worth noting that MLEFGN is not a blind denoising model; thus, we set $\sigma = 15$ and $\sigma = 50$ for gray-scale and color images, respectively. The choice of noise levels is based on our rough estimation. Different from MLEFGN, DnCNN-B is a blind Gaussian denoising model that sets the range of noise levels to $\sigma \in [0, 55]$, and FFDNet adopts an interactive strategy to handle real noisy images. These are all good strategies for blind image denoising, which can also be introduced into our model to further improve the capability of real image denoising. Recently, some blind or real image denoising models have been proposed, e.g., RIDNet [62], CBDNet [63], and VDNNet [64]. These methods usually use real images as training data sets or introduce specific learning strategies, which can also be introduced *in toto* our model. As discussed in SRRFN [65], the works on simulating degradation models are still meaningful, and an efficient network or framework is also crucial. In future works, we will further explore the performance of our proposed edge guidance framework on real image denoising.

TABLE VII
Accuracy Comparison of Training and Testing Data Sets in Different Modes

Train / Test	Noisy image	Clear image
Noisy image	83.50%	50.73%
Clear image	56.85%	**91.81%**

B. Exploring on High-Level Task

The quality of the denoised images will influence the performance of high-level computer vision tasks, such as image classification and segmentation. In this part, we aim to build a classifier to test the classification accuracy of clean, noisy, and denoised images. Especially, we use the VGG [66] model as our classifier and choose the 102 category flower data set [67] as the data set. First, we divide the data set into *train* and *test* parts. Then, we apply AWGN with noise level $\sigma = 50$ on them to obtain *train_noise* and *test_noise*, respectively. Finally, we use them to train the VGG classifier and test their classification accuracy. In Table VII, we provide accuracy results. According to the table, we can draw the following conclusions: 1) the model can achieve better results when the train and test data sets have the same data distribution and 2) the model achieves the best results when both the train and test data sets are clear images. Therefore, we hope to get clear enough images for image classification. However, the collected images are often accompanied by a lot of noise, which will seriously reduce the classification accuracy. In order to verify that image denoising can improve classification accuracy, we design two sets of experiments: 1) the model trained on clear images and tested on noisy images or the denoised images reconstructed by different denoising methods (ADNet, DNCNN, and our MLEFGN) and 2) the model trained on noisy images or the denoised images and tested on clear images. The accuracy results are presented in Fig. 19(a) and (b), respectively. It is obvious that using the denoised images reconstructed by MLEFGN for training or testing, we can get 72% accuracy, which is nearly 20% higher than the noise images and 10% higher than ADNet and DnCNN. This is a huge improvement, reflecting the fact that MLEFGN can reconstruct clearer images than other denoising methods. Although MLEFGN achieves better results than other methods, we should notice that the classification accuracy is still much lower than clear images (91%). This is because: 1) $\sigma = 50$ is a relatively high noise level, and it will



(a) (b)

Fig. 19. Exploration on image classification (AWGN, noise level: $\sigma = 50$). Model is trained on (a) clear images and tested on different mode data and (b) different mode data and tested on clear images.

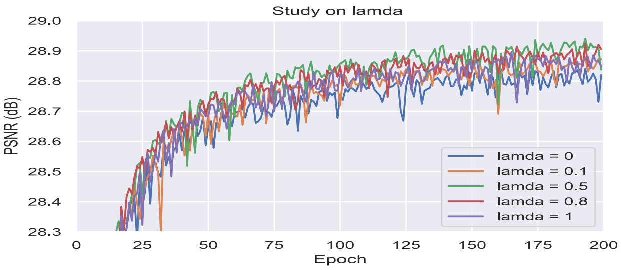


Fig. 20. Study of λ. When $\lambda = 0.5$, the model achieves the best results.

seriously destroy the content of the image so that reconstruct clear and accurate images is a difficult task and 2) the data set itself contains some real noise, making the denoising task more difficult. Unquestionably, the performance of current denoising methods can be further improved. In future work, we will combine the feedback from high-level tasks (e.g., image classification and image segmentation) to further improve the denoising effect.

C. Exploring the Hyperparameter λ

The total loss of MLEFGN consists of two parts, a content loss, and an edge loss. Among them, we introduce the hyperparameter λ to control the proportion of the edge loss. If λ is too small, it is difficult to play the role of edge guidance. In contrast, if λ is too big, the reconstructed images will be over-sharpened. In order to found a better λ, we design an ablation study in Fig. 20. It is obvious that MLEFGN is robust to the choice of λ, and the model achieves the best performance when $\lambda = 0.5$. Although it has achieved good results, we believe that there exists a better λ that can further improve our model. We intend to introduce the dynamic learning strategy to achieve automatic learning of λ, thereby solving this problem.

VIII. Conclusion

In this article, we proposed an MLEFGN for image denoising, which can reconstruct high-quality denoised images with clear and sharp edges. MLEFGN is a new edge guidance framework that integrates edge detection, edge guidance, and image denoising in an end-to-end model. To achieve this, we proposed an Edge-Net to provide edge priors and build a dual-path network to extract the image and edge features. In addition, we introduced the multilevel guidance mechanism to make full use of edge features. Extensive experiments show that our MLEFGN achieves competitive results on AWGN and real noisy image denoising tasks. Although the current model and strategy have achieved great results, further improvements can be made at high noise levels. In the future, we will explore a better edge extraction method and guidance strategy to further improve model performance. Meanwhile, we will verify the effectiveness of the edge guidance mechanism in other image restoration tasks, such as image dehazing, image deblurring, and image enhancement.

References

[1] F. Liu, L. Jiao, and X. Tang, “Task-oriented GAN for PolSAR image classification and clustering,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2707–2719, Sep. 2019.

[2] W. Luo, J. Li, J. Yang, W. Xu, and J. Zhang, “Convolutional sparse autoencoders for image classification,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3289–3294, Jul. 2018.

[3] J. Liu, Y. Wang, Y. Li, J. Fu, J. Li, and H. Lu, “Collaborative deconvolutional neural networks for joint depth estimation and semantic segmentation,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5655–5666, Nov. 2018.

[4] W. Guo and P. Aarabi, “Hair segmentation using heuristically-trained neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 25–36, Jan. 2018.

[5] K.-H. Shih, C.-T. Chiu, J.-A. Lin, and Y.-Y. Bu, “Real-time object detection with reduced region proposal network via multi-feature concatenation,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2164–2173, Jun. 2020.

[6] J. Liu, M. Gong, K. Qin, and P. Zhang, “A deep convolutional coupling network for change detection based on heterogeneous optical and radar images,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 545–559, Mar. 2018.

[7] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proc. 6th Int. Conf. Comput. Vis.*, Jan. 1998, pp. 839–846.

[8] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2005, pp. 60–65.

[9] K. He, J. Sun, and X. Tang, “Guided image filtering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

[10] S. G. Chang, B. Yu, and M. Vetterli, “Adaptive wavelet thresholding for image denoising and compression,” *IEEE Trans. Image Process.*, vol. 9, no. 9, pp. 1532–1546, Sep. 2000.

[11] X.-C. Tai, S. Borok, and J. Hahn, “Image denoising using TV-Stokes equation with an orientation-matching minimization,” in *Proc. Int. Conf. Scale Space Variational Methods Comput. Vis.*, 2009, pp. 490–501.

[12] W. Tansey, J. Thomason, and J. G. Scott, “Maximum-variance total variation denoising for interpretable spatial smoothing,” in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2460–2467.

[13] A. Buades, B. Coll, and J.-M. Morel, “Nonlocal image and movie denoising,” *Int. J. Comput. Vis.*, vol. 76, no. 2, pp. 123–139, Feb. 2008.

[14] W. Dong, L. Zhang, G. Shi, and X. Li, “Nonlocally centralized sparse representation for image restoration,” *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.

[15] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[16] S. Gu, L. Zhang, W. Zuo, and X. Feng, “Weighted nuclear norm minimization with application to image denoising,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2862–2869.

[17] J.-K. Im, D. W. Apley, and G. C. Runger, “Tangent hyperplane kernel principal component analysis for denoising,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 4, pp. 644–656, Apr. 2012.

[18] K. W. Jorgensen and L. K. Hansen, “Model selection for Gaussian kernel PCA denoising,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 163–168, Jan. 2012.

[19] A. Creswell and A. A. Bharath, “Denoising adversarial autoencoders,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 968–984, Apr. 2019.

[20] H. C. Burger, C. J. Schuler, and S. Harmeling, “Image denoising: Can plain neural networks compete with BM3D?” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2392–2399.

[21] J. Mairal, M. Elad, and G. Sapiro, “Sparse representation for color image restoration,” *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[22] Y. Chen, W. Yu, and T. Pock, "On learning optimized reaction diffusion processes for effective image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5261–5269.

[23] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, 2010.

[24] C. Tian, Y. Xu, L. Fei, J. Wang, J. Wen, and N. Luo, "Enhanced CNN for image denoising," *CAAI Trans. Intell. Technol.*, vol. 4, no. 1, pp. 17–23, Mar. 2019.

[25] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep CNN with batch renormalization," *Neural Netw.*, vol. 121, pp. 461–473, Jan. 2020.

[26] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[27] D. Yang and J. Sun, "BM3D-Net: A convolutional neural network for transform-domain collaborative filtering," *IEEE Signal Process. Lett.*, vol. 25, no. 1, pp. 55–59, Jan. 2018.

[28] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[29] W. Guo and W. Yin, "Edge guided reconstruction for compressive imaging," *SIAM J. Imag. Sci.*, vol. 5, no. 3, pp. 809–834, Jan. 2012.

[30] T. N. Canh, K. Q. Dinh, and B. Jeon, "Edge-preserving nonlocal weighting scheme for total variation based compressive sensing recovery," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2014, pp. 1–5.

[31] Y. Lu, J. Zhao, and G. Wang, "Edge-guided dual-modality image reconstruction," *IEEE Access*, vol. 2, pp. 1359–1363, 2014.

[32] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, and J. Jiang, "Edge-enhanced GAN for remote sensing image superresolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5799–5812, Aug. 2019.

[33] S. Chupraphawan and C. A. Ratanamahatana, "Deep convolutional neural network with edge feature for image denoising," in *Proc. Int. Conf. Comput. Inf. Technol.*, 2019, pp. 169–179.

[34] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, 1986. [Online]. Available: https://ieeexplore.ieee.org/document/4767851

[35] J. M. Prewitt, "Object enhancement and extraction," *Picture Process. Psychopictorics*, vol. 10, no. 1, pp. 15–19, 1970.

[36] M. K. Ng, H. Shen, S. Chaudhuri, and A. C. Yau, "Zoom-based super-resolution reconstruction approach using prior total variation," *Opt. Eng.*, vol. 46, no. 12, pp. 127003–1–127003–11, 2007.

[37] Q. Zhou, S. Chen, J. Liu, and X. Tang, "Edge-preserving single image super-resolution," in *Proc. 19th ACM Int. Conf. Multimedia (MM)*, 2011, pp. 1037–1040.

[38] J. Xie, R. S. Feris, and M.-T. Sun, "Edge-guided single depth image super resolution," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 428–438, Jan. 2016.

[39] W. Yang *et al.*, "Deep edge guided recurrent residual learning for image super-resolution," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5895–5907, Dec. 2017.

[40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jul. 2017, p. 3.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[42] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.

[43] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.

[44] X. Yang, Y. Xu, Y. Quan, and H. Ji, "Image denoising via sequential ensemble learning," *IEEE Trans. Image Process.*, vol. 29, pp. 5038–5049, Mar. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9032360

[45] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3929–3938.

[46] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu, "Attention-guided CNN for image denoising," *Neural Netw.*, vol. 124, pp. 117–129, Apr. 2020.

[47] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. Curves Surf.*, 2010, pp. 711–730.

[48] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 860–867.

[49] Rich Franzen. *Kodak Lossless True Color Image Suite*. Accessed: Nov. 15, 1999. [Online]. Available: http://r0k.us/graphics/kodak

[50] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, Jul. 2001, pp. 416–423.

[51] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5197–5206.

[52] M. Lebrun, M. Colom, and J.-M. Morel, "The noise clinic: A blind image denoising algorithm," *Image Process. Line*, vol. 5, pp. 1–54, Jan. 2015.

[53] M. Lebrun, M. Colom, and J.-M. Morel, "The noise clinic: A blind image denoising algorithm," *IPOL J.*, vol. 5, pp. 1–54, 2015. [Online]. Available: http://www.ipol.im/pub/art/2015/125/

[54] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 479–486.

[55] T. Plötz and S. Roth, "Neural nearest neighbors networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1087–1098.

[56] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2802–2810.

[57] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A persistent memory network for image restoration," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4539–4547.

[58] C. Cruz, A. Foi, V. Katkovnik, and K. Egiazarian, "Nonlocality-reinforced convolutional neural networks for image denoising," *IEEE Signal Process. Lett.*, vol. 25, no. 8, pp. 1216–1220, Aug. 2018.

[59] T. N. Canh, K. Q. Dinh, and B. Jeon, "Compressive sensing reconstruction via decomposition," *Signal Process., Image Commun.*, vol. 49, pp. 63–78, Nov. 2016.

[60] W. Guo, J. Qin, and W. Yin, "A new detail-preserving regularization scheme," *SIAM J. Imag. Sci.*, vol. 7, no. 2, pp. 1309–1334, Jan. 2014.

[61] W. Zuo, L. Zhang, C. Song, D. Zhang, and H. Gao, "Gradient histogram estimation and preservation for texture enhanced image denoising," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2459–2472, Jun. 2014.

[62] S. Anwar and N. Barnes, "Real image denoising with feature attention," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3155–3164.

[63] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1712–1722.

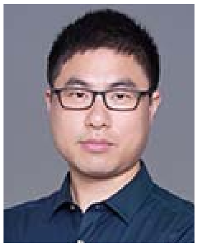
[64] Z. Yue, H. Yong, Q. Zhao, D. Meng, and L. Zhang, "Variational denoising network: Toward blind noise modeling and removal," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1688–1699.

[65] J. Li, Y. Yuan, K. Mei, and F. Fang, "Lightweight and accurate recursive fractal network for image super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3814–3823.

[66] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[67] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. 6th Indian Conf. Comput. Vis., Graph. Image Process.*, Dec. 2008, pp. 722–729.

[68] H. Liu, Z. Fu, J. Han, L. Shao, S. Hou, and Y. Chu, "Single image super-resolution using multi-scale deep encoder–decoder with phase congruency edge map guidance," *Inf. Sci.*, vol. 473, pp. 44–58, 2019.



Faming Fang received the Ph.D. degree in computer science from East China Normal University, Shanghai, China, in 2013.

He is currently an Associate Professor with the School of Computer Science and Technology, East China Normal University. His main research areas are image processing using variational methods, partial differential equations (PDEs), and deep learning.

Juncheng Li received the B.Sc. degree in computer science from Jiangxi Normal University, Nanchang, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, East China Normal University, Shanghai, China.

His main research interests are image processing, machine learning, and deep learning.

Yiting Yuan received the B.Sc. degree in computer science from China University Mining and Technology, Jiangsu, China, in 2017. She is currently pursuing the master's degree with the School of Computer Science and Technology, East China Normal University, Shanghai, China.

Her research interests include image stitching and image fusion.

Tieyong Zeng received the B.S. degree from Peking University, Beijing, China, in 2000, the M.S. degree from École Polytechnique, Palaiseau, France, in 2004, and the Ph.D. degree from the University of Paris XIII, Paris, France, in 2007.

He worked as a Post-Doctoral Researcher with ENS de Cachan, Cachan, France, from 2007 to 2008, and an Assistant/Associate Professor with Hong Kong Baptist University, Hong Kong, from 2008 to 2018. He is currently an Associate Professor with the Department of Mathematics, The Chinese University of Hong Kong, Hong Kong. His research interests are image processing, machine learning, and scientific computing.

Guixu Zhang received the Ph.D. degree from the Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou, China, in 1998.

He is currently a Professor with the Department of Computer Science and Technology, East China Normal University, Shanghai, China. His research interests include hyperspectral remote sensing, image processing, artificial intelligence, machine learning, and scientific computing.