Cascading and Enhanced Residual Networks for Accurate Single-Image Super-Resolution

Rushi Lan, Long Sun, Zhenbing Liu, Huimin Lu, Zhixun Su, Cheng Pang, and Xiaonan Luo

***Abstract*—Deep convolutional neural networks (CNNs) have contributed to the significant progress of the single-image super-resolution (SISR) field. However, the majority of existing CNN-based models maintain high performance with massive parameters and exceedingly deeper structures. Moreover, several algorithms essentially have underused the low-level features, thus causing relatively low performance. In this article, we address these problems by exploring two strategies based on novel local wider residual blocks (LWRBs) to effectively extract the image features for SISR. We propose a cascading residual network (CRN) that contains several locally sharing groups (LSGs), in which the cascading mechanism not only promotes the propagation of features and the gradient but also eases the model training. Besides, we present another enhanced residual network (ERN) for image resolution enhancement. ERN employs a dual global pathway structure that incorporates nonlocal operations to catch long-distance spatial features from the the original low-resolution (LR) input. To obtain the feature representation of the input at different scales, we further introduce a multiscale block (MSB) to directly detect low-level features from the LR image. The experimental results on four benchmark datasets have demonstrated that our models outperform most of the advanced methods while still retaining a reasonable number of parameters.**

***Index Terms*—Convolutional neural network, multiscale learning, residual learning, single-image super-resolution (SISR).**

Manuscript received May 6, 2019; revised August 21, 2019; accepted October 25, 2019. Date of publication February 21, 2020; date of current version December 22, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61702129, Grant 61772149, Grant 61562013, Grant 61866009, Grant 61572099, and Grant U1701267, in part by the China Post-Doctoral Science Foundation under Grant 2018M633047, and in part by the Guangxi Science and Technology Project under Grant 2019GXNSFFA245014, Grant AD18281079, Grant 2017GXNSFDA198025, Grant AD18216004, Grant AB17195057, and Grant AA18118039. This article was recommended by Associate Editor H. Lu. *(Corresponding author: Zhenbing Liu.)*

R. Lan is with the Guangxi Key Laboratory of Image and Graphic Intelligent Processing, Guilin University of Electronic Technology, Guilin 541004, China, and also with the School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China.

L. Sun, Z. Liu, C. Pang, and X. Luo are with the Guangxi Key Laboratory of Image and Graphic Intelligent Processing, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: zbliu2011@163.com).

H. Lu is with the Department of Mechanical and Control of Engineering, Kyushu Institute of Technology, Kitakyushu 8048550, Japan.

Z. Su is with the Guangxi Key Laboratory of Image and Graphic Intelligent Processing, Guilin University of Electronic Technology, Guilin 541004, China, also with the School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China, and also with the National Local Joint Engineering Research Center of Satellite Navigation and Location Service, Guilin University of Electronic Technology, Guilin 541004, China.

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

Digital Object Identifier 10.1109/TCYB.2019.2952710

I. Introduction

Super-resolution (SR) image reconstruction is widely used in many practical cases, such as military surveillance, medical diagnostics, satellite images, and video applications, and the demand for high-resolution (HR) images has dramatically increased recently. In practice, the quality of image resolution is limited by physical constraints. Much of the SR algorithms have been proposed to address this problem, and they can be broadly divided into models developed for still images or for video sequences. In this article, we focus on single-image super-resolution (SISR). The task of recovering super-resolved HR images I^{SR} from low-resolution (LR) versions I^{LR} is ill-posed since a number of HR solutions can map to any LR image. Therefore, numerous approaches have been developed so far, including interpolation-based, reconstruction-based, and learning-based methods [31], [45], [51], respectively.

The interpolation-based algorithms, such as bicubic interpolation [17], are very fast but suffer from lower accuracy and are limited in applications. More advanced reconstruction-based SR algorithms [29], [36] are proposed by introducing prior knowledge to limit the possible solution space. These methods can recover sharp details but rapidly degrade as scale factors increase; subsequently, the learning-based methods [4], [15], [26], [43], [44], [50], [56] are employed that exploit machine learning algorithms to analyze relationships between the I^{LR} image and the corresponding I^{HR} image by training substantial examples. Although such learning-based methods are outstanding, they involve time-consuming optimization operations.

Currently, deep convolutional neural networks (CNNs) have contributed to the significant progress of the SISR field because of their superior ability of feature representation. Dong *et al.* [7], [8] first proposed a convolutional model to solve the SISR problem in 2014, which became a milestone in the image restoration area. Since then, more complicated networks were designed to enhance the performance [10], [16], [18], [19], [21], [23], [25], [39], [40], [58]. Lim *et al.* [25] proposed a very deep and wide model with residual blocks and achieved satisfactory performance in terms of both peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [48]. Although these networks present promising results, there are some limitations to the CNN-based models: 1) the state-of-the-art models [23], [25], [47], [57], [58] mainly concentrate on improvements obtained via substantially increasing the depth or the width; thus, they have massive parameters and consume increasingly more computational resources, time, and tricks during training and 2) most of the CNN-based models do not fully use the hierarchical features from the original LR image.

To address these drawbacks, we explore two strategies to effectively extract features for accurate SISR. First, we propose a cascading residual network (CRN) for more efficient feature extraction. Specifically, we introduce a cascading connections mechanism for better feature fusion and gradient propagation. With such a mechanism, our network can incorporate features from multiple layers at both the local and global level. Moreover, a locally sharing group (LSG) structure is proposed, in which the local wider residual blocks (LWRBs) are stacked to exploit the feature of the I^{LR} image and allow the abundant low-level features to be passed.

Second, we present another enhanced residual network (ERN) for SISR. In this method, we introduce a dual global pathway structure for a more powerful feature expression. This schema incorporates nonlocal operations to catch long-distance spatial features from original LR input. Meanwhile, by stacking LWRBs, we can boost the feature representation ability. To fully use the low-level information, we additionally introduce a multiscale block (MSB) that directly extracts low-level features from the original I^{LR} image at different scales.

As the key component of our proposed networks, the LWRB contains two convolutional layers and a nonlinear layer ReLU. We exploit wider channels before the ReLU layer for building an inverted residual block, and it leads to significant improvements, due to the fact that using the activation function in bottlenecks indeed hurts the performance [34]. As discussed above, the latest state-of-the-art models [25], [58] maintain high performance with massive parameters and exceedingly deeper structures (e.g., over 100 layers). Compared with these models, our methods are more efficient since the parameters of the proposed models are only approximately 1/4 and 1/2 of those of the referenced algorithms, respectively, and the proposed models are considerably lower than them in depth. Experimentally, our methods show gain similar superior results regarding PSNR and SSIM.

The main contributions of this article are summarized as follows.

1) We propose the LWRB, which not only effectively preserves features via expanding the low-dimensional representation to high dimension before the activation function but also utilizes the information of all layers within a block via an identity connection.
2) We introduce a cascading schema to effectively boost feature fusion and gradient propagation. Such a mechanism enables our network to incorporate the features from multiple layers. Furthermore, an LSG structure is used to build the network and enhance its future expression.
3) We present an ERN for accurate SISR, which mainly contains the dual global pathway and several LWRBs. The global structure incorporates nonlocal operations to catch long-distance spatial features from original LR input. Meanwhile, stacking the residual blocks can enhance the representational capability.

The remainder of this article is arranged as follows. In Section II, we present a brief review of the relevant works on SISR. In Section III, we provide the architecture of the proposed networks in detail. In Section IV, we show extensive results to evaluate the proposed methods. Finally, we conclude the proposed methods in Section V.

II. Related Works

In this section, we briefly introduce some works that are related with our proposed models.

A. SISR Using Convolutional Neural Networks

Recently, CNN-based models have achieved dramatic success against traditional methods in image recovery, especially, super-resolution, given their powerful ability of feature expression. Dong *et al.* [7] first proposed a CNN-based algorithm to directly learn an end-to-end mapping between the I^{LR} image and the I^{SR} image. In their work, the model called SRCNN consists of three convolutional layers and shows an impressive performance over the conventional methods, such as sparse coding [29] and bicubic interpolation [17]. Later, many advanced models were developed by designing more complex CNN architectures. VDSR [18] introduced residual learning to increase the depth of the network and proved that this strategy can improve reconstruction performance and accelerate convergence. DRCN [19], a deeply recursive neural network for SISR, uses the same convolutional kernel in the reference network 16 times. By doing so, it can efficiently reduce the number of parameters. Notice that all of these methods use the interpolated image as input; this behavior not only leads to detail-smoothing effects but also relatively increases the computational cost and time consumption.

To address the problem of computational efficiency, several algorithms were proposed to automatically learn a mapping from I^{LR} to I^{SR}. FSRCNN [9] and ESPCN [35] explored two different active upsampling modules to reconstruct the low-quality image. The former used the standard deconvolution layer [53], which upsamples the previous features with an arbitrary interpolation operator and a subsequent convolution operator with a stride of 1. Rather than increasing resolution by inserting zero values, the latter introduced a subpixel convolution layer, which expanded the channels of the output features and then reshaped them to generate the HR output through a specific mapping criterion. It has been proven that the subpixel layer provides more contextual information and the interpolation is more efficient. Thanks to these merits, most of the following works also adopted this module, such as SRResNet [23], EDSR [25], and RDN [58]. Although impressive results have been achieved by these mentioned methods, most of them tend to consume a lot of computational resources.

B. Skip Connections

The concept of skip connections is first introduced in ResNet [11] and has been widely employed in diverse computer visual tasks, such as image restoration [40] and semantic segmentation [5], [6], [30], [41]. Since the plain SR network is hard to go deeper, various skip connections were introduced and achieved additional gain in performance. This strategy can be roughly divided into two categories, that is, global or local residual connections and dense connections.

1) Global or Local Residual Connections: The LR image is highly connected to the HR image in such an image-to-image translation task. Learning the residual map between these two images can capture the missing high-frequency details. VDSR [18], the first residual model used in super-resolution, proved the assumption that residual learning can improve the representation ability and accelerate convergence. Thus, this approach is widely used in the SR models [22], [47], [49].

2) Dense Connections: DenseNet [12], an effective network based on skip connections, allows the current layer to be connected with all the preceding layers. This schema provides richer information for recovering high-resolution details. Consequently, dense connections were introduced into the SR field [1], [10], [40], [46], [47], [58].

Memnet [40], proposed by Tai *et al.*, stacks memory blocks and adds the dense connections among each block. Based on this construction, the approach keeps a short and long memory of low-level features. RDN [58] used a similar architecture but is more useful to extract hierarchical features. Different from the aforementioned models, CARN [1] implemented a cascading connection mechanism to improve the SR performance and decrease operations. Haris *et al.* [10] proposed D-DBPN, which performs iterative upscaling and downscaling operations with dense connections and provides an error feedback mechanism for tuning the high-resolution results. This schema further improves the SR performance, especially, in a large enlargement such as ×8 SR.

C. Multiscale Learning

To optimize the sparse local features in a convolutional module, Szegedy *et al.* [38] proposed the inception module. This architecture processes the input data at various scales and then aggregates those information as input of the next stage to gain different abstract features. Inspired by [38] and [37], MSRB [24] was introduced as a multiscale residual block that used a 3×3 and a 5×5 kernel to adaptively extract local features and a 1×1 Conv layer to fuse the feature maps. It showed that performing different kernel operations could provide better extraction capability. However, this manner cannot cover a large range of receptive fields and generate more detailed layerwise multiscale representations.

III. Proposed Approach

In this section, we present a detailed description of the design methodology of our proposed networks and, then, discuss the difference between our methods and other state-of-the-art ones.

A. Network Architectures

The VGG-like algorithms of SISR do not make full use of the feature information from low-level layers, such as ESPCN [35] and FSRCNN [9]. The deeper models usually contain massive parameters for gaining the state-of-the-art performance. To better address the mentioned problems, we introduce two different strategies: 1) cascading connection structure and 2) globally dual residual path. The pipeline of our models includes three steps. Taking an I^{LR} image as input, a feature extraction module is used to obtain features from the low-quality image, and then these features are sent to the mapping stages. Finally, a simple upsampling block contains a convolutional layer, and a pixel-shuffle layer is adopted to enlarge the LR image. The main difference between these models is the mapping stages.

Specifically, let us denote I^{LR} and I^{SR} as the input and output of our models. We use two convolutional layers to extract low-level information from I^{LR} inputs

$$F_{\text{ext}} = H_{\text{ext}}\big(I^{\text{LR}}\big) \tag{1}$$

where $H_{\text{ext}}(\cdot)$ means the convolution operation, and then F_{ext} is sent to the mapping stages for higher level feature abstraction, we have

$$F_{\text{map}} = H_{\text{map}_m}\big(H_{\text{map}_{m-1}}\big(\cdots\big(H_{\text{map}_1}(F_{\text{ext}})\big)\cdots\big)\big) \tag{2}$$

where $H_{\text{map}}(\cdot)$ denotes our proposed mapping function and $H_{\text{map}_{m-1}}$ and H_{map_m} are the input and output of the mth LWRB, respectively. Finally, these features are upscaled via a single upsampling block

$$F_{\text{up}} = H_{\text{up}}\big(F_{\text{map}}\big) \tag{3}$$

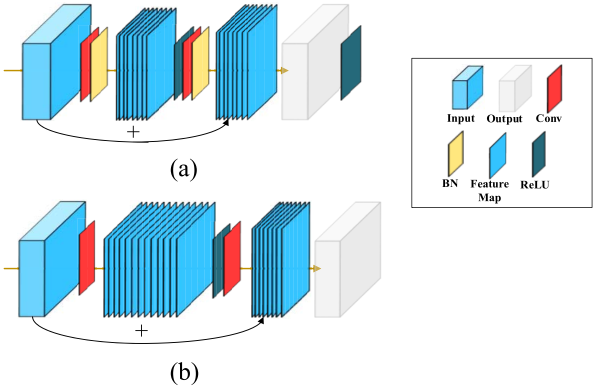
where $H_{\text{up}}(\cdot)$ represents an upscale module. There are many strategies to enlarge features, such as pre-upsampling [7]; post-upsampling [9], [35]; progressive upsampling [21]; and iterative upsampling and downsampling [10]. The post-upscaling method is used in this article and we chose the subpixel convolution layer [35] as the magnification function, which is proven effective to increase resolution. Therefore, our approach can be formulated as

$$I^{\text{SR}} = H_{\text{up}}\big(H_{\text{map}}\big(\cdots\big(H_{\text{ext}}\big(I^{\text{LR}}\big)\big)\big)\cdots\big). \tag{4}$$

B. Local Wider Residual Block

Residual networks [11], [18], [32] have exhibited noteworthy performance in computer vision areas, ranging from low-level to high-level problems. Although Lim *et al.* [25] successfully improved the ResNet architecture to address the SISR problem with EDSR, we further explore a better residual block to enhance the performance.

As shown in Fig. 1, the central building component of our proposed architectures is the basic residual block, which includes two convolutional layers and has been studied in [25]. This component is utilized to map the low-level features to HR space. We compare the building block from the original ResNet [11] and our proposed block. The modifications include the following: 1) removing BN layers; 2) reducing ReLU layers reasonably; and 3) expanding features before the ReLU layer.



(a)

(b)

Fig. 1. Comparison of residual blocks in original ResNet and ours. We remove the BN layers and expand features before the ReLU activation layer. We experimentally show that this simple modification substantially reduces the number of parameters and computational costs while regularly achieving superior results.

Recently, most of the PSNR-oriented tasks, including SISR, tend to remove BN layers [14] because it has been proven that the BN layers have a side effect on the final results of image super-resolution while increasing computational complexity [23], [25], [58]. The BN layers normalize the features among minibatches by using the mean and variance in a batch during training or testing. Regarding SISR, the LR input image and the reconstructed image have a similar structure; this layer makes it difficult to estimate the target image since the BN layers tend to introduce artifacts and limit the flexibility of networks. Note that we also tried to introduce other normalization methods to boost performance (e.g., weight normalization (WN) [33] and switchable normalization (SN) [28]), and experimentally showed that this is a time-consuming trick that causes extra computation but does not lead to better performance than an approach without normalization. Thus, we avoid using any normalization layers. Generally, the activation layer follows a specific convolutional layer to maintain the high nonlinearity of deep neural networks. However, we only use ReLU after the first convolutional layer in each basic block, given the assumption that the nonlinear ReLUs prevent the information in the low-level layer flow into deeper layers [34]. Moreover, we expand wider channels before the activation layer to capture more spatial information. Experimentally, these adjustments substantially reduce the number of parameters and the consumption of computational materials while achieving superior results.

Besides, the proposed block is different from EDSR [25]. A wider channel is used throughout the block in EDSR (e.g., it increased the channel up to 256), which dramatically increases the number of parameters and poses a challenge to train the model. In our models, we expand features before the ReLU activation layer and the low-level channel following it. Empirically, we found that it does not affect the great performance of the SR models while reducing a large number of parameters.

C. Cascading Residual Network

We now present our cascading residual network. Cascading connections have been widely applied to various computer vision tasks [1], [12], [27] since they allow the propagation of information across multiple paths. In Fig. 2, the mapping stages of our cascading network include G LSGs with skip connections. Each LSG further contains B LWRBs.

Expressed formally, let G_{out_i} be the output of the G_ith group. To increase the receptive field of the feature extraction module and reduce the number of parameters, we stacked small kernel sizes (e.g., 1×1 and 3×3) rather than directly using a large kernel size (e.g., 7×7 and 11×11). The low-level features F_{ext} are attained via the module and then sent to the G_ith group and final upsampling block. The output G_{out_i} of the G_ith group flows into one of the subsequent groups. Finally, a simple upscaling block is adopted to merge hierarchical features and enlarge the LR image.

Locally Sharing Group: It has been proven that stacking residual blocks is useful to build a deep architecture [18], [25]. However, a very deep plain network generally suffers from training difficulty due to the problem of the vanishing or exploding gradient. Thus, we propose an LSG as the basic unit. Given that stacking the residual blocks within a reasonable range can gain better performance, we accordingly investigate the number of LWRB included in the group. Then, the G_ith group can be expressed as

$$G_{out_i} = H_{lsg}\big(G_{out_{i-1}}\big) \tag{5}$$

where $H_{lsg}(\cdot)$ is the function of the G_ith group. $G_{out_{i-1}}$ and G_{out_i} denote the input and output of the G_ith group, respectively.

D. Enhanced Residual Network

As previously discussed above, the low-level features from the original input play a significant role in the SR task and many previous CNN-based methods ignore their importance. Based on this perception, we utilized an enhanced residual structure that contains a dual pathway structure.

Similar to the process of CRN, the low-level features are extracted by the feature extraction module and the MSB simultaneously. The output of the feature extraction block F_{ext} is sent to the mapping stages, which consist of several LWRBs to enhance the deep feature representations and the final upsampling module. The output of MSB (F_{msb}) directly operates an elementwise sum with F_{map} and F_{ext} via the long skip connections so that the features can be fully used in the reconstruction step. Subsequently, these refined features flow into the upsampling module for enlargement. We define the process as follows:

$$I^{\text{SR}} = H_{\text{up}}\big(F_{\text{ext}} + F_{\text{map}} + F_{\text{msb}}\big). \tag{6}$$

Dual Global Pathway: The global residual paths are shown in Fig. 3, where the top branch is designed to extract the low-level information with different kernel sizes (see Fig. 4) and the bottom path is a typical global connection to ensure a deeper network. This global structure incorporates nonlocal operations to catch long-distance spatial features from the original LR input; thus, we can take advantage of the low-level features to improve performance.

MSB: To optimize the sparse local features in a convolutional module with different scales, we propose the MSB to

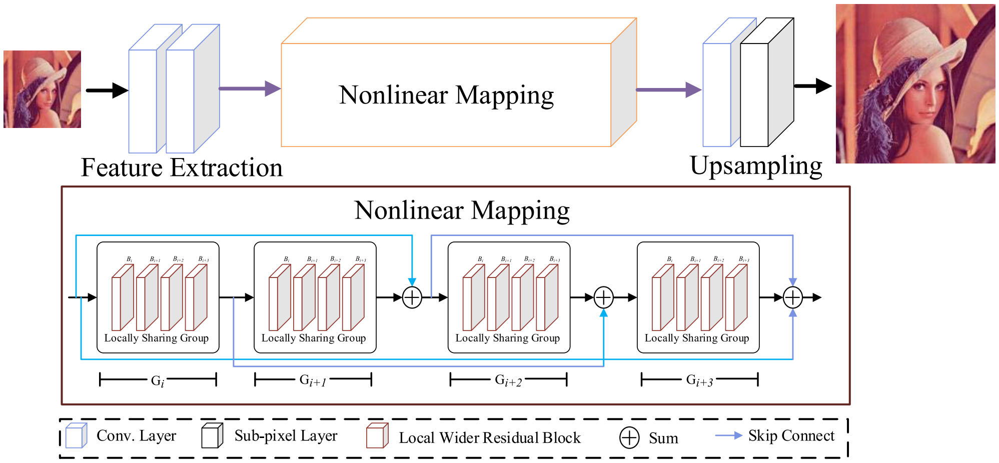


Fig. 2. Architecture of the CRN. Our model consists of a low-level feature extraction module for extracting information from original input, nonlinear mapping subnetwork for enhancing representation ability, and upsampling convolutional layers for upsampling feature maps and images. The blue arrows indicate the cascading connections.

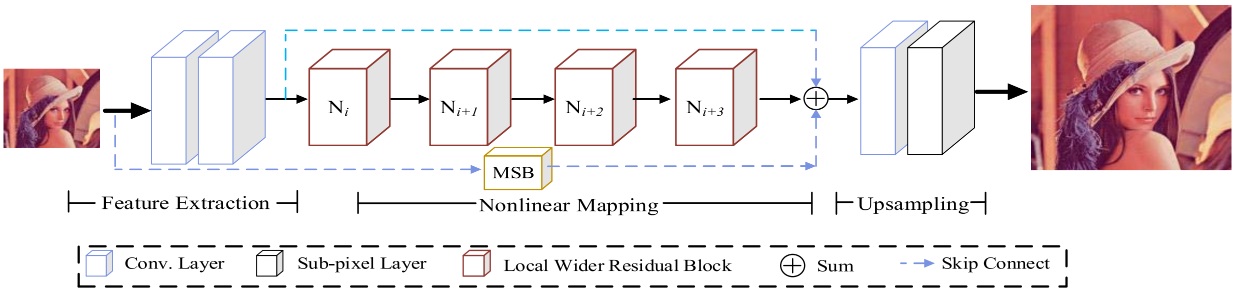


Fig. 3. Detailed network architecture of the ERN. Our model has two parallel branches, where the first branch exploits input data I^{LR} to gain high-level feature maps, and the second branch extracts hierarchical information from the original image to catch low-level representations. Then, fusing those features to recover the final high-resolution result.

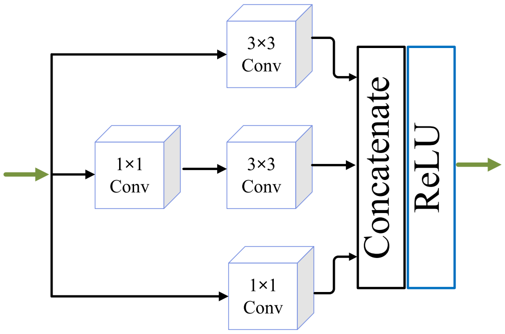


Fig. 4. Structure of the MSB. We use three parallel branches with different kernel sizes to exploit input data, and then concatenate them in the channel dimension for multiscale feature representations.

exploit the low-level information. This feature extraction block consists of three Conv layers with different kernel sizes, and the results of these Conv kernels are concatenated. In addition, we take the multiscale output to employ elementwise feature fusion with F_{map}. The ablation study reveals that benefits are achieved in the restoration.

E. Discussion

To further clarify the significance of the proposed models, this part discusses the differences between our models and the existing related ones.

1) Difference With Respect to CARN: Although the proposed CRN and CARN [1] are both based on ResNet [11], there are some differences between them. The CARN model is mainly constructed on the local and global cascading modules. The output of cascading blocks is cascaded into the higher layers. A cascading block contains several Residual-E blocks and 1×1 convolutional layers that are much more complex than the counterpart of our model. In CRN, each group is stacked with several blocks without extra connections and the block is only based on residual learning. Fewer connections undoubtedly mean fewer operations. For the global cascading connections, the output of each cascading block flows into all of the subsequent 1×1 convolutional layers via shortcut connections in CARN. However, our proposed model has different rules to use cascading connections. Specifically, the output of the low-level feature extraction module connects to the last group and one of the intermediary groups. The feature maps of the intermediary group pass to the following group in a specific gap.

2) Difference With Respect to EDSR: There are three main differences between the proposed ERN and EDSR [25]. The first difference is the design of the basic residual block. In EDSR, it utilizes the same wider input/output channel within the block, and this behavior comes with a large number of parameters. However, in ERN, we only expand the feature maps before the ReLU activation layer. Experiments revealed that this simple alteration leads to an advanced performance while reducing the parameters. The second difference lies in that there is no MSB in the EDSR model. Considering the multiscale features of the LR image in final reconstruction, we introduce the block to fully extract hierarchical features with different kernel sizes. The third difference concerns the component modification as follows: 1) our model stacks two Conv layers to extract low-level features; 2) we use relatively few modules in the non-linear mapping stage but obtain comparable feature representation ability; and 3) we simplified the reconstruction part that includes a Conv layer and an upscale layer.

3) Difference With Respect to WDSR-A: In addition to the different choice of the normalization layer, that is, WN used in WDSR-A [52], we mainly conclude another three differences between WDSR-A and our ERN network. First, the low-level feature extraction module is dissimilar. WDSR-A simply extracts the low-level feature by a single Conv layer while we stack two Conv layers to enlarge the receptive field of the hierarchical features. Second, we introduce a dual global structure because this approach is more effective to catch long-distance spatial features from the original LR input and promote the propagation of the gradient. In contrast, WDSR-A only considers a residual path. Third, WDSR-A uses a single convolutional layer with a 5 × 5 Conv kernel that directly detects low-level features of the original image. However, in ERN, we utilize a MSB that consists of different kernel sizes. We find that this modification improves the accuracy of our proposed SR model.

IV. Experimental Results

In this section, we first describe the implementation and training details of the proposed models and, then, we briefly depict the used benchmark datasets as well as the strategy to generate the LR images; model analysis follows this step. Finally, we compare our models with several state-of-the-art algorithms on four benchmark datasets.

A. Implementation and Training Details

In the proposed models, we set 3 × 3 as the filter size of all convolutional layers except those in the low-level feature extraction module and the multiscale branch. For the cascading model, experiments showed that the mapping module with $G = 4$ groups and $B = 4$ blocks led to a better performance. Meanwhile, from our observations, the mapping part of ERN that consists of $N = 16$ LWRBs possessed great representation ability.

We chose L1 loss as our loss function instead of L2 loss to train our models. The L2 loss has been widely used in the SR task due to its close connection with PSNR. However, recent work [25] indicated that L1 loss provides more powerful accuracy and convergence. During the training process, we use a batch size of 16 with size 96 × 96. Each epoch employs 1000 iterations of backpropagation. For optimization, we use the ADAM [20] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The learning rate is initially set to 1e-4 for all layers and is decreased to half every 200 epochs for a total of 850 epochs. It takes about two days to train the proposed models while EDSR takes eight days. All experiments were implemented with PyTorch and training on an NVIDIA Tesla P100 GPU.

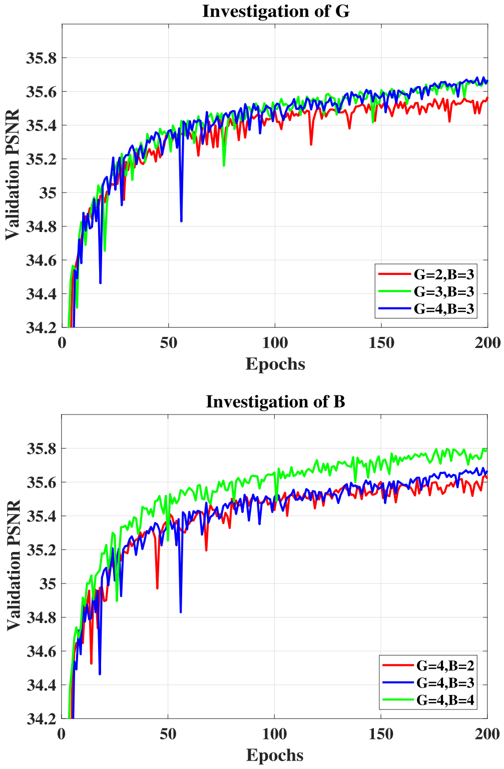


Fig. 5. Training results with different values of G and B. We investigate the linear combinations of G and B. An empirical formula $G = B = 4$ is a good tradeoff between performance and efficiency through this article.

B. Datasets

The DIV2K dataset [42] is a new high-resolution RGB image dataset with a large diversity of contents that includes 800 training images, 100 validation images, and 100 test images, respectively. In this article, we train the proposed models with 800 training images and select ten validation images to evaluate in the training process. During testing, we use four standard benchmark datasets: 1) Set5 [3]; 2) Set14 [54]; 3) B100 [2]; and 4) Urban100 [13]. The Set5 [3], Set14 [54], B100 [2] testsets mainly consist of natural scenes (i.e., landscapes, animals, and flowers) and the Urban100 [13] set collects 100 urban scenes images with a variety of real-world structure.

Following the previous work [58], two widely used quality metrics, PSNR and SSIM, are calculated on the final I^{SR} images on the Y channel of the transformed YCbCr color space. I^{LR} is downscaled from the corresponding I^{HR} image using bicubic downsampling.

C. Model Analysis

1) Comparison on Different Network Depths: In this section, we thoroughly investigate the basic parameters of our proposed models. For the CRN model, we present a comparison of the different numbers of group (G), block (B). As shown in Fig. 5, we first set $G = 2, 3, 4$ and $B = 3$ to investigate

TABLE I

QUANTITATIVE EVALUATION OF THE LINEAR COMBINATIONS OF G AND B. WE TEST THE PROPOSED MODEL WITH DIFFERENT G AND B ON THE SET5 AND SET14 DATASETS FOR ×2 SR

Model	Depth	Set5	Set14
G2B2	12	37.88	33.45
G2B3	16	38.04	33.62
G2B4	20	38.05	33.69
G3B2	16	37.98	33.60
G3B3	22	38.04	33.67
G3B4	28	38.11	33.78
G4B2	20	38.02	33.67
G4B3	28	38.12	33.77
G4B4	**36**	**38.17**	**33.84**

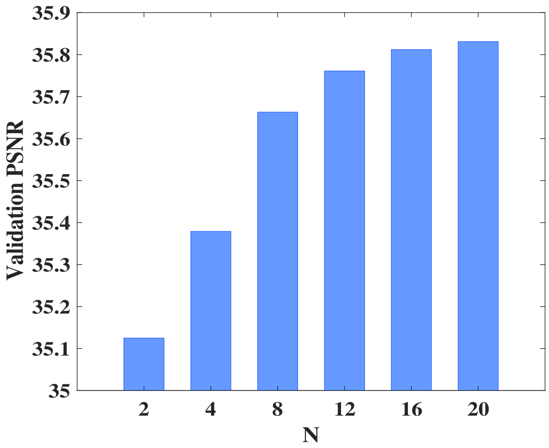


Fig. 6. Quantitative evaluation of the number of LWRBs. We build ERN with different network depth by varying the values of N on the validation dataset for ×2 SR.

the choice of G and then fixed $G = 4$ and $B = 2, 3, 4$ to explore the selection of B. The PSNR results on the DIV2K validation image with a scale factor of 2 describe the fact that a larger G and B can boost the performance. Meanwhile, we present the quantitative evaluation of different network depth in Table I. While the $G3B4$ and $G4B3$ models perform comparably, the $G4B4$ method achieves the best reconstruction accuracy. Therefore, we chose $G = B = 4$ to obtain a balance between performance and depth.

For the ERN model, we studied the network depth by varying the number of LWRBs (denoted as N for short). We set $N = 2, 4, 8, 12, 16, 20$, and the experimental results (the best performance on the validation dataset within 200-epoch training) are shown in Fig. 6. In general, the deep network achieves better results than the low-level ones; however, it is worth noting that the growth of PSNR is significantly less when $N > 10$ (e.g., it only increased by 0.019 dB when N increased from 16 to 20.) Under a certain parameter budget, we chose $N = 16$ for our SR network because the PSNR value is approximately equivalent to the state-of-the-art models, and it achieves 35.812 dB for ×2, which is better than the results 0.433 dB and 0.149 dB at $N = 4, 8$, respectively.

2) Effect of Multiscale Block: To demonstrate the effect of the MSB in the ERN model, we set up an ablation study with two scenarios: 1) with the block and 2) without the block.

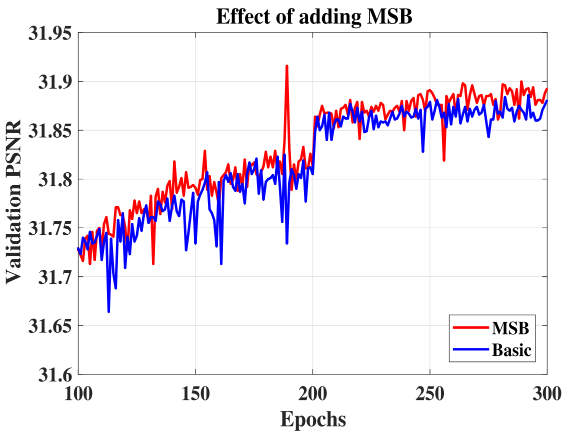


Fig. 7. Adding MSB can enhance the final results. The curves are based on the PSNR (dB) on DIV2K(val) (×3) in 300 epochs.

TABLE II

ABLATION STUDY OF NORMALIZATION LAYER. WE TRAIN THE PROPOSED MODELS WITH WN OR NO NORMALIZATION AND OBSERVE PERFORMANCE (PSNR/SSIM) DROP ON TWO BENCHMARKS: B100 AND URBAN100 WITH SCALING FACTORS 3 AND 4

Scale	Model	WN	B100	Urban100
×3	CRN	×	29.20/0.8081	28.62/0.8620
		✓	29.22/0.8083	28.63/0.8621
	ERN	×	29.21/0.8080	28.61/0.8614
		✓	29.20/0.8081	28.59/0.8614
×4	CRN	×	27.66/0.7395	26.44/0.7967
		✓	27.67/0.7397	26.45/0.7976
	ERN	×	27.70/0.7398	26.43/0.7966
		✓	27.65/0.7396	26.39/0.7963

For the formal expression, we denote the former model as MSB and the latter model as Basic. Fig. 7 presents the PSNR values of Basic, indicating they are relatively low on the validation dataset with a scaling factor ×3 under the same configuration. The statistical results of SSIM also show a matching trend. These comparisons demonstrate that MSB can improve the performance of super-resolution.

3) Number of Parameters: We show comparisons about the performance and number of parameters in Fig. 8. Compared with these mentioned algorithms, our models have considerable advantages. Although our networks have only approximately a quarter of the parameters of EDSR [25], they achieve roughly similar results on the benchmark B100(×4). Meanwhile, in comparison to MRSN [24], the state-of-the-art method proposed recently also has thousands of parameters, our CRN and ERN models increased by 0.14 dB and 0.18 dB, respectively. Moreover, our models achieve much better performance regarding PSNR and SSIM than the low-level networks, such as VDSR [18] and LapSRN [21]. This evidence indicates that our networks attain a better tradeoff between performance and model size.

4) Impacts of the Normalization Layer: We train our models with a WN layer and achieve advanced performance while removing all WN layers; the models gain comparable results

TABLE III
QUANTITATIVE EVALUATION OF THE STATE-OF-THE-ART SR METHODS. WE SHOW THE AVERAGE PSNR/SSIM FOR ×2, ×3, AND ×4 SR. *Red/blue* TEXT MEANS THE BEST/SECOND−BEST PERFORMANCE, RESPECTIVELY

Algorithm	**Scale**	**Parameters**	**Set5**	**Set14**	**B100**	**Urban100**
Bicubic	2	-	33.65/0.9300	30.34/0.8700	29.56/0.8440	26.88/0.8410
A+ [44]	2	-	36.54/0.9540	32.40/0.9060	31.22/0.8870	29.23/0.8940
SRCNN [7]	2	57K	36.65/0.9540	32.29/0.9030	31.36/0.8880	29.52/0.8950
FSRCNN [9]	2	12K	36.99/0.9550	32.73/0.9090	31.51/0.8910	29.87/0.9010
VDSR [18]	2	665K	37.53/0.9580	32.97/0.9130	31.90/0.8960	30.77/0.9140
LapSRN [21]	2	813K	37.52/0.9590	33.08/0.9130	31.80/0.8950	30.41/0.9100
EDSR [25]	2	40.7M	38.10/0.9602	33.91/0.9118	32.31/0.9013	32.93/0.9351
SRMDNF [55]	2	1.51M	37.79/0.9600	33.32/0.9150	32.05/0.8980	31.33/0.9200
CARN [1]	2	1.59M	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256
MSRN [24]	2	5.89M	38.08/0.9605	33.74/0.9170	32.23/0.9013	32.22/0.9326
CRN (ours)	2	9.47M	38.17/0.9610	33.84/0.9203	32.30/0.9012	32.69/0.9334
ERN (ours)	2	9.48M	38.18/0.9610	33.88/0.9195	32.30/0.9011	32.66/0.9332
Bicubic	3	-	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349
A+ [44]	3	-	32.60/0.9080	29.24/0.8210	28.30/0.7840	26.05/0.7980
SRCNN [7]	3	57K	32.76/0.9080	29.41/0.8230	28.41/0.7870	26.24/0.8000
FSRCNN [9]	3	12K	33.15/0.9130	29.53/0.8260	28.52/0.7900	26.42/0.8070
VDSR [18]	3	665k	33.66/0.9210	29.77/0.8340	28.83/0.7980	27.14/0.8290
LapSRN [21]	3	813K	33.82/0.9207	29.89/0.8304	28.82/0.7950	27.07/0.8298
EDSR [25]	3	43.1M	34.63/0.9280	30.53/0.8462	29.25/0.8093	28.80/0.8653
SRMDNF [55]	3	1.53M	34.12/0.9250	30.04/0.8370	28.97/0.8030	27.57/0.8400
CARN [1]	3	1.59M	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8439
MSRN [24]	3	6.07M	34.38/0.9262	30.34/0.8395	29.08/0.8041	28.08/0.8554
CRN (ours)	3	9.49M	34.60/0.9286	30.48/0.8455	29.20/0.8081	28.62/0.8620
ERN (ours)	3	9.50M	34.62/0.9285	30.51/0.8450	29.21/0.8080	28.61/0.8614
Bicubic	4	-	28.42/0.8100	26.10/0.7040	25.96/0.6690	23.15/0.6590
A+ [44]	4	-	30.30/0.8590	27.43/0.7520	26.82/0.7100	24.34/0.7200
SRCNN [7]	4	57K	30.49/0.8620	27.61/0.7540	26.91/0.7120	24.53/0.7240
FSRCNN [9]	4	12K	30.71/0.8650	27.70/0.7560	26.97/0.7140	24.61/0.7270
VDSR [18]	4	665k	31.35/0.8820	28.03/0.7700	27.29/0.7260	25.18/0.7530
LapSRN [21]	4	813K	31.54/0.8850	28.19/0.7720	27.32/0.7280	25.21/0.7560
EDSR [25]	4	43.7M	32.46/0.8968	28.80/0.7876	27.71/0.7420	26.64/0.8033
SRMDNF [55]	4	1.55M	31.96/0.8930	28.35/0.7770	27.49/0.7340	25.68/0.7730
CARN [1]	4	1.59M	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837
MSRN [24]	4	6.33M	32.07/0.8903	28.60/0.7751	27.52/0.7273	26.04/0.7896
CRN (ours)	4	9.51M	32.34/0.8971	28.74/0.7855	27.66/0.7395	26.44/0.7967
ERN (ours)	4	9.53M	32.39/0.8975	28.75/0.7853	27.70/0.7398	26.43/0.7966

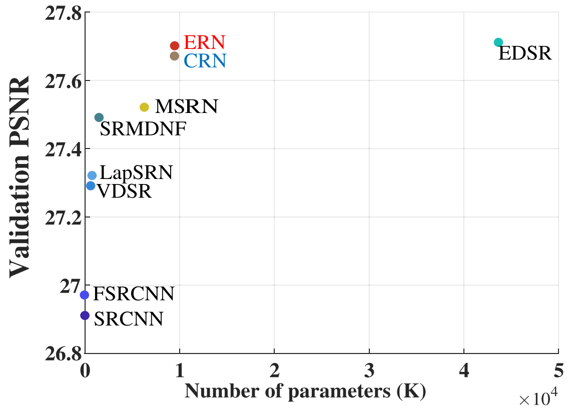


Fig. 8. Performance versus number of parameters. The results are calculated on the B100 dataset with a scale factor of 4. The proposed methods stride a balance between reconstruction accuracy and parameters.

on the test dataset. Regarding the CRN model, training with the WN layers obtains slightly better results than without normalization layers. Unfortunately, the ERN network shows an opposite trend when we carry out the same experiment. Specifically, from the results on ×3 and ×4 enlargement, the former model with the WN layers is only better by 0.02 dB and 0.01 dB on test datasets with an upsampling factor of 3, respectively. The ×4 upscaling results are similar. When we train the ERN network without using the normalization layers, it outperforms the same model with the WN layers for ×3 and ×4 enlargement on different datasets. Therefore, we chose to remove all WN layers from our models. As shown in Table II, this modification does not degrade the performance but saves the computational resources and memory usage.

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

Finally, we compared our proposed networks with nine state-of-the-art methods: 1) A+ [44]; 2) SRCNN [7]; 3) FSRCNN [9]; 4) VDSR [18]; 5) LapSRN [21]; 6) EDSR [25]; 7) SRMDNF [55]; 8) CARN [1]; and 9) MSRN [24]. These methods are evaluated on four aforementioned datasets as in the technical literature.

Table III illustrates the performance of all the above algorithms. It can be observed that our networks outperform the comparative models by a large margin on different scaling factors except EDSR [25]. The performance of CRN and ERN are entirely close to or even better than those of EDSR on some datasets, but the number of parameters of EDSR is about four times that of CRN or ERN.

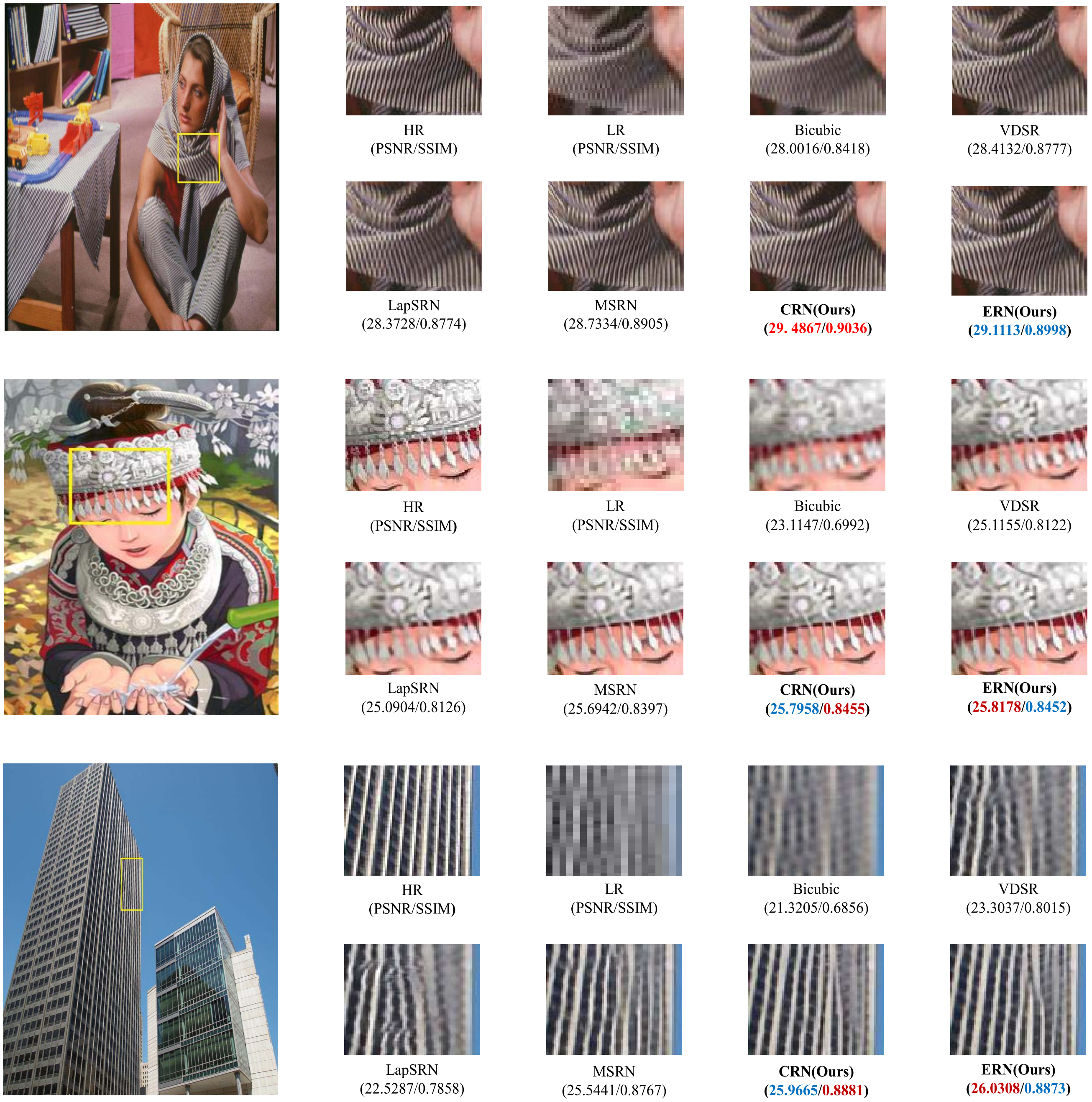


Fig. 9. Visual comparison on benchmark testsets. From top to bottom are ×2, ×3, and ×4 super-resolved results, respectively. The SR results are for images *barbara* and *comic* from Set14 and *img*_096 from Urban100. Our methods tend to generate more faithful and clear details.

1) Results on Set5: Our models outstrip the current state-of-the-art networks on ×2 enlargement and obtain an even larger margin of improvements for other upsampling factors except EDSR [25], which is only better by 0.01 dB and 0.07 dB than that obtained by our ERN model, respectively.

2) Results on Set14: Similar to the aforementioned results, for all upscaling factors, the ERN network achieves 33.88 dB, 30.51 dB, and 28.75 dB, which is better by 0.14 dB, 0.17 dB, and 0.15 dB than that achieved by MSRN [24], respectively. In addition, ERN achieves an improvement of approximately 0.32 dB over CARN [1] on different scales.

3) Results on B100: On this dataset, the CRN model achieves superior performance in terms of PSNR and SSIM for different enlargements. In detail, an average increase of approximately 0.6 dB using the proposed method was achieved over the deeper networks such as [18] and [21].

4) Results on Urban100: The Urban100 dataset consists of 100 building images. As stated in [10], EDSR tends to recover regular shapes, such as stripes or circles, and the basic elements in Urban100 are these patterns. Therefore, it achieves approximately 0.2 dB higher than CRN for all enlargements. Undoubtedly, our methods outperform other models by a large margin.

In Fig. 9, we present the visual results of four representative algorithms (Bicubic, VDSR, LapSRN, and MSRN) and the proposed ones. All methods here are tested on different upscaling factors, and the test images are selected from the Set14 and Urban100 datasets. The corresponding PSNR and

SSIM values are also reported for each method. From our observations, most of the comparative models tend to produce blurred edges. In contrast, the proposed networks can recover shapes and clear images. These obtained results indicate that the LWRBs are able to gather more information and the cascading connection or MSB fully uses the low-level features.

V. Conclusion

In this article, we proposed two CNN architectures, namely, CRN and ERN, to address the SISR problem. Compared with the existing CNN-based models, the proposed CRN takes account of the cascading mechanism to boost feature fusion and gradient propagation, while the ERN employs a dual global pathway to catch long-distance spatial features from the original LR input. Extensive benchmark evaluations showed that our proposed models present both quantitative and visible improvements compared with the previous state-of-the-art methods.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valued comments and constructive suggestions that significantly improved the quality of this article.

References

[1] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 256–272.

[2] 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, May 2011.

[3] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–10.

[4] C. L. P. Chen, L. Liu, L. Chen, Y. Y. Tang, and Y. Zhou, "Weighted couple sparse representation with classified regularization for impulse noise removal," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4014–4026, Nov. 2015.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[6] D. Dai, Y. Wang, Y. Chen, and L. Van Gool, "Is image super-resolution helpful for other vision tasks?," in *Proc. IEEE Win. Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–9.

[7] 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.

[8] 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.

[9] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 391–407.

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

[11] 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.

[12] 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, pp. 2261–2269.

[13] 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.

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, Lille, France, Jul. 2015, pp. 448–456. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html

[15] J. Jiang, Y. Yu, S. Tang, J. Ma, A. Aizawa, and K. Aizawa, "Context-patch face hallucination based on thresholding locality-constrained representation and reproducing learning," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 324–337, Jan. 2020.

[16] J. Jiang, Y. Yu, Z. Wang, S. Tang, R. Hu, and J. Ma, "Ensemble super-resolution with a reference dataset," *IEEE Trans. Cybern.*, to be published.

[17] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, no. 6, pp. 1153–1160, Dec. 1981.

[18] 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.

[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] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[21] 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.*, 2017, pp. 624–632.

[22] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Fast and accurate image super-resolution with deep laplacian pyramid networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2599–2613, Nov. 2018.

[23] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.

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

[25] 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. (CVPR) Workshops*, Jul. 2017, pp. 1132–1140.

[26] L. Liu, L. Chen, C. L. P. Chen, Y. Y. Tang, and C. M. Pun, "Weighted joint sparse representation for removing mixed noise in image," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 600–611, Mar. 2017.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[28] P. Luo, J. Ren, Z. Peng, R. Zhang, and J. Li, "Differentiable learning to-normalize via switchable normalization," in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, May 2019. [Online]. Available: https://openreview.net/forum?id=ryggIs0cYQ

[29] A. Marquina and S. J. Osher, "Image super-resolution by TV-regularization and Bregman iteration," *J. Sci. Comput.*, vol. 37, no. 3, pp. 367–382, 2008.

[30] V. Nekrasov, C. Shen, and I. D. Reid, "Light-weight refinenet for real-time semantic segmentation," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Newcastle upon Tyne, U.K., Sep. 2018, p. 125. [Online]. Available: http://bmvc2018.org/contents/papers/0494.pdf

[31] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: A technical overview," *IEEE Signal Process. Mag.*, vol. 20, no. 3, pp. 21–36, May 2003.

[32] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput. Assist. Intervent. (MICCAI)*, 2015, pp. 234–241.

[33] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 901–909.

[34] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4510–4520.

[35] W. Shi *et al.*, "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.

[36] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2008, pp. 1–8.

[37] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, Feb. 2017, pp. 4278–4284. [Online]. Available: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806

[38] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[39] 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. 2790–2798.

[40] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Oct. 2017, pp. 4549–4557.

[41] K. Tang, Z. Su, Y. Liu, W. Jiang, J. Zhang, and X. Sun, "Subspace segmentation with a large number of subspaces using infinity norm minimization," *Pattern Recognit.*, vol. 89, pp. 45–54, May 2018.

[42] R. Timofte *et al.*, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jul. 2017, pp. 1110–1121.

[43] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 1920–1927.

[44] R.Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, 2015, pp. 111–126.

[45] R. Timofte, R. Rothe, and L. Van Gool, "Seven ways to improve example-based single image super resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1865–1873.

[46] 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.

[47] X. Wang *et al.*, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, Sep. 2018, pp. 63–79.

[48] 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.

[49] Y. Wen, B. Sheng, P. Li, W. Lin, and D. D. Feng, "Deep color guided coarse-to-fine convolutional network cascade for depth image super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 994–1006, Feb. 2019.

[50] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[51] W. Yang, X. Zhang, Y. Tian, W. Wang, J. Xue, and Q. Liao, "Deep learning for single image super-resolution: A brief review," *IEEE Trans. Multimedia*, vol. 21, no. 12, pp. 3106–3121, Dec. 2019.

[52] J. Yu, Y. Fan, J. Yang, N. Xu, X. Wang, and T. S. Huang, "Wide activation for efficient and accurate image super-resolution," 2018. [Online]. Available: arXiv:1808.08718.

[53] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2018–2025.

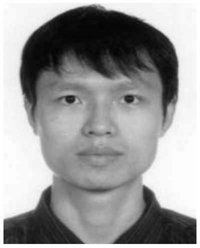
[54] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Curves and Surfaces*, J.-D. Boissonnat *et al.*, Eds. Heidelberg, Germany: Springer, 2012, pp. 711–730.

[55] K. Zhang, W. Zuo, and L. Zhang, "Learning a single convolutional super-resolution network for multiple degradations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 3262–3271.

[56] Y. Zhang, F. Shi, J. Cheng, L. Wang, P.-T. Yap, and D. Shen, "Longitudinally guided super-resolution of neonatal brain magnetic resonance images," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 662–674, Feb. 2019.

[57] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 294–310.

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



Rushi Lan received the B.S. and M.S. degrees from the Nanjing University of Information Science and Technology, Nanjing, China, and the Ph.D. degree from the University of Macau, Macau, China.

He is currently an Associate Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. His research interests include image classification, image restoration, and medical image processing.

Long Sun received the B.S. degree from the Yunnan University of Finance and Economics, Kunming, China, in 2018. He is currently pursuing the M.S. degree with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China.

His current research interests include image/video restoration, computational photography, and machine learning.

Zhenbing Liu received the B.S. degree from Qufu Normal University, Qufu, China, and the M.S. and Ph.D degrees from the Huazhong University of Science and Technology, Wuhan, China.

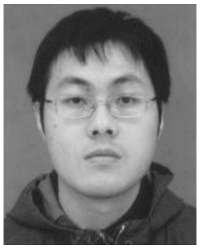
He was a Visiting Scholar with the Department of Radiology, University of Pennsylvania, Philadelphia, PA, USA, in 2015. He is currently a Professor and a Doctoral Supervisor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. His main research interests include image processing, machine learning, and pattern recognition.

Huimin Lu received the M.S. degree in electrical engineering from the Kyushu Institute of Technology, Kitakyushu, Japan, and Yangzhou University, Yangzhou, China, in 2011, and the Ph.D. degree in electrical engineering from the Kyushu Institute of Technology in 2014.

From 2013 to 2016, he was a JSPS Research Fellow with the Kyushu Institute of Technology, where he is currently an Associate Professor and an Excellent Young Researcher of MEXT, Tokyo, Japan. His research interests include computer vision, robotics, artificial intelligence, and ocean observation.

Zhixun Su received the B.S. degree in mathematics from Jilin University, Changchun, China, the M.S. degree in computer science from Nankai University, Tianjin, China, and the Ph.D. degree from the Dalian University of Technology, Dalian, China.

He is currently a Professor with the School of Mathematical Sciences and the Director of the Key Laboratory of Computational Geometry, Graphics and Images, Dalian University of Technology. His research interests include computer graphics, image processing, computational geometry, and computer vision.



Cheng Pang received the B.S. degree in computer science and the M.S. and Ph.D. degrees in computer technology from the Harbin Institute of Technology, Harbin, China, in 2011, 2013, and 2018, respectively.

He is currently with the Faculty of the Guilin University of Electronic Technology, Guilin, China. His interests include pattern recognition, image processing, machine learning, and computer vision.

Xiaonan Luo received the B.S. degree in computational mathematics from Jiangxi University, Nanchang, China, the M.S. degree in applied mathematics from Xidian University, Xi'an, China, and the Ph.D. degree in computational mathematics from the Dalian University of Technology, Dalian, China.

He is currently a Professor with the School of Computer and Information Security, Guilin University of Electronic Technology, Guilin, China. He received the National Science Fund for Distinguished Young Scholars granted by the National Natural Science Foundation of China. He was the Director of the National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou, China. His current research interests include computer graphics, machine learning, and pattern recognition.