Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8116846, 12 pages
https://doi.org/10.1155/2022/8116846

Research Article

DFAN: Dual Feature Aggregation Network for Lightweight Image Super-Resolution

Shang Li,[1,2] **Guixuan Zhang**,[2] **Zhengxiong Luo**,[1,2] **and Jie Liu**[2]

[1]*School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), China*
[2]*Institute of Automation, Chinese Academy of Sciences (CASIA), China*

Correspondence should be addressed to Jie Liu; jie.liu@ia.ac.cn

Received 12 October 2021; Revised 12 November 2021; Accepted 6 December 2021; Published 24 January 2022

Academic Editor: Ming Yan

With the power of deep learning, super-resolution (SR) methods enjoy a dramatic boost in performance. However, they usually have a large model size and high computational complexity, which hinders the application in devices with limited memory and computing power. Some lightweight SR methods solve this issue by directly designing shallower architectures, but it will adversely affect the representation capability of convolutional neural networks. To address this issue, we propose the dual feature aggregation strategy for image SR. It enhances feature utilization via feature reuse, which largely improves the representation ability while only introducing marginal computational cost. Thus, a smaller model could achieve better cost-effectiveness with the dual feature aggregation strategy. Specifically, it consists of Local Aggregation Module (LAM) and Global Aggregation Module (GAM). LAM and GAM work together to further fuse hierarchical features adaptively along the channel and spatial dimensions. In addition, we propose a compact basic building block to compress the model size and extract hierarchical features in a more efficient way. Extensive experiments suggest that the proposed network performs favorably against state-of-the-art SR methods in terms of visual quality, memory footprint, and computational complexity.

1. Introduction

Single image super-resolution (SISR) aims to reconstruct a visually natural high-resolution (HR) image from its low-resolution (LR) counterpart, which is an inherently ill-posed inverse problem. Due to the essential role in video processing [1], surveillance system [2], and object restoration [3], super-resolution (SR) is still an active research area.

Recently, deep learning-based image super-resolution methods [4–7] have shown prominent performance over conventional methods such as Bicubic interpolation and Lanczos resampling. After the proposal of residual learning [8], which simplifies the optimization of deep convolutional neural networks (CNNs), SR networks tend to become even deeper and larger. However, it is impractical to simply pursue performance gains without considering the model size and computational complexity. For devices with limited memory and battery capacity, cost-effective methods are preferred, which encourages the design of lightweight SR models. To reduce the number of parameters, some approaches adopt a recursive manner or parameter sharing scheme [9, 10]. However, to compensate for the performance drop, these methods have to increase the network width or depth, thus, resulting in high computational complexity as shown in Figure 1. Some other methods directly design shallower network architectures, which reduce parameters and calculations simultaneously. For example, [11, 12] are such compact models with fewer than 40 layers. However, their representation ability is restricted by the shallow architecture.

Towards these drawbacks, we propose Dual Feature Aggregation Network (DFAN) that can strike a better trade-off between SR performance and computational cost as illustrated in Figure 1. The key component of DFAN is the dual feature aggregation strategy. It aggregates local features and global features in a coarse-to-fine manner and

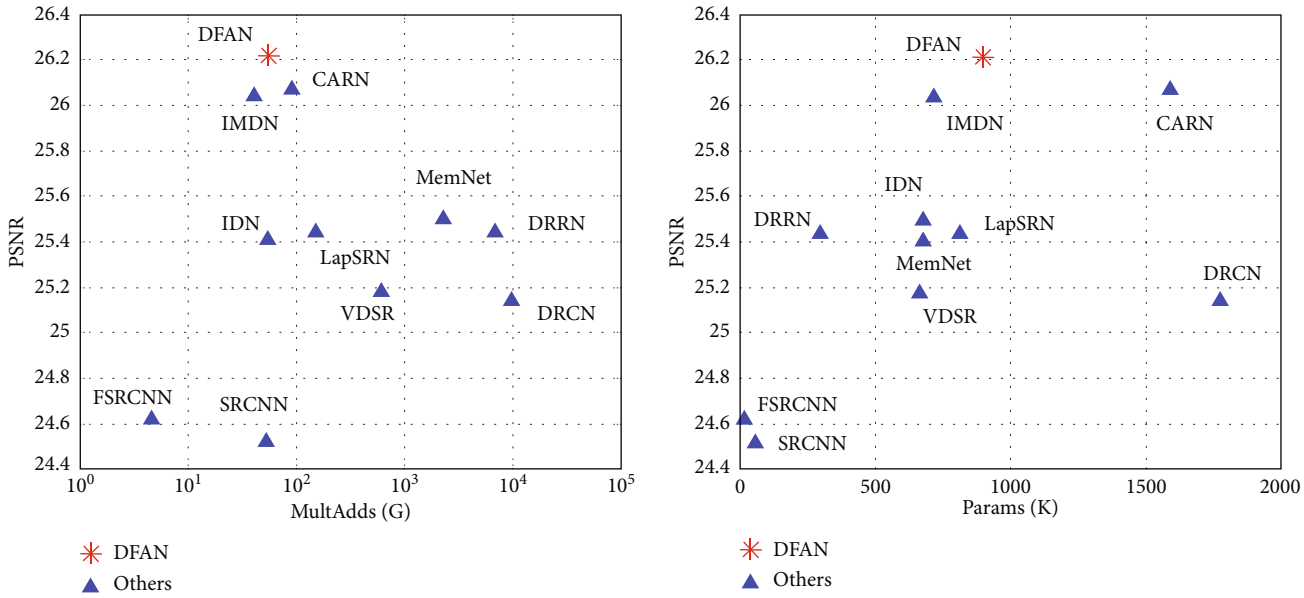


Figure 1: Average PSNR on Urban100 for 4× SR with respect to the number of multiply-adds (MultAdds) and parameters (Params). DFAN achieves the best performance with relatively fewer parameters and lower computational complexity.

could largely improve feature utilization via feature reuse. Specifically, the dual feature aggregation strategy consists of two modules: Local Aggregation Module (LAM) and Global Aggregation Module (GAM). LAM uses an efficient connection method and one convolutional layer to adaptively fuse hierarchical features along the channel dimension. Then, GAM further fuses the local aggregated features along the spatial dimension in an iterative manner. This progressive aggregation strategy fully leverages all hierarchical features, which enables the lightweight model to achieve better SR performance. In this paper, we also design an Efficient Convolutional Block (ECB) as the basic building block of DFAN. It comprises group convolutional layers with channel shuffle operation. Although ECB is compact, DFAN can still achieve competitive results with the help of the dual feature aggregation strategy.

In summary, our main contributions are as follows:

(i) We propose DFAN, which can achieve better SR performance with limited computational cost. It is more practical in real applications

(ii) We propose the dual feature aggregation strategy which aggregates local and global features in a progressive manner. It could make full use of all hierarchical features through feature reuse, which enhances the feature utilization while introducing only marginal computation cost. With our dual feature aggregation strategy, the lightweight SR model can achieve better cost-effectiveness

(iii) We also propose ECB as the basic building structure, which can extract hierarchical features in a computationally economical way

(iv) We show through extensive experiments that our model can achieve competitive results against state-of-the-art methods with relatively fewer parameters and calculations

2. Related Work

2.1. Lightweight SISR. Since Dong et al. [4] first applied CNNs to design Super-Resolution Convolutional Neural Network (SRCNN) and achieved significant improvement, deep learning based SISR methods have been actively explored and shown great advantages in representation capability. To obtain more powerful features for image reconstruction, they continue to enlarge the model size or network depth. Most existing SR methods have hundreds of convolutional layers, such as Residual Channel Attention Network (RCAN) [13], Residual Dense Network (RDN), and Deep Alternating Network (DAN) [14]. However, these methods are computationally expensive for real application. Thus, more and more lightweight SR methods are proposed. Deep Recursive Residual Network (DRRN) [9] and Memory Network (MemNet) [10] introduce recursive learning or weight sharing schemes to reduce parameters. However, they need to increase the computational complexity to compensate for the performance drop. Another idea is to build relatively shallower models, which can cut down the model size and calculations at the same time. Cascading Residual Network (CARN) [11], Information Distillation Network (IDN) [12], and Information Multi-Distillation Network (IMDN) [15] are all lightweight networks that have fewer than 40 layers. However, the shallow architecture could restrict their representation ability to some extent. For our method, we improve the feature utilization through dual feature aggregation, which can better balance the SR performance and computational cost.

2.2. Group Convolution. There has been rising interest in designing small and efficient neural networks [16–19] since

many deep and complicated neural networks are infeasible in practical applications. Group convolution is an important method for designing efficient neural networks. The application of the group convolution method dates back to [20] where the model is distributed over two GPUs, resulting in gains in accuracy and convergence speed. Depthwise convolution is a special case of group convolution and is originally introduced in [21]. In depthwise convolution, the number of groups is equal to the number of channels. Based on the depthwise convolution, Mobile Network (MobileNet) [18] gains state-of-the-art results among lightweight models in many visual tasks. Then, group convolution and depthwise convolution are generalized in a novel form in [22]. Channel shuffle operation is also proposed in [22] to overcome the side effect of group convolution. Recently, group convolution has been used in some lightweight image super-resolution methods. Ahn et al. [11] proposed efficient residual block containing group convolutional layers, and Hui et al. [12] introduced group convolution to some specific layers. However, there is still room for improvement in the reconstruction performance of these two models. In our DFAN, group convolution is used as a basic building unit without affecting the reconstruction performance.

2.3. Deep Feature Aggregation. As the feature representation capability of a single network layer is limited [23, 24], deep feature aggregation is typically used to fuse features of different layers, which can improve the representation capability in a computationally economical way. For instance, the Densely Connected Network (DenseNet) [25] and the Feature Pyramid Network (FPN) [26] are the dominant architectures for semantic feature aggregation and spatial feature aggregation [27]. DenseNet can better propagate features and gradients through dense connections that connect each layer to every other layer in a feed-forward fashion. FPN can equalize resolution and standardize semantics across the levels of a pyramidal feature hierarchy through top-down and lateral connections. Besides, Residual Network (ResNet) [8] is also a typical feature aggregation method which aggregates features via simple element-wise summation. Recently, Yu et al. [28] proposed an iterative aggregation method and a hierarchical aggregation method, which can further improve the performance of the aforementioned dominant architectures in many visual tasks. Inspired by this work, we introduce an iterative and adaptive global feature aggregation module to DFAN, obtaining more comprehensive information and improving reconstruction performance.

3. Proposed Method

3.1. Network Architecture. As depicted in Figure 2(a), DFAN mainly consists of four parts: the shallow feature extraction layer, stacked local feature aggregation modules, the global feature aggregation module, and the upsampling module.

The shallow feature extraction layer contains only one convolutional layer. It extracts shallow features F_0 from the LR image. Then, F_0 is input into the stacked LAMs for global residual learning. There are M stacked LAMs, and the local aggregated feature from the m^{th} LAM can be formulated as

$$F_m = f_{\text{LAM}}^m(F_{m-1}) = f_{\text{LAM}}^m\left(f_{\text{LAM}}^{m-1}\left(\cdots\left(f_{\text{LAM}}^1(\text{F}_0)\right)\right)\right), \quad (1)$$

where f_{LAM}^m refers to the operation of the m^{th} LAM, and F_m is the local aggregated feature from it. As shown in Figure 2, each LAM is composed of a series of ECBs, therefore, f_{LAM}^m can be viewed as a composite function.

After that, GAM fully leverages local aggregated features from LAMs in an iterative way, which can be expressed as

$$F_A = f_{\text{GAM}}(F_1, F_2, \cdots, F_G), \quad (2)$$

where F_A is the global aggregated feature. f_{GAM} denotes the operation of GAM. Then, the global long skip connection adds F_0 to F_A, obtaining the final aggregated feature F. The global skip connection can better propagate information and gradients, thus, stabilizing the training of DFAN.

Finally, we use an upscale module proposed in [29] to restore the final SR image I_{SR}. That is,

$$I_{\text{SR}} = f_{\text{conv}}\left(f_{\uparrow}\left(f_{g\text{conv}}(F)\right)\right) = f_{\text{DFAN}}(I_{\text{LR}}), \quad (3)$$

where $f_{g\text{conv}}$ denotes the group convolution, f_{conv} indicates the standard convolution, and $f_{\uparrow}$ is the upscaling module.

3.2. Local Feature Aggregation. Since features of different layers contain different weighted information, adaptively aggregating all hierarchical features could effectively improve the representation ability. Referring to [28], the key axes of feature fusion are semantic and spatial, which are closely related to channel and spatial dimensions, respectively. Thus, we propose the dual feature aggregation strategy, in which features are locally aggregated along the channel dimension, and then globally aggregated along the spatial dimension. In this subsection, we first explain the local feature aggregation.

3.2.1. Efficient Convolutional Block. As depicted in Figure 2(c), ECB is the basic building block of LAM. ECB is a residual learning module consisting of two group convolutional layers with channel shuffle operation [22] and a channel attention module [7]. Group convolution with channel shuffle operation can extract useful features in a computationally economical way. Assuming the group size of an $s \times s$ group convolutional kernel is g, the parameter amount and computation complexity of this group convolutional kernel will be both $1/g$ of an $s \times s$ standard convolutional kernel. Moreover, the channel shuffle operation enhances the information exchange among channels without extra parameters and calculations. There are B ECBs in each LAM. LAM fuses hierarchical features from ECBs by exploring the interchannel relationship. The local aggregated feature F_m from the m^{th} LAM can be obtained by

$$F_m = f_{\text{conv}}\left(\left[F_m^1, F_m^2 \cdots, F_m^B\right]\right) + F_{m-1}, \quad (4)$$

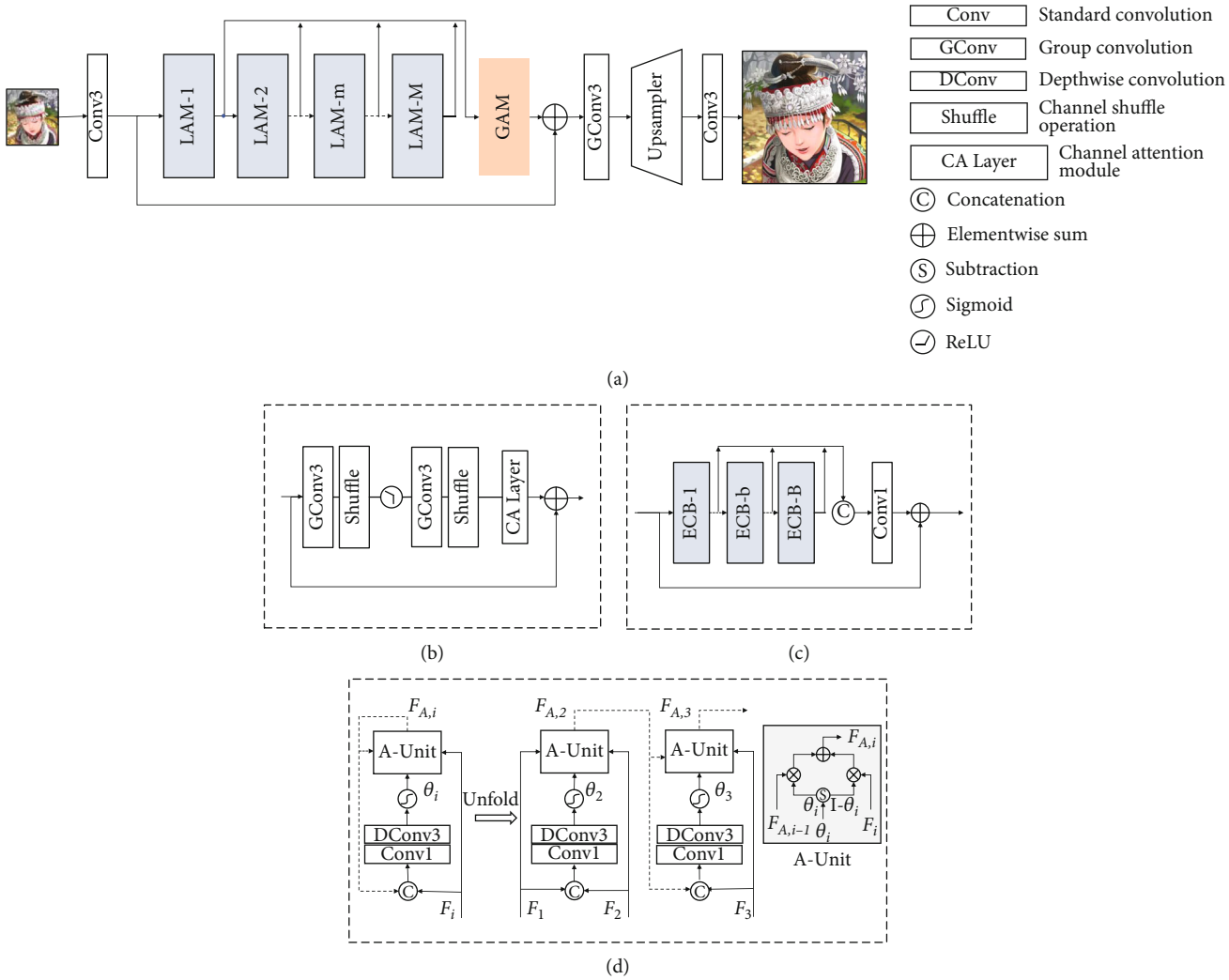


Figure 2: Framework of DFAN and its submodules. (a) Architecture of DFAN. (b) Efficient convolutional block (ECB). (c) Local aggregation module (LAM). (d) Global aggregation module (GAM).

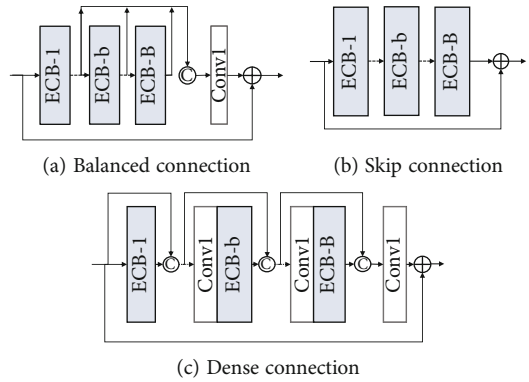
where $[F_m^1, F_m^2 \cdots, F_m^B]$ represents the concatenation of local features from ECBs in the m^{th} LAM.

3.2.2. Balanced Connection. The connection method in LAM is what we call balanced connection. As shown in Figure 3, compared with two commonly used connection methods in SR, i.e., skip connection and dense connection, our balanced connection is more flexible than skip connection and more lightweight than dense connection. The analysis is as follows:

(1) *Difference to Skip Connection.* As shown in Figure 3(b), for each LAM, if we only use skip connection which makes the elementwise sum of the hierarchical feature maps, all hierarchical features will contribute equally to the final aggregated feature. It may be inflexible since different features contain information of different importance. Our balanced connection can simply solve this issue by a 1×1 convolutional kernel. This convolutional kernel assigns specific learned weights to each pixel of local features, thus, adaptively aggregating them along the channel dimension

(2) *Difference to Dense Connection.* As shown in Figure 3(c), dense connection connects each ECB and all preceding ECBs to be concatenated and compressed as inputs to all subsequent ECBs, which requires more 1×1 convolutional kernels and harms the overall efficiency. However, our balanced connection directly connects each ECB for feature aggregation, which not only fully uses local features but also greatly reduces the number of parameters and computation operations

3.3. Global Feature Aggregation. The spatial dimension is orthogonal to the channel dimension. Thus, further fusing local aggregated features along the spatial dimension could supplement more information. Besides, since local aggregated features contain abundant information, it could be suitable to aggregate them in a coarse to fine fashion.



(a) Balanced connection (b) Skip connection

(c) Dense connection

Figure 3: Comparisons of three different connection methods.

Therefore, we design GAM, which can further fuse local aggregated features with spatial attention mechanism in an iterative manner.

In Figure 2(d), $F_{A,i}$ represents the global aggregated feature in the i^{th} iteration, and F_i represents the output of the i^{th} LAM. The iterative fusion of GAM can be formulated as

$$F_{A,i} = \begin{cases} F_1, & i = 1, \\ f_G([F_{A,i-1}, F_i]), & i > 1, \end{cases} \tag{5}$$

where $F_{A,i}$ is initialized with F_1, which is the output of the first LAM. f_G represents the global aggregation of GAM.

The main parts of GAM are (1) spatial attention generation and (2) iterative feature aggregation. First, the spatial attention θ_i is generated by the following operation,

$$\theta_i = \sigma(f_{d\text{conv}}(f_{\text{conv}}([F_{A,i-1}, F_i]))), \tag{6}$$

where f_{conv} denotes a 1×1 convolutional kernel that reduces the channel number of $[F_{A,i}, F_i]$ by half. $f_{d\text{conv}}$ denotes a 3×3 depthwise convolutional kernel to extract spatial information. Depthwise convolution applies a single filter to each input channel, which is more efficient than common convolution in terms of memory and computation. σ is the Sigmoid activation function constraining the spatial attention to $(0, 1)$. The spatial attention θ_i is the same size as F_i and $F_{A,i-1}$. Second, as shown in Figure 2(d), the feature fusion in A-Unit can be formulated as

$$F_{A,i} = \theta_i \otimes F_{A,i-1} + (I - \theta_i) \otimes F_i, \tag{7}$$

where $\otimes$ denotes the Hadamard product. I is the tensor with all elements being 1.

After the G^{th} iteration, we obtain the final global aggregated feature F_A. The overall iterative global aggregation can be summarized as follows,

$$\begin{cases} \sum_{i=1}^{G} \Theta_i = I, \\ F_A = \sum_{i}^{G} \Theta_i \otimes F_i, \end{cases} \tag{8}$$

where Θ_i denotes the final spatial attention for F_i, which is determined by all the local aggregated features from LAMs, thus, highly comprehensive. Additionally, Eq. (8) indicates that the global feature aggregation strategy satisfies the convex combination.

Table 1: Investigation of group convolution and standard convolution. We observe the best performance on PSNR with scaling factor $\times 4$.

	Conv	GConv	Set5	Set14	BSD100	Urban100
DFAN_W	✓	—	32.00	28.45	27.48	25.80
DFAN_D	✓	—	32.27	28.61	27.58	26.20
DFAN	—	✓	32.29	28.65	27.60	26.22

4. Experiments

4.1. Experimental Setup

4.1.1. Datasets and Metrics. We use the training set of DIV2K [30] to train all of our models. For testing, we use five standard benchmark datasets: Set5 [31], Set14 [32], BSD100 [33], Urban100 [34], and Manga109 [35]. The visual quality of SR results is evaluated with Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) [36] on the Y channel (i.e., luminance) of transformed YCbCr space. We also represent the number of parameters and multiply-adds to evaluate the memory footprint and computation complexity, respectively.

4.1.2. Degradation Models. To fully demonstrate the effectiveness of our DFAN, we use two degradation models to

Table 2: Study on the dual feature aggregation strategy. PSNR for 4× SR. "LA" denotes local feature aggregation, and "GA" denotes global feature aggregation.

	LAM	GAM	Set5	Set14	BSD100	Urban100
LAM0_GAM0	—	—	32.11/0.8937	28.50/0.7797	27.52/0.7343	25.96/0.7809
LAM1_GAM0	✓	—	32.16/0.8944	28.57/0.7814	27.57/0.7361	26.11/0.7864
LAM1_GAM1	✓	✓	32.29/0.8964	28.65/0.7832	27.60/0.7373	26.22/0.7898

simulate LR images. The first is the bicubic degradation model. The bicubic degradation model simulates LR images on scale $\times 2$, $\times 3$, and $\times 4$. The second is the blur-down degradation model that blurs HR images by 7×7 Gaussian kernel with a standard deviation 1.6. The blurred image is then downsampled on scale $\times 3$.

4.1.3. Training Details. The size of LR patches is 48×48. During training, we randomly rotate input images by 90°, 180°, or 270° and flip them horizontally or vertically. The batch size is 32. We use $L1$ loss as the loss function. We use Adam as the optimizer. The initial learning rate is $2e-4$, decayed by half every 200 epochs. We train our model for 1000 epochs.

4.2. Study on Efficient Convolutional Block. Different from most of the super-resolution networks, our DFAN uses group convolutional kernels instead of standard convolutional kernels in an ECB to extract features. Since group convolution is a basic operation of our ECB, we design DFAN_W and DFAN_D to validate the effectiveness of ECB. These two models have the same structure as DFAN, but group convolutional kernels in ECBs are replaced with standard convolutional kernels. All three models have similar number of parameters and computation operations, i.e., approximately 900 K and 60 G, respectively.

We denote the number of ECBs in each LAM as B, the number of LAMs as M, and the number of channels of each intermediate feature as C. For our DFAN, we set B, M, and C to 10, 6, and 64, respectively, and the group number of each group convolutional kernel in ECBs is 8. We set B, M, and C to 3, 2, and 64, respectively, for DFAN_W, and these hyperparameters to 10, 6, and 27, respectively, for DFAN_D. In other words, the width of DFAN_W is the same as DFAN. While the depth of DFAN_D is the same as DFAN.

As shown in Table 1, group convolution makes an outstanding trade-off between representation capability and computational costs. Compared with standard convolution, group convolution can make the model deeper or wider with limited parameters and calculations, which is beneficial to obtain richer hierarchical information.

4.3. Study on Dual Feature Aggregation. In this section, we experimentally investigate the effectiveness of the dual feature aggregation strategy. LAM0_GAM0 is the baseline network by removing balanced connections in LAM and GAM from DFAN. LAM1_GAM0 is built by removing GAM from DFAN. LAM1_GAM1 has both LAM and GAM, which is the same as DFAN. As shown in Table 2, when only LAM is added, PSNR is improved by approximately 0.05 dB. When both LAM and GAM are added, the performance is improved by a large margin (PSNR: +0.15 dB on Set14).

4.3.1. LAM Analysis. To intuitively show the effectiveness of LAM, we plot the training curves of LAM0_GAM0 and LAM1_GAM0 in Figure 4(a). Benefitting from the balanced connection in LAM, gradients could be better propagated. The margin between the two curves indicates that LAM could not only help the network converge faster but also help it converge to a better point. Additionally, the weight distribution is visualized in Figure 4(b). This indicates how much information of each ECB in an LAM contributes to the local aggregated feature generated by this LAM. Features from different ECBs contribute differently to local aggregated features, which suggest that LAM could adaptively aggregate hierarchical features to improve the final performance.

4.3.2. GAM Analysis. We experimentally prove that GAM also works well for some other networks. We use a shallower RCAN [7] as the baseline network (denoted as sRCAN). To facilitate network training, we set the RG number to 3, and the RCAB number to 5 for sRCAN. Then, we apply our GAM to sRCAN, which is denoted as sRCAN_GAM. As Table 3 shows, with only a small increase in parameters and computational complexity (Paramerters:+9 K, MultAdds:+0.9G), GAM can significantly improve the SR performance on all the benchmark datasets with scaling factor $\times 4$. Therefore, GAM could be used as a general lightweight tool to improve the performance of some existing SR methods.

To better understand the adaptive and iterative aggregation strategy of GAM, we visualize the spatial attention heatmaps generated by GAM in Figure 5. The 3D attention is transformed to 2D by taking the absolute mean along the channel dimension and then normalized to $[0, 1]$ over the spatial dimension. We can see that (1) spatial attention for different LAMs focuses on regions of different frequencies. For example, the spatial attention for LAM_1 (Figure 5(a)) focuses on low-frequency regions such as the background. While the spatial attention for LAM_6 (Figure 5(f)) focuses more on high-frequency regions with rich textures. Thus, both high-frequency and low-frequency information is important for SR. (2) Although some spatial attention focuses on high-frequency regions, they emphasize different parts. In LAM_6, more attention is given to regions of the main object. But in LAM_5, high-frequency regions in the background are emphasized. It indicates that GAM provides

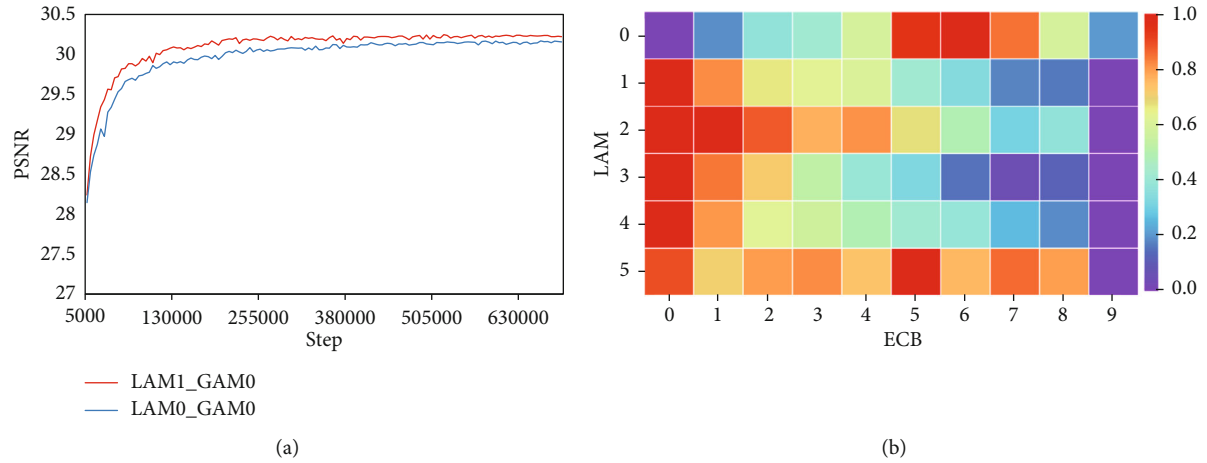


Figure 4: (a) Training curves of LAM0_GAM0 and LAM1_GAM0. (b) Weight distribution of ECBs in each LAM. The color pixel (b, m) encodes the average $L1$ norm of the weights connecting the feature from ECB-b in LAM-m to the local aggregated feature of LAM-m.

Table 3: Study on GAM. The computation cost and average PSNR/SSIM for $4\times$ SR.

Method	sRCAN	sRCAN_GAM
Params(K)/MultAdds(G)	208/16.1	217/17.0
Set5	31.69/0.8878	31.84/0.8901
Set14	28.29/0.7741	28.36/0.7763
BSD100	27.38/0.7292	27.42/0.7311
Urban100	25.53/0.7667	25.60/0.7702
Manga109	29.59/0.8955	29.73/0.8985

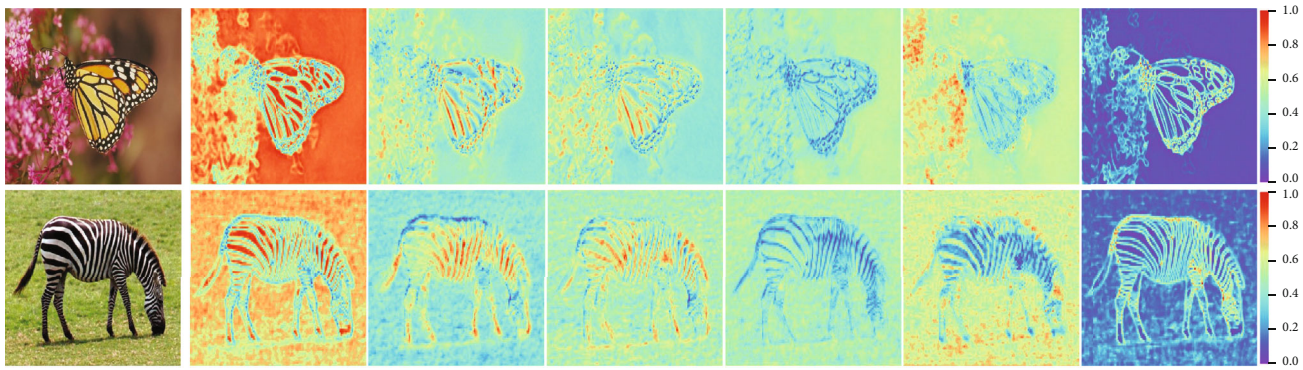


Figure 5: Spatial attention heatmaps generated by GAM for each LAM.

additional flexibility to deal with different types of information, which could enhance the representation capability.

4.4. Results with Bicubic Degradation Model. We compare DFAN with other state-of-the-art methods: SRCNN [4], Fast Super-Resolution Convolutional Neural Network (FSRCNN) [37], Very Deep Super-Resolution(VDSR) [5], Deeply-Recursive Convolutional Network (DRCN) [38], DRRN [9], MemNet [10], CARN [11], IDN [12], and IMDN [15].

4.4.1. Quantitative Results. We evaluate the average PSNR and SSIM on five benchmark datasets. In particular, we also calculate the number of parameters and multiply-adds of these models by assuming the HR image size to be 720p (1280×720). In Table 4, the proposed DFAN performs favorably against these methods on all benchmark datasets for $2\times$, $3\times$, and $4\times$ SR. Note that the number of parameters of our method is inconsistent for different scales because we apply the pixelshuffle operation [29] for upscaling, and the convolutional kernels in the upscaling module are of different sizes for different scales. CARN [11] used to be a strong baseline for lightweight SR models, but our DFAN outperforms it by a large margin (PSNR:+0.17 dB,

Table 4: Quantitative results with the bicubic degradation model. Best results are highlighted.

Method	Scale	Params (K)	MultAdds (G)	Set5		Set14		BSD100		Urban100		Manga109	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	×2	—	—	33.66	0.9299	30.24	0.8688	29.56	0.8431	26.88	0.8403	30.80	0.9339
SRCNN [4]	×2	57	52.7	36.66	0.9542	32.42	0.9063	31.36	0.8879	29.50	0.8946	35.60	0.9663
FSRCNN [37]	×2	12	6.0	37.00	0.9558	32.63	0.9088	31.53	0.8920	29.88	0.9020	36.67	0.9710
VDSR [2]	×2	665	612.6	37.53	0.9587	33.03	0.9124	31.90	0.8960	30.76	0.9140	37.22	0.9750
DRCN [38]	×2	1774	9788.7	37.63	0.9588	33.04	0.9118	31.85	0.8942	30.75	0.9133	37.55	0.9732
LapSRN [39]	×2	813	29.9	37.52	0.9590	33.08	0.9130	31.80	0.8950	30.41	0.9100	37.27	0.9740
DRRN [9]	×2	297	6796.9	37.74	0.9591	33.23	0.9136	32.05	0.8973	31.23	0.9188	37.88	0.9749
MemNet [10]	×2	677	2262.4	37.78	0.9597	33.28	0.9142	32.08	0.8978	31.31	0.9195	37.72	0.9740
IDN [12]	×2	677	168.4	37.83	0.9600	33.30	0.9148	32.08	0.8985	31.27	0.9196	38.01	0.9749
IMDN [15]	×2	694	158.8	38.00	0.9605	33.63	0.9177	32.19	0.8996	32.17	0.9283	38.88	0.9774
CARN [11]	×2	1592	222.8	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.51	0.9312	38.36	0.9765
DFAN	×2	877	201.4	38.11	0.9609	33.74	0.9189	32.21	0.9002	32.36	0.9305	38.93	0.9774
Bicubic	×3	—	—	30.39	0.8682	27.55	0.7742	27.21	0.7385	24.46	0.7349	26.95	0.8556
SRCNN [4]	×3	57	52.7	32.75	0.9090	29.28	0.8209	28.41	0.7863	26.24	0.7989	30.48	0.9117
FSRCNN [37]	×3	12	5.0	33.16	0.9140	29.43	0.8242	28.53	0.7910	26.43	0.8080	31.10	0.9210
VDSR [5]	×3	665	612.6	33.66	0.9213	29.77	0.8314	28.82	0.7976	27.14	0.8279	32.01	0.9340
DRCN [38]	×3	1774	9788.7	33.82	0.9226	29.76	0.8311	28.80	0.7963	27.15	0.8276	32.24	0.9343
DRRN [9]	×3	297	6796.9	34.03	0.9244	29.96	0.8349	28.95	0.8004	27.53	0.8378	32.71	0.9379
MemNet [10]	×3	677	2262.4	34.09	0.9248	30.00	0.8350	38.96	0.8001	27.56	0.8376	32.51	0.9369
IDN [12]	×3	677	84.4	34.11	0.9253	29.99	0.8354	28.95	0.8031	27.42	0.8359	32.71	0.9381
IMDN [15]	×3	703	71.5	34.36	0.9270	30.32	0.8417	29.09	0.8046	28.17	0.8519	33.61	0.9445
CARN [11]	×3	1592	118.8	34.29	0.9255	30.29	0.8407	29.06	0.8034	27.38	0.8404	33.50	0.9440
DFAN	×3	900	92.6	34.49	0.9280	30.37	0.8426	29.13	0.8059	28.31	0.8551	33.78	0.9458
Bicubic	×4	—	—	28.42	0.8104	26.00	0.7027	25.96	0.6675	23.14	0.6577	24.89	0.7866
SRCNN [4]	×4	57	52.7	30.48	0.8628	27.49	0.7503	26.90	0.7101	24.52	0.7221	27.58	0.8555
FSRCNN [37]	×4	12	4.6	30.71	0.8657	27.59	0.7535	26.98	0.7150	24.62	0.7280	27.90	0.8610
VDSR [5]	×4	665	612.6	31.35	0.8838	28.01	0.7674	27.29	0.7251	25.18	0.7524	28.83	0.8870
DRCN [38]	×4	1774	9788.7	31.53	0.8854	28.02	0.7670	27.23	0.7233	25.14	0.7510	28.93	0.8854
LapSRN [39]	×4	813	149.4	31.54	0.8850	28.19	0.7720	27.32	0.7280	25.44	0.7638	29.09	0.8900
DRRN [9]	×4	297	6796.9	31.68	0.8888	28.21	0.7720	27.38	0.7284	25.44	0.7638	29.45	0.8946
MemNet [10]	×4	677	2262.4	31.74	0.8893	28.26	0.7723	27.40	0.7281	25.50	0.7630	29.42	0.8942
IDN [12]	×4	677	54.9	31.82	0.8903	28.25	0.7730	27.41	0.7297	25.41	0.7632	29.41	0.8942
IMDN [15]	×4	715	40.9	32.21	0.8948	28.58	0.7811	27.56	0.7353	26.04	0.7838	30.45	0.9075
CARN [11]	×4	1592	90.9	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837	30.47	0.9084
DFAN	×4	896	55.8	32.30	0.8964	28.65	0.7832	27.60	0.7373	26.22	0.7898	30.59	0.9097

SSIM:+0.0024 on Set5) with 42% fewer parameters and 37% fewer MultAdds on scale ×4. It indicates that our method can achieve a better trade-off between computational cost and effectiveness. Therefore, feature aggregation has promising prospects in the research of lightweight image SR.

4.4.2. Visual Results. In Figure 6, we show visual comparisons on scale ×4. Our method restores the letter "*g*" in "*ppt3*" more clearly, while most other methods encounter artifacts or edge distortion. For "*img030*" in Urban100 and "*img86000*" in BSD100, most methods do not reconstruct the contour of the window well, but our method can reconstruct these edges better.

4.5. Results with Blur-Down Degradation Model. As mentioned in the main submission, we further apply our method

Figure 6: Visual results for 4× SR with the bicubic degradation model.

Table 5: Quantitative results for 3× SR with the blur-down degradation model. Best results are highlighted.

Model	Scale	Params (K)	MultAdds (G)	Set5		Set14		BSD100		Urban100		Manga109	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	×3	—	—	28.78	0.8308	26.38	0.7271	26.33	0.6918	23.52	0.6862	25.46	0.8149
SRCNN [4]	×3	57	52.7	32.75	0.9090	29.28	0.8209	28.41	0.7863	26.24	0.7989	29.47	0.8924
FSRCNN [37]	×3	12	5.0	33.16	0.9140	29.43	0.8242	28.53	0.7910	26.43	0.8080	23.04	0.7927
VDSR [5]	×3	665	612.6	33.66	0.9213	29.77	0.8314	28.82	0.7976	27.14	0.8279	31.06	0.9234
IDN [12]	×3	677	84.4	34.30	0.9256	30.33	0.8400	29.08	0.8034	27.91	0.8455	33.54	0.9427
IMDN [15]	×3	703	71.5	34.38	0.9263	30.32	0.8401	29.09	0.8038	28.03	0.8477	33.70	0.9438
CARN [11]	×3	1592	118.8	34.40	0.9264	30.39	0.8415	29.13	0.8046	28.07	0.8489	33.69	0.9442
DFAN	×3	900	92.6	34.50	0.9274	30.43	0.8419	29.17	0.8058	28.27	0.8526	33.99	0.9456

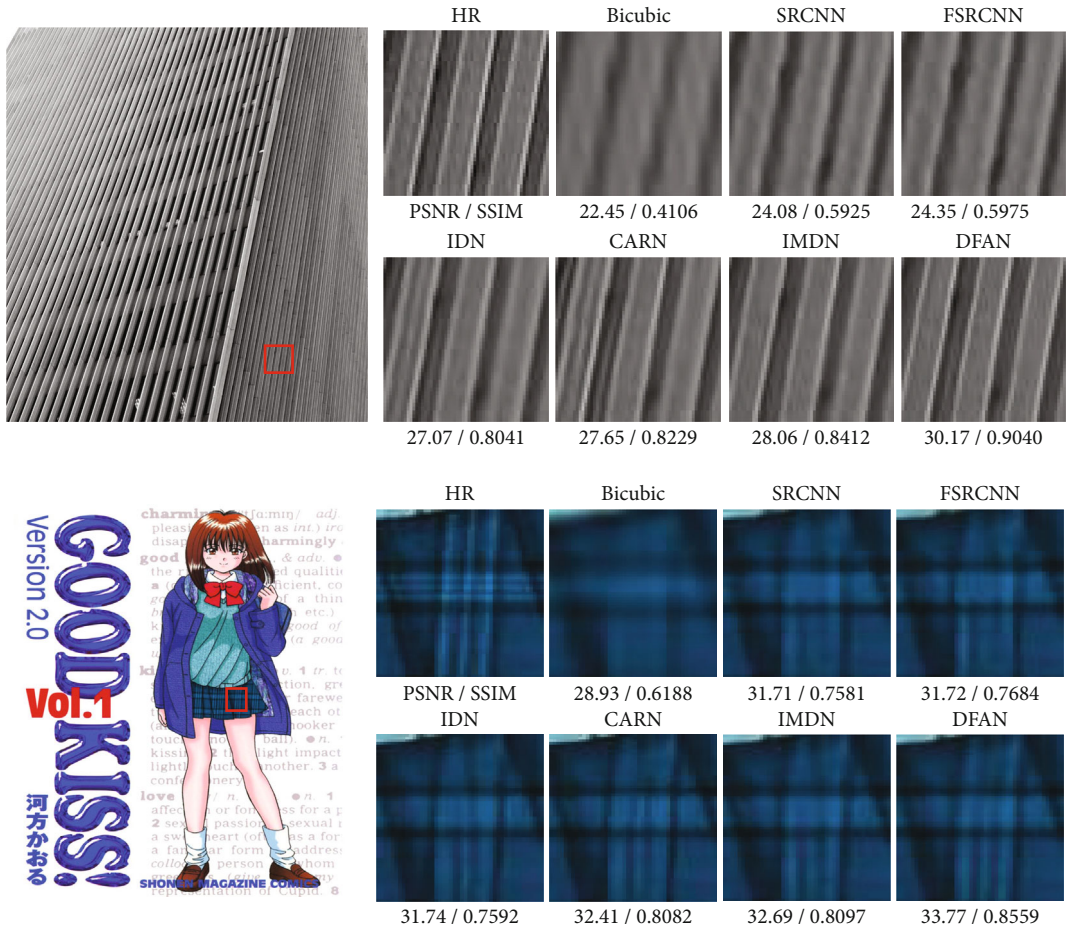


Figure 7: Visual results for 3× SR with the blur-down degradation model.

to super-resolve images with blur-down degradation, which is also commonly used in [7, 13]. We compare DFAN with SRCNN [4], FSRCNN [37], VDSR [5], CARN [11], IDN [12], and IMDN [15].

As shown in Table 5, compared with the networks that are stacked by several elaborately designed building blocks, such as CARN, IDN, and IMDN, our lightweight network with the dual feature aggregation strategy can better leverage the hierarchical features. In addition, the visual comparison in Figure 7 also demonstrates the superiority of our method.

5. Conclusions

We propose DFAN that can strike a better trade-off between SR performance and computational cost. The proposed dual feature aggregation strategy makes local and global feature aggregations adaptively. Through feature reuse, it could simultaneously improve feature utilization and representation ability. Benefitting from the dual feature aggregation strategy, our network achieves competitive performances with fewer parameters and lower computational complexity, which is more practical for real applications.

Data Availability

The image datasets supporting this work are from previously reported studies and datasets, which have been cited. The processed data are available at the repository: BasicSR(https://github.com/xinntao/BasicSR/blob/master/docs/DatasetPreparation.md#Image-Super-Resolution).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China (2019YFB1406200) and was also the research achievement of the Key Laboratory of Digital Rights Services. It is based on our previous teamwork, Lightweight Image Super-Resolution via Dual Feature Aggregation Network, presented in 2021 at the 2nd International Conference on Culture-oriented Science & Technology (ICCST).

References

[1] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy, "Edvr: video restoration with enhanced deformable convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Long Beach California, USA, 2019.

[2] J. Kim, G. Li, I. Yun, C. Jung, and J. Kim, "Edge and identity preserving network for face super-resolution," *Neurocomputing*, vol. 446, pp. 11–22, 2021.

[3] J. Peng, K. Fu, Q. Wei, Y. Qin, and Q. He, "Improved multiview decomposition for single-image high-resolution 3D object reconstruction," *Wireless Communications and Mobile Computing*, vol. 2020, 14 pages, 2020.

[4] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[5] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1646–1654, Las Vegas Nevada, USA, 2016.

[6] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 136–144, Honolulu Hawaii, USA, 2017.

[7] 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*, pp. 286–301, 2018.

[8] 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*, pp. 770–778, Las Vegas Nevada, USA, 2016.

[9] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3147–3155, Honolulu, Hawaii, 2017.

[10] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: a persistent memory network for image restoration," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4539–4547, Venice, Italy, 2017.

[11] 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*, pp. 252–268, Munich, Germany, 2018.

[12] 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*, pp. 723–731, Salt Lake City Utah, 2018.

[13] 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*, pp. 2472–2481, Salt Lake City Utah, 2018.

[14] Z. Luo, Y. Huang, S. Li, L. Wang, and T. Tan, "Unfolding the alternating optimization for blind super resolution," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.

[15] Z. Hui, X. Gao, Y. Yang, and X. Wang, "Lightweight image super-resolution with information multidistillation network," in *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2024–2032, Nice, France, 2019.

[16] M. Wang, B. Liu, and H. Foroosh, "Factorized convolutional neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 545–553, Venice, Italy, 2017.

[17] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and<0.5 mb model size," 2016, https://arxiv.org/abs/1602.07360.

[18] A. G. Howard, M. Zhu, B. Chen et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[19] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4820–4828, Las Vegas Nevada, USA, 2016.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[21] L. Sifre and S. Mallat, "Rigid-motion scattering for texture classification," 2014, https://arxiv.org/abs/1403.1687.

[22] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: an extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856, Salt Lake City Utah, USA, 2018.

[23] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, pp. 818–833, Springer, Cham, 2014.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, Boston Massachusetts, USA, 2015.

[25] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, Honolulu Hawaii, USA, 2017.

[26] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, 2017.

[27] J. Liu, R. Jia, W. Li, F. Ma, and X. Wang, "Image dehazing method of transmission line for unmanned aerial vehicle inspection based on densely connection pyramid network," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8857271, 9 pages, 2020.

[28] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2403–2412, Salt Lake City Utah, USA, 2018.

[29] W. Shi, J. Caballero, F. Huszár et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883, Las Vegas Nevada, USA, 2016.

[30] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: dataset and study," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 126–135, Honolulu Hawaii, USA, 2017.

[31] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "Low-complexity single-image superresolution based on nonnegative neighbor embedding," in *British Machine Vision Conference*, BMVA Press, 2012.

[32] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International Conference on Curves and Surfaces*, pp. 711–730, Springer, 2010.

[33] 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 *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV*, pp. 416–423, Vancouver, BC, Canada, 2001.

[34] J.-B. Huang, A. Singh, and N. Ahuja, "Single image superresolution from transformed self-exemplars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5197–5206, Boston Massachusetts, USA, 2015.

[35] Y. Matsui, K. Ito, Y. Aramaki et al., "Sketch-based manga retrieval using manga109 dataset," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21811–21838, 2017.

[36] 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.

[37] C. Dong, C. C. Loy, and X. Tang, "Accelerating the superresolution convolutional neural network," in *European Conference on Computer Vision*, pp. 391–407, Springer, 2016.

[38] J. Kim, J. Kwon Lee, and K. Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1637–1645, Las Vegas Nevada, USA, 2016.

[39] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 624–632, Honolulu Hawaii, USA, 2017.