This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3005025, IEEE Transactions on Multimedia

Accurate and Efficient Image Super-Resolution via Global-Local Adjusting Dense Network

Xinyan Zhang, Peng Gao, Sunxiangyu Liu, Kongya Zhao, Guitao Li, Liuguo Yin, *Member, IEEE* and ChangWen Chen, *Fellow, IEEE*

***Abstract*—Convolutional neural network-based (CNN-based) method has shown its superior performance on the image super-resolution (SR) task. However, several researches have shown that obtaining a better reconstruction result often leads to the significant increase in parameters and computation. To alleviate the burden in computational needs, we propose a novel global-local adjusting dense super-resolution network (GLADSR) to build a powerful yet lightweight CNN-based SR model. To enhance the network capacity, we present a global-local adjusting module (GLAM) which can adaptively reallocate the processing resources with local selective block (LSB) and global guided block (GGB). The GLAMs are linked with nested dense connections to make better use of the global-local adjusted features. In addition, we also introduce a separable pyramid upsampling (SPU) module to replace the regular upsampling operation, and thus brings a substantial reduction of its parameters and obtains better results. Furthermore, we show that the proposed refinement structure is capable of reducing image artifacts in SR processing. Extensive experiments on benchmark datasets show that the proposed GLADSR outperforms the state-of-the-art methods with much fewer parameters and much less computational cost.**

***Index Terms*—Image super-resolution, global-local adjusting, separable pyramid upsampling, refinement structure.**

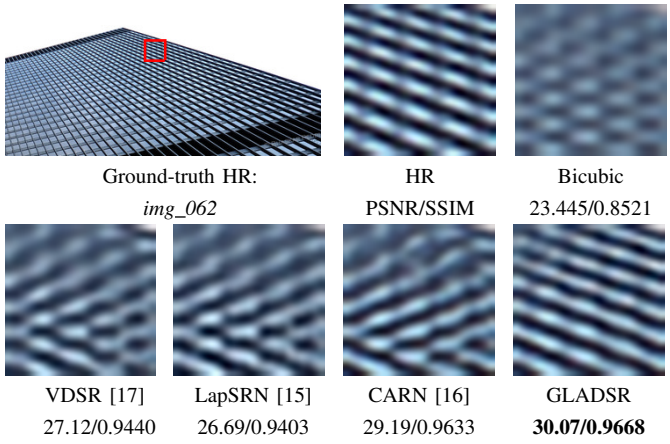


Fig. 1. Visual result of our method compared with existing methods on the image from Urban100.

I. Introduction

SINGLE image super-resolution (SR) is an image enhancement technique, aiming at reconstructing a high-resolution (HR) image from its low-resolution (LR) version. The technique has been widely used in multimedia applications, since it could be a less expensive option for zooming factors in digital cameras or imaging software [1]. SR is an ill-posed inverse procedure and therefore the solution is not unique for any LR input. To tackle this problem, many learning-based algorithms, such as Neighbor Embedding [2], Sparse Representation [3], [4], and Local Neighborhood Regression [5], [6], are proposed to learn the mapping from massive corresponding LR/HR image pairs.

In recent years, with the development of deep learning, the convolutional neural network-based (CNN-based) SR methods [7], [8], [9], [10] have shown their preferable results and thus became the mainstream approaches. However, for most CNN-based methods (e.g., EDSR [11], RCAN [12]), they can improve their performance by increasing the network depth, which results in the significant increase in parameter numbers and computational complexity. Therefore, it is still a challenge for the CNN-based SR methods to reduce the parameters and computation while keeping performance optimal. To answer this challenge, the latest works have tried to build lightweight and powerful SR models. Tai *et al.* [13] use a recurrent unit to obtain images with a few parameters. Yang *et al.* [14] also utilize recurrent residual blocks to extract multi-level features for reconstruction. Lai *et al.* [15] apply a progressive upsampling structure for recovering with large factors. Ahn *et al.* [16] introduce a cascading structure upon the residual network for better results. However, these methods also have their own limits: (1) The methods that use recurrent architecture still require heavy computational cost, although it can reduce parameters. (2) The network resources are not well balanced, since the extracted features in those models are equally treated, which restricts the overall network capacity due to resource allocation on redundant information.

To address these issues, we propose a novel CNN-based scheme, named global-local adjusting dense super-resolution network (GLADSR). The proposed scheme can significantly reduce both the parameter number and computational complexity, compared to the above methods. It is reasonable to assume that recovering a texture-rich area of the LR image by SR operation is more difficult than using the same operation in a smooth region. Therefore, an efficient SR model should allocate more computational resources to preserve detail-rich information, such as edges, corners, and textures in the images.

X.Y Zhang, S.X.Y Liu, K.Y Zhao and G.T. Li are with School of Aerospace Engineering, Tsinghua University, Beijing 100084, China. X.Y Zhang, S.X.Y Liu, K.Y Zhao and L.G. Yin are also with Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China. P. Gao is with College of Engineering, Beijing 100871, China. C.W. Chen is with Department of Computer Science and Engineering, University at Buffalo, NY 14260 USA and Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230026, China.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3005025, IEEE Transactions on Multimedia

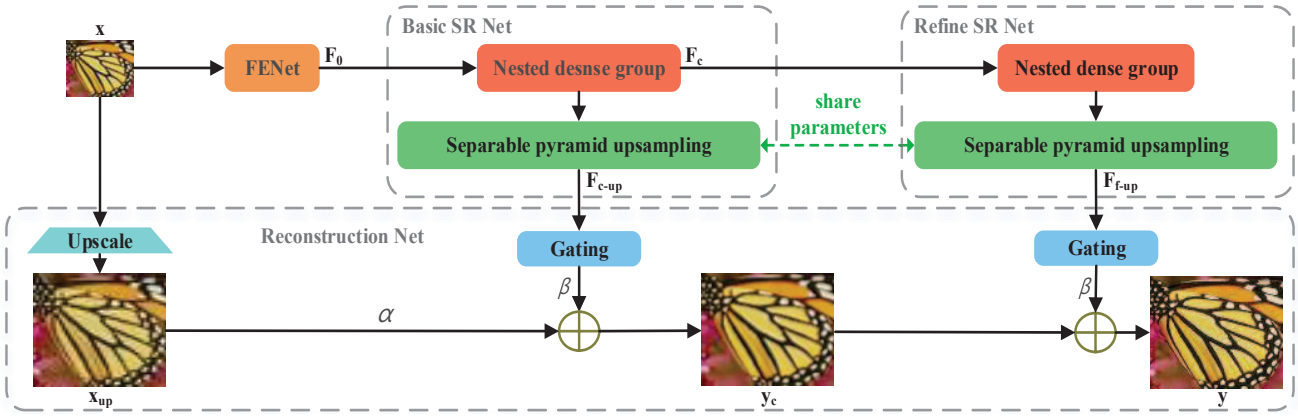


Fig. 2. Overall architecture of our global-local ajusting dense super-resolution network (GLADSR). The GLADSR consists of four parts: feature extraction net (FENet), basic SR net, fine SR net and reconstruction net. Specially, the parameters of separable pyramid upsampling (SPU) in both basic and refine SR net are shared.

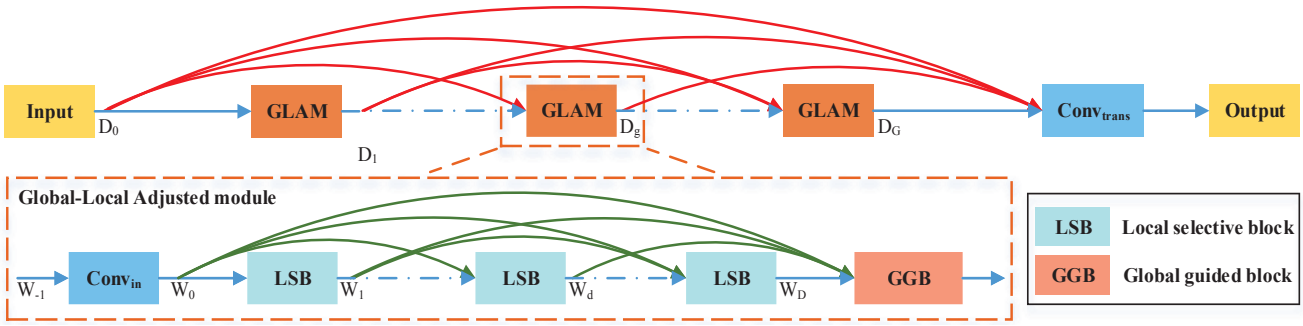


Fig. 3. Network structure of nested dense group (NDG). Global-local adjusting module (GLAM) in NDG, which is comprised of a global guided block (GGB) and multi-stacked local selective blocks (LSB), is the core of the proposed GLADSR.

This key idea and main design principle have led to the success of the proposed scheme. Specifically, the structure of GLADSR is mainly comprised of four parts: a feature extraction net (FENet), a basic SR net, a fine SR net, and a reconstruction net. The basic SR net and the fine SR net, as the core of the proposed scheme, share a similar structure. They are both comprised of a separable pyramid upsampling module (SPU) and a nested dense group (NDG). SPU is a novel upsampling design, which combines the concept of group and perception. It can obtain the upscaled features with much fewer parameters. The NDG is comprised of multiple global-local adjusting modules (GLAMs) which are linked with dense skip connections. Applying such a connecting strategy can keep information flows more flexible, and it can make full use of the adjusted features extracted by each GLAM.

As the basic module of GLADSR, the GLAM with soft-attention mechanisms is comprised of a global guided block (GGB) and multiple stacked local selective blocks (LSB) in a dense structure. GGB is designed to modify the focus regions of each module in a global way, while LSBs are designed to select more helpful features in a local way. The scheme of GLAM thus reduces the redundancy of features in the network and generates more powerful feature-maps. In the final stage of GLADSR, the input LR image is recovered by using a stepwise refinement strategy. The first step of the strategy is to generate a basic result that is directly reconstructed from the LR image. Then, in the next step, the final SR image is generated by refining the basic result from the first step. The proposed refinement structure can effectively accelerate convergence and reduce artifacts. As Fig. 1 shows, the result of the proposed GLADSR has sharper edges and fewer artifacts, compared to other state-of-the-art methods.

The main contributions of the proposed scheme can be summarized as follows,

- The proposed GLADSR is capable of achieving better performance with much fewer parameters and less computational cost than the previous state-of-the-art methods.
- The global-local adjusted module (GLAM) is a simple but powerful structure, which is capable of optimizing resource allocation to increase model capacity via local selective block (LSB) and global guided block (GGB).
- The separable pyramid upsampling (SPU) module obtains superior results while dramatically reduce the parameters. The proposed refinement structure further diminishes image distortion almost without additional parameters.

The rest of the paper is organized as follows. Related work is discussed in Section II. Section III describes the detailed design of the proposed scheme. Experimental results and discussions are presented in Section IV. Finally, we conclude this paper in Section V.

II. Related Work

Recently, deep learning (DL)-based methods have been widely adopted in super-resolution and have achieved better performance than traditional approaches [18], [19], [20], [21], [22], [23], [24]. In this section, we focus on works related to DL-based image super-resolution methods and attention mechanisms, which are closely related to the proposed scheme in this research.

A. Image Super-Resolution

As the pioneer CNN model for SR, Dong *et al.* [7] propose SRCNN, which effectively learns a non-linear mapping from LR to HR by using 3 convolutional layers. Motivated by SRCNN, Kim *et al.* [17] propose a deeper network with 20 layers named VDSR, which increases network depth with residual learning strategy. Despite their good performance, the pre-processing stage of these methods requires to upscale LR images to the target size, and this significantly increases computation complexity. To solve this problem, some approaches have been proposed to use the post-upscaling strategy in the network. ESPCN [25] directly uses the original LR image as input, and introduces a sub-pixel convolution layer (pixel-shuffle operation) for final upsampling step. Cai *et al.* [26] also utilize shuffle upsampling operation, and further learn per-pixel degradation kernels on reconstruction process for real-word SR. FSRNN [27] uses a deconvolution layer to upscale LR features at the end of the network. Shi *et al.* [28] further apply a deconvolutional module to upscale the LR feature-maps in a content-adaptive way to preserve the structural details. Hu *et al.* [29] adopt a meta-upscale module in the last step to generate the HR image of arbitrary size. In this paper, we propose a separable pyramid upsampling (SPU) module to further improve the performance of the post-upscaling step for lightweight super-resolution.

SRResnet [30] applies the residual blocks [31] in its model, which can increase the network depth with residual skip connections. EDSR [11] further improves the reconstruction performance for SR by removing the batch normalization layer in residual blocks and increasing the width of the network. Tong *et al.* [32] introduce the dense blocks from Densenet [33], which connects the features in the model with dense skip connections to alleviate the vanishing gradient problem and enhance the network ability. MMCNN [34] also adopts dense skip connections in a very deep network for feature extraction and reconstruction, and achieves promising results. Thus it can be found that skip connections in CNN models can effectively enhance the flow of information and optimize the network structure by building multi-paths from a lower level to a higher level. Instead of the above-mentioned methods, we propose a nested dense structure to make better use of the global/local features and reduce network redundancy.

Although deeper networks show better performance [11], [17], they also need more parameters. To control the model size, DRCN [35] has recourse to a recursive layer. DRRN [13] learns the residual unit recursively to further enhance the performance, and MemNet [36] stacks recursive memory blocks with a densely connected structure. Li *et al.* [37] further design an RNN with a feedback block to generate powerful high-level representations for SR. However, a common weak point shared by the above-mentioned recursive methods is that they need huge computing cost to compensate performance loss. To reduce both parameters and computational cost, LapSRN [15] employs a set of cascaded sub-networks, which progressively reconstruct the sub-band residuals in a coarse-to-fine mode. Hui *et al.* [38] introduce the information distillation network (IDN) which extracts local long and short-path features with a distillation block. Ahn *et al.* [16] present the CARN, which applies the cascading block upon the residual network.

B. Attention Mechanisms

Attention mechanism has achieved great interest and shown good performance in many tasks, such as image captioning [39], [40], image classification [41], and image detection [42]. Chen *et al.* [39] propose SCA-CNN which incorporates spatial and channel-wise attention in the multi-layers of CNN for image captioning. Hu *et al.* [41] introduce a Squeeze-and-Excitation(SE) block to model the interdependencies of channel-wise to improve network performance for image classification. CBAM [42] makes use of a spatial attention module and optimizes the SE block to compute channel attention for classification and detection tasks. However, studies on the effect of the attention mechanism for image restoration are scanty. As a case, Zhang *et al.* [12] incorporate the channel attention mechanism [41] into local residual blocks, and then propose RNAN [43] with the mixed attention to improve network performance. Dai *et al.* [44], in addition, propose a second-order channel attention to enhance network ability.

Inspired by the role of attention in those vision tasks, our approach is to enhance the network capacity by adaptively modifying the feature distribution with both spatial and channel-wise attention mechanisms. To the best of our knowledge, it is the first attempt to investigate the effect of both attention mechanisms based on the global and local patterns for image super-resolution.

III. Methodology

A. Network Architecture

As shown in Fig. 2, the proposed GLADSR is comprised of four parts: a feature extraction net (FENet), a basic SR net, a refine SR net and a reconstruction net. Here, we use x and y to denote the input LR image and the output HR image.

FENet is the first stage of GLADSRs pipeline. A convolutional layer (Conv) is employed in FENet to extract the shallow features (denoted by F_0) directly from the input x.

Then, the basic SR net uses F_0 as the input, and generates the coarse upsampled features F_{c-up}. Specifically, the nested dense group (NDG) extracts an intermediate coarse features F_c from F_0 at first, and a separable pyramid upsampling module (SPU), as a subsequent step of NDG, creates the F_{c-up}.

$$F_{c-up} = f_{spu}\left(F_c\right) = f_{spu}\left(f_{ndg\_c}\left(F_0\right)\right), \tag{1}$$

where $f_{ndg\_c}(\cdot)$, $f_{spu}(\cdot)$ denote the functions of NDG and SPU in the basic net.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3005025, IEEE Transactions on Multimedia

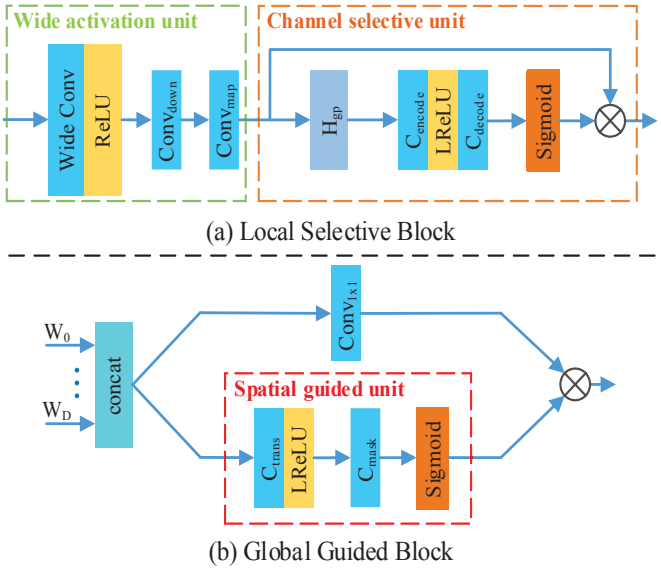


Fig. 4. The implementation of the local selective block (LSB) and global guided block (GGB).

The next step is the refine SR net. It has the same structure as the above mentioned basic SR net, and utilizes the coarse features F_c to generate the refined upsampled features F_{f-up}.

$$F_{f-up} = f_{spu}\left(f_{ndg_f}\left(F_c\right)\right), \tag{2}$$

where $f_{ndg_f}(\cdot)$ denotes function of NDG in the refine net. In addition, the parameters of SPU are shared in both basic and refine net, and more details about SPU will be given in Section III-D.

At the final stage, a refinement structure is designed for the reconstruction net. Since obtaining the HR image directly from the LR input is difficult and often has distortion, a stepwise strategy is introduced in the reconstruction net. We firstly generate a basic SR image y_c from the LR input by using F_{c-up} with residual learning strategy, and then obtain the final result y by refining the intermediate coarse image y_c. Moreover, we further use the learnable variables α, β to combine the upscaled LR image x_{up} and the residual coarse/fine images for stable training,

$$y_c = \alpha x_{up} + \beta f_{re-c}\left(F_{c-up}\right), \tag{3}$$

$$y = y_c + \beta f_{re-f}\left(F_{f-up}\right), \tag{4}$$

where $f_{re-c}(\cdot)$, $f_{re-f}(\cdot)$ denote the gating layers to reconstruct the residual basic/fine images. More analysis of the proposed refinement structure will be given in Section IV-D3.

Given a training set $\{x^{(i)}, \tilde{y}^{(i)}\}_{i=1}^{N}$, where N is the number of training images and $\tilde{y}^{(i)}$ is the corresponding ground-truth image, our GLADSR aims to minimize the $L1$ loss function with the parameter set Θ,

$$\begin{aligned}\mathcal{L}\left(\Theta\right) = &\frac{\gamma}{N}\sum_{i=1}^{N}\left\|\tilde{y}^{(i)} - y_c^{(i)}\right\|_1 \\ &+\frac{1-\gamma}{N}\sum_{i=1}^{N}\left\|\tilde{y}^{(i)} - y^{(i)}\right\|_1,\end{aligned} \tag{5}$$

where γ is the loss weight for the outputs of different stages.

B. Nested Dense Group

We now present the details of the nested dense group (NDG), which is consists of G global-local adjusting modules (GLAM) and a bottleneck layer (see Fig. 3). Each GLAM is comprised of a *global guided block* (GGB) and B *local selective blocks* (LSB) with local dense connections. Furthermore, NDG uses *nested dense connections* (NDC) to concatenate the output dense features of each GLAM, and the dense features from all GLAMs are adaptively fused with the bottleneck layer. The idea behind NDC is to further preserve the information from all the GLAMs and utilize the features more effectively. We summarize the differences between NDG and original dense network [33] in Section IV-C.

In NDG, the outputs of the previous GLAM have direct connections to all subsequent modules and blocks, which can not only fully utilize the extracted features, but also help the information flow. Local dense connections inside GLAM (green lines in Fig. 3) make full use of the adjusted features in its current module, while nested dense connections (NDC, red lines in Fig. 3) between any two GLAMs are used to further improve network performance. Specifically, the input for g-th GLAM is the concatenation of all previous GLAM outputs $[D_1, D_2, \ldots, D_{g-1}]$ and the input of NDG D_0. The NDC can learn multi-level representations in a global pattern, and reuse the dense adjusted features of each GLAM. We will also demonstrate the effectiveness of nested dense connections in Section IV-D2.

C. Global-Local Adjusting Module

Global-Local Adjusting Module (GLAM), whose structure is shown in Figure 3, is the core of the GLADSR. Specifically, a Conv operation is applied at the beginning of the GLAM, and obtains channel pooling features W_0. This operation sets the channel dimension to K_0. After that, local selective block (LSB) is stacked with dense connections, and thus the input of the d-th LSB is a concatenation of all previous outputs.

$$W_d = f_{lsb}\left(\left[W_0, W_1, \ldots, W_{d-1}\right]\right), \tag{6}$$

where $f_{lsb}(\cdot)$ denotes LSB function and W_d denotes the output of d-th LSB which has K feature-maps. Finally, GGB guides the fusion of multi-level LSB outputs,

$$D_g = f_{ggb}\left(\left[W_0, W_1, \ldots, W_D\right]\right), \tag{7}$$

where $f_{ggb}(\cdot)$ denotes GGB function, and D_g denotes g-th GLAM output which also has K_0 feature-maps. To demonstrate the effectiveness of LSB and GGB, a detailed analysis is laid on Section IV-D2. It can also be found that training the deep model with the global-local adjusting structure is more stable.

Local selective block is comprised of a wide activation unit (WAU) and a channel selective unit (CSU), as Fig. 4(a) shows. The WAU, inspired by wide activation mechanism [45], is composed of a wide Conv, a downward Conv, and a mapping Conv. The wide Conv is laid first to expand the features with the expansion factor τ, since using wider features before the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3005025, IEEE Transactions on Multimedia

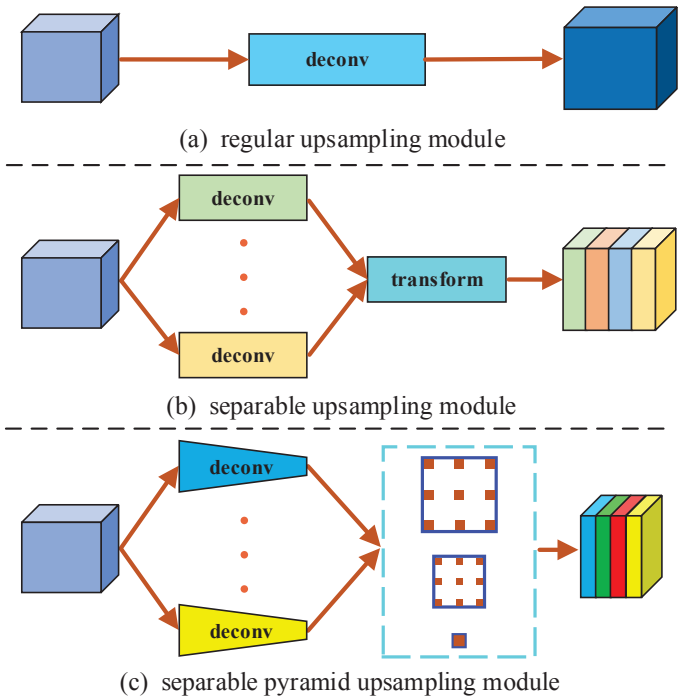


Fig. 5. The proposed separable pyramid upsampling (SPU) extracts dimension-reduced features with separble deconvolution and further enhances upscaled features with pyramid convolutions.

ReLU activation can improve the SR performance. The output dimension of the wide Conv is then denoted by K_w, which is equal to $\tau \times K$. In order to improve efficiency, the dimension of features then reduces back to K with the downward Conv, and we finally use the mapping Conv to extract spatial-wise features and enlarge receptive fields of the network.

In our design, each WAU is followed by a CSU. This design aims to improve model ability by selecting to strengthen the helpful local features, and suppressing other useless ones. We use $U = [u_1, u_2, \ldots, u_K]$ to denote the outputs of WAU. The CSU firstly obtains the statistics of input features U, and then uses an encode-decode structure to get interdependencies of the features. Specially, we adopt the Leaky ReLU (LReLU) [46] instead of the commonly-used ReLU, because it effectively avoids generating "dead features" [47] in CSU, and we can obtain more powerful and stable attention maps when the network goes deeper.

$$R_c = S\left(C_{decode}(\delta_l\left(C_{encode}(H_{gp}\left(U\right)\right)\right), \tag{8}$$

where H_{gp} denotes global pooling function, δ_l and S denote LReLU and sigmoid activation function. C_{encode} and C_{decode} are the weight of Conv, which sets channel dimension to K/r and K. R_c denotes the channel-wise selective weights, which represent the importance of corresponding features in U. Consequently, we finally combine the WAU outputs U and their corresponding weights R_c with channel-wise multiplication, and it can thus reweight the distribution of local channel-wise features with LSB.

Global guided block is illustrated in Fig. 4(b). For image super-resolution task, the input image is varied from different spatial regions, so it is helpful to guide the network focus on different regions at different levels. For example, at a higher level, the model should pay more attention to edges and textures instead of the plain regions, which are easier to be reconstructed. Therefore, to generate a spatial guided mask for multi-level features of LSBs, we design a simple but powerful unit, spatial guided unit (SGU). Specifically, we firstly use a transmission layer C_{trans} to adaptively learn weights for local hierarchical features. Then a mask layer C_{mask} is applied to get the guided mask R_s,

$$R_s = S\left(C_{mask}\delta_l\left(C_{trans}\left([W_0, W_1, \ldots, W_D]\right)\right)\right), \tag{9}$$

where C_{trans} is weight of a 1×1 Conv with channel dimension K and C_{mask} is the weight of a 3×3 Conv. The LReLU activation is also applied in SGU to obtain the spatial attention maps. Through the above learning process, the output guided mask R_s can adaptively represent the focus weights of spatial regions in each GLAM. Besides, to reduce the input dimensions of GGB ($K_0 + DK$), a Conv is employed to reduce the number of output channels for improving computing efficiency. The final output of GGB is obtained by combing the compressive input features and corresponding spatial guided mask through an element-wise multiplication. LSB is designed to take responsibility for improving local features with channel-wise relationships, while GGB is presented to further modify the features in a global way with spatial characteristics.

D. Separable Pyramid Upsampling

Deconvolution layers (deconv) have been widely used to upsample low-resolution features in recent SR methods [27], [38], [32]. Although these methods have achieved good results with deconv operation (see Figure 5(a)), the parameter numbers are still too large, especially for a lightweight SR model. Therefore, inspired by the concept of depth-wise separable convolution [48], a separable upsampling module is designed in the present work to effectively reduce parameters in the upscaling process. As shown in Figure 5(b), the designed module firstly maps the input features into ρ different groups, and upscales each group by a single deconvolution layer. Then, in the transform layer, the outputs of each group are fused to create a linear combination with a 1×1 convolutional layer.

Furthermore, we propose a novel separable pyramid upsampling module (SPU) to optimize the separable upsampling module (as shown in Fig. 5(c)). Firstly, we further reduce the parameters by decreasing the output channel dimension of each deconvolution group with ratio d, while it can also degrade upscaling performance. Therefore, to improve the output features of SPU, we then concatenate each group, and adopt the parallel spatial pyramid structure to obtain the features with different receptive fields. Specifically, we implement a three-level pyramid operation with parallel convolutional layers (two 3×3 dilated Convs with different sampling rates and a 1×1 Conv), and the dimension of each parallel output is reduced by half. The outputs extracted with multiple receptive fields can provide more powerful information for reconstruction, and we thus fuse them to generate the final output of SPU.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3005025, IEEE Transactions on Multimedia

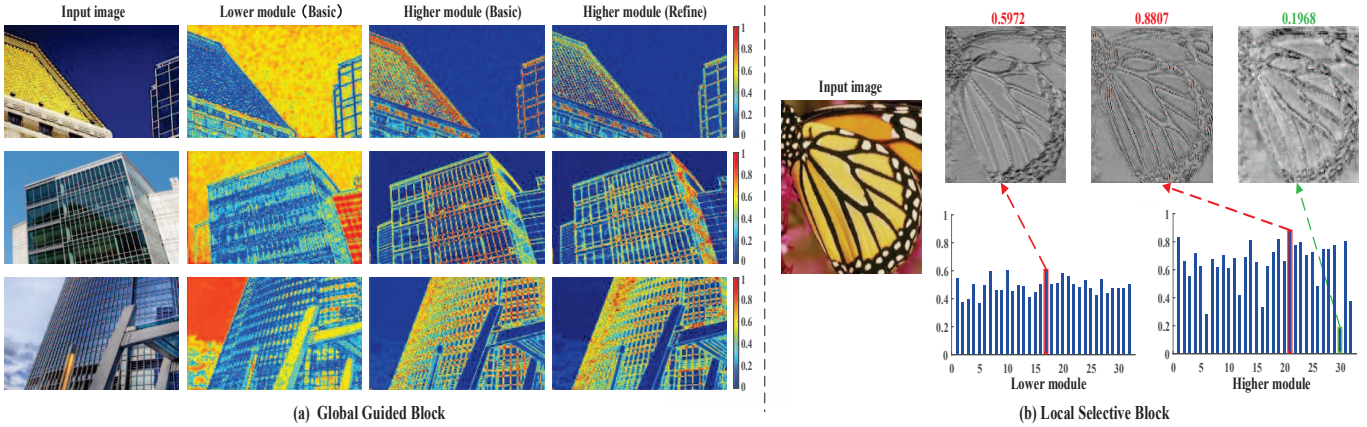


Fig. 6. Illustrations of global guided block (GGB) and local selective block (LSB). The heatmaps in (a) show global guided mask of GGB in different modules (warmer color indicates larger weight). The bar graphs in (b) show the weights of local selective unit in different LSB.

Here, n is to denote the input of the upsampling module and k is the kernel size of the deconvolution layer. For the regular upsampling module, the dimension of output is equal to the input. For the proposed SPU, the dimension is reduced to n/d with separable deconvolution, and the final output is further reduced to $n/2d$ with spatial pyramid operation. Therefore, the change of parameters from regular upsampling module to SPU can be calculated by the following equation:

$$n^2k^2 \rightarrow \frac{n^2k^2}{d\rho} + \frac{n^2(2\cdot 3^2+1^2)}{2d^2} = \frac{n^2}{d}(\frac{k^2}{\rho}+\frac{19}{2d}). \quad (10)$$

For example, when the scale factor is 4, we set $\rho = 4$, $d = 4$, and $k = 8$ in this work. Therefore, the reduction ratio of SPU is 7.2%, which is much less than the regular module. More analysis of SPU is laid in section IV-D4.

IV. Experiments

A. Datasets

In the training stage, we use DIV2K dataset [49] which is widely used in [12], [16] for SR. Our models are trained with 800 training images from the dataset, and the images are also augmented by randomly flipping horizontally and rotating with 90°, 180°, 270°. In testing stage, four benchmark datasets are used: Set5 [50], Set14 [51], BSD100 [52], Urban100 [53]. Set5, Set14, and BSD100 have many natural images while Urban100 contains many detail-rich urban images. We use PSNR and SSIM [54] on Y-channel of YCbCr color space, to evaluate the quality of the SR results.

B. Implementation details

In our GLADSR, the structure of nested dense groups in both basic and refine SR nets is the same. Each NDG contains G GLAMs and B LSBs. The growth rate in GLAM is set to $K = K_0 = 32$, the expansion factor τ in WAU equals to 3. The bottleneck layer in NDG and the Conv layer in FENet has 64 filters. The sampling rates of SPU are 2 and 4. Furthermore, the gating layers in reconstruction net have 3 filters, since our model uses color images as its input and output. LR images of different scale factors are generated by downsampling the training images with the bicubic kernel. The LR images are then cropped into a sub-image set of 48×48. The batch size is set to 16. We use Adam [55] to optimize models with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The learning rate is initialized to 10^{-4} and then decreases to half at every 2×10^5 iterations.

C. Difference with Prior Works

In this subsection, we show more details about the differences between the proposed GLADSR and several related representative works.

Difference to DenseNet. First, DenseNet [33] is often used to treat high-level computer vision tasks (e.g. image classification), while GLADSR is specially presented for image SR without utilizing some operations in DenseNet, such as max pooling and batch normalization. Second, DenseNet introduces the densely connected structure inside the local dense block. While GLADSR further adopts the nested dense connections (NDC) in a global way, and it can make better use of the hierarchical features through the network. Third, we propose the stepwise refinement structure to obtain better performance, rather than the direct way in DenseNet.

Difference to RCAN. First, RCAN [12] is a very large model (over 400 layers) for image SR, while our GLADSR aims to build a lightweight model with much fewer parameters and operations. Second, RCAN only considers the channel-wise attention in the network, and neglects the spatial attention, which is also important for image SR. While our GLADSR learns both spatial and channel-wise attention in the network, and the spatial attention is very helpful to distinguish various areas (e.g., smooth regions, and complex textures). Third, GLADSR adopts Leaky ReLU [46] for spatial and channel-wise attention, instead of the ReLU operation in RCAN. It can effectively avoid the "dead features" in the attention branch when the network goes deeper, and make the training of the network more stable.

Difference to RNAN. Although RNAN [43] incorporates the attention mechanisms in the network for image restoration, there are still distinct differences between RNAN and GLADSR. First, RNAN adopts the attention mechanisms in a local way (i.e., only for the features with short-term memory), while GLADSR adopts the attention mechanisms on the global

and local patterns. GLADSR considers the attention for both long/short-term memories to achieve more representational ability. Second, there are no dense connections in RNAN, and most local residual blocks have no direct access to their previous attention blocks. While GLADSR adopts the nested dense structure across the adjusted modules, which can make better use of the hierarchical features. Third, GLADSR introduces a novel module, named separable pyramid upsampling (SPU), in the upscaling step for lightweight super-resolution. It can bring better performance than the upsampling operation (sub-pixel convolution) used in RNAN.

D. Network Investigation

1) Analysis of Global-Local Adjusting Structure: In this section, we analyze why the proposed global-local adjusting structure in GLADSR can benefit the image super-resolution, and illustrate the effectiveness of the network as well. In general, it is harder to recover high-frequency information than low-frequency one, thus the SR results will be better if the models can optimize the focuses on various features. However, previous methods often equally treat the extracted features and neglect to consider the distribution of network resources to various information, hence limiting the representation ability. By contrast, the present work introduces the spatial guided mask in global guided block (GGB), and the channel-wise selective weight in local selected block (LSB), both of which can reweight different features with the proper distribution.

For the GGB, in the low-level module, the spatial mask mainly focuses on low-frequency regions, and it is thus helpful to recover the outlines at the beginning of the network (as Fig. 6(a) shows). As the network goes deeper, it is more reasonable to allocate more computational resources on high-frequency details, which are harder to recover. Therefore, in the high-level module, the focus areas of GGB switch to image details, such as edges and textures. The heatmaps between the lower and higher modules show that GGB can adaptively optimize the weights of the interest regions from low-level to high-level. Comparing the heatmaps between the basic and refine net in the high-level module, it can be observed that such a tendency is still preserved. Moreover, the heatmaps of the spatial mask in the refine net are finer, and pay more attention to the textual areas that may be neglected in the basic net. GGB guides the network to modify the spatial attention in a global way as well as effectively reallocate network resources.

For the LSB, it tends to rescale channel-wise features in a local way. As shown in Fig. 6(b), in the lower module, LSB selects and distributes larger weights to the features with clearer outlines with the guidance of GGB. While, in the higher module, the features with sharper edges and more detailed textures are selected and allocated with larger weights. In contrast, blurry or noisy features are relatively suppressed. Moreover, we can observe that the distribution of channel-wise weights in the higher module has larger variance than that in the lower module, since the extracted features can show a greater level of specificity when the network goes deeper [41]. LSB can well discriminate the characteristic of features at different levels, and selects more useful features for reconstruction.

Table I
ABLATION STUDY ON LOCAL SELECTIVE BLOCK (LSB), GLOBAL GUIDED BLOCK (GGB) AND NESTED DENSE CONNECTIONS (NDC). MODELS ARE ALL EVALUATED ON SET5/URBAN100 IN 3×10^5 ITERATIONS WITH SCALE FACTOR $\times 2$. RED DENOTES THE BEST PSNR VALUE.

Ablation Settings			PSNR (dB)
LSB	GGB	NDC	
×	×	×	37.45/30.79
×	✓	×	37.75/31.52
✓	×	×	37.76/31.55
✓	✓	×	37.82/31.72
×	×	✓	37.56/31.11
×	✓	✓	37.81/31.71
✓	×	✓	37.84/31.75
✓	✓	✓	37.89/31.80

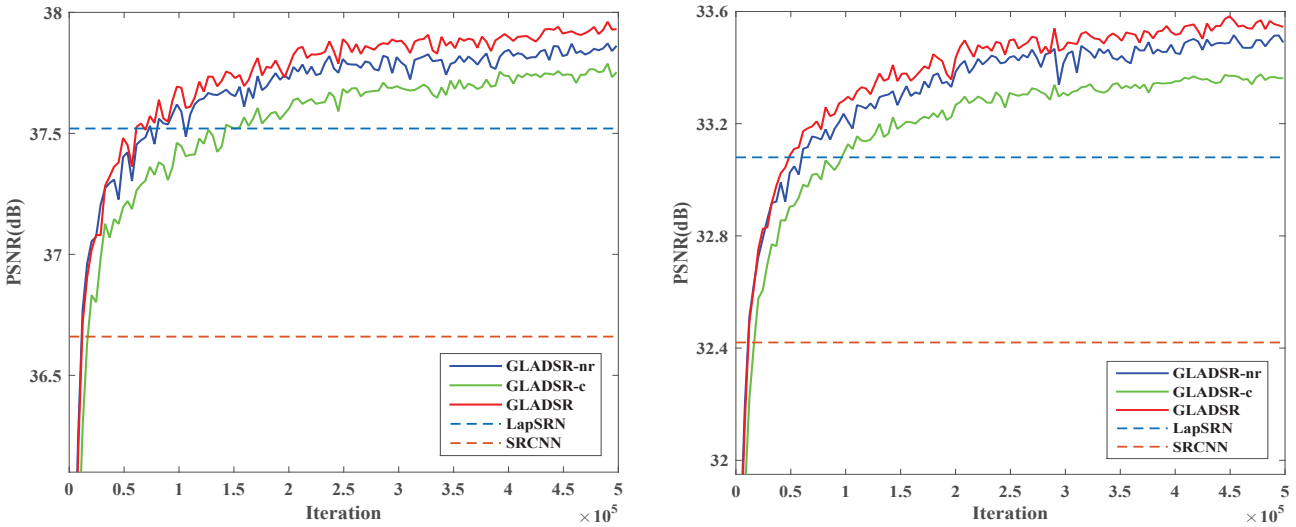
In summary, GGB adjusts the focus regions of each module in a global way, while LSB modifies the channel-wise distribution of corresponding blocks in a local way. The global-local adjusting module (GLAM) can effectively extract more powerful information by using both of them. The results obtained also lead us to suggest that the principle of adjusting the distribution of network resources on global and local mode is also applicable to a broader class of image processing tasks.

2) Ablation study: Table I shows the ablation study on the effects of local selective block (LSB), global guided block (GGB) and nested dense connections (NDC). The eight models have the same G and B ($G = 4, B = 4$). To demonstrate the effect of global and local blocks, we carry out experiments by removing the channel selective unit or/and spatial guided unit in each GLAM. It is observed that both LSB and GGB can improve network performance independently. Specifically, the PSNR on Set5 ($\times 2$) increases from 37.45 dB to 37.76 dB with LSB and 37.75 dB with GGB even without NDC. Therefore, it can be concluded that using both of them can achieve better performance, no matter what NDC is used or not. It can be attributed that the combination of local and global information with their spatial and channel-wise inter-dependencies can adaptively modify multi-level features, and optimize the network performance. The above experimental results further demonstrate that the proposed LSB and GGB structure are effective, and it is helpful to integrate both of them to adjust the network allocation for SR.

To investigate the effect of nested dense connections (NDC), we carry a control experiment without using NDC. Comparing the results of the first 4 rows and last 4 rows in Table I, it can be found that the networks with NDC perform better than those without NDC. Specially, even adopting both LSB and GGB, NDC can still improve the performance from 37.73dB to 37.80dB on Urban100 dataset. These results prove that the proposed NDC can further reuse the dense features extracted from each GLAM, and further improve the network representation ability. They also indicate the effectiveness of the nested dense structure.

3) Benefit of refinement structure: We introduce a refinement strategy to reconstruct the HR images in GLADSR. We set $G = 4$ and $B = 4$ for GLADSR, and denote GLARSR-c as the intermediate coarse output of GLADSR, which is obtained from the features of basic SR Net. As the control

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3005025, IEEE Transactions on Multimedia



(a) Set5 (b) Set14

Fig. 7. Convergence analysis on the refinement structure. The curves are evaluated by using PSNR on Set5 and Set14 ($\times 2$).

Fig. 8. Effect of the refinement structure. Compared to GLADSR-nr, our GLADSR correctly reconstruct the lines by refining the intermediate basic output (GLADSR-c).

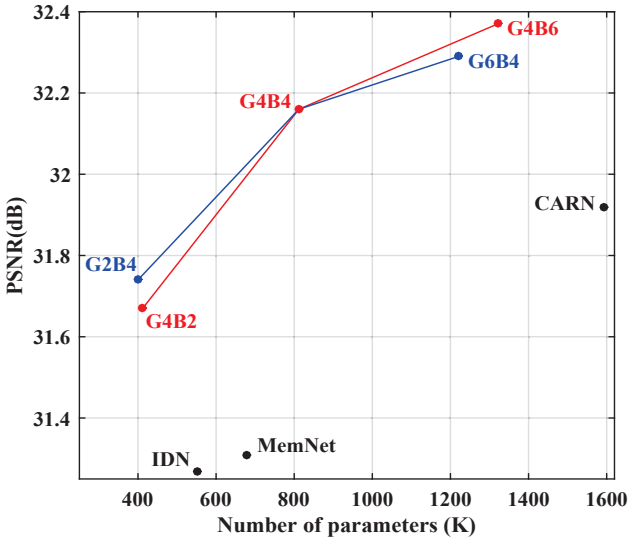


Fig. 9. Trade-off between parameters and PSNR with scale factor $\times 2$. The models are evaluated on Urban100.

group, GLADSR-nr is designed to remove the refinement reconstruction net in GLADSR, and the final result is directly generated through the upscaled LR image and the output of refine SR Net with residual strategy. Specially, the parameters of GLADSR and GLADSR-nr are almost the same, since the SPU in both basic and refine SR net is shared. As shown in Fig. 7, the convergence curves of GLADSR and GLADSR-nr on Set5 and Set14 datasets are evaluated. The converge speed of GLADSR is faster than GLADSR-nr does, and it also achieves better performance on both datasets. In addition, for Set5 dataset (see Fig. 7(a)), it can be found that GLADSR (the red curve) performs better than SRCNN [7] within 1.5×10^4 iterations and LapSRN [15] within 8×10^4 iterations.

In addition, we also show the visual comparison in Fig. 8. It can be observed that GLADSR-nr generates artifacts without using the refinement structure, and similar distortions also occur in GLADSR-c. However, in contrast, GLADSR compensates artifacts in the coarse output and obtains more accurate results. Due to adopting the stepwise refinement strategy, GLADSR can effectively optimize the basic coarse result with the refining step. Therefore, to sum up, the proposed refinement structure is beneficial to the SR task.

4) Study on the separable pyramid upsampling: We adopt the separable perception upsampling (SPU) module to improve the efficiency of the upsampling step in GLADSR. Differing from the depthwise separable convolution [48], we map the input features into ρ groups and apply a single deconvolution filter to each separate group with dimension reduced ration d. When the numbers of ρ and d are fixed, the reduction in parameters mainly determined by the kernel size of deconvolution filters. We set $\rho = 4$, $d = 4$ as the default settings. Therefore, the reduction ratios are 9.9%, 7.5%, 7.2% for different scale factors $\times 2$, $\times 3$, $\times 4$ with corresponding kernel sizes (4, 7, 8).

As shown in Table IV, on scale factors $\times 2$, compared to the regular upsampling (deconvolution), the proposed separable upsampling structure significantly reduces the number of parameters, and achieves better performance with much less computational cost. SPU can further decrease the parameters

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3005025, IEEE Transactions on Multimedia

Table II
QUANTITATIVE RESULTS OF STATE-OF-THE-ART METHODS. AVERAGE PSNR/SSIMS FOR SCALE FACTOR $\times 2$, $\times 3$ AND $\times 4$ ON DATASETS SET5, SET14, BSD100 AND URBAN100. PARAMETER IN RED DENOTES THE BEST PERFORMANCE AND THE BLUE ONE DENOTES THE SECOND BEST PERFORMANCE.

Method	Scale	Params.	Mult-Adds	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM
SRCNN [7]	×2	57K	52.7G	36.66/0.9542	32.42/0.9063	31.36/0.8879	29.50/0.8946
FSRCNN [27]	×2	12K	2.9G	37.00/0.9558	32.63/0.9088	31.53/0.8920	29.88/0.9020
VDSR [17]	×2	665K	612.6G	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140
DRCN [35]	×2	1,774K	17,974.3G	37.63/0.9588	33.04/0.9118	31.85/0.8942	30.75/0.9133
LapSRN [15]	×2	813K	29.9G	37.52/0.9590	33.08/0.9130	31.80/0.8950	30.41/0.9100
DRRN [13]	×2	297K	6,796.9G	37.74/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188
MemNet [36]	×2	677K	2,662.4G	37.78/0.9597	33.28/0.9142	32.08/0.8979	31.31/0.9195
IDN [38]	×2	553K	127.4G	37.83/0.9600	33.30/0.9148	32.08/0.8985	31.27/0.9196
GLADSR-L(ours)	×2	400K	94.6G	37.84/0.9600	33.43/0.9162	32.04/0.8981	31.74/0.9243
CARN [16]	×2	1,592K	222.8G	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256
GLADSR(ours)	×2	812K	187.2G	37.99/0.9608	33.63/0.9179	32.16/0.8996	32.16/0.9283
SRCNN [7]	×3	57K	52.7G	32.75/0.9090	29.28/0.8209	28.41/0.7863	26.24/0.7989
FSRCNN [27]	×3	12K	1.3G	33.16/0.9140	29.43/0.8242	28.53/0.7910	26.43/0.8080
VDSR [17]	×3	665K	612.6G	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279
DRCN [35]	×3	1,774K	17,974.3G	33.82/0.9226	29.76/0.8311	28.80/0.7963	27.15/0.8276
DRRN [13]	×3	297K	6,796.9G	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378
MemNet [36]	×3	677K	2,662.4G	34.09/0.9248	30.00/0.8350	28.96/0.8001	27.56/0.8376
IDN [38]	×3	553K	56.6G	34.11/0.9253	29.99/0.8354	28.95/0.8013	27.42/0.8359
GLADSR-L(ours)	×3	409K	47.0G	34.18/0.9253	30.22/0.8393	28.99/0.8026	27.93/0.8470
CARN [16]	×3	1,592K	118.8G	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493
GLADSR(ours)	×3	821K	88.2G	34.41/0.9272	30.37/0.8418	29.08/0.8050	28.24/0.8537
SRCNN [7]	×4	57K	52.7G	30.48/0.8628	27.49/0.7503	26.90/0.7101	24.52/0.7221
FSRCNN [27]	×4	12K	0.7G	30.71/0.8657	27.59/0.7535	26.98/0.7150	24.62/0.7280
VDSR [17]	×4	665K	612.6G	31.35/0.8838	28.01/0.7674	27.23/0.7229	25.26/0.7547
FEQE [56]	×4	96K	5.5G	31.32/0.8754	28.09/0.7660	27.29/0.7251	25.18/0.7524
DRCN [35]	×4	1,774K	17,974.3G	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510
LapSRN [15]	×4	813K	149.4G	31.54/0.8850	28.19/0.7720	27.32/0.7280	25.21/0.7560
DRRN [13]	×4	297K	6,796.9G	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638
MemNet [36]	×4	677K	2,662.4G	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630
IDN [38]	×4	553K	33.6G	31.82/0.8903	28.25/0.7730	27.41/0.7297	25.41/0.7632
GLADSR-L(ours)	×4	413K	29.4G	31.94/0.8909	28.51/0.7782	27.49/0.7327	25.88/0.7776
CARN [16]	×4	1,592K	90.9G	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837
GLADSR(ours)	×4	826K	52.6G	32.14/0.8940	28.62/0.7813	27.59/0.7361	26.12/0.7851

of the upsampling step than the deconvolution operation in about 10 times while keeping comparable performance. In addition, we also compared the proposed SPU with the sub-pixel convolution, which is widely used in recent work as well [11], [19]. The sub-pixel convolution operation is set with the same input and output dimension as SPU for a fair comparison. SPU can achieve superior performance with much fewer parameters and less computational cost, which can also demonstrate the powerful ability of SPU.

5) Investigation of parameters settings: To further investigate the scalability of the GLADSR, we train more networks with different parameter settings: the GLAM number G in NDG and the LSB number B in each GLAM. A baseline network is constructed with $G = 4, B = 4$ (present as G4B4 for short). As Figure 9 shows, we carry out experiments by independently changing the number of G (the red line) and the number of B (the blue line). Increasing either G or B can improve the performance of the network, which again validate the well-known fact that deeper network has better performance. Although the model with a larger G or B requires more parameters to achieve better performance, our G6B4 and G4B6 still outperform CARN [16] under fewer parameters. On the other hand, our model with smaller G and B can dramatically reduce the parameters with acceptable performance loss. In addition, the more lightweight networks (G2B4, G4B2) even generate better results with fewer parameters than IDN [38] and MemNet [36]. The results thus show that GLADSR is much more effective and efficient than those state-of-the-art methods. Furthermore, comparing the red and blue lines in Fig. 9, it is also shown that the increasing number of B is more helpful for improving network performance.

6) Running time Analyses: We compare running time and performance with different methods on Set5($\times 4$) in Table V. Compared to CARN [16], our GLADSR only uses about half amount of parameters as well as achieves faster speed and better results. Moreover, the more lightweight version GLADSR-L(G2B4) is about $200\times$ faster than MemNet [13] and $4\times$ faster than VDSR [17] with fewer parameter. We can thus conclude that our model can achieve a good trade-off between the running time and performance.

E. Comparisons with State-of-the-Art Methods

In this section, GLADSR is evaluated with state-of-the-art methods. To balance the performance and efficiency, we select GLADSR (G4B4) as the basic network and GLADSR-L (G2B4) as a more lightweight version. In addition, the computational cost of each method is measured by the numbers of Multi-Adds operations. The Multi-Adds are all calculated with 960×960 output.

Table II lists the quantitative comparison results with the lightweight methods in comparable FLOPs. It can be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3005025, IEEE Transactions on Multimedia

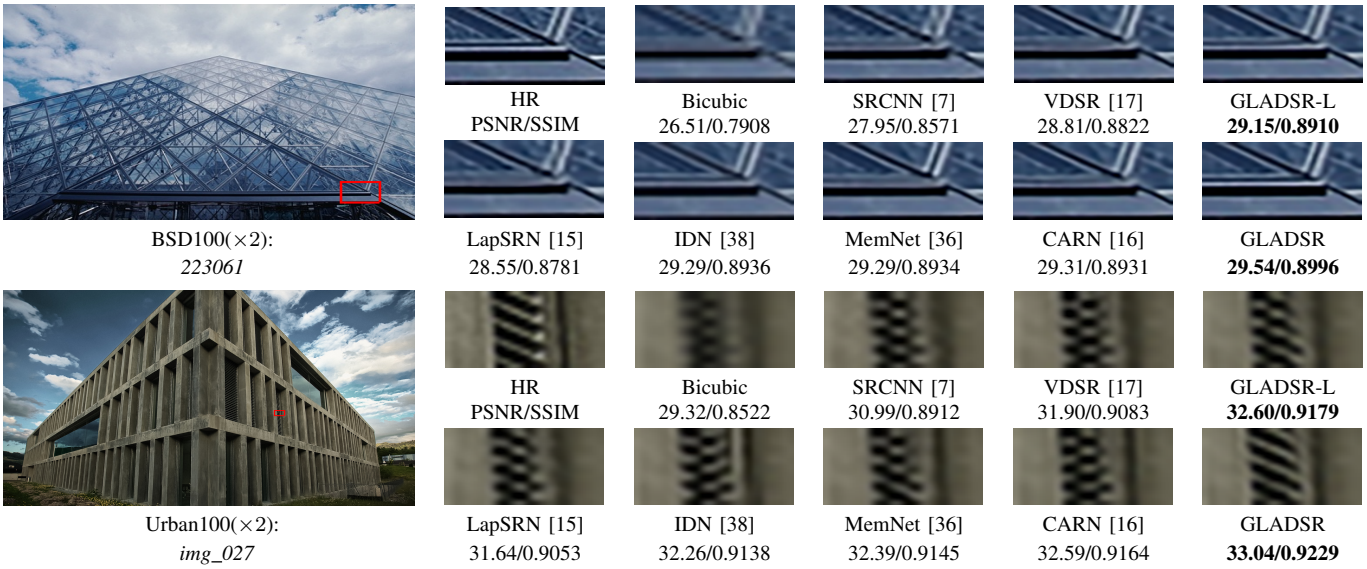


Fig. 10. Qualitative comparison on BSD100 and Urban100 dataset with scale factor ×2.

Fig. 11. Qualitative comparison on Set14 and Urban100 datasets with scale factor ×3.

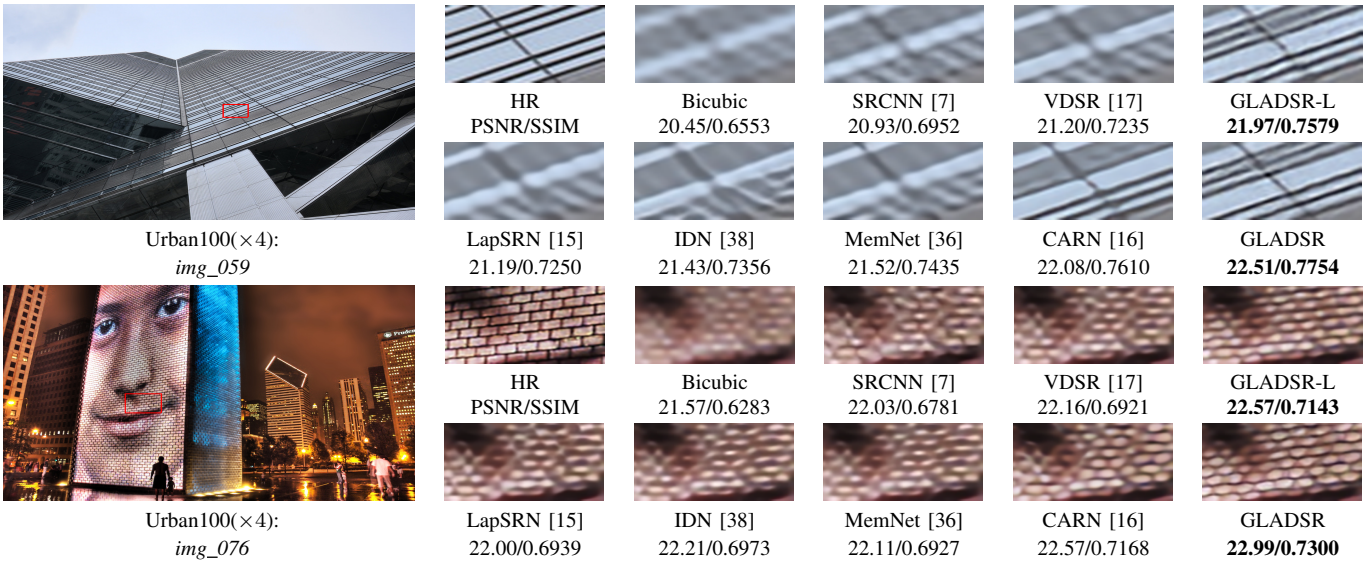


Fig. 12. Qualitative comparison on Urban100 datasets with scale factor ×4.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3005025, IEEE Transactions on Multimedia

Table III
QUANTITATIVE RESULTS OF LARGE SR MODELS WITH SCALE FACTOR $\times 2$ ON DATASETS SET5, SET14, BSD100 AND URBAN100. PARAMETER IN RED DENOTES THE BEST PERFORMANCE AND THE BLUE ONE DENOTES THE SECOND BEST PERFORMANCE..

Method	Scale	Params.	Set5 PSNR / SSIM	Set14 PSNR / SSIM	BSD100 PSNR / SSIM	Urban100 PSNR / SSIM
EDSR [11]	$\times 2$	43M	38.11 / 0.9602	33.92 / 0.9195	32.32 / 0.9013	32.93 / 0.9351
RDN [19]	$\times 2$	22M	38.24 / 0.9614	34.01 / 0.9212	32.34 / 0.9017	32.89 / 0.9353
RCAN [12]	$\times 2$	16M	38.27 / 0.9614	34.12 / 0.9216	32.41 / 0.9027	33.34 / 0.9384
D-DBPN [18]	$\times 2$	5.8M	38.09 / 0.9600	33.85 / 0.9190	32.27 / 0.9000	32.55 / 0.9324
RNAN [43]	$\times 2$	7.5M	38.17 / 0.9611	33.87 / 0.9207	32.32 / 0.9014	32.73 / 0.9340
GLADSR (large)	$\times 2$	**5.4M**	38.19 / 0.9612	33.89 / 0.9202	32.33 / 0.9016	32.80 / 0.9345
GLADSR+ (large)	$\times 2$	**5.4M**	38.25 / 0.9614	34.02 / 0.9211	32.38 / 0.9021	33.01 / 0.9360

Table IV
STUDY ON THE EFFECT OF THE PROPOSED SEPARABLE PYRAMID UPSAMPLING (SPU). MODELS ($G = 4, B = 4$) ARE ALL EVALUATED ON SET5/URBAN100 IN 3×10^5 ITERATIONS WITH SCALE FACTOR $\times 2$.

Module Settings	Params. (ratio)	Mult-Adds	PSNR(dB)
Deconvolution	65.5K (100%)	15.1G	37.81/31.70
Sub-pixel convolution	55.3K (84.4%)	12.7G	37.85/31.76
Separable upsampling	20.5K (31.3%)	3.8G	37.87/31.79
SPU	**6.5K (9.9%)**	**3.2G**	**37.89/31.80**

Table V
RUNNING TIME, PSNR AND PARAMETER NUMBERS COMPARISONS. WE REPORT RUNNING TIME FOR REFERENCE, BECAUSE THE TIME IS RELATED TO IMPLEMENTATION PLATFORM AND CODE. THE PSNR VALUES ARE TESTED ON SET5 WITH SCALE FACTOR $\times 4$.

Methods	VDSR [17]	MemNet [36]	CARN [16]	GLADSR-L (Ours)	GLADSR (Ours)
Params.	665K	677K	1592K	413K	826K
PSNR (dB)	31.35	31.74	32.13	31.94	**32.14**
Time (s)	0.15	7.93	0.10	**0.04**	0.07

found that our GLADSR outperforms the state-of-the-arts on all datasets at all scale factors. Compared to CARN [16], GLADSR achieves superior performance with fewer parameters and less computational cost. Specifically, with scale factor $\times 3$, GLADSR presents the best results with 821K parameters and 88.2G Mult-Adds (while CARN [16] uses 1592K and 118.8G). Compared to the recursive networks (e.g., DRRN [13], DRCN [35], and MemNet [36], GLADSR consumes much less computational cost and gets better results. Furthermore, GLADSR-L, adopting fewer parameters, also achieves much better results against the popular lightweight methods, such as LapSRN [15] and IDN [38]. These comparisons highlight the efficiency of GLADSR again.

Fig. 10, 11 and 12 illustrate visual results of our model with scale factors $\times 2, \times 3, \times 4$ compared to the state-of-the-art methods. It can be found that both of our models (GLADSR and GLADSR-L) perform better than the corresponding methods. More importantly, we observe that most of the compared methods cannot accurately recover the details of LR images, and sometimes generate the artifacts (e.g., "*img_027*" in Fig. 10 and "*img_076*" in Fig. 12). In contrast, our GLADSR can alleviate the artifacts and recover correct textures with the use of refinement structure. Furthermore, as shown in "*223061*"(Fig. 10) and "*ppt3*"(Fig. 11), both of GLADSR and GLADSR-L can recover much sharper and clearer edges, which are more close to the ground truth. For the images such as "*059*"(Fig. 12), even human beings cannot easily distinguish the textures well in the LR images. Most of the methods cannot recover them either. However, our models can reconstruct better visual results with the benefit of our network designs.

To further demonstrate the scalability of GLADSR, especially the capability compared to the models with more parameters, we also train a large GLADSR, named GLADSR (large). We set GLADSR (large) with $G = 8$, $B = 10$. Similar to [11], [43], we adopt the self-ensemble strategy to further improve the reconstruction performance, and denote the model as GLADSR+. As Table III shows, GLADSR+ can achieve the second-best performance on four benchmarks with fewer parameters. Even without the self-ensemble strategy, GLADSR (large) can also obtain better results than D-DBPN [18] and RNAN [43] do. Moreover, GLADSR (large) performs better than EDSR [11], and only uses 12.5% parameters as EDSR does. Compared to RCAN [12], the performance of our model is remarkable as well, because the parameter number of GLADSR (large) is 5.4M, which is much smaller than that of RCAN (16M). Comparisons on large models indicate that GLADSR can make better use of the network resources, and thus become more effective.

F. Discussions

1) Training with Small Dataset: The experimental results shown in Table II and III are based on DIV2K training set. To test the performance of GLADSR on a small dataset, a model named GLADSR (SR291) is retrained. We use the widely-used SR291 dataset as the training set, which is only comprised of 291 images, respectively from the previous work [4], [52]. We compare GLADSR (SR291) with state-of-the-art methods, such as VDSR [17] and MemNet [36], which also trained with the same small dataset. As Table VI shows, even with the small training set, GLADSR outperforms the other methods on four benchmarks with scale factor $\times 2$, $\times 3$, and $\times 4$. These experiments demonstrate the effectiveness of GLADSR as well. Moreover, compared to the results of GLADSR in Table II, it can be concluded that our SR model can get better performance when training with a larger number of high-resolution images.

2) Super-resolution on Real-world Images: To further evaluate the ability of GLADSR on real-world images, we super-

Table VI
QUANTITATIVE RESULTS OF STATE-OF-THE-ART METHODS, WHICH IS TRAINED ON SMALL DATASET WITH SCALE FACTOR $\times 2$, $\times 3$, AND $\times 4$. PARAMETER IN RED DENOTES THE BEST PERFORMANCE AND THE BLUE ONE DENOTES THE SECOND BEST PERFORMANCE.

Method	Set5			Set14			BSD100			Urban100		
	$\times 2$	$\times 3$	$\times 4$	$\times 2$	$\times 3$	$\times 4$	$\times 2$	$\times 3$	$\times 4$	$\times 2$	$\times 3$	$\times 4$
SRCNN [7]	36.66	32.75	30.48	32.42	29.28	27.49	31.36	28.41	26.90	29.50	26.24	24.52
FSRCNN [27]	37.00	33.16	30.71	32.63	29.43	27.59	31.53	28.53	26.98	29.88	26.43	24.62
VDSR [17]	37.53	33.66	31.35	33.03	29.77	28.01	31.90	28.82	27.23	30.76	27.14	25.26
DRRN [13]	37.74	34.03	31.68	33.23	29.96	28.21	32.05	28.95	27.38	31.23	27.53	25.44
MemNet [36]	37.78	34.09	31.74	33.28	30.00	28.26	32.08	28.96	27.40	31.31	27.56	25.50
GLADSR (SR291)	37.87	34.24	31.86	33.36	30.12	28.35	32.15	29.03	27.47	31.36	27.59	25.54

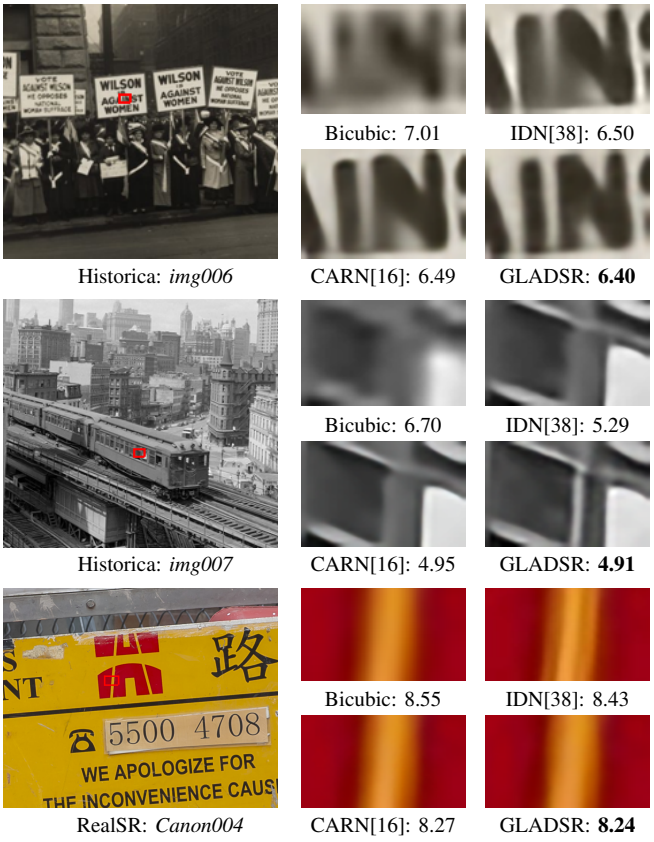


Fig. 13. Visual results on real-world images on scale factor $\times 4$. The values of each sub-images are NIQE scores.

resolve images of real-world scenes from historical photographs [57] and RealSR dataset [26] by scale factor $\times 4$. The historical images are with JPEG compression artifacts, and the RealSR dataset is captured by digital cameras (e.g., Canon 5D3). We compare GLADSR with two state-of-the-art methods (IDN [38] and CARN [16]), and evaluate the SR results with the no-reference image quality score NIQE [58]. The smaller score of NIQE indicates the better perceptual quality. As shown in Figure 13, GLADSR outputs the images with better perceptual quality than the other methods. For the image "*Cannon004*", we do not retrain the models with the training set of RealSR, while our GLADSR can still effectively avoid the distortions generated in IDN. Moreover, for images "*img006*" and "*img007*", both of the IDN and CARN suffer from blurring artifacts. In contrast, GLADSR reconstructs much sharper and more accurate results. These

Table VII
BENEFITS OF GLADSR. THE BASE_NETWORK ($G = 4, B = 4$) IS DESIGNED WITH THE REFINEMENT STRUCTURE. WE INCREASE THE DEPTH OF DEEPER_BASE_NETWORK WITH $B = 5$. UTILIZING THE SCHEME OF GLAM AND SPU, THE PROPOSED MODEL PERFORMS MUCH BETTER THAN THE DEEPER BASE NETWORK WITH LESS PARAMETERS. MODELS ARE ALL EVALUATED ON SET5/URBAN100 IN 3×10^5 ITERATIONS WITH SCALE FACTOR $\times 2$.

Module Settings	Params.	PSNR(dB)
base_network	726K	37.45/30.79
deeper_base_network	1028K	37.59/31.16
base_network + GLAM	871K	37.81/31.70
base_network + GLAM + SPU	**812K**	**37.89/31.80**

comparisons indicate the benefits of utilizing the global-local adjusting structure, making our GLADSR robustly for different degradation kernels.

3) Benefits of GLADSR: GLADSR is built on the Global-Local Adjusting Module (GLAM), where the features from the local and global components are well reweighted. With the design of the Separable Pyramid Upampling Module (SPU), the parameters on the upscaling operation are dramatically reduced. In addition, benefit from the refinement structure, GLADSR can alleviate the artifacts and reconstruct better results. As shown in Table VII, we denote the model only with refinement structure as base_network. As the other CNN-based methods did, we simply increase the depth of base_network to see how the performance changes (denoted as deeper_base_network). The deeper_base_network merely improves the performance even with 300K additional parameters. However, with both GLAM and SPU, the proposed GLADSR can achieve much better performance than base_network, but only consumes about 90K parameters. Besides, only with the scheme of GLAM, the performance can also increase 0.91dB on Urban100 dataset with fewer parameters. In conclusion, the scheme of GLAM helps the model to reduce the redundancy of extracted features and generate powerful features robustly. Furthermore, SPU uses the concept of group and perception to improve the efficiency of the upsampling step. With the help of the proposed structures, our method outperforms the previous state-of-the-arts on both aspects of efficiency and performance, and meanwhile consumes fewer resources.

V. CONCLUSION

The present work proposes a novel global-local adjusting dense super-resolution network (GLADSR) for single image super-resolution. The highlights of our scheme can be sum-

marized as follows: (1) GLADSR uses a stepwise strategy rather than a direct mapping to generate the final SR image, which can significantly alleviate the artifacts during SR processing. (2) GLADSR introduces a global-local adjusting module (GLAM). It can make full use of the global/local information, and distribute computational resources in a more reasonable way. (3) GLADSR can reconstruct SR images with much fewer parameters and operations, compared to other lightweight methods. While the performance can be further improved by the specific designed upsampling structure (SPU). Therefore, GLADSR is worthy of being applied in practice and extends the single image SR performance to a new state-of-the-art level. For future research, our GLADSR may further benefit from adversarial training, which may help to generate more photo-realistic images. We also plan to extend the proposed scheme to other ill-posed image processing tasks, such as denoising and deblurring.

REFERENCES

[1] N. Kumar and A. Sethi, "Fast learning-based single image super-resolution," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1504–1515, 2016.

[2] H. Chang, D. Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *CVPR*, vol. 1, 2004, pp. I–I.

[3] Z. Zhu, F. Guo, H. Yu, and C. Chen, "Fast single image super-resolution via self-example learning and sparse representation," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2178–2190, 2014.

[4] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[5] J. Jiang, X. Ma, C. Chen, T. Lu, Z. Wang, and J. Ma, "Single image super-resolution via locally regularized anchored neighborhood regression and nonlocal means," *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 15–26, 2016.

[6] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *ACCV*, 2014, pp. 111–126.

[7] C. Dong, C. L. Chen, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2014.

[8] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 114–125.

[9] Z. Shao, L. Wang, Z. Wang, and D. Juan, "Remote sensing image super-resolution using sparse representation and coupled sparse autoencoder," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. PP, pp. 1–12, 07 2019.

[10] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, "Deep learning for single image super-resolution: A brief review," *IEEE Transactions on Multimedia*, 2019.

[11] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1132–1140.

[12] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 286–301.

[13] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *CVPR*, 2017, pp. 2790–2798.

[14] X. Yang, H. Mei, J. Zhang, K. Xu, B. Yin, Q. Zhang, and X. Wei, "Drfn: Deep recurrent fusion network for single-image super-resolution with large factors," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 328–337, 2018.

[15] W. S. Lai, J. B. Huang, N. Ahuja, and M. H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *CVPR*, 2017.

[16] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 252–268.

[17] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *CVPR*, 2016, pp. 1646–1654.

[18] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1664–1673.

[19] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2472–2481.

[20] Z. Zhang, Z. Wang, Z. Lin, and H. Qi, "Image super-resolution by neural texture transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7982–7991.

[21] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, and J. Jiang, "Edge-enhanced gan for remote sensing image superresolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, pp. 1–14, 03 2019.

[22] L. Zhou, Z. Wang, Y. Luo, and Z. Xiong, "Separability and compactness network for image recognition and superresolution," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, pp. 1–12, 01 2019.

[23] K. Jiang, Z. Wang, P. Yi, G. Wang, K. Gu, and J. Jiang, "Atmfn: Adaptive-threshold-based multi-model fusion network for compressed face hallucination," *IEEE Transactions on Multimedia*, vol. PP, pp. 1–1, 12 2019.

[24] P. Yi, Z. Wang, K. Jiang, Z. Shao, and J. Ma, "Multi-temporal ultra dense memory network for video super-resolution," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, pp. 1–1, 07 2019.

[25] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," pp. 1874–1883, 2016.

[26] J. Cai, H. Zeng, H. Yong, Z. Cao, and L. Zhang, "Toward real-world single image super-resolution: A new benchmark and a new model," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3086–3095.

[27] C. Dong, C. L. Chen, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *European Conference on Computer Vision*, 2016, pp. 391–407.

[28] Y. Shi, K. Wang, C. Chen, L. Xu, and L. Lin, "Structure-preserving image super-resolution via contextualized multitask learning," *IEEE transactions on multimedia*, vol. 19, no. 12, pp. 2804–2815, 2017.

[29] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun, "Meta-sr: a magnification-arbitrary network for super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1575–1584.

[30] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[32] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4799–4807.

[33] G. Huang, Z. Liu, L. v. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 00, July 2017, pp. 2261–2269. [Online]. Available: doi.ieeecomputersociety.org/10.1109/CVPR.2017.243

[34] Z. Wang, P. Yi, K. Jiang, J. Jiang, Z. Han, T. Lu, and J. Ma, "Multi-memory convolutional neural network for video super-resolution," *IEEE Transactions on Image Processing*, vol. PP, pp. 1–1, 12 2018.

[35] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *CVPR*, 2016, pp. 1637–1645.

[36] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *ICCV*, 2017, pp. 4549–4557.

[37] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, "Feedback network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3867–3876.

[38] Z. Hui, X. Wang, and X. Gao, "Fast and accurate single image super-resolution via information distillation network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 723–731.

[39] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T. S. Chua, "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning," in *CVPR*, 2017, pp. 6298–6306.

[40] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *ECCV*, 2016, pp. 451–466.

[41] H. Jie, S. Li, S. Gang, and S. Albanie, "Squeeze-and-excitation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, 2017.

[42] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[43] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," in *International Conference on Learning Representations*, 2019.

[44] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 065–11 074.

[45] J. Yu, Y. Fan, J. Yang, N. Xu, Z. Wang, X. Wang, and T. Huang, "Wide activation for efficient and accurate image super-resolution," *arXiv:1808.08718*, 2018. [Online]. Available: http://arxiv.org/abs/1808.08718

[46] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.

[47] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, 2014.

[48] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv:1704.04861*, 2017.

[49] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Los Alamitos, CA, USA: IEEE Computer Society, jul 2017, pp. 1122–1131. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPRW.2017.150

[50] M. Bevilacqua, A. Roumy, C. Guillemot, and A. Morel, "Low-complexity single image super-resolution based on nonnegative neighbor embedding," *BMVC*, 2013.

[51] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International Conference on Curves and Surfaces*, 2012, pp. 711–730.

[52] 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 *ICCV*, 2002, pp. 416–423 vol.2.

[53] J. B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *CVPR*, 2015, pp. 5197–5206.

[54] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[55] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980*, 2014.

[56] T. Vu, C. Van Nguyen, T. X. Pham, T. M. Luu, and C. D. Yoo, "Fast and efficient image quality enhancement via desubpixel convolutional neural networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[57] L. Wei-Sheng, H. Jia-Bin, A. Narendra, and Y. Ming-Hsuan, "Fast and accurate image super-resolution with deep laplacian pyramid networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1.

[58] A. Mittal, Fellow, IEEE, R. Soundararajan, and A. C. Bovik, "Making a 'completely blind' image quality analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2013.