

# Multi-level 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 the 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 Multi-level Edge Features Guided Network (MLEFGN). Firstly, 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 image and edge features, respectively. Finally, we introduce a multi-level 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-Net, edge guidance, image denoising.

## I. INTRODUCTION

**I**MAGE denoising is an extremely hot topic in computer vision, which aims to reconstruct a clean image from the noisy one (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 as a challenging inverse problem. In order to elegantly solve the problem of image denoising, we

The work of Faming Fang was supported 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 by the NSFC under Grant 11671002, in part by the CUHK Start-Up, and in part by the CUHK DAG under Grant 4053342, Grant 4053405, Grant RGC 14300219, RGC 14302920, and Grant NSFC/RGC N\_CUHK 415/19. The work of Guixu Zhang was supported by the Key Project of the National Natural Science Foundation of China under Grant 61731009.

F. Fang, J. Li, Y. Yuan, and G. Zhang are with the Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai, China, and also with the school of Computer Science and Technology, East China Normal University, Shanghai, China. (E-mail: cvjunchengli@gmail.com, {fmfang, gxzhang}@cs.ecnu.edu.cn)

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



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

aim to build a flexible approach with the following desirable properties: (a) it is able to perform image denoising in an end-to-end manner; (b) it is efficient and can achieve superior performance; (c) it can integrate image priors to accelerate model convergence and improve the visual effects of the denoised images.

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: (i) spatial filtering methods, such as bilateral filters [7], nonlocal means filters [8], and guided filters [9]; (ii) model-based methods, such as total variation (TV) approaches [11], [12], non-local self-similarity (NSS) models [13], and sparse dictionary learning models [14]; (iii) wavelet transform-based methods [10]. At that time, the block matching 3D 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 huge amount of calculation and low efficiency; (2) these methods usually use hand-designed priors like sparsity [21] or nonlocal self-similarity [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 results with the blossom of deep convolutional neural networks (CNNs). The success of 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 Gaussian image denoising, which took advantage of batch normalization and residual learning to achieve competitive denoising results. The BM3D-Net [27] is a non-local based network that introduced BM3D into CNN by wavelet shrink-

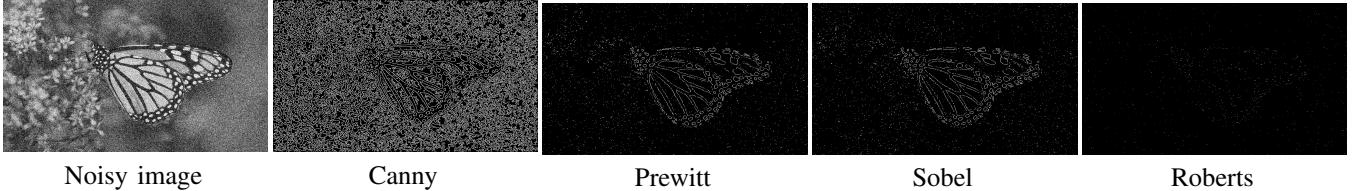


Fig. 2. The results of different edge detectors acting on a noisy image (noise level:  $\sigma = 50$ ). Obviously, these edge detectors are extremely sensitive to noise.

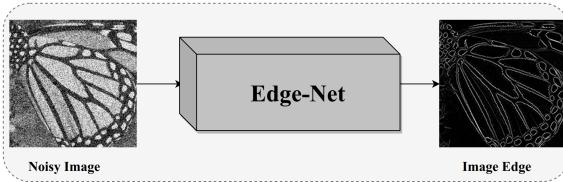


Fig. 3. The principle of edge detection. The Edge-Net is used to reconstruct clear image edges from the noisy image.

age. 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 the 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 total variation (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 researches introduced edges into CNN model to guide image reconstruction or enhance the reconstructed image. For instance, Yang et al. [32] proposed an Edge Guided Recurrent Residual Network (DEGRRN) for single-image super-resolution. Chupraphawan et al. [33] proposed a Deep Convolutional neural network 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 Multi-level Edge Features Guided Network (MLEFGN) for image denoising. Specially, 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 multi-level edge features guidance mechanism is introduced to reconstruct the denoised image.

In summary, the main contributions of this paper are:

(i) 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.

(ii) We propose an edge reconstruction network (Edge-Net). Edge-Net is the first CNN model that can directly reconstruct clear edges from the noisy observation.

(iii) We propose a Multi-level Edge Features Guided Network (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.

**Image Edges 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 (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 reconstruction network (Edge-Net) to directly predict the clear and accurate image edges from the noisy image.

**Edge-guided Image Reconstruction:** In the past few decades, prior-guided image reconstruction methods have received great attention. For example, the total variation (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

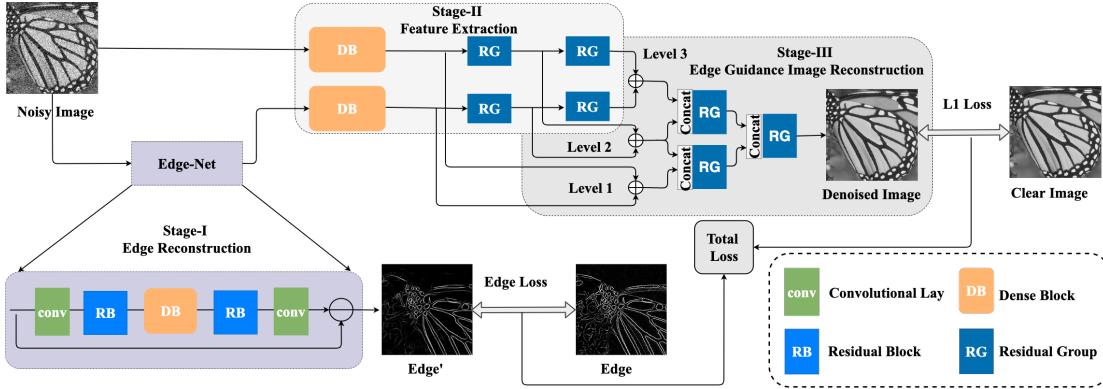


Fig. 4. The complete architecture of our proposed Multi-level Edge Features Guided Network (MLEFGN). The green, orange, sky blue, navy blue, and purple blocks represent the convolutional layer, dense block, residual block, residual group, and EdgeNet, respectively.

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 non-local structure and histogram of the natural image in the gradient domain for compressive sensing recovery; Guo et al. [29] proposed an edge guided compressive sensing reconstruction approach, which can recover high-quality images from fewer measurements. Recently, some researches introduced image edges into the CNN model to further improve the quality of reconstructed images. For instance, Yang et al. [39] proposed an edge guided recurrent residual network (DEGRRN) for single image super-resolution (SISR); Liu et al. presented a phase congruency edge map guided multi-scale deep encoder-decoder network for SISR; Chupraphawan [33] designed a deep convolutional neural network with edge feature (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 exist 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, Prewitt, and so on) on the degraded image. This is because existing edge detectors (whether binarized or non-binarized edge detector) are extremely sensitive to other factors (e.g., noise, 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. MULTI-LEVEL EDGE FEATURES GUIDED NETWORK

The degradation model of image denoising can be formulated as  $y = x + v$ , where  $y$  is the noisy image obtained from the clear image  $x$ , and  $v$  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 datasets. That is, given a training dataset  $\{x^i, y^i\}_{i=1}^M$ , we need to solve

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

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

Following previous works, we also consider this as a learning problem and propose a Multi-level Edge Features Guided Network (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 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 multi-level 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

$$I'_{edge} = E(I_{noisy}), \quad (2)$$

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 [40] and two residual groups to extract the corresponding features

$$f_{image} = F_{Top}(I_{noisy}), \quad (3)$$

$$f_{edge} = F_{Bottom}(I'_{edge}), \quad (4)$$

where  $F_{Top}(\cdot)$  and  $F_{Bottom}(\cdot)$  denote the corresponding path network,  $f_{image}$  and  $f_{edge}$  represent the extracted image low-frequency 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

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

where  $F_{EGM}(\cdot)$  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 which only learn the mapping between noisy and clear images, we need to ensure the accuracy of the reconstructed edges. Hence, given a training dataset  $\{I_{noisy}^i, I_{clear}^i, I_{edge}^i\}_{i=1}^M$ , we need to solve

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

where  $\theta$  denotes the parameter set of our MLEFGN,  $\lambda$  is a hyperparameter,  $E(\cdot)$  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 sub-model 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 the following subsections.

### 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 (Fig. 2). To address this problem, we propose an edge reconstruction network (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 dense block (DB). Among them, the convolutional layers are used to change the spatial dimension of images, the residual and dense blocks are used to extract image features. In Fig. 5, (a) and (c) represent the residual block [41] and dense block [40], respectively. Different from the original residual block, 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 feed-forward fashion. Consequently, the  $p$ -th layer receives feature maps of all preceding layers ( $X_0, X_1, \dots, X_{p-1}$ ) as input

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

where  $[X_0, X_1, \dots, X_{p-1}]$  denotes the concatenation of feature maps produced in layers  $0, \dots, p-1$ . Here,  $p \leq P$ , and  $P$  represents the number of layers in the dense block. Meanwhile,  $H_p$  indicates the current convolutional layer, and  $X_p$  represents the output of this layer. Note that dense blocks are known for its powerful feature extraction capabilities. However, this dense connection mechanism will bring a large number of parameters. To solve this issue, a  $1 \times 1$  bottleneck layer is introduced at the tail of the block to reduce model parameters.

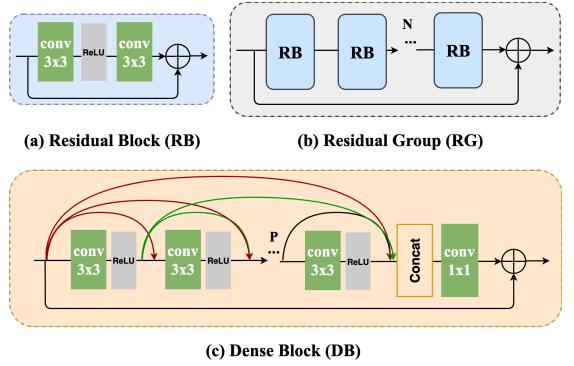


Fig. 5. The architecture of residual block, residual group, and dense block.

Furthermore, we also introduce local residual learning to improve the information flow 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. Specifically, 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  $R(\cdot)$ , the operations of the Edge-Net can be rewritten as

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

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

$$\mathcal{L}_{edge} = \|I'_{edge} - I_{edge}\|_1, \quad (9)$$

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 the 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 hinder their widespread application. Therefore, we recommend retaining more edge information by eliminating the binarization strategy. To achieve this, we propose a new curvature formula for edge detection, which do not use the binarization strategy

$$I_{edge} = \text{div}(u_a, u_b), \quad (10)$$

where  $u_i = \frac{\nabla_i I_{clear}}{\sqrt{1 + |\nabla I_{clear}|^2}}$ ,  $i = \{a, b\}$ ,  $a$  and  $b$  represent horizontal and vertical directions, respectively. Meanwhile,  $\nabla$  and  $\text{div}(\cdot)$  represents the gradient and divergence operation, respectively. Compared to existing edge extractors, our curvature formula can accurately describe the changes in the gradient regions and obtain more high-frequency details (Fig. 6), thus we can get more precise edges as our training labels. In this paper, we use Eq. (10) to obtain the label for training because of its simplicity and feasibility. However, it is worth noting that Eq. (10) can be replaced by any better edge detector.

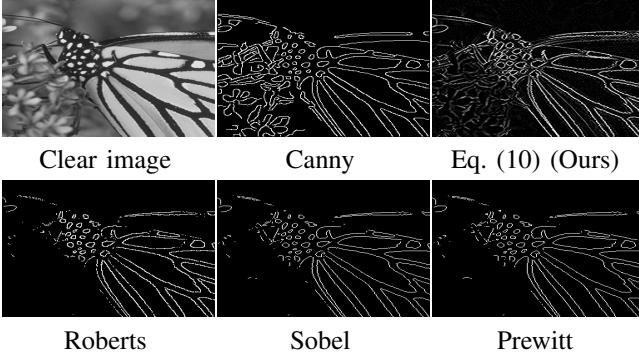


Fig. 6. The results of different edge detectors acting on a clear image. Obviously, our method can obtain more high-frequency details.

This means that one can directly use other non-binarized edge extraction methods or achieve edge extraction by suppressing the binarization operation of the off-the-shelf edge detector (e.g., Sobel, Canny, Prewitt, and so on).

### 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 dense block (DB) and two residual groups (RGs). Among them, the dense block is used to extract features and residual groups are used to further refine the extracted features. As shown in Fig. 5, we use  $N$  residual blocks (RBs) to form a residual group (RG). However, we observe that simply stacking RB will achieve sub-optimal results. Therefore, we apply a long skip connection for global residual learning like 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 multi-level 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  $i$ -th level image and edge features fusion can be defined as

$$f_{\text{fuse}}^i = f_{\text{image}}^i + f_{\text{edge}}^i, \quad (11)$$

where  $f_{\text{image}}^i$  and  $f_{\text{edge}}^i$  denote the image and edge feature maps extracted in the  $i$ -th level ( $i = 1, 2, 3$ ), respectively. After that, we can obtain three outputs  $f_{\text{fuse}}^1$ ,  $f_{\text{fuse}}^2$ , and  $f_{\text{fuse}}^3$ .

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 the clear image. However, we find this method can not 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 ( $f_{\text{fuse}}^1$ ,  $f_{\text{fuse}}^2$ ,  $f_{\text{fuse}}^3$ ) will be sent to the module, and the residual group (RG) will be used to concatenate and fuse the features in adjacent levels:

$$I_{\text{IPM}}^1 = F_{\text{RG}}^1([f_{\text{fuse}}^1, f_{\text{fuse}}^2]), \quad (12)$$

$$I_{\text{IPM}}^2 = F_{\text{RG}}^2([f_{\text{fuse}}^2, f_{\text{fuse}}^3]), \quad (13)$$

$$I_{\text{IPM}}^3 = F_{\text{RG}}^3([I_{\text{IPM}}^1, I_{\text{IPM}}^2]), \quad (14)$$

where  $[ \cdot ]$  is the concatenation operation,  $F_{\text{RG}}^i$  denotes the  $i$ -th RG, and  $I_{\text{IPM}}^i$  represents the corresponding output. Finally, we use  $L_1$  as the content loss to minimize the difference between the denoised and clear images

$$\mathcal{L}_{\text{content}} = \|I'_{\text{clear}} - I_{\text{clear}}\|_1, \quad (15)$$

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

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

Overall, we propose a Multi-level Edge Features Guided Network (MLEFGN) for image denoising. Specifically, 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.

## IV. EXPERIMENTS

### A. Datasets

In this paper, we choose the additive white Gaussian noise (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 datasets. Following the previous works [26], [28], we use Set12, BSD68, and Urban100 to evaluate the performance of MLEFGN on grayscale images, and use 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.

TABLE I

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

Images	C.ma	House	Peppers	Starfish	Monarch	Airplane	Parrot	Lena	Barbara	Boat	Man	Couple	Average
<b>Noise Level</b>													
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 $^6_{7 \times 7}$ [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]	<b>32.81</b>	35.22	<b>33.49</b>	32.17	33.17	<b>31.86</b>	<b>31.96</b>	34.71	32.80	32.57	32.47	32.58	32.98
MLEFGN (Ours)	32.56	<b>35.41</b>	33.42	<b>32.29</b>	<b>33.44</b>	31.82	31.90	<b>34.80</b>	<b>33.05</b>	<b>32.60</b>	<b>32.51</b>	<b>32.67</b>	<b>33.04</b>
<b>Noise Level</b>													
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 $^6_{7 \times 7}$ [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	<b>31.24</b>	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]	<b>30.34</b>	33.41	<b>31.14</b>	29.41	30.39	29.17	<b>29.49</b>	32.61	30.25	30.37	30.08	30.24	30.58
MLEFGN (Ours)	30.29	<b>33.61</b>	30.98	<b>29.66</b>	<b>30.52</b>	<b>29.25</b>	29.46	<b>32.76</b>	30.57	<b>30.40</b>	<b>30.13</b>	<b>30.30</b>	<b>30.66</b>
<b>Noise Level</b>													
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]	<b>28.80</b>	31.92	28.75	27.27	28.13	27.10	27.69	30.73	<b>29.48</b>	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	<b>32.47</b>	<b>29.37</b>	<b>27.77</b>	<b>28.70</b>	<b>27.62</b>	<b>28.03</b>	<b>31.36</b>	29.09	<b>29.00</b>	<b>28.79</b>	<b>28.88</b>	<b>29.15</b>
<b>Noise Level</b>													
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	<b>25.77</b>	26.88	25.90	<b>26.58</b>	29.68	26.48	27.32	27.30	27.07	27.32
ADNet [46]	27.31	30.59	<b>27.69</b>	25.70	26.90	25.88	26.56	29.59	26.64	27.35	27.17	27.07	27.37
MLEFGN (Ours)	27.15	<b>31.00</b>	27.63	<b>25.77</b>	<b>27.01</b>	<b>26.05</b>	26.56	<b>29.85</b>	<b>27.37</b>	<b>27.40</b>	<b>27.32</b>	<b>27.35</b>	<b>27.54</b>

### B. Implementation Details

**Training setting:** Before training, we first apply Eq. (10) on the training dataset 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 grayscale 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 grayscale images and  $S = 64$  for color images) as inputs and the learning rate is initialized as  $10^{-4}$ . In addition, MLEFGN is implemented with PyTorch framework and updated with the Adam optimizer.

**Model setting:** In this paper, we propose a multi-level edge features guided network (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  $\mathcal{L}_{edge}$  according to plenty of experiments. In the final model, each residual group (RG) contains  $5$  ( $N = 5$ ) residual blocks (RBs), each dense block (DB) contains  $8$  ( $P = 8$ ) convolutional layers, and each layer contains  $64$  ( $C = 64$ ) channels. Meanwhile, the kernel sizes of all the convolutional layers are all set as  $3 \times 3$  except for the bottleneck layer, whose kernel size is  $1 \times 1$ .

TABLE II  
THE AVERAGE PSNR (DB) RESULTS OF DIFFERENT IMAGE DENOISING METHODS ON GRayscale IMAGES (**BSD68**) WITH NOISE LEVELS  $\sigma=15, 25, 35$ , AND  $50$ .

Method	$\sigma=15$	$\sigma=25$	$\sigma=35$	$\sigma=50$
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
NLED $^6_{7 \times 7}$ [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 <sup>3</sup> Net [55]	N/A	29.30	N/A	<b>26.39</b>
MLEFGN (Ours)	<b>31.81</b>	<b>29.34</b>	<b>27.85</b>	<b>26.39</b>

### 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 $^6_{7 \times 7}$  [44], NN3D [58], WNNM [16], IRCNN [45], DnCNN [26], FFDNet [28], ADNet [46], and N<sup>3</sup>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.

**PSNR Results on Grayscale Images:** In TABLES I, II, and IV, we report the average PSNR results of different

TABLE III

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

Method	Kodak24 [49]				CBSD68 [50]				Urban100 [51]			
	$\sigma=10$	$\sigma=30$	$\sigma=50$	$\sigma=70$	$\sigma=10$	$\sigma=30$	$\sigma=50$	$\sigma=70$	$\sigma=10$	$\sigma=30$	$\sigma=50$	$\sigma=70$
Noise Level												
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)	<b>37.04</b>	<b>31.67</b>	<b>29.38</b>	<b>27.94</b>	<b>36.37</b>	<b>30.56</b>	<b>28.21</b>	<b>26.75</b>	<b>36.42</b>	<b>31.32</b>	<b>28.92</b>	<b>27.28</b>

TABLE IV

THE AVERAGE PSNR (DB) RESULTS OF DIFFERENT IMAGE DENOISING METHODS ON GRAYSCALE IMAGES (**URBAN100**) WITH NOISE LEVELS  $\sigma=15, 25$ , AND  $50$ .

Method	$\sigma=15$	$\sigma=25$	$\sigma=50$
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 <sup>3</sup> Net [55]	N/A	30.19	26.82
WNNM [16]	32.97	30.39	26.83
MLEFGN (Ours)	<b>33.21</b>	<b>30.64</b>	<b>27.22</b>

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<sup>3</sup>Net are the most classic and famous CNN-based methods, ADNet and NLED<sub>7×7</sub><sup>6</sup> are the latest methods proposed in 2020. According to these tables, we can clearly observe that our MLEFGN achieves the best results on all datasets. Compared with the classic CNN-based denoising method (e.g., DnCNN, FFDNet), the performance of MLEFGN has been significantly improved. In addition, the results of our MLEFGN are slightly better than N<sup>3</sup>Net on BSD68. It is worth noting that MLEFGN achieves excellent performance on Urban100, which is 0.4dB higher than N<sup>3</sup>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 grayscale images.

**PSNR Results on Color Images:** As for color image denoising, we employ three datasets, namely Kodak24 [49], CBSD68 [50], and Urban100 [51]. Among them, CBSD68 is the corresponding color version of the grayscale BSD68 database. In TABLE III, we provide the average results of different image denoising methods on these three datasets. Different from grayscale 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. Obviously, our MLEFGN achieves the best results on all datasets. This further demonstrates the

effectiveness and robustness of MLEFGN, which can achieve excellent performance in different color spaces, noise levels, and datasets.

**Visual Comparisons:** In Figs. 7 and 8, we provide the visual results of the denoised images with noise level 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 remain 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 is a lot of noises and artifacts under the eaves, and the face is blurred and unclear. In contrast, our MLEFGN can reconstruct high-quality 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 grayscale 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 image and edge features effectively; (3) the introduced multi-level 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

Image edges are the key components of high-frequency details. Therefore, many researchers introduced it into their methods for image reconstruction. Among them, we found a denoising method that introduces image edges into CNN model, which named DCEF [33]. DCEF was proposed by Chupraphawan et al. in 2019 and the method can be divided into three steps: (a) extracting image edges from the noisy input using Canny detector; (b) adding the noisy image and the extracted edges to obtain the new input; (c) building a CNN model to reconstruct the final denoised image. Different from this method, we aim to explore a new edge guidance

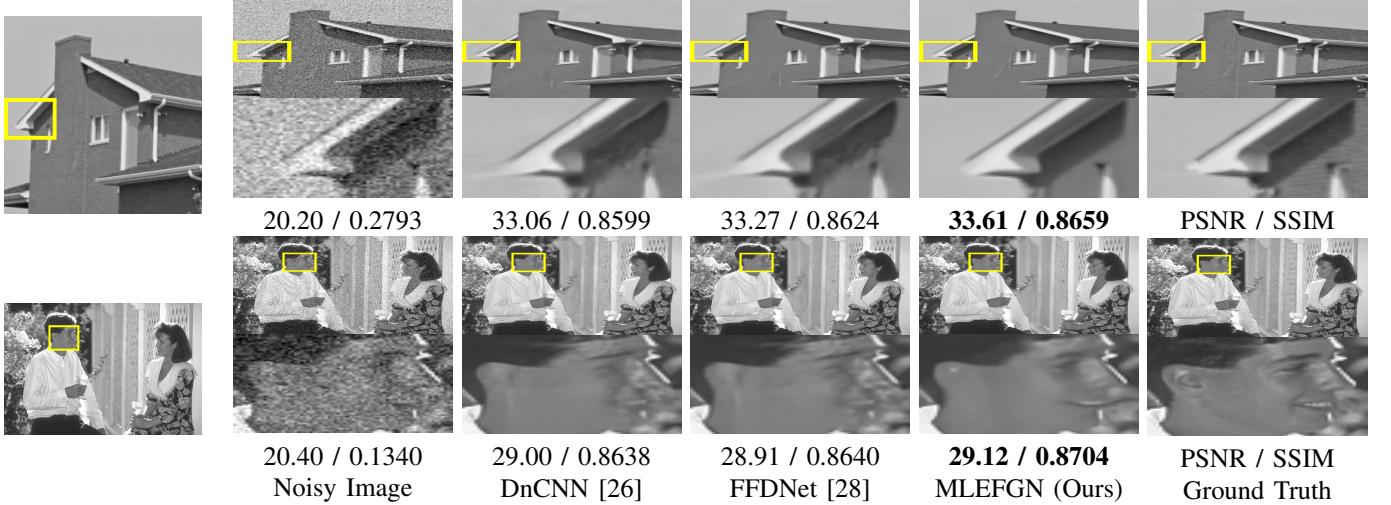
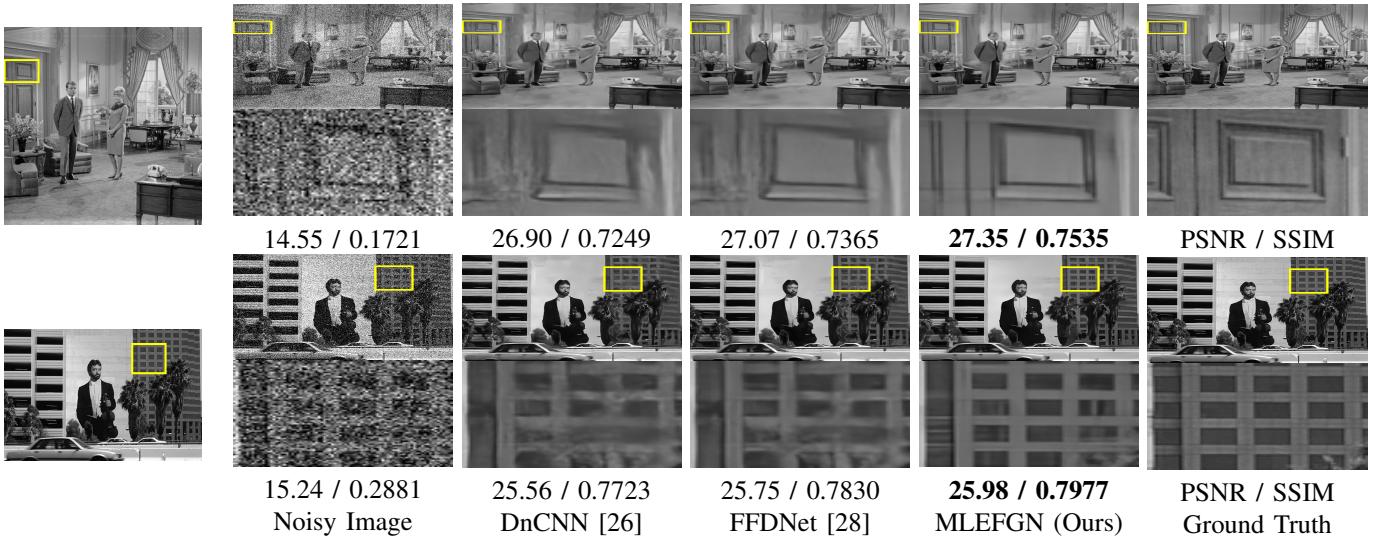
Fig. 7. Visual comparison of MLEFGN with DnCNN [26] and FFDNet [28] on grayscale images (noise level:  $\sigma = 25$ ).Fig. 8. Visual comparison of MLEFGN with DnCNN [26] and FFDNet [28] on grayscale images (noise level:  $\sigma = 50$ ).

TABLE V  
PSNR (dB) COMPARISON WITH DCEF ON SET12 AND BSD68.

Datasets	Set12			BSD68		
	$\sigma=15$	$\sigma=25$	$\sigma=50$	$\sigma=15$	$\sigma=25$	$\sigma=50$
Noisy Level	32.65	33.33	26.92	31.50	29.14	26.05
DCEF	<b>33.04</b>	<b>33.66</b>	<b>27.54</b>	<b>31.81</b>	<b>29.34</b>	<b>26.39</b>
MLEFGN (Ours)						

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. Obviously, 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.

### 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 due to small textural variation cannot be well differentiated from noise [59], [61]. In Figs. 7 and 8, we have provided the 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 Means 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.

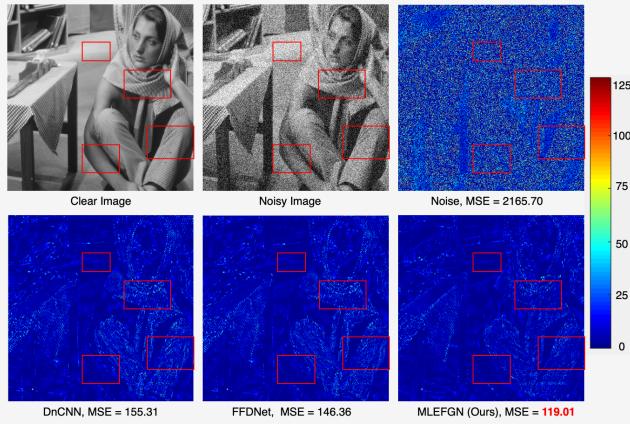


Fig. 9. Recovery performance comparison (noise level:  $\sigma=50$ ). Please zoom in to view details.

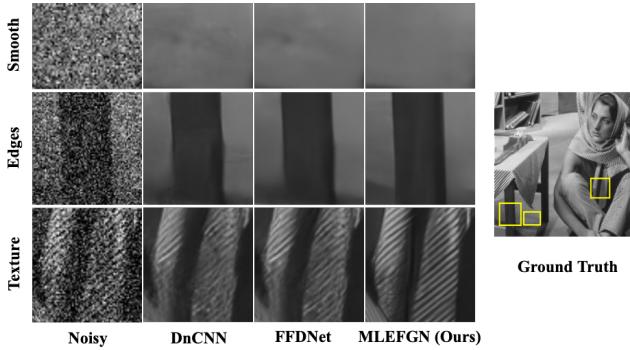


Fig. 10. Image region visualization (noise level:  $\sigma=50$ ).

(2). In Fig. 10, we provide detailed visualization results of image region including smooth, edges, and texture. Obviously, 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 grayscale 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: (1) the PSNR gap between BM3D and CNN-based methods (DnCNN, FFDNet, and MLEFGN) further increases under high noise level; (2) at relatively low noise levels, the performance of DnCNN is comparable to FFDNet, or even better than it. But 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  $\sigma = 75$ . Obviously, 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 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 excellent visual effect, which have 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 region can also be observed in our results. This might be caused by the loss 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 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 high noise level image denoising task and commit to proposing a more effective edge reconstruction model and image denoising model.

TABLE VI  
PSNR COMPARISON ON GRayscale IMAGES (NOISE LEVEL:  $\sigma=75$ ).

Noise Level $\sigma=75$				
Methods	BM3D	DnCNN	FFDNet	MLEFGN (Ours)
Set12	24.91	25.20	25.49	<b>25.69</b>
BSD68	24.21	24.64	24.79	<b>24.85</b>



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

### VI. ABLATION STUDY

In this paper, we propose a Multi-level Edge Features Guided Network for image denoising. In order to study the effectiveness of the proposed method, we provide a series of ablation studies in this section.

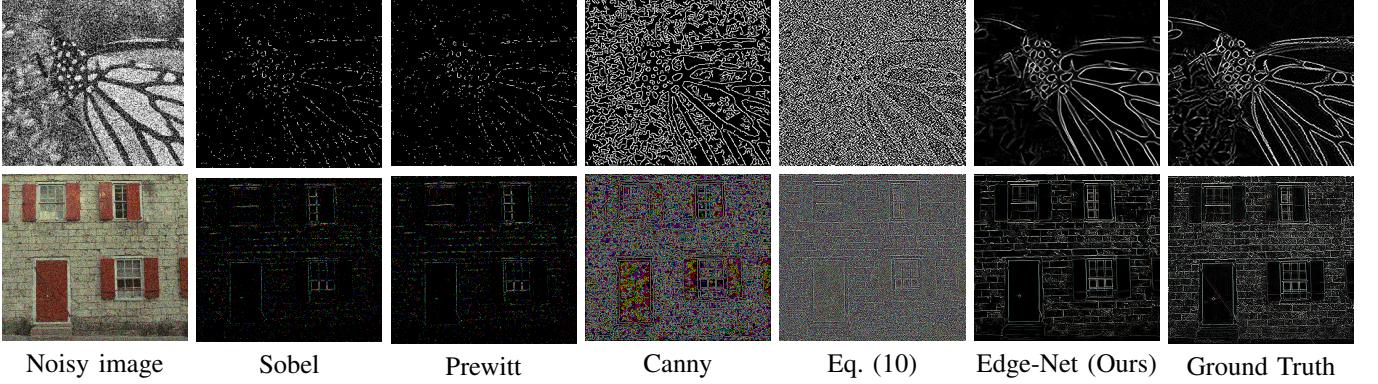


Fig. 12. Visual comparison of image edges extracted by different methods. Noise levels of grayscale and color images are set to  $\sigma = 75$  and 50, respectively.

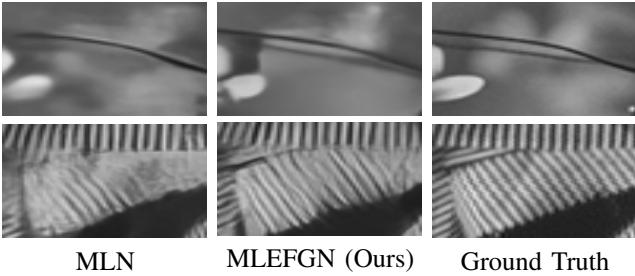


Fig. 13. Visual comparison of MLN and MLEFGN (noise level  $\sigma = 75$ ).

#### 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.

(i). 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 Eq. (10), and our Edge-Net. Obviously, 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 Eq. (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 to other methods, image edges reconstructed by the Edge-Net are amazing. This fully illustrates the effectiveness of Edge-Net in edge extraction.

(ii). 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 Multi-level Network (MLN). MLN uses the framework of MLEFGN as the backbone

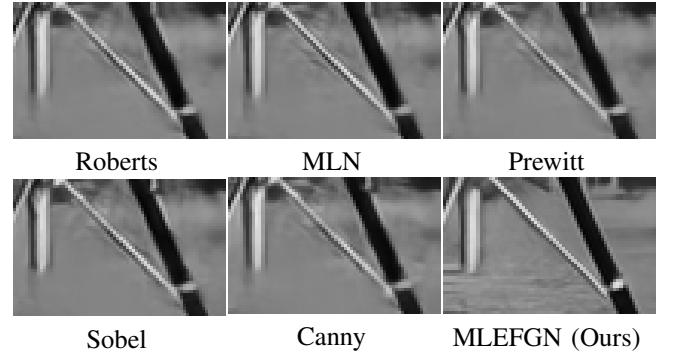


Fig. 14. Visual comparison of the denoised images with  $\sigma = 50$ .

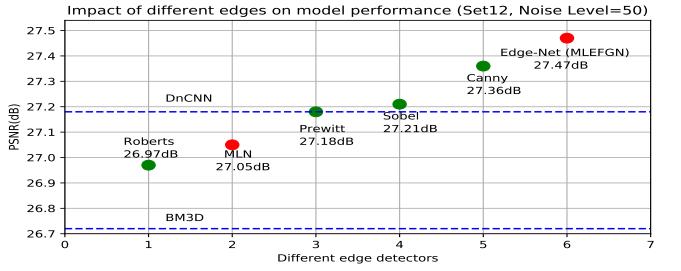


Fig. 15. Study of different edge detectors on Set12 (noise level:  $\sigma = 50$ ).

and removes the Edge-Net in the new network. Therefore, both paths of the network in MLN receives 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.

(iii). All the aforementioned experiments demonstrate the effectiveness of the edge guidance mechanism. In this 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 original

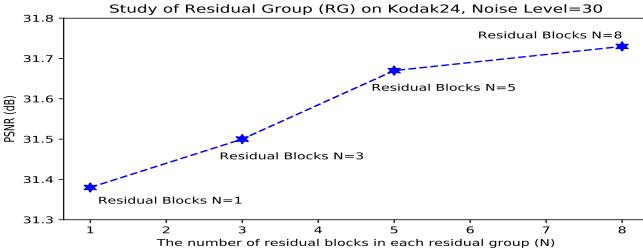


Fig. 16. Study of residual group on Kodak24 (noise level:  $\sigma = 30$ ). We set  $N = 1, 3, 5$ , and 8 in each residual group, respectively.

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 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 (ii). In Fig. 15, we provide the PSNR results of the reconstructed images guided by these edge detectors. According to the figure, we draw 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; (4) with the guidance of edges provided by the Edge-Net, the model achieves the best results. In addition, Fig. 14 shows the visual comparison of the denoised images reconstructed by these models. Obviously, 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 mechanism 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 residual groups (RGs) for feature extraction and fusion, respectively. RG is an efficient module, which contains  $N$  residual blocks. In order to explore the impact of different numbers of residual blocks 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.

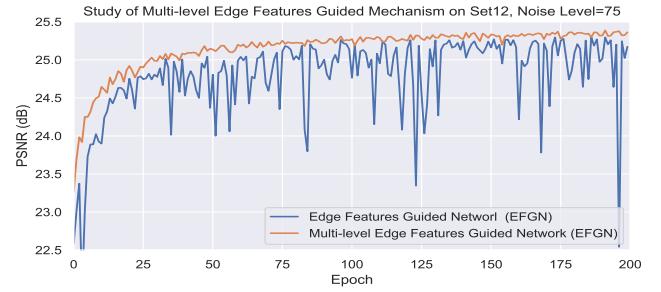


Fig. 17. Study of the Multi-level Guidance Mechanism (noise level  $\sigma = 75$ ).

### C. Study of the Multi-level Guidance Mechanism

In order to make full use of different levels of image and edge features, we propose a multi-level 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 are fused ( $f_{\text{fuse}}^3 = f_{\text{image}}^3 + f_{\text{edge}}^3$ ); (2) the Inverted Pyramid Module (IPM) is decomposed and moved to the end of the model to ensure that the parameters of the two models are consistent; (3) the recombined IPM takes the  $f_{\text{fuse}}^3$  as input to reconstruct the final denoised image. In Fig. 17, we provide the performance changes of these two models during training. Obviously, the curve of our 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 multi-level 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 multi-level 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 of these noises are usually non-uniform 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 datasets, 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 the visual comparison of these methods. Obviously, 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 grayscale 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



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

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 datasets or introduce specific learning strategies, which can also be introduced into 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.

### 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. Specifically, we use the VGG [66] model as our classifier and choose the 102 category flower dataset [67] as the dataset. Firstly, we divide the dataset 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 the accuracy results. According to the table, we can draw the following conclusions: (i) the model can achieve better results when the train and test datasets have the same data distribution; (ii) the model achieves the best results when both the train and test datasets are clear images. Therefore, we hope to get clear enough

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%	<b>91.81%</b>

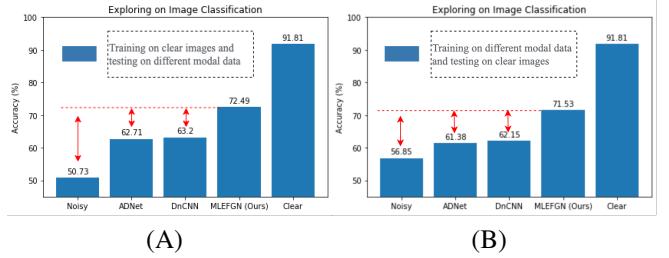


Fig. 19. Exploration on image classification (AWGN, noise level  $\sigma = 50$ ).

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: (a) 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); (b) the model trained on noisy images and tested on clear images. The accuracy results are presented in Fig. 19 (A) and (B), respectively. Obviously, using the denoised images reconstructed by MLEFGN for training or testing 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 (i)  $\sigma = 50$  is a relatively high noise level, it will seriously destroy the content of the image so that reconstruct clear and accurate images is a difficult task; (ii) the dataset 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, image segmentation) to further improve the denoising effect.

### C. Exploring the Hyperparameter $\lambda$

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

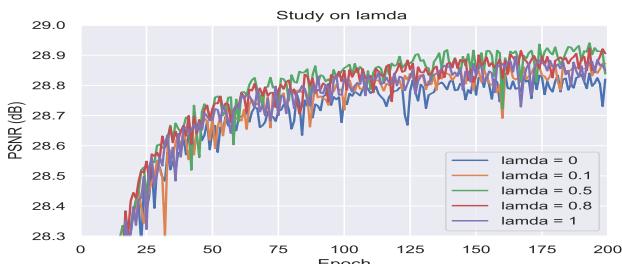


Fig. 20. Study of  $\lambda$ . When  $\lambda = 0.5$ , the model achieves the best results.

intend to introduce the dynamic learning strategy to achieve automatic learning of  $\lambda$ , thereby solving this problem.

## VIII. CONCLUSION

In this paper, we proposed a Multi-level Edge Features Guided Network (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 reconstruction network (Edge-Net) to provide edge priors and build a dual-path network to extract image and edge features. In addition, we introduced the multi-level 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] Fang Liu, Licheng Jiao, and Xu Tang. Task-oriented gan for polar image classification and clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2707–2719, 2019.
- [2] Wei Luo, Jun Li, Jian Yang, Wei Xu, and Jian Zhang. Convolutional sparse autoencoders for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):3289–3294, 2017.
- [3] Jing Liu, Yuhang Wang, Yong Li, Jun Fu, Jiangyun Li, and Hanqing Lu. Collaborative deconvolutional neural networks for joint depth estimation and semantic segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 29(11):5655–5666, 2018.
- [4] Wenzhangzhi Guo and Parham Aarabi. Hair segmentation using heuristically-trained neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 29(1):25–36, 2016.
- [5] Kuan-Hung Shih, Ching-Te Chiu, Jiou-Ai Lin, and Yen-Yu Bu. Real-time object detection with reduced region proposal network via multi-feature concatenation. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.
- [6] Jia Liu, Maoguo Gong, Kai Qin, and Puzhao Zhang. A deep convolutional coupling network for change detection based on heterogeneous optical and radar images. *IEEE Transactions on Neural Networks and Learning Systems*, 29(3):545–559, 2016.
- [7] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 839–846, 1998.
- [8] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 60–65, 2005.
- [9] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):1397–1409, 2013.
- [10] S Grace Chang, Bin Yu, and Martin Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 9(9):1532–1546, 2000.
- [11] Xue-Cheng Tai, Sofia Borok, and Jooyoung Hahn. Image denoising using tv-stokes equation with an orientation-matching minimization. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 490–501, 2009.
- [12] Wesley Tansey, Jesse Thomason, and James G Scott. Maximum-variance total variation denoising for interpretable spatial smoothing. pages 2460–2467, 2018.
- [13] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Nonlocal image and movie denoising. *International Journal of Computer Vision*, 76(2):123–139, 2008.
- [14] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013.
- [15] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.
- [16] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2862–2869, 2014.
- [17] Joon-Ku Im, Daniel W Apley, and George C Rung. Tangent hyperplane kernel principal component analysis for denoising. *IEEE Transactions on Neural Networks and Learning Systems*, 23(4):644–656, 2012.
- [18] Kasper Winther Jorgensen and Lars Kai Hansen. Model selection for gaussian kernel pca denoising. *IEEE Transactions on Neural Networks and Learning Systems*, 23(1):163–168, 2011.
- [19] Antonia Creswell and Anil Anthony Bharath. Denoising adversarial autoencoders. *IEEE Transactions on Neural Networks and Learning Systems*, 30(4):968–984, 2018.
- [20] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2392–2399, 2012.
- [21] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2008.
- [22] Yunjin Chen, Wei Yu, and Thomas Pock. On learning optimized reaction diffusion processes for effective image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5261–5269, 2015.
- [23] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.
- [24] Chunwei Tian, Yong Xu, Lunke Fei, Junqian Wang, Jie Wen, and Nan Luo. Enhanced cnn for image denoising. *CAAI Transactions on Intelligence Technology*, 4(1):17–23, 2019.
- [25] Chunwei Tian, Yong Xu, and Wangmeng Zuo. Image denoising using deep cnn with batch renormalization. *Neural Networks*, 2019.
- [26] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.
- [27] Dong Yang and Jian Sun. BM3D-Net: a convolutional neural network for transform-domain collaborative filtering. *IEEE Signal Processing Letters*, 25(1):55–59, 2018.
- [28] Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.
- [29] Weihong Guo and Wotao Yin. Edge guided reconstruction for compressive imaging. *SIAM Journal on Imaging Sciences*, 5(3):809–834, 2012.
- [30] Thuong Nguyen Canh, Khanh Quoc Dinh, and Byeungwoo Jeon. Edge-preserving nonlocal weighting scheme for total variation based compressive sensing recovery. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–5, 2014.
- [31] Yang Lu, Jun Zhao, and Ge Wang. Edge-guided dual-modality image reconstruction. *IEEE Access*, 2(11):1359–1363, 2014.

- [32] Kui Jiang, Zhongyuan Wang, Peng Yi, Guangcheng Wang, Tao Lu, and Junjun Jiang. Edge-enhanced gan for remote sensing image superresolution. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8):5799–5812, 2019.
- [33] Supakorn Chupraphawan and Chotirat Ann Ratanamahatana. Deep convolutional neural network with edge feature for image denoising. In *International Conference on Computing and Information Technology*, pages 169–179, 2019.
- [34] John Canny. A computational approach to edge detection. In *Readings in Computer Vision*, pages 184–203. 1987.
- [35] Judith MS Prewitt. Object enhancement and extraction. *Picture Processing and Psychopictorics*, 10(1):15–19, 1970.
- [36] Michael K Ng, Huanfeng Shen, Subhasis Chaudhuri, and Andy C Yau. Zoom-based super-resolution reconstruction approach using prior total variation. *Optical Engineering*, 46(12):127003–1–127003–11, 2007.
- [37] Qiang Zhou, Shifeng Chen, Jianzhuang Liu, and Xiaou Tang. Edge-preserving single image super-resolution. In *Proceedings of the ACM International Conference on Multimedia*, pages 1037–1040, 2011.
- [38] Jun Xie, Rogerio Schmidt Feris, and Ming-Ting Sun. Edge-guided single depth image super resolution. *IEEE Transactions on Image Processing*, 25(1):428–438, 2016.
- [39] Wenhan Yang, Jiashi Feng, Jianchao Yang, Fang Zhao, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep edge guided recurrent residual learning for image super-resolution. *IEEE Transactions on Image Processing*, 26(12):5895–5907, 2017.
- [40] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 3, 2017.
- [41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [42] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [43] Y. Chen and T Pock. Trainable nonlinear reaction diffusion: a flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 39(6):1256–1272, 2017.
- [44] Xuhui Yang, Yong Xu, Yuhui Quan, and Hui Ji. Image denoising via sequential ensemble learning. *IEEE Transactions on Image Processing*, 29:5038–5049, 2020.
- [45] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017.
- [46] Chunwei Tian, Yong Xu, Zuoyong Li, Wangmeng Zuo, Lunke Fei, and Hong Liu. Attention-guided cnn for image denoising. *Neural Networks*, 124:117–129, 2020.
- [47] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International Conference on Curves and Surfaces*, pages 711–730, 2010.
- [48] S. Roth and M. J. Black. Fields of experts: a framework for learning image priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 860–867, 2005.
- [49] Rich Franzen. Kodak lossless true color image suite. *source: <http://r0k.us/graphics/kodak>*, 4, 1999.
- [50] David Martin, Charless Fowlkes, Doron Tal, Jitendra Malik, et al. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Proceedings of the IEEE International Conference on Computer Vision*, 2001.
- [51] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015.
- [52] Marc Lebrun, Miguel Colom, and Jean Michel Morel. The noise clinic: a blind image denoising algorithm. *Image Processing on Line*, 5:1–54, 2015.
- [53] Marc Lebrun, Miguel Colom, and Jean-Michel Morel. The noise clinic: a blind image denoising algorithm, 2015.
- [54] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *2011 International Conference on Computer Vision*, pages 479–486, 2011.
- [55] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *Advances in Neural Information Processing Systems*, pages 1087–1098, 2018.
- [56] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in Neural Information Processing Systems*, pages 2802–2810, 2016.
- [57] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017.
- [58] Cristovao Cruz, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Nonlocality-reinforced convolutional neural networks for image denoising. *IEEE Signal Processing Letters*, 25(8):1216–1220, 2018.
- [59] Thuong Nguyen Canh, Khanh Quoc Dinh, and Byeungwoo Jeon. Compressive sensing reconstruction via decomposition. *Signal Processing: Image Communication*, 49:63–78, 2016.
- [60] Weihong Guo, Jing Qin, and Wotao Yin. A new detail-preserving regularization scheme. *SIAM journal on imaging sciences*, 7(2):1309–1334, 2014.
- [61] Wangmeng Zuo, Lei Zhang, Chunwei Song, David Zhang, and Huijun Gao. Gradient histogram estimation and preservation for texture enhanced image denoising. *IEEE Transactions on Image Processing*, 23(6):2459–2472, 2014.
- [62] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3155–3164, 2019.
- [63] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1712–1722, 2019.
- [64] Zongsheng Yue, Hongwei Yong, Qian Zhao, Deyu Meng, and Lei Zhang. Variational denoising network: Toward blind noise modeling and removal. In *Advances in Neural Information Processing Systems*, pages 1688–1699, 2019.
- [65] Juncheng Li, Yiting Yuan, Kangfu Mei, and Faming Fang. Lightweight and accurate recursive fractal network for image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019.
- [66] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [67] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008.



**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 area is image processing using the variational methods, 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 a Ph.D. student 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 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** is currently an Associate Professor in the Department of Mathematics, The Chinese University of Hong Kong. He received the B.S. degree from Peking University, Beijing, China, M.S. degree from Ecole Polytechnique, Palaiseau, France, and Ph.D. degree from the University of Paris XIII, Paris, France, in 2000, 2004, and 2007, respectively. He worked as a postdoctoral researcher at ENS de Cachan in 2007-2008 and Assistant/Associate Professor in Hong Kong Baptist University in 2008-2018. 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, and artificial intelligence. His research interests are image processing, machine learning and scientific computing.