Image Superresolution via Dense Discriminative Network

Jiayi Ma, Xinya Wang, and Junjun Jiang

***Abstract*—Deep convolutional neural networks have recently made a considerable achievement in the single-image superresolution (SISR) problem. Most CNN architectures for SISR incorporate long or short connections to integrate features, and treat them equally. However, they neglect the discrimination of features, and consequently, achieving relatively poor performance. To address this problem, in this article, we propose a dense discriminative network that is composed of several aggregation modules (AM). Specifically, the AM merges extraction and integration nodes in a tree structure, which can aggregate features progressively in an efficient way. In particular, we compress and rescale the densely connected information in the aggregation node by modeling the interaction between channels, which shares the same insight with the attention mechanism for improving the discriminative ability of network. Extensive experiments conducted on several publicly available datasets have demonstrated the superiority of our model over state-of-the-art in objective metrics and visual impressions.**

***Index Terms*—Aggregation, attention mechanism, convolutional neural network (CNN), densely connection, superresolution (SR).**

I. Introduction

SINGLE-IMAGE superresolution (SISR) aiming to reconstruct a high-resolution (HR) image from a single low-resolution (LR) input image, has been widely studied in computer vision ranging from medical and satellite imaging to security and surveillance imaging. At present, high-performance and low-cost superresolution (SR) techniques are still in high demand by many related industrial applications such as the video industry and display device industry [1].

SISR is essentially to recover high-frequency details from low-frequency data, and it constitutes an ill-posed and challenging problem due to the inevitable loss of information in the image degradation process. To solve this problem, various methods have been investigated in the past decades, including interpolation-, reconstruction-, and learning-based methods [2]. Learning-based methods using a large number of extra images attempts to generate a mapping function from LR to HR, such as manifold learning [3], sparse coding [4], linear regression [5], [6], random forest [7], and convolutional neural networks (CNNs) [8]–[12], [12], [13]. Possessing the strong nonlinear expressiveness, CNN-based methods have become increasingly popular in recent years for solving the aforementioned ill-posed problem. To learn a nonlinear LR–HR mapping, Dong *et al.* [8] first introduced CNN into SISR in an end-to-end manner, called super resolution convolutional neural network (SRCNN), which has shown its superiority to nondeep learning methods even if it relies on just a small receptive field. Subsequently, residual learning is introduced in the CNN-based method by incorporating long or short and multipath skip connections to deepen networks, such as VDSR [9], DRCN [11] and DRRN [12]. In general, a deeper and wider network holds better nonlinearity expressiveness and larger receptive fields, and many effective building models for SR have achieved better performance, such as EDSR [14], MemNet [15] and RDN [16]. However, as the network growing deeper, the features extracted by convolutional layers would be hierarchical [16], and how to make full use of such features to recover more details is still an open problem.

In the recent past, to better propagate information, researchers have been focusing on designing a network connecting the features densely. DCSCN [17] first introduced the densely connection in extraction network by only one dense block, and subsequently skip connections between several dense blocks in SRDenseNet [13] were utilized, performing better reconstruction results. In order to further improve the performance, MemNet [15] introduced the unit gate in dense blocks for mining persist memory and stacked several memory blocks with a densely connected structure. Furthermore, the local fusion is employed for dense blocks in RDN [16] and all outputs of the blocks are collected for reconstruction. Nevertheless, the existing scenarios typically involve two major disadvantages. On the one hand, when the extracted features from different outputs are densely connected for the next operation, they are treated equally assuming that each feature map have the same contribution for the new representations, which neglects the discrimination of the information. On the other hand, the densely connection in the above methods are always arranged in a chain structure, while the pyramidal structure

Manuscript received January 6, 2019; revised April 24, 2019; accepted July 21, 2019. Date of publication August 14, 2019; date of current version March 4, 2020. This work was supported by the National Natural Science Foundation of China under Grants 61773295 and 61971165. *(Corresponding author: Junjun Jiang.)*

J. Ma and X. Wang are with the Electronic Information School, Wuhan University, Wuhan, 430072, China (e-mail: jyma2010@gmail.com; wangxinya@whu.edu.cn).

J. Jiang is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001 China, and also with Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: junjun0595@163.com).

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

Digital Object Identifier 10.1109/TIE.2019.2934071

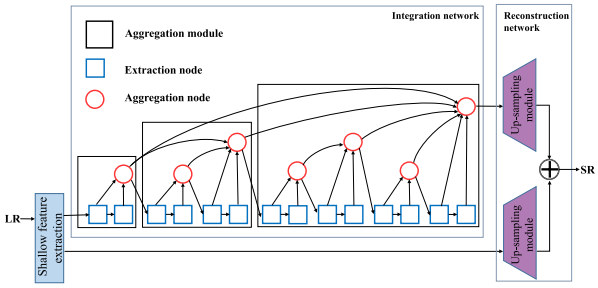


Fig. 1. Whole structure of the proposed SRDDN.

could deepen the representations by nonlinear and progressive fusion [18].

In this article, we proposed a dense discriminative network for SR named SRDDN to address the aforementioned drawbacks. In particular, the aggregation module (AM) is set as the building module for SRDDN, in which we unify the aggregation and extraction nodes in a tree structure to learn richer combinations from the feature hierarchy. The aggregation node collect the output of the extraction nodes and explicitly model nonlinear correlations between channels. Before sending features to the next stage, this node would learn to select features of relative importance by attention mechanism, so that the network is discriminative and simultaneously offers a more efficient use of computation resource. Sharing the same insight with pyramidal structure, our SRDDN progressively merges the hierarchical features through several AMs, as shown in Fig. 1. Rather than concating the features at different levels in a chain structure, we increase the depth of the AM step by step and progressively aggregate and deepen the representations. After extracting dense hierarchical features, we conduct global fusion to reconstruct the high-frequency details in an effective way. Our method is evaluated on standard benchmark datasets, which outperforms state-of-the-art approaches in terms of peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM).

Contributions: Our main contributions include the following three aspects. First, we propose a novel dense discriminative network for SISR. This network aggregates dense hierarchical features progressively in a tree structure, which is able to outperform the state-of-the-art. Second, we introduce the attention mechanism for feature selection in the aggregation node. Therefore, the accumulated features of more importance are emphasized and preserved for deep representation, enhancing the discriminative learning ability. Due to the effective use of the dense features, our SRDDN has a small network scale. Third, we upscale the shallow features in the LR space and fuse them with deep feature by global residual learning for reconstruction.

The remainder of this article is organized as follows. Section II describes background material and related work. In Section III, we describe the proposed SRDDN in detail. We discuss the differences between our approach and the state-of-the-art in Section IV. Section V illustrates the performance of our method in comparison with other approaches on several publicly available datasets. Section VI concludes this article.

II. Related Works

Over the years, SISR algorithms have been extensively studied in the literature. In the following, we briefly review the learning-based methods, which can be divided into three main categories [19]. In addition, we also discuss the attention mechanism that our approach bases on.

A. SR Algorithms Based on Internal Databases

Several methods [20]–[22] for SR utilize the self-similarity property of the natural image and erect scale-space pyramid of the input image for LR–HR matching pairs. Although the internal dictionary generated from the input image is more relevant than that in external database, the number of LR–HR patch pairs is inadequate for matching patches containing complex structures.

To address the drawback, Singh *et al.* [23] decomposed local image structure into different sub-band component patches and found matches independently instead of seeking similar patches directly in the image domain. Huang *et al.* [24] expanded the internal patch search space by incorporating the perspective geometry and affine transformations. These SR algorithms based on internal databases are time consuming for searching patches, which is not feasible for real-time applications.

B. SR Algorithms Based on External Databases

The SR methods based on external datebases attempt to learn a mapping function from LR to HR images by supervised machine learning algorithms, including nearest neighbor [25], manifold embedding [3], sparse representation [4] and kernel ridge regression [26]. Rather than fitting the entire data space, recent methods group the external datasets into clusters by K-means [27], sparse dictionary [6] or random forest [7], and find linear regression for each cluster. While these approaches are effective and efficient, the extracted features and mapping functions are manually designed, which may not be optimal for generating high-quality SR images.

C. CNN-Based Methods for SR

Most recently, CNN-based methods have achieved significant improvement against the aforementioned methods. Dong *et al.* [8] initiated the SRCNN to solve the SISR problem in an end-to-end manner. To further improve the performance, Kim *et al.* [9] stacked 20 convolutional layers, and residual learning combined with gradient clipping was used to ease the training difficulty of the deep network. Then, recursive learning and multipath connections in DRCN [11] and DRRN [12] were employed to construct a wider network. However, these methods should preprocess the LR image to the desired HR space before feeding them into the network, which not only put extra demands on computation, but also lose some details of the original image. Consequently, a deconvolutional layer in FSRCNN [28] and a subpixel layer in ESPCN [29] were exploited for unsampling after extracting features from LR input. These methods have been proven effective to increase the

spatial resolution and replace preprocessing operation. Recently, Huang *et al.* [10] proposed a novel network called DenseNet, which passing all its previous feature maps to the later layers by concatenating them. This dense connection was utilized in DCSCN [17], SRDenseNet [13], MemNet [15], RDN [16] and DBPN [30]. Although there are several distinction among them, features from different level were concatenated directly, and equalized for extraction or reconstruction. Some generalized features might be trivial during the intermediate procedure but be treated equally, which causes the computational resource consumption and constrains the discriminative expression ability.

In our method, after densely connecting operation, we propose a mechanism that allows the aggregation node to perform feature selection, emphasizing informative features, and suppressing less useful ones, which can improve the representational power of the network.

D. Attention Mechanism

Numerous CNN-based methods for SISR have derived innovations from high-level vision tasks. Recently, attention mechanism is popular in these areas. Attention principle focuses on special parts of the signal to conduct computation, rather than using all available information, a large part of it being irrelevant. Specifically, it can be regarded as an effective way to allocate available resources. A similar idea, concentrating on informative parts of the inputs has been applied in deep learning for speech recognition [31], translation [32] and reasoning [33]. In recent years, attention mechanism followed by a gate function was employed in vision tasks to allocate the processing resources toward the most informative components. For image classification, Wang *et al.* [34] introduced a trunk-and-mask attention mechanism into the deep residual networks. Hu *et al.* [35] posed a squeeze-and-excitation block with a lightweight gating mechanism to learn channel-wise dependencies, which can be integrated for residual and inception module. For image generation, Parmar *et al.* [36] integrated self-attention mechanism into an autoregressive model for image generation, and prepixel attention has been explored in the context of generative adversarial networks to model the internal representation of images [37]. In contrast, attention mechanism can also be effective in low-level vision tasks.

Our proposed aggregation node is specialized to learn channel-wise relationships for densely connected features, which can be viewed as a reweighed gate. When connecting the feature hierarchies, feature selection has been performed, through which it can learn to use the global information to ease the learn process, and significantly enhance the discriminative power of the network.

III. Proposed Method

In this section, we describe the proposed SRDDN in detail, including the network architecture, loss function, and implementation details.

A. Overall Architecture

As shown in Fig. 1, our SRDDN consists of three parts: low-level feature extraction network, integration network, and reconstruction network. The output is I^{SR} when a LR image I^{LR} is taken as the initial input. First, we adopt 64 filters in a single convolution layer to extract low-level feature F_0. Each map of the F_0 display specific feature response, such as corners, edges, color conjunctions, and so on

$$F_0 = H_{3\times3}(I^{\mathrm{LR}}) \tag{1}$$

where $H_{3\times3}$ denotes the convolutional operation with kernel size 3×3. The nature of the features in the network is hierarchical [38] and high-level feature could give more clues for reconstruction. Subsequently, the shallow feature F_0 is sent to the integration network, as the rich feature F_{RF} is obtained as

$$F_{\mathrm{RF}} = A(F_0) \tag{2}$$

where A denotes the operation of the deep integration network, which contains several AMs, merging aggregation nodes, and extraction nodes. After extracting rich high-level features, global residual fusion is used for reducing the complexity of the task for integration network. Therefore, we upscale the shallow and deep features into HR space separately in the reconstruction network by

$$F_{\mathrm{GUF}} = U(F_0) \tag{3}$$

$$F_{\mathrm{RUF}} = U(F_{\mathrm{RF}}) \tag{4}$$

where U donates the upscale module, F_{GUF} and F_{RUF} are the global and residual upscaled feature, respectively. There are several alternatives for upscaling, such as deconvolutional layer in [28], nearest-neighbor upsampling followed by convolution in [39], and subpixel layer in [29]. Such postupscaling strategy has been demonstrated to be more efficient for computation complexity and can achieve higher performance than preupscaling SR methods. We choose the ESPCN in [29] followed by a convolutional layer. Finally, the SR result is reconstructed by

$$I^{\mathrm{SR}} = F_{\mathrm{GUF}} + F_{\mathrm{RUF}}. \tag{5}$$

B. Integration Network

Due to the fact that hierarchical feature combinations are beneficial for reconstruction when the network is growing deeper [16], we introduce a hierarchical deep aggregation tree as the integration network, as shown in Fig. 1. Specifically, in basic AM, extraction node and aggregation node are assembled in a tree-structured way to aggregate different levels of representation. To acquire the deep representation, we increase the depth of the AM step by step instead of connecting them densely in a chain structure. Particularly, inspired by deep layer aggregation (DLA) in [18], the output of the middle aggregation node is sent to the backbone to merge deeper feature iteratively. In each AM, the last aggregation node has access to all previous outputs, which makes sure the information propagation and progressively merges the deep representation. Our integration network shows great importance on the short paths and reuse,

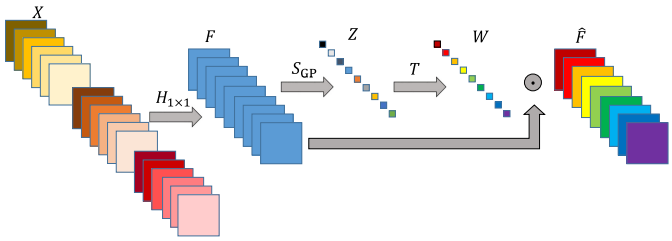


Fig. 2. Structure of the aggregation node.

and extends skip connections with a tree that crosses different levels rather than concatenation.

1) Aggregation Node: When the features are densely connected in a channel-wise way and used for the next expression, they are treated equally supposing all of them contribute to the new extraction. However, whether this assumption is true or not remains to be seen. Sharing the same insight of the attention mechanism, the aggregation node aims to merge representations from different levels and select informative features simultaneously, as shown in Fig. 2. At the beginning, cross-channel correlations are mapped as new combinations of features, jointly by using standard convolutional filters with 1×1 kernel. To decrease the parameters of the network, the dimension of the input is halved. Given an input X, consisting of the densely connected features from different extraction nodes, the output is obtained by

$$F = H_{1\times1}(X). \tag{6}$$

Due to the fact that low-frequency information extracted in LR space is abundant even trivial, taking all features for information abstraction would consume too much resource. In order to make the network distinguish the informative feature, we should find the global representation of each feature map and exploit the channel-wise interaction. First, considering a single unit in each feature map generated from the learned filter with a local receptive field, it cannot acquire the contextual information outside this region. Therefore, global average pooling is utilized to collect the global statistics into a descriptor for expressing the whole image. Let $F = [f_1, \ldots, f_C]$ be the input, which has C feature maps with size of $H \times W$. The channel-wise statistic $Z = [z_1, \ldots, z_C] \in R^C$ is then generated by shrinking F through the spatial dimension $H \times W$

$$Z = S_{GP}(F) \tag{7}$$

where S_{GP} represents the channel-wise global average pooling operation. Specifically, the cth element of z is obtained by

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} f_c(i,j), \quad c = 1, \ldots, C \tag{8}$$

where $f_c(i,j)$ is the value at the position (i,j) of the cth feature f_c. Second, for capturing channel-wise dependencies, the nonlinear and nonmutually-exclusive interactions should be learned adaptively to ensure the discrimination on multiple channels [35]. It can be formulated as

$$W_c = \mathcal{T}(Z) \tag{9}$$

where $\mathcal{T}$ denotes the modeling process. Different from the fully connected (FC) layers in [35], we employ two convolutional layers followed by a sigmoid function to acquire the final channel-wise weight. In more detail

$$W_c = s(H_{1\times1}(\sigma(H_{1\times1}(Z)))) \tag{10}$$

where s and σ represent the sigmoid and ReLU activations, respectively. The sigmoid function can allow the multiple channels to be emphasised instead of one-hot activation. To further control the parameter of the network, we set a reduction factor r for the feature map of two layers. It means when the input Z has C channels, the output feature map of the first convolutional layer is reduced to $\frac{C}{r}$ and that of the second layer is restored to C for consistency. Finally, the feature F is reweighed along the cth channel to perform feature selection through emphasizing informative features and suppressing less useful ones

$$\hat{F} = F \cdot W_c. \tag{11}$$

Through the aggregation node, we adaptively distinguish the dense feature for the new representation, which enhances the discrimination of the whole network.

2) Extraction Node: Over the years, numerus building blocks have been developed for extracting features, ranging from residual blocks in SRResNet [40] and EDSR [14] to dense blocks in SRDesNet [13] and RDN [16]. These basic blocks are all compatible in our extraction node. In order to ease the training process, we use the simplified residual blocks in EDSR [14]. In the extraction node, if we have G residual blocks, donate F_{in} as the original input, the residual block abstracts the feature by

$$F_{\text{LF}} = R_G(R_{G-1} \cdots (R_2(R_1(F_{\text{in}})))) \tag{12}$$

where R_G and F_{LF} represent the operation of residual block and local feature, respectively. After extracting sufficient local features, local residual learning is introduced to further improve the information flow. The final output F_{out} of the extraction node can be obtained by

$$F_{\text{out}} = F_{\text{LF}} + F_{\text{in}}. \tag{13}$$

C. Loss Function

Our method is optimized with ℓ_1 loss function instead of ℓ_2. In general, minimizing ℓ_2 is preferred to maximize the value of PSNR. However, ℓ_1 loss provides better convergence than ℓ_2 [14]. Given the training set $\{(I_i^{\text{LR}}, I_i^{\text{HR}})\}_{i=1}^N$, which consists of N LR patches and corresponding HR ones. Thereby, the goal of training SRDAN is to minimize the ℓ_1 loss between the SR results and their corresponding HR counterparts

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left\| I_i^{\text{SR}} - I_i^{\text{HR}} \right\|_1. \tag{14}$$

D. Implementation Details

In the aggregation node, the number of the output maps is cut in half of the input. That is to say, the input X in (6) has

$2C$ feature maps. The reduction factor is set to 16 which does not resulting in too many parameters for the whole network. In the extraction node, we set the number of residual blocks as 12, i.e., $G = 12$. In each residual block, the filter size keeps the same with the input for local fusion. In the shallow feature extraction network, the convolutional layer has 64 filters. As we output color images, the last layer in up-sampling module is equipped with three filters. To keep the size of all the feature maps identical to the input image before the upscaling layer, zero padding is used around the boundary of each input before the convolutional operation. For the upscaling process, we use ESPCN followed by a convolutional layer to upscale the coarse resolution features to the expected ones.

IV. Discussions

A. Difference to DLA

Inspired from DLA [18], we adopt the hierarchical deep aggregation as the AM in our proposed integration network. In general, DLA is used for high-level computer vision tasks (e.g., visual recognition). While the integration network is designed to aggregate hierarchical feature for SISR. Moreover, we remove the downsampling operation between the AM, as the downsampling operation could discard some pixel-level information for reconstruction, and hence hinder performance of the network for SR problem. Furthermore, the aggregation node in DLA is compatible with any layer and block. While in our network, we redefine the aggregation node with attention mechanism for selecting features, where the collected features are reweighed depending on their correlations. As a result, we can emphasize the important features and restrain the useless ones, which would enhance the discrimination of the network. Last but not least, we densely concatenate the output of the AM to fully use hierarchical information, which has been neglected in DLA.

In view of these changes, more pixel-level details would persist by the integration network, which is suitable for the SISR problem. The qualification of the aggregation node makes the network discriminative to the features, weakening the influence of the useless feature for reconstruction and producing more faithful results.

B. Difference to EDSR

There are three main differences between EDSR [14] and our SRDDN. The first one is the design of basic building block. EDSR simplifies the residual block from SRResNet [40] and repeats them in the chain structure to deepen the network. In our method, we employ the residual block in the extraction node and local residual fusion is utilized to further encourage the flow of information and gradient. In addition, the extraction nodes and the aggregation nodes are assembled in a tree structure to integrate the representations progressively. Through the progressive aggregation, more hierarchical features are fully exploited and emphasised so that we can acquire better results than EDSR. The second one is that there is no concatenation in EDSR. Instead, we make the output of the AM concatenate for feature reuse. The low-level features are beneficial for the extraction of high-level features. Therefore, the densely connection could deepen the representation and improve the performance of our method. The third one is that EDSR adopts global residual learning before the upscaling module. Whereas we upscale the shallow features and the deep features, respectively, before using global residual fusion for reconstructing the SR result, which could ease the training process.

C. Difference to RCAN

In spite of the same use of the channel attention, we mainly summarize some differences between RCAN and our SRDDN. Sharing the same whole framework with EDSR, RCAN directly connects the residual channel attention blocks (RCAB) in the residual in residual structure so the last two points of difference to EDSR also exit in that to RCAN. The most important point is that RCAN adopt the channel attention in the residual structure to construct the RCAB. In this way, the interaction that learned by attention mechanism is related to the features from the same level. While, SRDDN equips the aggregation node with attention mechanism in a tree structure and it is expected to select the informative features and restricted the useless ones for next expression. The hierarchical nature of the features could make the network more discriminative.

V. Experimental Results

To evaluate the performance of the proposed method, in this section, we first provide the experimental settings, and then give an experimental model analysis and discuss the attention effect. Subsequently, we conduct qualitative and quantitative comparisons with the state-of-the-art competitors. Finally, we test the performance on degradation model.

A. Experimental Settings

Our training data include 800 training images from the DIV2K dataset, same as that in EDSR [14] and RDN [16] without argumentation. DIV2K [41] is a newly collected dataset of RGB images with a large diversity of contents, which is divided into: 800 images for training, 100 images for validation, 100 images for testing. We utilize the bicubic downsampling method at the corresponding scale factor to obtain the LR training patches with reference to existing SR methods [8], [9]. We randomly extract 16 color patches with the size of 48×48 for each training iteration. Our model is optimized by ADAM [42] with $\beta_1 = 0.9$, $\beta_2 = 0.9999$, $\epsilon = 10^{-8}$. The initial leaning rate is set to 10^{-4} and then decreases to half every 2×10^5 iterations of back-propagation. We use PyTorch to implement our models with a GTX 1080Ti and the initialization strategy is the default method in PyTorch, which means we adopt the uniform distribution to initialize the parameter according to the method described in [43]. Four publicly available datasets are used for evaluation including Set5 [44], Set14 [45], BSD100 [46], and Urban100 [24]. As we process the color image for both training and testing, the SR results are evaluated with PSNR and SSIM on Y channel (i.e., luminance) of transformed YCbCr space, which are calculated in MATLAB.

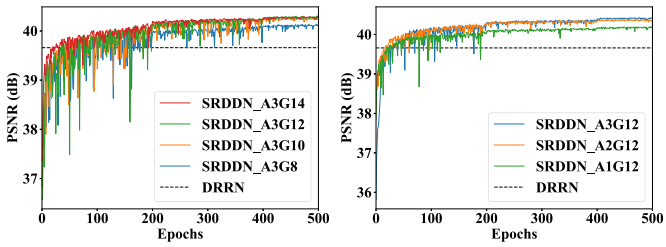


Fig. 3. PSNR learning curves of our method at different G (left) and A (right) conducted at scale factor $\times 2$.

TABLE I
PERFORMANCE ON DIFFERENT SETTINGS OF AGGREGATION NODE EVALUATED ON DIV2K VALIDATION DATASET AT SCALE FACTOR $\times 4$

	SRDDN	SRDDN (no attention)
Loss	5.46	5.68
PSNR (dB)	34.328	34.210

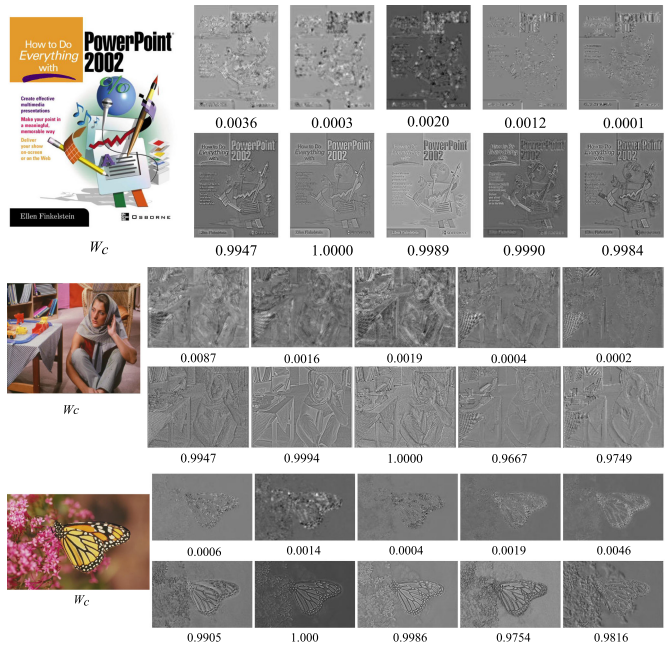


Fig. 4. Examples of W_c and corresponding feature maps of three pictures "ppt," "barbara," and "butterfly" in Set14.

B. Model Analysis

The effective way to deepen our integration network is to increase the number of the residual blocks in the extraction node. Therefore, we investigate the influence of the hyperparameter G, the number of which is increased from 8 to 14. We use the performance of DRRN [12] as a reference. The convergence curves of the PSNR value during training process up to 500 epoches are shown in Fig. 3. From the results, we observe that as G increases from 12 to 14, the convergence of and PSNR is quite similar, and $G = 14$ performs slightly better. Compared with the performance of $G = 12$, the model of $G = 14$ has faster convergence rate but it ultimately reach to nearly the same level. Therefore, we set $G = 12$ as the default value to balance the parameter and the performance.

In order to explore the effectiveness of the AM in the integration network, we also conduct the experiment on different number of AM with the same setting $G = 12$, donated as different A in the right one of the Fig. 3. As observed in the graph, when we adopt three AMs, the model is converged slower than others but it ultimately achieve better performance. Furthermore, even if we only use one AM with nearly 50 layers in the whole network, our method still outperform the DRRN [12].

C. Attention Effect

To explore the effect of the aggregation node in our method, we replace the attention mechanism for comparison. That is, we discard the global average pooling and the reweighing operation in the aggregation node and use the three same convolutional layers followed by the activation function. As shown in Table I, the SRDDN model with the attention mechanism achieves better performance than the model without the attention mechanism at the same level of model size. In particular, the PSNR value of the SRDDN is improved from 34.210 to 34.328 by adding attention mechanism. This is mainly because the network could learn the interdependencies between features and enjoy more freedom to choose them [37]. Therefore, more useful features would be employed for reconstruction. In addition, the comparison of ultimate loss demonstrates that the attention mechanism could contribute to the convergence.

To better understand the effectiveness of the aggregation node, we test our model on the dataset Set14 and show several typical feature maps with attention values in Fig. 4. We randomly select the value of W_c close to 0 or 1 and visualize their corresponding feature maps in the last aggregation node, since the output of this node is employed for reconstruction. In each image, the color one is the HR image and the grey maps are the extracted feature maps before rescaling by the weights. We observe that the network learns to allocate attention according to the texture. For instance, in the image "ppt3," the aggregation node attends to keep the feature map whose texture details are sharp and clear and sets the W_c close to 1. As for the W_c close to 0, some features are interpreted by the extra noise and the others have less details, and hence, after feature selection, these feature maps are restrained. In the image "barbara," it can be seen that the map partly with remarkable information would also be remained and the weights of blurred feature are learned to 0. These observations further demonstrate that the attention mechanism works well in our network.

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

We compare our SRDDN with six state-of-the-art SR methods, including SRCNN [8], VDSR [9], DRRN [12], LapSRN [47], EDSR [14] and RDN [16]. We do not compare our model with MDSR in [14] and RCAN [48]. The formal uses the multiscale patches as the training input, which allow the network to grasp more information from different scales. The later trains a very deep network up to 400 layers to acquire high performance. As we concentrate on the effectiveness of the network, the deeper network and multiscale inputs are not adopted in our

TABLE II
PUBLIC BENCHMARK TEST RESULTS (PSNR/SSIM) AT SCALE FACTORS ×2, ×3, ×4

Dataset	Scale	Bicubic	SRCNN [8]	VDSR [9]	DRRN [12]	LapSRN [47]	EDSR [14]	RDN [16]	SRDDN	SRDDN+
Set5	×2	33.65/0.9299	36.66/0.9542	37.53/0.9590	37.74/0.9591	37.52/0.9591	38.11/0.9602	38.23/0.9613	38.21/0.9612	38.27/0.9615
	×3	30.39/0.8682	32.75/0.9090	33.67/0.9210	34.03/0.9244	33.82/0.9227	34.65/0.9280	34.70/0.9295	34.72/0.9297	34.80/0.9300
	×4	28.42/0.8104	30.48/0.8626	31.35/0.8830	31.68/0.8888	31.54/0.8855	32.46/0.8968	32.46/0.8988	32.51/0.8987	32.67/0.9005
Set14	×2	30.24/0.8688	32.45/0.9067	33.05/0.9130	33.23/0.9136	33.08/0.9130	33.92/0.9195	34.00/0.9212	33.90/0.9203	34.06/0.9213
	×3	27.55/0.7742	29.30/0.8215	29.78/0.8320	29.96/0.8349	29.79/0.8320	30.52/0.8462	30.56/0.8468	30.56/0.8465	30.68/0.8482
	×4	26.00/0.7027	27.50/0.7531	28.02/0.7680	28.21/0.7721	28.19/0.7720	28.80/0.7876	28.80/0.7870	28.81/0.7876	28.93/0.7896
BSD100	×2	29.56/0.8431	31.36/0.8879	31.90/0.8960	32.05/0.8973	31.80/0.8950	32.32/0.9013	32.33/0.9016	32.33/0.9015	32.38/0.9021
	×3	27.21/0.7385	28.41/0.7863	28.83/0.7990	28.95/0.8004	28.82/0.7973	29.25/0.8093	29.25/0.8092	29.25/0.8093	29.33/0.8100
	×4	25.96/0.6675	26.90/0.7101	27.29/0.7260	27.38/0.7284	27.32/0.7280	27.71/0.7420	27.71/0.7418	27.72/0.7416	27.80/0.7432
Urban100	×2	26.88/0.8403	29.50/0.8946	30.77/0.9140	31.23/0.9188	30.41/0.9101	32.93/0.9351	32.87/0.9325	32.91/0.9351	33.13/0.9369
	×3	24.46/0.7349	26.24/0.7969	27.14/0.8290	27.53/0.8378	27.07/0.8272	28.80/0.8653	28.79/0.8652	28.81/0.8658	29.01/0.8682
	×4	23.14/0.6577	24.52/0.7221	25.18/0.7540	25.44/0.7638	25.21/0.7553	26.64/0.8033	26.62/0.8027	26.66/0.8040	26.90/0.8090

Red: the best.

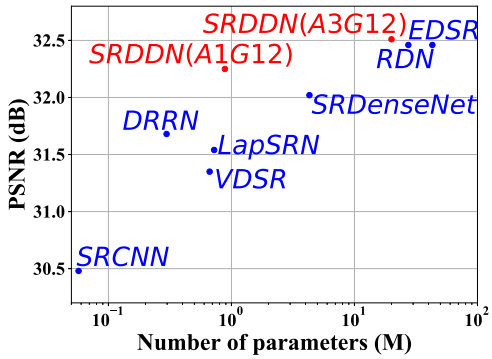


Fig. 5. Number of parameters in different models evaluated at scale factor ×4.

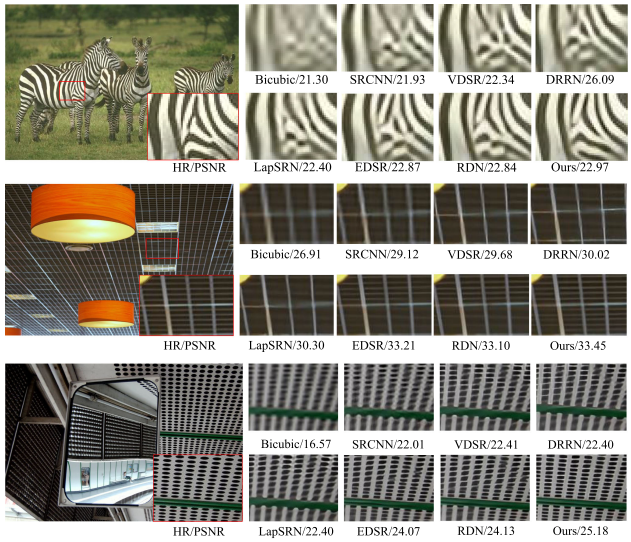


Fig. 6. Qualitative comparison of SR results for "253027" (top) from BSD100, "img044" (middle), and "img004" (bottom) from Urban100 at scale factor ×4.

method. Similar to [14], [16], we also adopt the self-ensemble strategy [14] to further improve our SRDDN and denote the self-ensembled SRDDN as SRDDN+. The source code provided by the authors are conducted to generate the SR results. Two widely used image quality metrics PSNR and SSIM are evaluated on the SR results at the upscaling factors 2, 3, and 4. Four publicly available datasets are used for evaluation such as Set5 [44], Set14 [45], BSD100 [46] and Urban100 [24]. Table II reports the quantitative comparisons for these methods. Although the PSNR values of our SRDDN are only slightly better than the latest results, RDN, our model has less parameters, as shown in Fig. 5. We also visualize the number of parameters and performance of our method with one aggregation node ($A = 1$) and 12 residual blocks ($G = 12$), donated as SRDDN (A1G12). The lightweight model is superior than the comparison algorithms with the same level parameters, such as LapSRN, SRDenseNet. Therefore, our SRDDN is able to reach a better tradeoff between the model size and the performance. Moreover, our SRDDN can achieve further improvement with self-ensemble. According to Table II, our model has more advantages on the upscale factor 4. The basic idea of the proposed method is to progressively extract the hierarchical features. Therefore, on the one hand, through the extraction of hierarchical features, it can be ensured that features of different levels are explored during reconstruction. On the other hand, due to this progressive reconstruction strategy, our method can progressively recover the missing high-frequency details in the input image.

In Fig. 6, we display visual comparisons of all methods on three representative images from BSD100 and Urban100. For the image "253027," it can be seen that most compared methods would generate noticeable artifacts and produce blurred edges. In contrast,owing to the attention mechanism for feature selection, our SRDDN could keep the texture syllabify. As for images "004" and "044" from the dataset Urban100, due to the benefit of progressive aggregation, our SRDDN can reconstruct more detail information than the others. Furthermore, we observe that the state-of-the-art methods might generate some false details in most of the reconstruction results, which do not exist in the ground truth images. Comparatively, the results of our method would produce less fake information because of the restraining on useless features in the aggregation node. Specifically, we visualize three representative comparisons in Fig. 7. For the top image, although our method does not reconstruct the barbed wire in the SR result, it does not produce the fake details and the results of our methods are more faithful to the ground truth. In images "078" and "092," we reconstruct more real structure than the others. There still exist blocking effect, detail loss and distortion in our results, since SR is a challenging task,

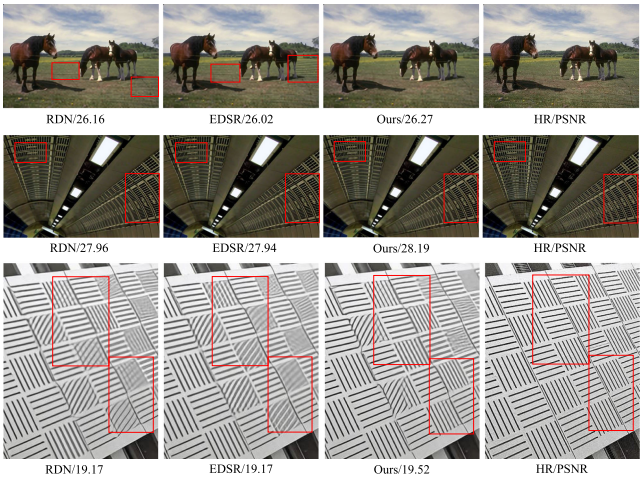


Fig. 7. Qualitative comparison of SR results for "197017" (top) from BSD100, "img078" (middle), and "img092" (bottom) from Urban100 at scale factor $\times 4$.

TABLE III
PUBLIC BENCHMARK TEST RESULTS (PSNR/SSIM) AT SCALE FACTOR $\times 3$

Testset	SPMSR [50]	IRCNN [52]	SRMD [51]	SRDDN (ours)
Set5	31.11/0.8837	33.16/0.9156	34.09/0.9243	34.69/0.9277
Set14	28.15/0.7924	29.54/0.8270	30.11/0.8364	30.62/0.8443
BSD100	27.56/0.7553	28.49/0.7885	28.98/0.8010	29.32/0.8067
Urban100	24.92/0.7545	26.47/0.8081	27.49/0.8371	28.60/0.8569

Red: the best.

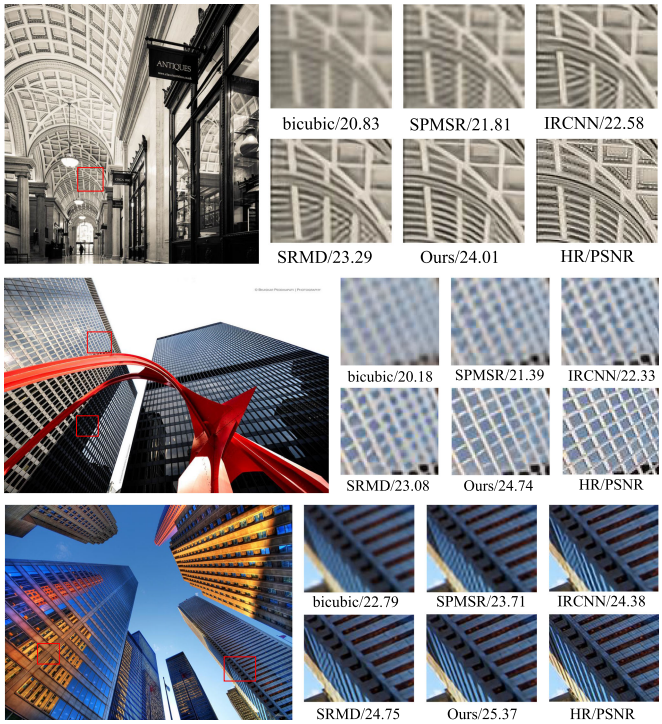


Fig. 8. Qualitative comparison of degradation model for "img083" (top), "img012" (middle), and "img062" (bottom) from Urban100 at scale factor $\times 3$.

especially perfect reconstruction of high-frequency details. However, compared with other state-of-the-art method, our method still has better visual results.

E. Performance on Degradation Model

To further show the scalability of the proposed method, we also conduct our model to superresolve another widely used degradation [49] which involves a 7×7 Gaussian kernel with width 1.6, and a direct downsampler with scale factor 3. Our SRDDN is compared with three state-of-the-art methods including SPMSR [50], SRMD [51], and IRCNN [52]. Table III shows the average PSNR and SSIM results on Set5, Set14, BSD100, and Urban100 at the scaling factor 3. Our SRDDN performs the best on all the datasets. The performance gains over other state-of-the-art methods are consistent with the visual results in Fig. 8.

VI. CONCLUSION

In this article, we proposed a novel dense discriminative network called SRDDN for SISR, which could achieve the state-of-the-art performance in terms of PSNR and SSIM. A key characteristic of our SRDDN was to integrate the extraction node and the aggregation node into a tree. This enabled us to interactively abstract the features and selectively preserve the useful ones, which not only improved the nonlinear and discriminable ability, but also decreased the model size. The qualitative and quantitative results on four publicly available datasets revealed the superiority of our method over the state-of-the-art.

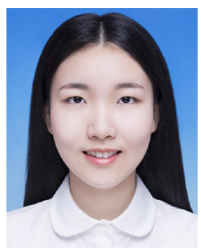
REFERENCES

[1] W. Jia, Y. Zhao, R. Wang, S. Li, H. Min, and X. Liu, "Are recent SISR techniques suitable for industrial applications at low magnification," *IEEE Trans. Ind. Electron.*, vol. 66, no. 12, pp. 9828–9836, 2019.
[2] J. Tian and K.-K. Ma, "A survey on super-resolution imaging," *Signal, Image Video Process.*, vol. 5, no. 3, pp. 329–342, 2011.
[3] H. Chang, D. Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 2004, pp. 275–282.
[4] 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.
[5] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 1920–1927.
[6] R. Timofte, V. D. Smet, and L. V. Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. 12th Asian Conf. Comput. Vision*, 2014, pp. 111–126.
[7] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 3791–3799.
[8] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 184–199.
[9] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 1646–1654.
[10] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 4700–4708.
[11] J. Kim, J. Kwon Lee, and K. Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 1637–1645.
[12] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 2790–2798.

[13] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 4809–4817.
[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. Vision Pattern Recognit. Workshops*, 2017, pp. 136–144.
[15] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 4539–4547.
[16] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 2472–2481.
[17] Y. Jin, S. Kuwashima, and T. Kurita, "Fast and accurate image super resolution by deep CNN with skip connection and network in network," in *Proc. 24th Int. Conf. Neural Inf. Process.*, 2017, pp. 217–225.
[18] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 2403–2412.
[19] R. Al-falluji, A. Youssif, and S. Guirguis, "Single image super resolution algorithms: A survey and evaluation," *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 6, pp. 2278–1323, 2017.
[20] C.-Y. Yang, J.-B. Huang, and M.-H. Yang, "Exploiting self-similarities for single frame super-resolution," in *Proc. 10th Asian Conf. Comput. Vision*, 2010, pp. 497–510.
[21] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Trans. Graph.*, vol. 30, no. 2, 2011, Art. no. 12.
[22] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Proc. IEEE 12th Int. Conf. Comput. Vision*, 2009, pp. 349–356.
[23] A. Singh and N. Ahuja, "Super-resolution using sub-band self-similarity," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 552–568.
[24] J. B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 5197–5206.
[25] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, Mar./Apr. 2002.
[26] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 1127–1133, Jun. 2010.
[27] C.-Y. Yang and M.-H. Yang, "Fast direct super-resolution by simple functions," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 561–568.
[28] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. 14th Eur. Conf. Comput. Vision*, 2016, pp. 391–407.
[29] 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. Vision Pattern Recognit.*, 2016, pp. 1874–1883.
[30] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep backprojection networks for super-resolution," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 1664–1673.
[31] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 577–585.
[32] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2015.
[33] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiský, and P. Blunsom, "Reasoning about entailment with neural attention," in *Proc. Int. Conf. Learn. Represent.*, 2016.
[34] F. Wang *et al.*, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 3156–3164.
[35] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 7132–7141.
[36] N. Parmar *et al.*, "Image transformer," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4055–4064.
[37] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7354–7363.
[38] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vision*, Springer, 2014, pp. 818–833.
[39] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *Proc. Int. Conf. Learn. Represent.*, 2017.
[40] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 4681–4690.
[41] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2017, pp. 1122–1131.
[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.
[43] 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. Vision*, 2015, pp. 1026–1034.
[44] 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. Vision Conf.*, 2012, pp. 135.1–135.10.
[45] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. 7th Int. Conf. Curves Surf.*, 2010, pp. 711–730.
[46] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE Int. Conf. Comput. Vision*, 2001, pp. 416–423.
[47] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate superresolution," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit*, 2017, pp. 624–632.
[48] 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. Vision*, 2018, pp. 286–301.
[49] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.
[50] T. Peleg and M. Elad, "A statistical prediction model based on sparse representations for single image super-resolution," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2569–2582, Jun. 2014.
[51] K. Zhang, W. Zuo, and L. Zhang, "Learning a single convolutional super-resolution network for multiple degradations," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 3262–3271.
[52] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 3929–3938.

Jiayi Ma received the B.S. degree in information and computing science and the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2008 and 2014, respectively.

From 2012 to 2013, he was an Exchange Student with the Department of Statistics, University of California at Los Angeles, Los Angeles, CA, USA. He is currently a Professor with the Electronic Information School, Wuhan University, Wuhan, China. His research interests include the areas of computer vision, machine learning, and pattern recognition.



Xinya Wang received the B.S. degree from the Electronic Information School, Wuhan University, Wuhan, China, in 2018. She is currently working toward the master's degree with the Multispectral Vision Processing Lab, Wuhan University.

Her research interests include neural networks, machine learning, and image processing.

Junjun Jiang received the B.S. degree from the Department of Mathematics, Huaqiao University, Quanzhou, China, in 2009, and the Ph.D. degree from the School of Computer, Wuhan University, Wuhan, China, in 2014.

He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. His research interests include image processing and computer vision.