IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 15, 2021, accepted January 1, 2022, date of publication January 12, 2022, date of current version January 19, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3142510

Multiscale Recursive Feedback Network for Image Super-Resolution

XIAO CHEN[1,2], (Member, IEEE), AND CHAOWEN SUN[1]

[1]School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China
[2]Jiangsu Atmospheric Environment and Equipment Technology Collaborative Innovation Center, Nanjing University of Information Science and Technology, Nanjing 210044, China

Corresponding author: Xiao Chen (chenxiao@nuist.edu.cn)

This work was supported in part by the 333 High-Level Personnel Training Projects in Jiangsu Province of China, and in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

ABSTRACT Deep learning-based networks have achieved great success in the field of image super-resolution. However, many networks do not fully combine high-level and low-level information, and fuse local and global information. A multiscale recursive feedback network (MSRFN) for image super-resolution is proposed. First, multiscale convolution is integrated into the feedback network to propose multiscale projection units that adaptively capture image features of different scales by driving a multipath information flow. Next, recursive learning is applied to multiscale projection groups composed of up- and down-multiscale projection units to construct a feedback module that exploits high-level information to correct the low-level representation and refines the features in the early layers. Then, global residual learning and local residual feedback were combined to provide more contextual information for the final reconstruction. Experimental results demonstrate that MSRFN can predict more high-frequency details and alleviate the ringing effect and checkerboard artifacts inherently in CNN-based models. Even when the training datasets are relatively small, MSRFN is still superior to most state-of-the-art methods, especially for large scaling factors ($\times 8$).

INDEX TERMS Image super-resolution, feedback, multiscale convolution, large factors, back-projection, residual learning.

I. INTRODUCTION

Super-resolution (SR), an important image processing technology in the field of computer vision, is widely applied in medical imaging [1], security and surveillance [2], satellite remote sensing images [3], image compression [4] and small object detection [5], [6]. It aims to establish a suitable model for converting a low-resolution (LR) image to a high-resolution (HR) image [7]. Because a given LR image may correspond to a series of possible HR images rather than a single unique image, SR is a challenging ill-posed inverse problem. Currently, numerous SR methods have been proposed to address this problem, which are primarily divided into three types: interpolation-based, reconstruction-based, and learning-based methods [8], [9]. The SR model based on deep learning has gained wide attention in recent years owing to its superior reconstruction performance.

The associate editor coordinating the review of this manuscript and approving it for publication was Tony Thomas.

SRCNN [10], [11] is the first network that applies convolutional neural networks (CNNs) to SR, which directly learns the nonlinear mapping from interpolated LR images to HR images in an end-to-end manner. As a simple shallow linear network, its performance is superior to that of most traditional networks, which demonstrates the superiority of CNNs in solving the SR problem. Subsequently, a series of SR algorithms based on the SRCNN were proposed. Depth can provide larger fields and more contextual information as a key factor in deep neural networks. However, two problems were caused by deepening the network, including gradient disappearance/explosion and numerous parameters. To alleviate the gradient problem effectively, researchers have introduced residual learning [12] and succeeded in training deeper networks, including VDSR [13] and EDSR [14]. In addition, dense connections [15] are often employed, which enables networks not only to alleviate the gradient vanishing problem, but also encourage feature reuse, such as SR-DenseNet [16], RDN [17], and DBPN [18]. To reduce the network parameters, some networks, such as DRCN [19], DRRN [20], and

DRFN [21], employ recursive learning to facilitate weight sharing. Owing to these mechanisms, a growing number of algorithms tend to design more complex and deeper networks to obtain a higher peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [22].

The following problems exist in many present networks: First, many SR networks ignore the training difficulty in achieving excellent performance of depth models, resulting in a huge training setting, more training ticks, and more time. For example, DBPN [18] employs a very large training setting, including DIV2K (1,000 2 K resolution images, including 800 training images and 200 evaluation images) [23], Flickr2K [14] (2650 2 K resolution images), and ImageNet [24] dataset (over 14 million). Second, most SR networks learn hierarchical representations of LR images in a feedforward manner, which relies on their limited features. In addition, the pre-processed feedforward networks can only accommodate a single upsampling factor, and they require a large adjustment and retraining each time they migrate to other upsampling factors, which is extremely inflexible. Owing to the lack of feedback, feedforward networks such as DRRN [20] have difficulty with large scaling factors. Although MSRN [25] and LapSRN [26] with feedforward architectures can perform the experience of $\times 8$ enlargement, there is still an improvement in the $\times 8$ reconstruction performance. Third, a few SR studies introduced feedback mechanisms, but they obtained image features at a single scale without taking full use of image features. Due to the inadequate utilization of features, the features gradually disappear in the process of transmission, especially for large factors SR (such as $\times 8$ SR). Networks such as DBPN [18] and SRFBN [27], [28] fail to cope with the drawbacks of single-scale feedback networks and cannot learn feature mapping at multiple context scales.

To solve the above problems, we designed a novel multiscale recursive feedback network (MSRFN). The structure is illustrated in Fig. 1. MSRFN uses much fewer training datasets than DBPN with only 800 images from DIV2K, but it outperforms DBPN even on large scaling factors. Moreover, owing to the introduction of multiscale feedback, the MSRFN can not only learn rich hierarchical feature representations at multiple context scales, but also refine low-level information with high-level information and better represent the mutual relationships between LR-HR image pairs. In addition, the MSRFN can extend to any upscaling factors with only minor adjustments of the network, and it can also provide the flexibility to define and train networks with different depths, which benefits from a modular end-to-end structure. It is more exciting that MSRFN can effectively alleviate the ringing and jaggy effect at the edge structures and produce more competitive SR results, particularly for $\times 8$ enlargement.

The main contributions of our study are as follows.

First, a multiscale projection unit (MSPU) is proposed by incorporating a multiscale convolution kernel into the feedback connection. Different kernel sizes are introduced in each branch to drive the multipath information flow for up- or down-sampling operations. The MSPU can adaptively capture image features at different scales, which are regarded as local multiscale features. In addition, multiscale receptive fields and information sharing performed between different bypasses contribute to the full use of local features. Furthermore, the 1×1 convolution layer is applied to achieve dimensional reduction and cross-channel multiscale feature fusion; it also improves the generalization ability of the network by adding a nonlinear activation to the learning representation of the previous layer. This kind of local multipath learning enhances branch information communication, further increases the receptive field of the network, and improves guide reconstruction.

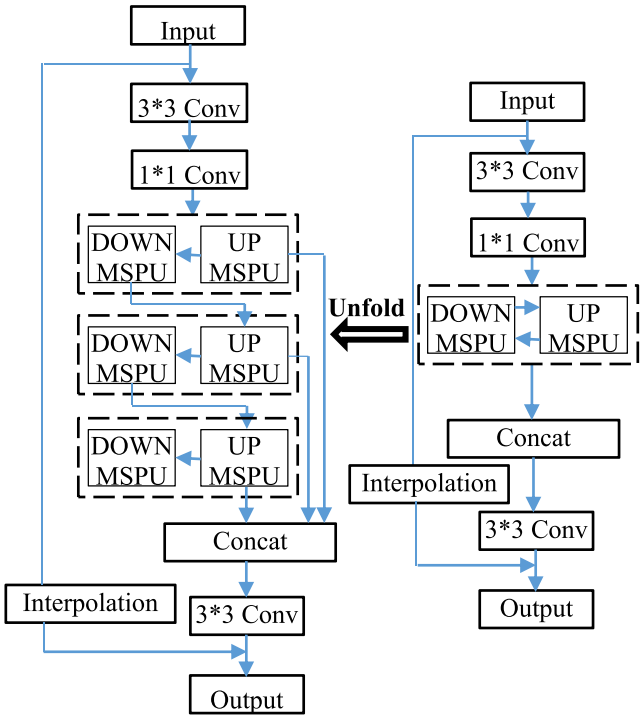

FIGURE 1. The structure of a multiscale recursive feedback network.

Second, in the MSRFN, a pair of up- and down-MSPUs constitutes a multiscale projection group (MSPG) that can realize the local feedback process. MSPG not only generates HR features from the LR input, but also projects them back to the LR spaces. Only one MSPG is used for recursive learning to form a feedback scheme. This kind of top-down work allows previous layers to access useful information from the following layers to refine low-level representation and enrich high-level features. Meanwhile, such a recurrent structure with feedback flow can not only constantly correct the mutual relationship between LR and HR features, but also effectively reduce the network parameters and support a deeper structure. MSRFN has a powerful early reconstruction ability. In addition, the reconstruction module concatenates the HR multiscale feature maps generated by MSPG, which can transfer more abundant elements for the reconstruction of the HR images.

Third, in addition to combining high-level and low-level information, we also combine local and global information by the fusion of local multiscale residual features and global residual features to maximize the utilization of image features and overcome defect features that disappear in the transmission process. On the one hand, MSRFN applies the iterative up- and down-sampling framework to provide local residual feedback for the multiscale projection residuals of MSPG, acquiring finer initial features in the early layers. On the other hand, the global residual skip connection adds the residual image to the global identity mapping from the LR input and helps the network recover the residual between the LR and HR images, greatly reducing the learning difficulty and promoting faster convergence of the network. The combination of local residual feedback and global residual learning helps feature reuse and provides more contextual information for creating SR images.

II. RELATED WORK

A. IMAGE SUPER-RESOLUTION

SR based on deep learning is a trainable data-driven model that can directly learn the non-linear mapping between LR and HR images in an end-to-end manner [11]. The upsampling operation is the key step because it determines how to generate the HR output from the LR input. In view of the different locations of upsampling operations in the model, SR frameworks are divided into four types [29]: pre-upsampling, post-sampling, progressive upsampling, and iterative up- and down-sampling frameworks.

SRCNN is a pioneering framework that adopts a pre-upsampling framework [10], [11]. It is characterized by the completion of the upsampling operation in the pre-processing step. The LR image is enlarged to the target size by the interpolation algorithm, and then the algorithm inputs the interpolation image into the network to establish the mapping relationship with the HR image. Hence, the pre-upsampling SR comes with the defect of poor scalability and difficulty in accommodating any scaling factors with minor adjustments to the network. Although the framework has a lower learning cost owing to its simple structure, it is subject to side effects, including additional noise from coarse images, noise amplification, blurring, and exponentially increasing computational complexity.

To avoid learning most mappings in high-dimensional space, researchers proposed a post-sampling framework that aims to integrate the upsampling layer at the end of the network and directly learn hierarchical feature representation from the LR input. FSRCNN [30] and ESPCN [31] are representative algorithms that improve the computational efficiency and quality of SR images compared with SRCNN. However, because of the limited learnable features in the LR images and the performance of the upsampling operation only once, it is difficult to characterize the complex mapping from the LR to HR images, which greatly increases the learning difficulty for large scaling factors of $\times 4$ and $\times 8$.

To overcome this drawback, LapSRN [26] employs a progressive upsampling framework that uses multiple upsampling modules to progressively reconstruct higher-resolution images. By adding a multi-stage design to the feed-forward network and upsampling the image to a higher resolution at each stage, the complex large-scale factor reconstruction can be decomposed into multiple simple small-scale reconstructions. The scheme of gradually reconstructing multiple SR images of different scales reduces the difficulty of learning and improves the SR performance on large scaling factors. However, its essence is the stacking of a single upsampling network, which is still limited by LR features and subjected to feature underutilization.

To address the above problems, Haris *et al.* innovatively proposed the DBPN algorithm and constructed an iterative up-down sampling framework, which better explores the mutual dependency of LR-HR images by introducing iterative back-projection [18]. The framework alternates up- and down-sampling operations to generate deeper HR features and combines HR images of different depths to produce the results. The authors also introduced a dense connection to improve the network accuracy. This scheme can capture the deep mapping relationship between LR and HR, which improves the reconstruction performance and successfully implements a large scaling factor. However, training this network requires an extremely large dataset and requires more training time and skills. In addition, the network only uses a single-scale convolution kernel, and it is difficult to extract feature information at different scales.

B. NETWORKS

Based on the above four SR frameworks, researchers have applied different network design strategies to construct various SR networks with distinctive characteristics.

DRCN [19] and DRRN [20] are typical models that apply recursive learning to the pre-sampling framework, which stacks multiple identical layers or units in a recursive manner to increase the network depth. Shared weights between recursive modules prompt the network to greatly reduce the introduced parameters and gain a larger receptive field to learn more features. However, recursive learning easily leads to the inherent degradation of deep networks, so it often needs to be combined with residual learning.

Residual learning only learns residual mappings to recover high-frequency information, which avoids direct conversion from LR to HR images. Therefore, it solves the overfitting problem of deep networks and improves the convergence speed. Unlike DRCN, DRRN replaces a recursive layer consisting of a single convolutional layer with a recursive block consisting of several residual units. ResNet [12] and VDSR [13] applied local residual learning and global residual learning to a pre-sampling framework, respectively. Inspired by this, DRRN introduces skip connections in both local residual units and the global network, which reduces the difficulty of training deep models and alleviates the vanishing or exploding gradient problem.

Compared with a simple linear network, the multipath structure designed by DRRN further facilitates learning, in which the residual path can learn high-frequency features, and the identity path transmits rich early image information to the later layers and promotes gradient back propagation. Based on the residual module proposed by Kim [12], Huang introduced dense connections [15]. Unfortunately, this results in an exponential increase in computational complexity and applies a single-size convolution kernel to both the residual and dense modules.

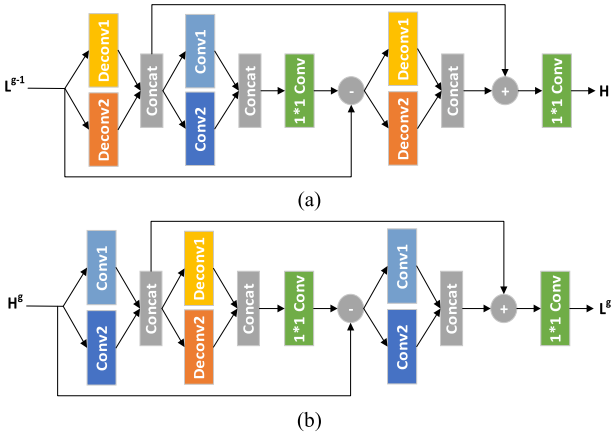

FIGURE 2. Multiscale projection units (MSPUs). (a) Up multiscale projection unit. (b) Down multiscale projection unit.

Multipath learning aims to transfer diverse feature settings through multiple branches of the model and fuse these elements to provide better performance. Under the progressive upsampling framework, LapSRN [26] introduced global multipath learning, which predicts the sub-band residuals with a feature extraction path and reconstructs different scaling HR images through multipath signal flow. Under the post-sampling framework, MSRN [25] introduced local multipath learning, which achieves adaptive detection of image features at different scales using the proposed multiscale feature extraction module. Multiple branches can extract image features of different aspects and continuously exchange information with each other during propagation, further enhancing the ability to learn and extract features.

However, all of the above SR networks learn one-way mapping from LR to HR in a feed-forward manner. This feed-forward structure prevents early layers from effectively utilizing useful information from later layers. Therefore, a few SR algorithms introduce a feedback mechanism that allows the model to convert the output into input to correct the previous state. DBPN [18] proposed an iterative error feedback based on iterative up- and down-sampling layers to enable the network to implement a self-correcting procedure. SRFBN [27] uses hidden states in an RNN with constraints to construct a feedback module to drive the feedback stream and generate powerful high-level representations. However, all of these feedback networks use a single-scale kernel to learn the mapping functions.

To the best of our knowledge, there is no model that integrates local multiscale feature learning into a feedback network for SR.

III. MULTISCALE RECURSIVE FEEDBACK NETWORK

We first focus on the details of MSPU in the network in Section 3.1, which is divided into up and down MSPUs (Fig. 2 (a) and (b)). The feedback module composed of the recursive multiscale projection group (MSPG) is described in Section 3.2. Finally, we divide into three main modules to specifically analyze the MSRFN in Section 3.3 (Fig. 1).

A. MULTISCALE PROJECTION UNIT

Inspired by the idea of GoogleNet [28], we introduce a multiscale convolution kernel in the projection unit, in which we construct two branch networks and apply different scale convolution kernels to different branches to capture image features at different scales. Such local multipath learning is introduced not only to make information sharing between different bypasses, but also to help make full use of the local features. According to the iterative up and down sampling framework, up and down MSPUs are designed for upsampling and downsampling operations, respectively.

1) UP MULTISCALE PROJECTION UNIT

As shown in Fig. 2(a), the up MSPU mainly consists of six steps to map the LR feature, L^{g-1}, to the HR feature, H^g. The details are as follows.

Step 1: Using the previously calculated LR feature map, L^{g-1}, as input, and, respectively, using deconvolution layers with kernels of different sizes, $D^{\uparrow}_{u1}$ and $D^{\uparrow}_{u2}$, to perform upsampling operations on two branches, L^{g-1} is mapped into the HR feature maps, H^g_{u1} and H^g_{u2}.

$$H^g_{u1} = D^{\uparrow}_{u1}(L^{g-1}) \tag{1}$$
$$H^g_{u2} = D^{\uparrow}_{u2}(L^{g-1}) \tag{2}$$

$D^{\uparrow}_{u1}$ and $D^{\uparrow}_{u2}$ represent Deconv1(k_1, n) and Deconv2(k_2, n), respectively; k_1 and k_2 represent the kernel size, and n represents the number of kernels.

Step 2: Concatenating the HR feature maps, H^g_{u1} and H^g_{u2}, and using convolution layers with kernels of different sizes, $C^{\downarrow}_{u1}$ and $C^{\downarrow}_{u2}$, to perform downsampling operations on two branches, the concatenated HR feature map is mapped into the LR feature maps, L^g_{u1} and L^g_{u2}.

$$L^g_{u1} = C^{\downarrow}_{u1}([H^g_{u1}, H^g_{u2}]) \tag{3}$$
$$L^g_{u2} = C^{\downarrow}_{u2}([H^g_{u1}, H^g_{u2}]) \tag{4}$$

$C^{\downarrow}_{u1}$ and $C^{\downarrow}_{u2}$ represent Conv1(k_1, 2n) and Conv2(k_2, 2n), respectively. Here, the number of channels in each branch is 2n.

Step 3: Concatenating the LR feature maps, L^g_{u1} and L^g_{u2}, and using a 1×1 convolution to perform feature pooling and dimension reduction, two LR maps are merged into the LR

feature map, L_u^g, to achieve cross-channel feature fusion.

$$L_u^g = C_u([L_{u1}^g, L_{u2}^g]) \tag{5}$$

C_u represents Conv(1, n), and the number of channels in each branch becomes n from 2n. In addition, the 1×1 convolution adds non-linear activation to the learning representation of the previous layer to improve the expression ability of the network.

Step 4: The residual, e_u^g, is obtained by calculating the difference between the observed LR map, L^{g-1}, and the reconstructed LR map, L_u^g.

$$e_u^g = L_{u1}^g - L^{g-1} \tag{6}$$

Step 5: Two deconvolution layers with kernels of different sizes, $D_{e1}^{\uparrow}$ and $D_{e2}^{\uparrow}$, are used to upsample the residual, e_u^g, on the two branches. The residual in the LR space is mapped to the HR space, producing new residual HR feature maps, H_{e1}^g and H_{e2}^g.

$$H_{e1}^g = D_{e1}^{\uparrow}(e_u^g) \tag{7}$$

$$H_{e2}^g = D_{e2}^{\uparrow}(e_u^g) \tag{8}$$

$D_{e1}^{\uparrow}$ and $D_{e2}^{\uparrow}$ represent Deconv1(k_1, n) and Deconv2(k_2, n), respectively, and the number of channels in each branch is n.

Step 6: Concatenating the residual HR feature maps, H_{e1}^g and H_{e2}^g, and summing with HR feature maps concatenated in step 2, the HR feature map, H^g, obtained by a 1×1 convolution is the final output of the up-MSPU.

$$H^g = C_h\left(\left[H_{u1}^g, H_{u2}^g\right] + \left[H_{e1}^g, H_{e2}^g\right]\right) \tag{9}$$

C_h represents Conv(1, n). The number of channels is 2n after summing, and then Conv(1, n) reduces the number of output channels to n, which is consistent with the input. Both the input and output of the MSPU have the same number of channels. This structure allows multiple MSPUs to be mutually connected.

2) DOWN MULTISCALE PROJECTION UNIT

As shown in Fig. 2(b), a down-multiscale projection unit was defined. Its function is to map the input HR feature, H^g, to the LR feature, L^g. Details are as follows.

Step 1: Taking the HR feature map, H^g, from the previous up MSPU as input, and using two convolution layers, $C_{d1}^{\downarrow}$ and $C_{d2}^{\downarrow}$, with kernels of different sizes to perform upsampling operations on two branches, H^g is mapped into the LR feature maps, L_{d1}^g and L_{d2}^g.

$$L_{d1}^g = C_{d1}^{\downarrow}\left(H^g\right) \tag{10}$$

$$L_{d2}^g = C_{d2}^{\downarrow}\left(H^g\right) \tag{11}$$

$C_{d1}^{\downarrow}$ and $C_{d2}^{\downarrow}$ represent Conv1(k_1, n) and Conv2(k_2, n), respectively, and k_1 and k_2 represent the size of the kernels.

Step 2: Concatenating the LR feature maps, L_{d1}^g and L_{d2}^g, and using deconvolution layers with kernels of different sizes, $D_{d1}^{\uparrow}$ and $D_{d2}^{\uparrow}$, to perform upsampling operations on two branches, the concatenated LR feature map is mapped into the HR feature maps, H_{d1}^g and H_{d2}^g.

$$H_{d1}^g = D_{d1}^{\uparrow}\left(\left[L_{d1}^g, L_{d2}^g\right]\right) \tag{12}$$

$$H_{d2}^g = D_{d2}^{\uparrow}\left(\left[L_{d1}^g, L_{d2}^g\right]\right) \tag{13}$$

$D_{d1}^{\uparrow}$ and $D_{d2}^{\uparrow}$ represent Deconv1(k_1, 2n) and Deconv2(k_2, 2n), respectively. The number of channels in each branch is 2n.

Step 3: The HR feature maps, H_{d1}^g and H_{d2}^g, are concatenated and sent to a 1×1 convolution to obtain the HR feature map, H_d^g.

$$H_d^g = C_d\left(\left[H_{d1}^g, H_{d2}^g\right]\right) \tag{14}$$

C_d represents Conv(1, n), and the number of channels in each branch is changed from 2n to n.

Step 4: The residual, e_d^g, is obtained by calculating the difference between the observed HR map, H_g, and the reconstructed HR map, H_d^g.

$$e_d^g = H_{d1}^g - H^{g-1} \tag{15}$$

Step 5: Two convolution layers with kernels of different sizes, $C_{e1}^{\downarrow}$ and $C_{e2}^{\downarrow}$, are used to downsample the residual, e_d^g, on the two branches. The residual in the HR space is mapped to the LR space, producing new residual LR feature maps, L_{e1}^g and L_{e2}^g.

$$L_{e1}^g = C_{e1}^{\downarrow}\left(e_d^g\right) \tag{16}$$

$$L_{e2}^g = C_{e2}^{\downarrow}\left(e_d^g\right) \tag{17}$$

$C_{e1}^{\downarrow}$ and $C_{e2}^{\downarrow}$ represent Conv1(k_1, n) and Conv2(k_2, n), respectively, and the number of channels in each branch is n.

Step 6: Concatenating the residual LR feature maps, L_{e1}^g and L_{e2}^g, and summing with LR feature maps concatenated in step 2, the LR feature map, L^g, obtained by a 1×1 convolution, is the output of the down-multiscale projection unit.

$$L^g = C_l\left(\left[L_{d1}^g, L_{d2}^g\right] + \left[L_{e1}^g, L_{e2}^g\right]\right) \tag{18}$$

C_l represents Conv(1, n). The number of channels is 2n after summing, and then Conv(1, n) reduces the number of output channels to n, which is the same as the input.

B. RECURSIVE MULTISCALE PROJECTION GROUP

The feedforward structure only maps the rich representation of the input space to the output space, and this one-way mapping is limited to the LR features from the input space. An up MSPU followed by a down MSPU constitutes a multiscale projection group, which can project LR multiscale features to HR space and then back to LR space. Let the output of the previous projection group modulate the input of the next iteration to form feedback. As the feedback flow alternates between the up- and down-sampling processes, the projection residual is fed into the sampling layer, and then local residual feedback is employed to change the solution to form a self-correcting process iteratively. Multiple recurrent MSPGs are considered an efficient iterative process to optimize reconstruction errors

to capture the interdependence between LR and HR images more deeply and enhance the utilization of local features. Significantly, our entire network uses only one MSPG, and recursive learning allows it to be shared among all recursive stages, which greatly increases the network depth without increasing the network capacity. In addition, our network can directly obtain the HR feature output from the MSPG at each stage and then fuse the HR features of different depths from each iteration.

To control the number of parameters and reduce the computational complexity, many network models use a 3×3 convolution to complete the feature mapping. This can avoid the increase in computational cost and the decrease in convergence speed caused by large-scale convolution kernels, but at the expense of a part of the reconstruction performance. However, recursive MSPG implements the iterative utilization of the MPU, which not only greatly promotes the shared weights and reduces the parameters, but also suppresses the limitation that the large-scale kernel brings slow convergence speed and may produce suboptimal results. This allows our network to design large-scale kernels with multibranched structures. Hence, each branch of our MSPU uses a large-scale kernel such as 10×10, which can extract more image features and improve the reconstruction result.

C. NETWORK STRUCTURE

The MSRFN is mainly divided into three components: feature extraction, feedback, and reconstruction modules, as shown in Fig. 1. Significantly, because global residual learning is applied, the entire network takes the original LR image as input and only needs to learn the residual image between the HR image and the interpolated LR image. Here, let conv(f, n) denote the convolution layer, where f is the size of the kernel and n is the number of channels. The introduction of these three modules is as follows:

The original LR image, I_{LR}, is input into the feature-extraction module to produce the initial LR feature map, L^0.

$$L^0 = f_0\left(I_{LR}\right) \tag{19}$$

The feature extraction module is composed of two convolution layers, conv(3, n_0) and conv(1, n). n_0 is the number of channels in the initial LR feature extraction layer. n is the number of input channels in MSPG. It first uses conv(3, n_0) to generate shallow features L^0 with LR image information from the input I_{LR}, and then uses conv(1, n) to reduce the number of channels from n_0 to n.

Subsequently, the initial LR feature map, L^0, flows into the feedback module formed by the recursive MSPG and outputs a series of HR feature maps, H^g.

$$\begin{aligned} &\text{For g in G,} \\ &H^g = f_{FM}^g\left(L^{g-1}\right), \quad 1 \le g \le G, \end{aligned} \tag{20}$$

where G represents the number of MSPGs equivalent to the total recursion time. f_{FM}^g represents the feature mapping process of the MSPG at the g-th stage in the feedback module.

TABLE 1. The settings of input patch sizes and network parameters.

Scale	×2	×3	×4	×8
Input size	60 × 60	50 × 50	40 × 40	20 × 20
f_0^0	Conv (3,1,1)			
f_0^1	Conv (1,1,1)			
Path 1	Conv1(6,2,2)	Conv1(7,3,2)	Conv1(8,4,2)	Conv1(12,8,2)
Path 2	Conv2(8,2,3)	Conv2(9,3,3)	Conv2(10,4,3)	Conv2(14,8,3)
f_{RM}	Conv(3,1,1)			
G	6			
Depth	63			

Note: f_0^0 and f_0^1 refer to the first layer and second convolutional layer in the initial feature extraction stages. Path 1 and Path 2 refer to the two bypasses in the MSPUs, respectively. f_{RM} denotes the last reconstruction layer and G denotes the number of MSPGs. In Conv (K, S, P), K represents the kernel size, S represents the stride, and P represents the padding.

When g is 1, the initial LR feature map L^0 is taken as the input of the first MSPG, and when g is greater than 1, the LR feature map L^{g-1} generated by the previous MSPG is taken as the current input.

The reconstruction module is expressed as follows:

$$I^{Res} = f_{RM}\left(\left[H^1, H^2, \cdots, H^g\right]\right) \tag{21}$$

Here, $\left[H^1, H^2, \cdots, H^g\right]$ represents the deep concatenation of multiple HR feature maps. f_{RB} represents the operation of the reconstruction module, which concatenates a series of HR feature maps generated in the feedback module and then flows across conv(3, 3) to generate a residual image, I^{Res}.

Through the global residual skip connection, the final output SR image can be expressed as

$$I^{SR} = I^{Res} + f_{US}\left(I^{LR}\right) \tag{22}$$

Here, f_{US} represents an upsampling operation with interpolation. According to the given scaling factor, bilinear interpolation is applied to enlarge the original input image I^{LR} to the target size (other interpolation algorithms may also be used, e.g., bicubic interpolation). Then, the interpolation LR image bypassing the main body of the network is transferred to the end of the network and summed with the reconstructed residual image I^{Res} to generate the final image I^{SR}.

As their name implies, different modules play different roles in our deep neural network, and the three major modules constitute our SR framework. Assuming that the number of MSPGs is g, the network contains a total of $(10g + 3)$ layers. Two layers were used for the feature extraction. $(5 + 5) * g$ layers were used for feature mapping in the feedback module, and one layer was used for the final reconstruction. We abstract these modules by defining multiple basic blocks and parameterizing the modules in the network in a concise manner. Owing to the introduction of modules in network design, we can change the depth of the network by only changing G, which makes it more convenient to train the network with different depths or different numbers of MSPGs. In addition, it is easier to migrate to any upsampling factor with only minor adjustments to the network parameters.

TABLE 2. Quantitative comparisons of the MSRFN with 20 algorithms for ×2, ×3, and ×4 SR. Red numbers denote the best performance.

Scale	Method	Set5	Set14	BSD100	Urban100	Manga109
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
×2	1. Bicubic	33.68/0.9304	30.24/0.8691	29.56/0.8435	26.88/0.8405	31.05/0.9350
×2	2. SRCNN[9][11]	36.66/0.9542	32.45/0.9067	31.36/0.8879	29.51/0.8946	35.72/0.9680
×2	3. FSRCNN[30]	36.98/0.9556	32.62/0.9087	31.50/0.8904	29.85/0.9009	36.62/0.9710
×2	4. SCN[39]	36.52/0.9530	32.42/0.9040	31.24/0.8840	29.50/0.8960	35.51/0.9670
×2	5. REDNet[40]	37.66/0.9599	32.94/0.9144	31.99/0.8974	-/-	-/-
×2	6. VDSR[13]	37.53/0.9587	33.05/0.9127	31.90/0.8960	30.77/0.9141	37.16/0.9740
×2	7. DRCN[19]	37.63/0.9588	33.06/0.9121	31.85/0.8942	30.76/0.9133	37.57/0.9730
×2	8. LapSRN[26]	37.52/0.9591	32.99/0.9124	31.80/0.8949	30.41/0.9101	37.53/0.9740
×2	9. DRRN[20]	37.74/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188	37.92/0.9760
×2	10. DnCNN[41]	37.58/0.9590	33.03/0.9128	31.90/0.8961	30.74/0.9139	-/-
×2	11. ZSSR[42]	37.37/0.9570	33.00/0.9108	31.65/0.8920	-/-	-/-
×2	12. MemNet[43]	37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195	37.72/0.9740
×2	13. CMSC[44]	37.89/0.9605	33.41/0.9153	32.15/0.8992	31.47/0.9220	-/-
×2	14. IDN[45]	37.83/0.9600	33.30/0.9148	32.08/0.8985	31.27/0.9196	38.02/0.9749
×2	15. CNF[46]	37.66/0.9590	33.38/0.9136	31.91/0.8962	-/-	-/-
×2	16. SRMDNF[47]	37.79/0.9601	33.32/0.9159	32.05/0.8985	31.33/0.9204	38.07/0.9761
×2	17. SelNet[48]	37.89/0.9598	33.61/0.9160	32.08/0.8984	-/-	-/-
×2	18. CARN[49]	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256	38.36/0.9764
×2	19. SRRAM[50]	37.82/0.9592	33.48/0.9171	32.12/0.8983	32.05/0.9264	-/-
×2	20. MRFN[51]	37.98/0.9611	33.41/0.9159	32.14/0.8997	31.45/0.9221	38.29/0.9759
×2	21. MFFB[52]	37.82/0.9599	33.35/0.9156	32.04/0.8980	31.49/0.9218	38.23/0.9762
×2	22. FGLRL[53]	37.35/0.9633	33.11/0.9133	32.20/0.8842	27.30/0.8280	-/-
×2	23. MSRN[25]	38.08/0.9605	33.74/0.9170	32.23/0.9013	32.22/0.9326	38.82/0.9868
×2	Ours	38.11/0.9609	33.74/0.9188	32.25/0.9004	32.38/0.9307	38.82/0.9775
×3	1. Bicubic	30.40/0.8686	27.54/0.7741	27.21/0.7389	24.46/0.7349	26.95/0.8560
×3	2. SRCNN	32.75/0.9090	29.29/0.8215	28.41/0.7863	26.24/0.7991	30.48/0.9120
×3	3. FSRCNN	33.16/0.9140	29.42/0.8242	28.52/0.7893	26.41/0.8064	31.10/0.9210
×3	4. SCN	32.62/0.9080	29.16/0.8180	28.33/0.7830	26.21/0.8010	30.22/0.9140
×3	5. REDNet	33.82/0.9230	29.61/0.8341	28.93/0.7994	-/-	-/-
×3	6. VDSR	33.66/0.9213	29.78/0.8318	28.83/0.7976	27.14/0.8279	32.01/0.9340
×3	7. DRCN	33.82/0.9226	29.77/0.8314	28.80/0.7963	27.15/0.8277	32.31/0.9360
×3	8. LapSRN	33.82/0.9227	29.79/0.8320	28.82/0.7973	27.07/0.8271	32.21/0.9350
×3	9. DRRN	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8377	32.74/0.9390
×3	10. DnCNN	33.75/0.9222	29.81/0.8321	28.85/0.7981	27.15/0.8276	-/-
×3	11. ZSSR	33.42/0.9188	29.80/0.8304	28.67/0.7945	-/-	-/-
×3	12. MemNet	34.09/0.9248	30.00/0.8350	28.96/0.8001	27.56/0.8376	32.51/0.9369
×3	13. CMSC	34.24/0.9266	30.09/0.8371	29.01/0.8024	27.69/0.8411	-/-
×3	14. IDN	34.11/0.9253	29.99/0.8354	28.95/0.8013	27.42/0.8359	32.69/0.9378
×3	15. CNF	33.74/0.9226	29.90/0.8322	28.82/0.7980	-/-	-/-
×3	16. SRMDNF	34.12/0.9254	30.04/0.8382	28.97/0.8025	27.57/0.8398	33.00/0.9403
×3	17. SelNet	34.27/0.9257	30.30/0.8399	28.97/0.8025	-/-	-/-
×3	18. CARN	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493	33.49/0.9440
×3	19. SRRAM	34.30/0.9256	30.32/0.8417	29.07/0.8039	28.12/0.8507	-/-

TABLE 2. *(Continued.)* **Quantitative comparisons of the MSRFN with 20 algorithms for ×2, ×3, and ×4 SR. Red numbers denote the best performance.**

×3	20. MRFN	34.21/0.9267	30.03/0.8363	28.99/0.8029	27.53/0.8389	32.82/0.9396
×3	21. MFFB	34.31/0.9265	30.29/0.8408	29.05/0.8035	27.94/0.8472	33.37/0.9433
×3	22. FGLRL	32.95/0.9125	29.51/0.8325	28.66/0.7870	27.30/0.8280	-/-
×3	23. MSRN	34.38/0.9262	30.34/0.8395	29.08/0.8041	28.08/0.8554	33.44/0.9427
×3	Ours	34.63/0.9289	30.53/0.8455	29.23/0.8085	28.67/0.8624	34.09/0.9476
×4	1. Bicubic	28.43/0.8109	26.00/0.7023	25.96/0.6678	23.14/0.6574	25.15/0.7890
×4	2. SRCNN	30.48/0.8628	27.50/0.7513	26.9/0.7103	24.52/0.7226	27.66/0.8580
×4	3. FSRCNN	30.70/0.8657	27.59/0.7535	26.96/0.7128	24.60/0.7258	27.89/0.8590
×4	4. SCN	30.39/0.8620	27.48/0.7510	26.87/0.7100	24.52/0.7250	27.39/0.8570
×4	5. REDNet	31.51/0.8869	27.86/0.7718	27.40/0.7290	-/-	-/-
×4	6. VDSR	31.35/0.8838	28.02/0.7678	27.29/0.7252	25.18/0.7525	28.82/0.8860
×4	7. DRCN	31.53/0.8854	28.03/0.7673	27.24/0.7233	25.14/0.7511	28.97/0.8860
×4	8. LapSRN	31.54/0.8866	28.09/0.7694	27.32/0.7264	25.21/0.7553	29.09/0.8900
×4	9. DRRN	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638	29.46/0.8960
×4	10. DnCNN	31.40/0.8845	28.04/0.7672	27.29/0.7253	25.2/0.7521	-/-
×4	11. ZSSR	31.13/0.8796	28.01/0.7651	27.12/0.7211	-/-	-/-
×4	12. MemNet	31.74/0.8893	28.26/0.7723	27.4/0.7281	25.50/0.7630	29.42/0.8942
×4	13. CMSC	31.91/0.8923	28.35/0.7751	27.46/0.7308	25.64/0.7692	-/-
×4	14. IDN	31.82/0.8903	28.25/0.7730	27.41/0.7297	25.41/0.7632	-/-
×4	15. SRMDNF	31.96/0.8925	28.35/0.7787	27.49/0.7337	25.68/0.7731	30.09/0.9024
×4	16. CNF	31.55/0.8856	28.15/0.7680	27.32/0.7253	-/-	-/-
×4	17. SelNet	32.00/0.8931	28.49/0.7783	27.44/0.7325	-/-	-/-
×4	18. CARN	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	30.40/0.9082
×4	19. SRRAM	32.13/0.8932	28.54/0.7800	27.56/0.7350	26.05/0.7834	-/-
×4	20. MRFN	31.90/0.8916	28.31/0.7746	27.43/0.7309	25.46/0.7654	29.57/0.8962
×4	21. MFFB	32.31/0.8963	28.71/0.7843	27.66/0.7383	26.30/0.7922	30.84/0.9126
×4	22. FGLRL	31.44/0.8846	28.05/0.7688	27.26/0.7244	25.13/0.7501	-/-
×4	23. MSRN	32.07/0.8903	28.60/0.7751	27.52/0.7273	26.04/0.7896	30.17/0.9034
×4	Ours	32.42/0.8979	28.81/0.7869	27.73/0.7408	26.59/0.8003	31.13/0.9158

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The performance of the MSRFN was evaluated using several benchmark datasets. We first introduce the experimental setting, evaluation metrics, and implementation details, then provide the quantitative comparison results with mainstream methods, and finally show the visualization results of different methods from the perspective of qualitative analysis. Comparative analysis of various SR models demonstrated the superiority of the MSRFN.

A. IMPLEMENTATION AND TRAINING DETAILS

1) EXPERIMENTAL PLATFORM

The operation system is win10, the CPU is Intel Core i5-7500, and the GPU is NVIDIA RTX-2080. All experiments were completed using the deep learning framework Pytorch 1.2.0, and the accelerator library was CUDA Toolkit 10.0.

2) DATASETS AND METRICS

We used 800 images in DIV2K [23] as the training set. DIV2K contains 800 2 K resolution train images collected from the Internet. Rotation and flipping are used for data augmentation to fully utilize the training data [14]. During the test, we selected PSNR and SSIM [22] as metrics to evaluate SR image quality on five standard benchmark datasets: Set5 [32], Set14 [33], BSD100 [34], Urban100 [35], and Manga109 [36]. The Set5 dataset has 5 images (“baby,” “bird,” “butterfly,” “head,” “woman”). The Set14 dataset is a dataset consisting of 14 images. The BSD100 dataset has 100 test images and it is composed of a large variety of images ranging from natural images to object-specific such as plants, people, food etc. The Urban100 dataset contains 100 images of urban scenes. Manga109 is composed of 109 manga volumes drawn by professional manga artists in Japan. They are commonly used for testing performance

TABLE 3. Quantitative comparisons of the MSRFN with 11 algorithms for ×8 SR. Red numbers denote the best performance.

Scale	Method	Set5	Set14	BSD100	Urban100	Manga109
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
×8	1. Bicubic	24.40/0.6580	23.10/0.5660	23.67/0.5480	20.74/0.5160	21.47/0.6500
×8	2. SRCNN	25.33/0.6900	23.76/0.5910	24.13/0.5660	21.29/0.5440	22.46/0.6950
×8	3. FSRCNN	20.13/0.5520	19.75/0.4820	24.21/0.5680	21.32/0.5380	22.39/0.6730
×8	4. SCN	25.59/0.7071	24.02/0.6028	24.30/0.5698	21.52/0.5571	22.68/0.6963
×8	5. VDSR	25.93/0.7240	24.26/0.6140	24.49/0.5830	21.70/0.5710	23.16/0.7250
×8	6. LapSRN	26.15/0.7380	24.35/0.6200	24.54/0.5860	21.81/0.5810	23.39/0.7350
×8	7. MemNet	26.16/0.7414	24.38/0.6199	24.58/0.5842	21.89/0.5825	23.56/0.7387
×8	8. DRFN[21]	26.22/0.7400	24.57/0.6250	24.60/0.5870	--/--	--/--
×8	9. MSLapSRN[26]	26.34/0.7558	24.57/0.6273	24.65/0.5895	22.06/0.5963	23.9/0.75640
×8	10. MSRN	26.59/0.7254	24.88/0.5961	24.70/0.5410	22.37/0.5977	24.28/0.7517
×8	11. EDSR	26.96/0.7762	24.91/0.6420	24.81/0.5985	22.51/0.6221	24.69/0.7841
×8	Ours	27.28/0.7874	25.15/0.6486	24.94/0.6028	22.80/0.6331	25.15/0.7980

of SR models. The larger the metric value, the better is the reconstruction performance. To be consistent with the existing network, all evaluations were performed only on the luminance channel (Y).

3) TRAINING SETTINGS

We set the batch size to 16. To take full advantage of the memory resources and contextual information from LR images, we feed RGB image patches with different patch sizes according to the upscaling factor (Table 1), which are used for training together with the corresponding HR patches. Bicubic down sampling is used as the degradation model to produce LR images from the ground truth HR image. We apply the method proposed by He *et al.* [37] to initialize the weights and use the ADAM [38] optimizer to optimize the parameters. The learning rate was initialized to 0.0001 and decayed by half per 200 epochs. We adopted L1 loss to train the network.

We designed different kernel sizes and padding in each branch of the MSPU and adjusted the sizes of the kernels and strides according to the corresponding scaling factors. Table 1 lists the network parameter settings for the different SR factors. Both the input and output of the network use RGB color channels. Except for the reconstruction layer at the end of the network, PReLU [37] was used as the activation function behind all the convolution and deconvolution layers.

B. COMPARISON OF RESULTS AND DISCUSSION

For four different scale factors (×2, ×3, ×4, and ×8), we qualitatively and quantitatively compared MSRFN with other latest SR models on five test sets with different characteristics. Set5, Set14, and B100 mainly contain natural scenes; Urban100 is composed of many regular patterns in urban scenes and focuses on man-made structures with details in different frequency ranges; Manga109 is comic datasets drawn by Japanese artists.

1) QUANTITATIVE ANALYSIS

Table 2 presents the results of quantitative comparisons. It can be seen that in these five datasets, MSRFN has higher objective evaluation metrics in terms of PSNR and SSIM. This proves that the MSRFN is not only inclined to construct regular artificial patterns, but also good at reconstructing irregular natural patterns. In particular, our training sets do not contain any comic images, but excellent experimental results are shown for Manga109, which indicates that the MSRFN has excellent performance in reconstructing images with fine-structure information such as comic characters. In short, the MSRFN is superior in adapting to various scene features and possesses remarkable SR results for images with different characteristics.

For small enlargement factors (×2, ×3, ×4), we compared the MSRFN with 21 advanced methods, as shown in Table 2. Because many models are not suitable for a large-scale factor SR (×8), the MSRFN is compared with 11 advanced methods on ×8, as shown in Table 3. For ×2 enlargement, MSRFN obtains the best PSNR results in five benchmark datasets, and the SSIM values of the MSRFN are only slightly lower than MSRN in BSD100, Urban100, and Manga109. However, for the ×3, ×4, and ×8 enlargements, the MSRFN is superior to all other models in terms of PSNR and SSIM. As the upscaling factor increased, the superiority became relatively more obvious. Especially for ×8 SR, it proves the effectiveness of MSRFN to enlarge the image with a large factor, which can generate HR components better than other networks.

2) QUANTITATIVE ANALYSIS

For qualitative analysis, Figs. 3 to 17 display the visual effects of multiple SR works in the above five datasets with different scaling factors.

For small SR factors (×2, ×3, and ×4), we compared the MSRFN with eight mainstream methods: bicubic, SRCNN,

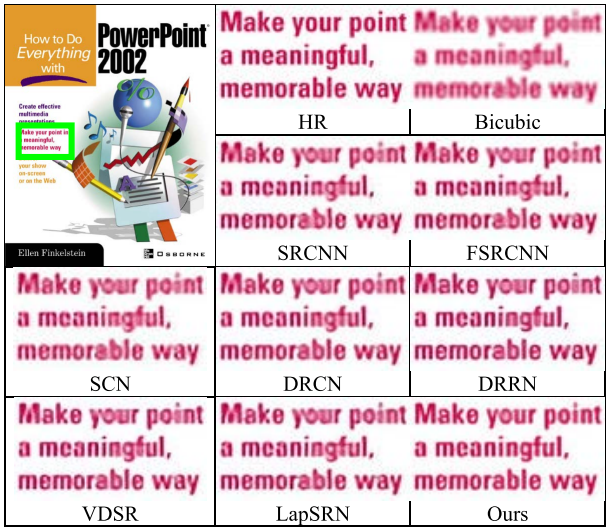

FIGURE 3. Visualization results for ppt3 (from Set14) for ×2 SR.

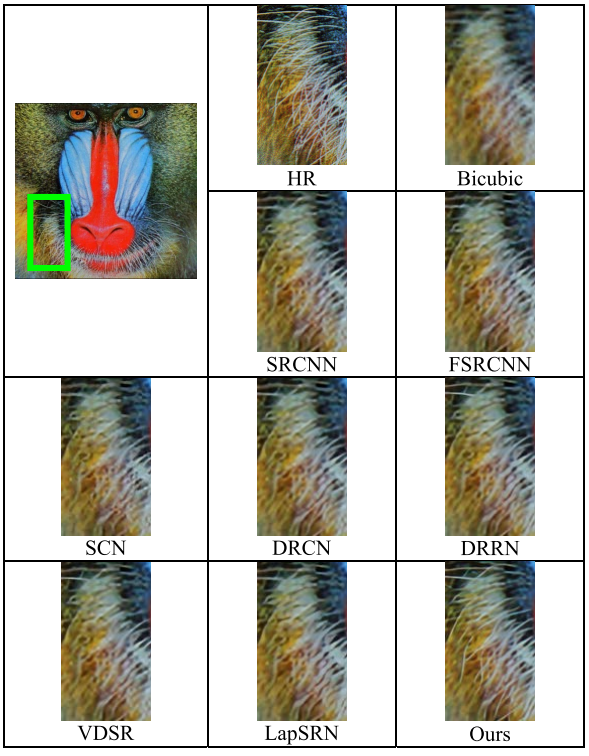

FIGURE 4. Visualization results for baboon (from Set14) for ×3 SR.

FSRCNN, SCN, DRCN, DRRN, VDSR, and LapSRN. Fig. 3 shows the visualization results for the ×2 SR. Owing to the low magnification, the gap between different models is subtle, but in contrast, the MSRFN still shows an obvious advantage. The text in our reconstructed image is clearer, there is no blur or adhesion between the letters, and the first letter "M" recovered from seven CNN-based networks (from SRCNN to LapSRN) has a crack that should not exist, but MSRFN has avoided this defect very well. Figs. 4 and 5 show

FIGURE 5. Visualization result for Belmondo (from Manga109) for ×3 SR.

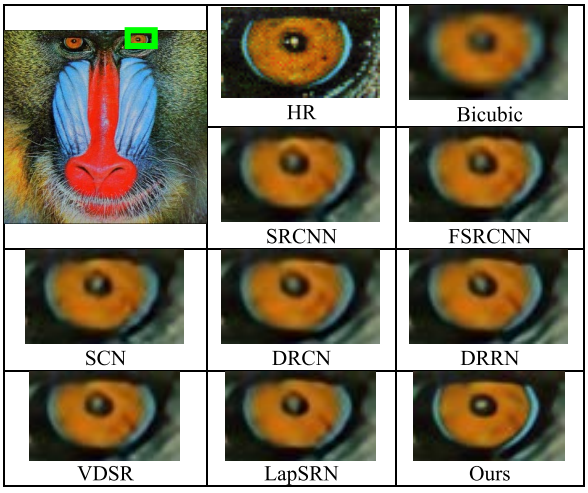

FIGURE 6. Visual result for Belmondo (from Set4) for ×4 SR.

the visualization results for the ×3 SR. For the natural image "baboon," the MSRFN restores sharper beard patterns than other models; for the comic image "Belmondo," the edges of the patterns reconstructed by other models have obvious blur artifacts, while the MSRFN accurately predicts the edges and details of patterns. Fig. 6 shows the visualization results for the ×4 SR. For the image "Belmondo" with irregular characteristics in Fig. 6, the eye patterns recovered by other models all suffer from different degrees of blurring, but MSRFN can recover more high-frequency information and details so that the reconstructed pattern contains sharp and accurate edges. For the image "img_096" with regular characteristics from Urban100 (Fig. 7), the edge features recovered by other models are obviously affected by the ringing effect and checkerboard artifacts, but MSRFN successfully eliminates these negative effects and reconstructs clearer patterns of building and window, which are very close to original HR image in comparison.

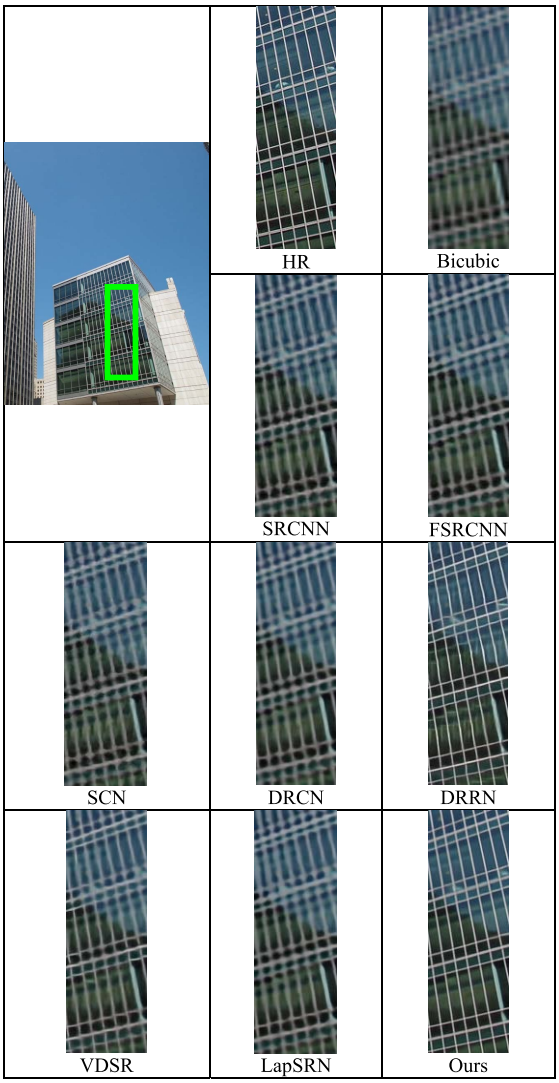

FIGURE 7. Visualization results for img_096 (from Urban100) for ×4 SR.

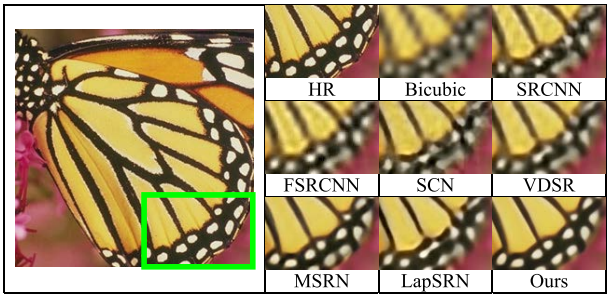

FIGURE 8. Visualization results for butterfly (from Set5) for ×8 SR.

For a large SR factor (×8), we compared the MSRFN with seven mainstream methods: bicubic, SRCNN, FSRCNN, SCN, VDSR, LapSRN, and MSRN in five benchmark datasets (Figs. 8-17). As shown in Fig. 8, the MSRFN has an excellent reconstruction effect for irregular speckle patterns, while the SR results from other models lose more edge details and have a relatively severe blurring. Fig. 12 shows that the MSRFN can reconstruct clear text even on large scaling factors, and other models have difficulty in estimating

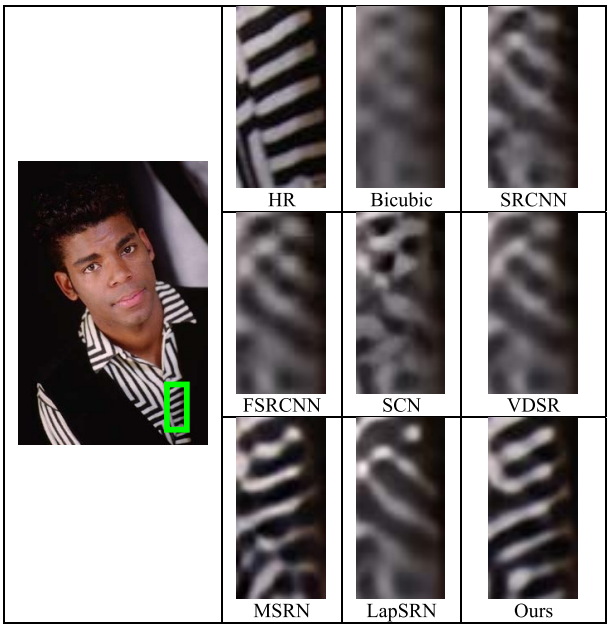

FIGURE 9. Visualization results for 302008 (from BSD100) for ×8 SR.

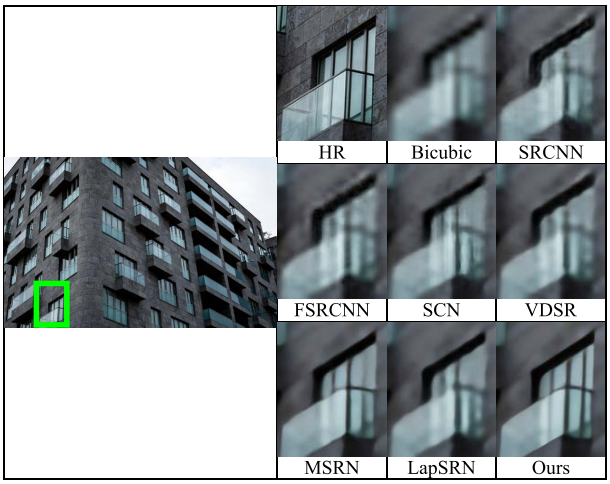

FIGURE 10. Visualization results for img_001 (from Urban100) for ×8 SR.

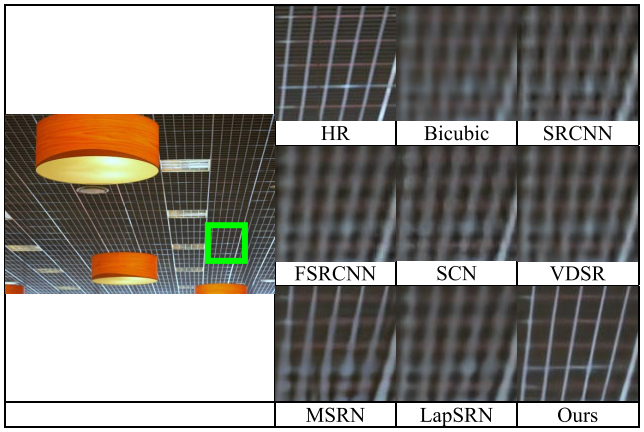

FIGURE 11. Visualization results for img_044 (from Urban100) for ×8 SR.

high-frequency information because of insufficient feature utilization, which reduces the ability to recover text details. In Fig. 9, the other models predict the wrong stripe

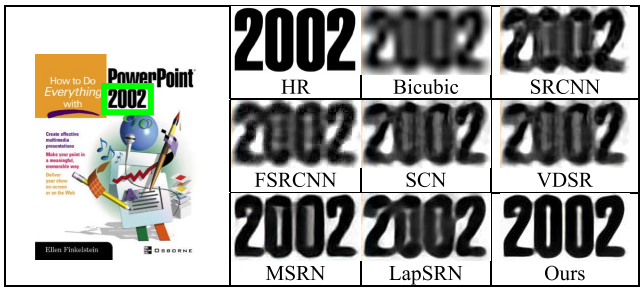

FIGURE 12. Visualization results for ppt3 (from Set14) for ×8 SR.

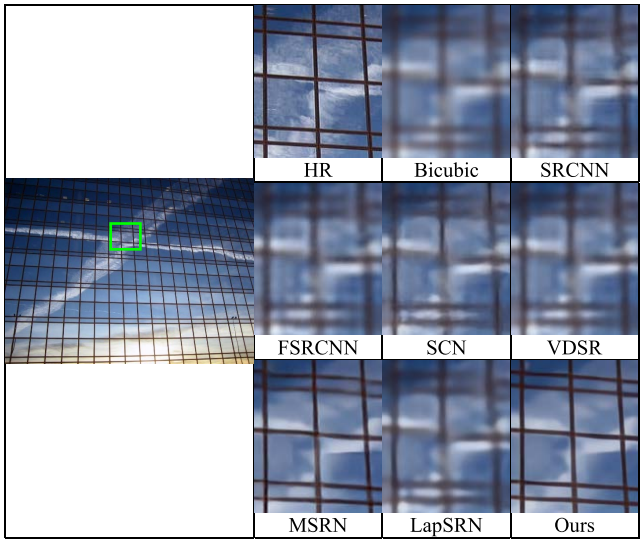

FIGURE 13. Visualization results for img_055 (form Urban100) for ×8 SR.

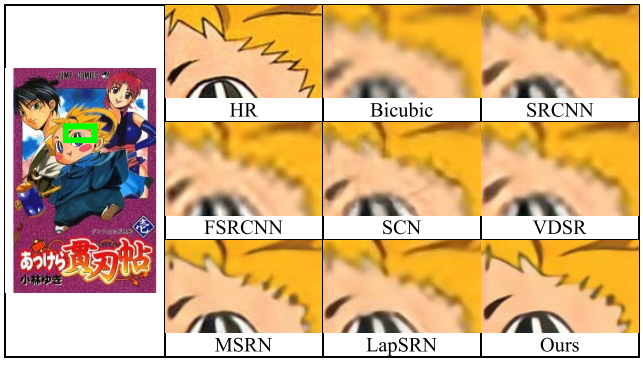

FIGURE 14. Visualization results for AkkeraKanjinchou (from Manga109) for ×8 SR.

direction owing to their weak ability to recover high-frequency components, but the MSRFN recovers the high-frequency texture details to the greatest extent and the correct direction. Figs. 10, 11, and 13, show the visualization results of images on Urban100, from which it can be seen that the MSRFN surpasses other advanced models in the reconstruction performance of images containing regular modes with more mid- and high-frequency information. Figs. 14 to 17 show the reconstruction results of the comic images with more complex and fine textures. Other methods have difficulty in estimating high-frequency details such that SR images have smooth edges and blur artifacts, but the MSRFN results have finer details such as sharper edges and contours.

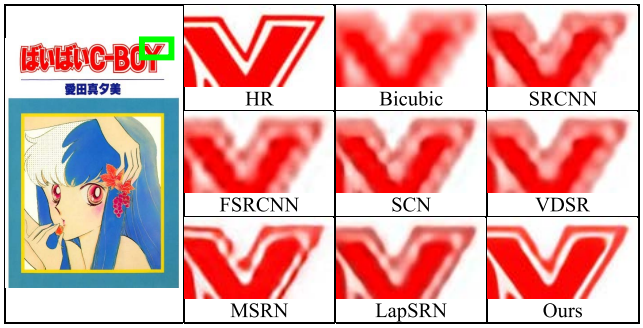

FIGURE 15. Visualization results for byebyec-boy (from Manga109) for ×8 SR.

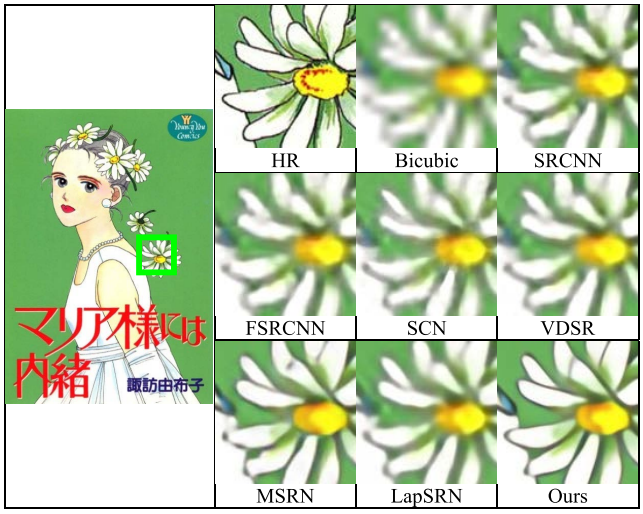

FIGURE 16. Visualization results for MariaSamaNihaNaisyo for ×8 SR.

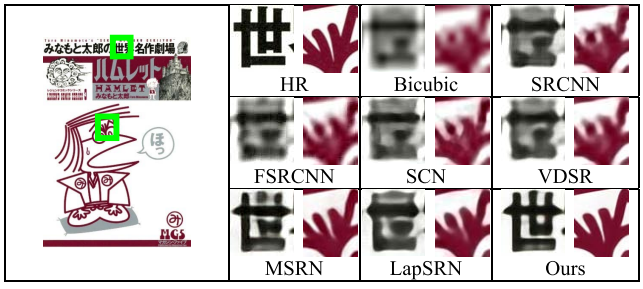

FIGURE 17. Visualization results for Hamlet (from Manga109) for ×8 SR.

Owing to the loss of information during image degradation, especially the loss of high-frequency information, these CNN-based SR models still recover smooth image edges. As the scaling factor increases, the edge blurring becomes more severe. However, MSRFN can suppress the smooth component and predict more high-frequency information, which can make SR images with sharper edges and contours, and to a great extent alleviate the interference of checkerboard artifacts and ringing effects. Surprisingly, MSRFN still retains this advantage at large scaling factors, generating the SR results closest to the ground truth in comparison.

The above qualitative and quantitative comparisons and analyses show that the MSRFN has a persuasive reconstruction performance. Compared with feed-forward networks, it focuses on refining well-developed information; compared with single-scale networks, it can focus on fine details and generate finer high-level representations. It can not only capture image features on multiple context scales and mine more mutual dependencies between LR and HR images, but also create contextual information from LR input, which can save HR features better, even in the face of large scaling factors.

V. CONCLUSION

We propose a multiscale recursive feedback network for image super-resolution. Unlike single-scale networks, the proposed multiscale projection unit can adaptively capture image features with different scales by constructing a two-bypass structure with different kernels, in which feature information can be shared between different bypasses to fully use the local features of images. Unlike feed-forward networks, we design recursive multiscale projection groups to form feedback modules that can effectively enhance features. We also combine local and global information by the fusion of local multiscale residual features and global residual features. The feedback flow exploits the high-level information extracted from deep layers to refine the low-level features from shallow layers, which improves the early reconstruction performance of the MSRFN. Furthermore, a combination of global residual learning and local residual feedback can encourage feature reuse and provide more contextual information for the final reconstruction. Therefore, MSRFN not only focuses on fusing local information and global information, but also pays attention to combining low-level details with high-level abstract semantics, which helps to produce more faithful results to the ground truth. The experimental results show that the MSRFN achieves encouraging performance and is superior to other advanced SR methods, especially for large-scale factors (such as $\times 8$).

Future research improvements mainly have the following directions. If there is noise in images, the performance of SR methods might become worse. We will study SR methods for noisy images by the integration of denoising methods [54]–[58]. In training, ADAM optimizer is commonly used in many SR studies. The use of other optimization algorithms [59]–[63], such as particle swarm optimization algorithm [64]–[66], might improve the SR performance in our future study. As the MSRFN has achieved good performance for $\times 8$ SR, we also intend to apply it to higher SR rates such as $\times 16$ and develop a single model performing multiscale super-resolution in our future study.

REFERENCES

[1] S. Ren, J. Li, K. Guo, and F. Li, "Medical video super-resolution based on asymmetric back-projection network with multilevel error feedback," *IEEE Access*, vol. 9, pp. 17909–17920, 2021.

[2] Y. Pang, J. Cao, J. Wang, and J. Han, "JCS-Net: Joint classification and super-resolution network for small-scale pedestrian detection in surveillance images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 12, pp. 3322–3331, Dec. 2019.

[3] Y. Luo, L. Zhou, S. Wang, and Z. Wang, "Video satellite imagery super resolution via convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2398–2402, Dec. 2017.

[4] Y. Tan, J. Cai, S. Zhang, W. Zhong, and L. Ye, "Image compression algorithms based on super-resolution reconstruction technology," in *Proc. IEEE 4th Int. Conf. Image, Vis. Comput. (ICIVC)*, Jul. 2019, pp. 162–166.

[5] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "SOD-MTGAN: Small object detection via multi-task generative adversarial network," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 206–221.

[6] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "Finding tiny faces in the wild with generative adversarial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 21–30.

[7] H. Lin and J. Yang, "Light weight IBP deep residual network for image super resolution," *IEEE Access*, vol. 9, pp. 93399–93408, 2021.

[8] H. Yang and Y. Wang, "An effective and comprehensive image super resolution algorithm combined with a novel convolutional neural network and wavelet transform," *IEEE Access*, vol. 9, pp. 98790–98799, 2021.

[9] Z. Yang, D. Pan, and P. Shi, "Joint image dehazing and super-resolution: Closed shared source residual attention fusion network," *IEEE Access*, vol. 9, pp. 105477–105492, 2021.

[10] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[11] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 184–199.

[12] 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.

[13] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646–1654.

[14] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 136–144.

[15] 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)*, Jul. 2017, vol. 1, no. 2, p. 3.

[16] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4809–4817.

[17] 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.

[18] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1664–1673.

[19] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1637–1645.

[20] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3147–3155.

[21] X. Yang, H. Mei, J. Zhang, K. Xu, B. Yin, Q. Zhang, and X. Wei, "DRFN: Deep recurrent fusion network for single-image super-resolution with large factors," *IEEE Trans. Multimedia*, vol. 21, no. 2, pp. 328–337, Feb. 2019.

[22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[23] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 126–135.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[25] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 527–542.

[26] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 624–632.

[27] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, "Feedback network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3867–3876.

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[29] Z. Wang, J. Chen, and S. C. H. Hoi, "Deep learning for image super-resolution: A survey," 2019, *arXiv:1902.06068*.
[30] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 391–407.
[31] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1874–1883.
[32] M. Bevilacqua, A. Roumy, C. Guillemot, and A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. 23rd Brit. Mach. Vis. Conf.*, 2012, pp. 1–12.
[33] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. Curves Surfaces*. Berlin, Germany: Springer, 2010, pp. 711–730.
[34] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, Aug. 2011.
[35] 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.
[36] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, "Sketch-based Manga retrieval using manga109 dataset," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 21811–21838, Oct. 2017.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[39] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 370–378.
[40] 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. 2810–2818.
[41] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
[42] A. Shocher, N. Cohen, and M. Irani, "Zero-shot super-resolution using deep internal learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3118–3126.
[43] 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.
[44] Y. Hu, X. Gao, J. Li, Y. Huang, and H. Wang, "Single image super-resolution via cascaded multi-scale cross network," 2018, *arXiv:1802.08808*.
[45] Z. Hui, X. Wang, and X. Gao, "Fast and accurate single image super-resolution via information distillation network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 723–731.
[46] H. Ren, M. El-Khamy, and J. Lee, "Image super resolution based on fusing multiple convolution neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1050–1057.
[47] K. Zhang, W. Zuo, and L. Zhang, "Learning a single convolutional super-resolution network for multiple degradations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3262–3271.
[48] J.-S. Choi and M. Kim, "A deep convolutional neural network with selection units for super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1150–1156.
[49] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 252–268.
[50] J.-H. Kim, J.-H. Choi, M. Cheon, and J.-S. Lee, "MAMNet: Multi-path adaptive modulation network for image super-resolution," 2018, *arXiv:1811.12043*.
[51] Z. He, Y. Cao, L. Du, B. Xu, J. Yang, Y. Cao, S. Tang, and Y. Zhuang, "MRFN: Multi-receptive-field network for fast and accurate single image super-resolution," *IEEE Trans. Multimedia*, vol. 22, no. 4, pp. 1042–1054, Apr. 2020.
[52] C. Sun and X. Chen, "Multiscale feature fusion back-projection network for image super-resolution," *Acta Autom. Sinica*, vol. 47, no. 7, pp. 1689–1700, 2021.
[53] J. Hou, Y. Si, and X. Yu, "A novel and effective image super-resolution reconstruction technique via fast global and local residual learning model," *Appl. Sci.*, vol. 10, no. 5, p. 1856, Mar. 2020.
[54] X. Chen and J. Li, "Noise reduction for ultrasonic Lamb wave signals by empirical mode decomposition and wavelet transform," *J. Vibroeng.*, vol. 15, no. 3, pp. 1157–1165, 2013.
[55] X. Chen, Y. Gao, and C. Wang, "Fractional derivative method to reduce noise and improve SNR for Lamb wave signals," *J. Vibroeng.*, vol. 17, no. 8, pp. 4211–4218, 2015.
[56] C. Xiao and W. Chen-Long, "Noise suppression for Lamb wave signals by tsallis mode and fractional-order differential," *Acta Phys. Sinica*, vol. 63, no. 18, 2014, Art. no. 184301.
[57] X. Chen and C. Wang, "Noise removing for Lamb wave signals by fractional differential," *J. Vibroeng.*, vol. 16, no. 6, pp. 2676–2684, 2014.
[58] N. Long and C. Xiao, "Mode separation for multimode Lamb waves based on dispersion compensation and fractional differential," *Acta Phys. Sinica*, vol. 67, no. 20, 2018, Art. no. 204301.
[59] X. Chen and C. J. Hu, "Adaptive medical image encryption algorithm based on multiple chaotic mapping," *Saudi J. Biol. Sci.*, vol. 24, no. 8, pp. 1821–1827, 2017.
[60] X. Chen and C. Wang, "Tsallis distribution-based fractional derivative method for Lamb wave signal recovery," *Res. Nondestruct. Eval.*, vol. 26, no. 3, pp. 174–188, Jul. 2015.
[61] X. Chen and W. Zhan, "Effect of transducer shadowing of ultrasonic anemometers on wind velocity measurement," *IEEE Sensors J.*, vol. 21, no. 4, pp. 4731–4738, Feb. 2021.
[62] X. Chen and D. Ma, "Mode separation for multimodal ultrasonic Lamb waves using dispersion compensation and independent component analysis of forth-order cumulant," *Appl. Sci.*, vol. 9, no. 3, p. 555, Feb. 2019.
[63] X. Chen, Y. Gao, and L. Bao, "Lamb wave signal retrieval by wavelet ridge," *J. Vibroeng.*, vol. 16, no. 1, pp. 464–476, 2014.
[64] X. Chen and B. Zhang, "Improved DV-hop node localization algorithm in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 8, Aug. 2012, Art. no. 213980.
[65] X. Chen and S. Zou, "Improved Wi-Fi indoor positioning based on particle swarm optimization," *IEEE Sensors J.*, vol. 17, no. 21, pp. 7143–7148, Nov. 2017.
[66] X. Chen and B. Zhang, "3D DV-hop localisation scheme based on particle swarm optimisation in wireless sensor networks," *Int. J. Sensor Netw.*, vol. 16, pp. 100–105, Jan. 2014.

XIAO CHEN (Member, IEEE) was born in Henan, China, in 1974. He received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2003.

He was with the Institute of Micro- and Sensor-Systems, Otto-Von-Guericke University Magdeburg, Germany, from 2004 to 2005. He was a Visiting Scholar with Imperial College London, U.K., from 2015 to 2016. He is currently a Professor at the Nanjing University of Information Science and Technology, Nanjing, China. He has authored a book and more than 100 articles, and holds more than 70 inventions. His research interests include modern electronic design, signal and information processing, image processing and communication, ultrasonic imaging, and ultrasonic NDE.

CHAOWEN SUN was born in Jiangsu, Nanjing, China, in 1991. She received the B.S. degree from the Zijin College, Nanjing University of Science and Technology, Nanjing, in 2018. She is currently pursuing the master's degree in electronics and communication engineering with the Nanjing University of Information Science and Technology, Nanjing. Her research interests include computer vision, deep learning, and image processing.