An Efficient Group Feature Fusion Residual Network for Image Super-Resolution

Pengcheng Lei and Cong Liu

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, 516 Jungong Road, Shanghai, 200093, China.

Abstract. Convolutional neural networks (CNNs) have made great breakthrough in the field of image super-resolution (SR). However, most current methods are usually to improve their performance by simply increasing the depth of their network. Although this strategy can get promising results, it is inefficient in many real-world scenarios because of the high computational cost. In this paper, we propose an efficient group feature fusion residual network (GFFRN) for image super-resolution. In detail, we design a novel group feature fusion residual block (GFFRB) to group and fuse the features of the intermediate layers. In this way, GFFRB can enjoy the merits of the lightweight of the group convolution and the high-efficiency of the skip connections, thus achieving better performance compared with most current residual blocks. Experiments on the benchmark test sets show that our models are more efficient than most of the state-of-the-art methods.

1 Introduction

Single image super-resolution (SR) is a classical low-level computer vision problem that tries to restore a high-resolution (HR) image from a single low-resolution (LR) image. Since the reconstructed HR image contains rich details, SR techniques have been widely used in the field of image processing such as face authentication, public relations security monitoring and so on [1, 2].

SR is an inherent ill-posed problem since a multiplicity of solutions exist for any given LR image. To solve this problem, numerous SR methods have been proposed, including interpolation-based methods [3], reconstruction-based methods [4] and learning-based methods [5, 6]. In recent years, the convolutional neural network (CNN) based SR methods, with their powerful nonlinear expression ability, have achieved dramatic success in the field of image SR.

Since Dong et al. [7] firstly proposed a three-layer CNN (SRCNN) for image SR, a large number of CNN based methods have emerged. The early CNN based SR methods used shallow networks (less than 10 layers) to learn the mapping function between LR and HR images, such as FSRCNN [8] and ESPCN [9]. Since He et al. [10] proposed ResNet to solve the convergence problem of deep networks, the SR methods began to grow in depth to improve the reconstruction accuracy. Kim et al. [11] utilized global residual learning to build a very deep network (VDSR, about 20 layers). Lim et al. [12] proposed an Enhanced Deep

Residual Network (EDSR, about 60 layers), which used a stack of residual blocks to gradually recover the high frequency details from the LR inputs. Zhang et al. [13] further proposed a Residual Dense Network (RDN, about 150 layers), which employed dense connections in the residual block to extract abundant local features. Zhang et al. [14] proposed a Residual Channel Attention Network (RCAN). By using residual in residual (RIR) structure, the depth of the network reached 400 layers.

All the methods introduced above show excellent reconstruction performance by increasing the depth of their network, however, they all have a huge number of parameters and computations, which will put a high demand on the hardware resources. Considering that the SR method may be operated on a mobile device, the computing and storage resources of them are limited. A huge network will consume more hardware resources and result in longer inference time, which will seriously affect the user experience. Another time demanding scenario is video streaming data SR because it contains a large number of images and a huge network will affect the real-time performance of video image processing. Therefore, it is particularly important to design a more efficient and lightweight network.

2 Related work

2.1 Lightweight Neural Network

Recent studies indicate that Skip connection, Recursive network and Group convolution are three widely used strategies in current lightweight SR networks. The details of them are introduced as follows.

(1) **Skip connection**. Skip connection can enhance the information flow between different convolutional layers, thus improving the reconstruction accuracy. The most representative lightweight methods with this strategy are MemNet [15] and CARN [16]. MemNet designed a memory module to adaptively retain the useful information of different residual blocks. CARN employed the cascading mechanism to fuse the information among different residual blocks, thus building an efficient cascade residual network. However, the skip connections of these networks are only conducted between different residual blocks, thus the improvement of reconstruction accuracy is very limited.

(2) **Recursive network**. Recursive network designs a recursive unit and makes the data pass the unit repeatedly thus building a more complex mapping function. Through parameter sharing between different recursive phases, this strategy reduces the model parameters effectively. The most representative recursive networks are DRCN [17], DRRN [18], MemNet [15] and SRFBN [19]. Although these methods can achieve good performance with fewer parameters, they also have some problems: 1) Most of them upsample the LR image before CNN. 2) These methods usually use very deep networks to compensate for the performance degradation caused by parameter sharing. Both the two problems increase the time complexity of the network.

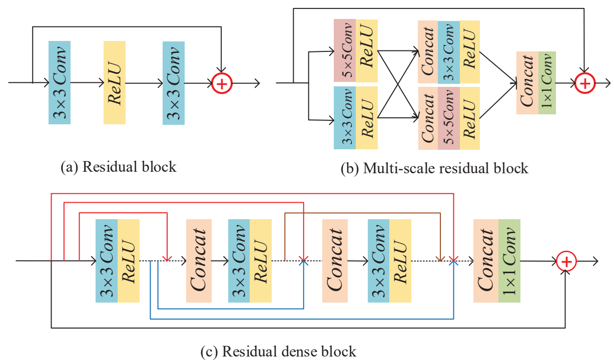

Fig. 1. The structure of current residual blocks: (a) Residual block, (b) Multi-scale residual block, (c) Residual dense block. The $\oplus$ operations are element-wise addition for residual learning.

(3) **Group convolution**. Group convolution groups the input feature maps and convolves them within each group. This strategy can reduce both the number of the parameters and the calculations of the model. AlexNet [20] firstly proposed the group convolution to solve the scarcity of hardware resources. MobileNet [21] designed a depthwise separable convolution network for mobile vision applications. In the field of image super-resolution, CARN-m [16] used this strategy to design a lightweight SR method for mobile devices. However, simply using group convolution to replace the traditional convolution will result in the decrease of the accuracy, so we need to combine it with some other strategies to build more efficient methods.

2.2 Recent Residual Blocks

To design an efficient network, a good way is to design a more efficient residual block. In recent years, many efficient residual blocks have been proposed to improve the reconstruction accuracy. Lim et al. [12] proposed a residual block (RB, as shown in figure 1(a)) by removing the BN operation of SRResNet [22] and got higher reconstruction accuracy. RB is really concise and effective, but its utilization of local information is limited. To solve this problem, Zhang et al. [13] proposed a dense residual block (RDB, as shown in figure 1(c)), which designed dense skip connections to continuously fuse the features of the current layer with those features of the previous layers. This structure has powerful nonlinear capability to fully extract the local information. However, the dense connection also introduces a large number of parameters and computations, which is not desirable for lightweight networks. Li et al. [23] proposed a multi-scale residual block (MSRB, as shown in figure 1(b)), which used convolution kernels of different sizes $(3 \times 3, 5 \times 5)$ to adaptively detect the features in different scales. However, the 5×5 filters do not seem efficient in lightweight models.

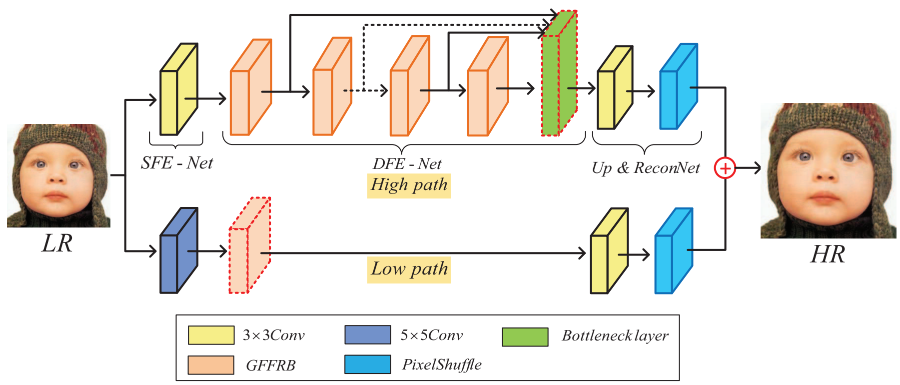

Fig. 2. The structure of the proposed group feature fusion residual network (GFFRN). The modules marked by the red dotted line represent the removed parts in GFFRN-L. GFFRN-L has the same structure with WDSR [24].

To solve these problems, in this paper, we propose a Group Feature Fusion Residual Network (GFFRN) and its lightweight version GFFRN-L. Both the two models consist of a series of Group Feature Fusion Residual Blocks (GFFRB). GFFRB is a newly proposed residual block in this paper, which takes advantage of group convolution and skip connection to fully extract abundant local features. More details are shown in Section 3.

The main contributions are as follows: **(1)** We propose a novel group feature fusion residual block (GFFRB), which combines the advantages of both the lightweight of the group convolution and the high-efficiency of the skip connections, thus achieving better performance compared with most current residual blocks. **(2)** Based on GFFRB, we propose an efficient two-path group feature fusion residual network (GFFRN), which achieves higher efficiency compared with most state-of-the-art methods. **(3)** To further reduce the number of parameters and computations, we also propose a lightweight network GFFRN-L by reducing the depth and the width of GFFRN. The proposed GFFRN-L achieves the best performance among the models that have less than 1M parameters.

3 Proposed Method

In this section, we will introduce the details of the proposed GFFRN, GFFRN-L and GFFRB respectively. Let's denote I_{LR} and $\widehat{I}_{HR}$ as the input and output of GFFRN respectively and both of them have C channels. We also denote $Conv(s, n)$ as a convolutional layer, where s represents the size of the filters and n represents the number of the filters.

3.1 Network Architecture of GFFRN

We first propose an efficient GFFRN for image SR. As shown in figure 2, GFFRN has two paths, e.t. a high-path and a low-path. The high-path of GFFRN has a powerful nonlinear ability, which uses multiple GFFRBs to restore the high-frequency information. It mainly consists of three parts: shallow feature extraction net (SFE-Net), deep feature extraction net (DFE-Net) and finally the upsampling and reconstruction net (Up&Recon-Net). The low-path has a relatively weak nonlinear ability, which uses a simple structure to restore the low-frequency information. The network structure of GFFRN is actually an improved version of WDSR [24]. Compared with WDSR, we mainly make improvements in two aspects. Firstly, we add a bottleneck layer [23, 13] in the high-path to make our network fully utilize the hierarchical features. Secondly, we add a GFFRB in the low-path to properly enhance its feature extraction ability. Note that the original WDSR only uses a 5×5 convolutional layer in the low-path to extract the low-frequency information.

In the high-path of GFFRN, the first part is SFE-Net and it uses one convolutional layer, $Conv(3, m)$, to extract shallow features and expand the number of channels from C to m, where m denotes the base channel number of the intermediate layers. The second part is DFE-Net and this part contains D GFFRBs and a bottleneck layer [23, 13]. The third part is Up&ReconNet. It consists of $Conv(3, C \times s^2)$ and a sub-pixel convolutional layer in sequence, where $s = (2, 3, 4)$ denotes the upscaling factor. In the low-path of GFFRN, it mainly consists of a convolutional layer $Conv(5, m)$, a GFFRB and an Up&ReconNet. The final HR image can be obtained by

$$\widehat{I}_{HR} = f_{HP}(I_{LR}) + f_{LP}(I_{LR}), \tag{1}$$

where $f_{HP}(\cdot)$ and $f_{LP}(\cdot)$ denote the operations of the high-path and the low-path respectively.

Loss function We employ $L1$ loss to optimize the proposed network. Given a training set $\{I^i_{LR}, I^i_{HR}\}^N_{i=1}$, which contains N LR images and their corresponding HR images. We need to minimize the $L1$ loss function between the reconstructed image $\widehat{I}_{HR}$ and the ground truth image I_{HR}. $L1$ loss function is shown in Eq. 2.

$$L(\Theta) = \frac{1}{N} \sum_{i=1}^{N} \|\widehat{I}^i_{HR} - I^i_{HR}\|_1, \tag{2}$$

where Θ represents the parameters of proposed network.

3.2 Lightweight GFFRN (GFFRN-L)

To further reduce the number of the parameters and computations, we also propose a lightweight network GFFRN-L (Less than 1M parameters). GFFRN-L has the same structure with WDSR [24]. Compared with GFFRN, there are

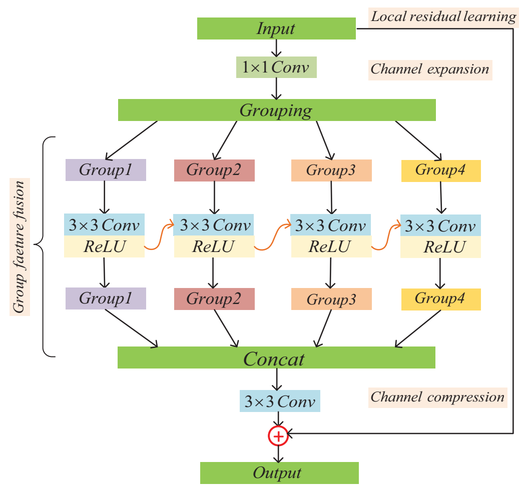

Fig. 3. The structure of the proposed group feature fusion residual block (GFFRB).

fewer residual blocks and channels. As shown in figure 2, the modules marked by red dotted line represent the removed components in GFFRN-L. By using these strategies, the number of parameters in GFFRN-L is reduced by 80% compared with GFFRN. More model details are shown in Section 4.2.

3.3 Group Feature Fusion Residual Block (GFFRB)

As mentioned above, current residual blocks have some drawbacks that cause them to be inefficient. Inspired by lightweight networks, we propose a more efficient group feature fusion residual block (GFFRB). As shown in figure 3, the GFFRB contains four parts: channel expansion layer, group feature fusion layer, channel compression layer and local residual learning. Next, we will describe the four parts in details.

Channel Expansion Layer Wider network allows more information to pass through [24]. However, simply increasing the width of the network will increase the computation complexity quadratically. To avoid this problem, we utilize a relatively small base channel number ($m = 64$). When the feature enters GFFRB, we first use an efficient convolutional layer $Conv(1, m \times e)$ to expand the channels from m to $m \times e$, where e denotes the expansion factor. Let F_{d-1} and F_d be the input and output of the d-th GFFRB respectively and both of them have m feature maps. This operation can be formulated as

$$F_E = f_{CE}(F_{d-1}), \tag{3}$$

where $f_{CE}(\cdot)$ denotes the convolutional layer, F_E denotes the feature maps after channel expansion.

Group Feature Fusion Layer To further control the computations, inspired by group convolution, we divide the wide feature into four groups along the channel axis. However, different from the group convolution used in previous methods [16], we design a novel group feature fusion layer, which gradually fuses the features of the current group with the output features of the last group. The group feature fusion operations have two advantages: **1)** It increases both the depth and the width of the network with fewer parameters. **2)** The skip connections in different groups are more sparse than the skip connections in dense block, which can not only improve the utilization of local features, but also control the number of parameters. In detail, we firstly divide the F_E into 4 feature groups (G_1, G_2, G_3, G_4) and each group has $(gc = m \times e/4)$ channels, where gc denotes the number of channels in each group. Then we use skip connections and convolution operations to fuse the features between different groups. Finally, all the fused feature maps are concatenated together. These operations are formulated as follows.

$$\begin{cases} [G_1, G_2, G_3, G_4] = Grouping(F_E), \\ G_{f1} = ReLU(f_{g1}(G_1)), \\ G_{f2} = ReLU(f_{g2}([G_2, G_{f1}])), \\ G_{f3} = ReLU(f_{g3}([G_3, G_{f2}])), \\ G_{f4} = ReLU(f_{g4}([G_4, G_{f3}])), \\ F_{fuse} = [G_{f1}, G_{f2}, G_{f3}, G_{f4}]. \end{cases} \tag{4}$$

$Grouping(\cdot)$ denotes the operation that averagely groups F_E into four groups in channel dimension. $f_{g1}(\cdot), f_{g2}(\cdot), f_{g3}(\cdot), f_{g4}(\cdot)$ denote the convolutional layers $Conv(3, gc)$ of the four groups. $G_{f1}, G_{f2}, G_{f3}, G_{f4}$ denote the fused feature maps of the four groups. $ReLU(\cdot)$ represents the ReLU activation function. F_{fuse} denotes the extracted features from the group feature fusion layer.

Channel Compression Layer After the group feature fusion layer, we use one convolutional layer $Conv(3, m)$ to fuse the features of the four groups. This operation can further improve the utilization of the local multilevel features and compress the number of channels at the same time. It can be formulated as

$$F_C = f_{Comp}(F_{fuse}), \tag{5}$$

where $f_{Comp}(\cdot)$ denotes the convolutional layer, F_C denotes the feature maps after channel compression.

Local Residual Learning This part we use the local residual learning to further improve the information flow. The final output of the d-th GFFRB can be obtained by

$$F_d = F_{d-1} + F_C. \tag{6}$$

4 Experimental Results

4.1 Datasets

In experiment, we apply DIV2K dataset [25] to train our models. This dataset is a newly-proposed high-quality image dataset, which contains 800 training images, 100 validation images and 100 test images. In the testing phase, we use five widely used benchmark datasets: Set5 [26], Set14 [27], BSDS100 [28], Urban100 [29] and Manga109 [30]. These datasets contain a variety of images, so they can fully validate our models.

4.2 Implementation and Training Details

Training details The training configurations of our models are similar to M-SRN [23]. We use the images of 1-800 and 801-810 in DIV2K dataset as our training set and validation set respectively. The training data is augmented by random scaling, rotation and flipping. In the training phase, we randomly select 16000 RGB input patches ($C = 3$) of size 48×48 from all the LR images in every epoch. The batch size is set to 16, thus every epoch has 1000 iterations of back-propagation. The model is trained 800 epochs. The learning rate begins with 1×10^{-4} and is halved every 200 epochs. Our model is optimized by Adam [31] by setting $\beta 1 = 0.9$, $\beta 2 = 0.999$ and $\epsilon = 10^{-8}$. The network is programmed by Pytorch and the experiment is performed on a PC with an i9-9900k CPU, 32GB RAM, and a RTX 2080Ti GPU.

Model details The proposed GFFRN and GFFRN-L consist of a series of GFFRBs. In the high-path of GFFRN, the number of GFFRB D is set to 12 and the expansion factor e is set to 4. In the low-path of GFFRN, D is set to 1, e is set to 2. In GFFRN-L, D is set to 6, e is set to 2. The base channel number m of the two models are all set to 64.

4.3 Comparisons with State-of-the-art Methods

In this section, we compare the proposed GFFRN and GFFRN-L with some other state-of-the-art SR methods including Bicubic [3], SRCNN [7], VDSR [11], DRRN [18], IDN [32], CARN [16], SRFBN-S [19], MSRN [23] and SRFBN [19]. In particular, GFFRN has the similar number of parameters to MSRN and GFFRN-L has the similar number of parameters to CARN. It should be noticed that GFFRN-L do not apply the multi-scale learning approach that used in CARN because of the two-path network structure. The widely used image evaluation methods, peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM), are used to evaluate the performance of the proposed methods. Furthermore, we also compare these models from other dimensions, including the number of parameters and computational complexity. Similar to CARN, we use the Multi-Adds to represent the computational complexity of

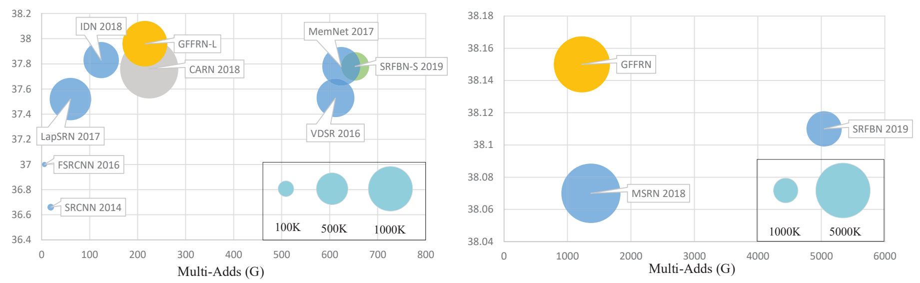

Fig. 4. Trade-off between accuracy vs. number of operations and parameters on Set5 $\times 2$ dataset. The x-axis and the y-axis denote the Multi-Adds and PSNR, and the area of the circle represents the number of parameters. The Multi-Adds are computed by assuming that the resolution of HR image is 720p.

the model and assume the HR image size to be 720p (1280×720) to calculate Multi-Adds. In figure 4, we compare our GFFRN and GFFRN-L against various state-of-the-art methods on the Set5 $\times 2$ dataset. As shown in the figure, when considering three aspects of speed, accuracy and the number of parameters, the proposed GFFRN and GFFRN-L achieve the best performance.

Table 1 lists the experiment results of PSNR and SSIM obtained by using different methods. Here we only compare models that have roughly similar number of parameters as ours[1]. We first to analyse the performance of GFFRN-L. Compared with CARN, the proposed GFFRN-L achieves comparable results with fewer parameters and computations. Compared with IDN, GFFRN-L has twice as many parameters, but the benefits are also huge. GFFRN-L outperforms IDN by a margin of 0.1-0.5 PSNR on different benchmark test sets. Compared with SRFBN-S, the proposed GFFRN-L has fewer computations but gets higher reconstruction accuracy. Especially when the scaling factor is $\times 4$, GFFRN-L obtains higher accuracy with only 5% computations of SRFBN-S. Secondly, we analyse the performance of GFFRN. Compared with MSRN, even with fewer parameters and computations, the proposed GFFRN outperforms it by a large margin on different benchmark test sets. SRFBN outperforms GFFRN, however, its benefits mainly come from the recursive structure, which introduces a huge number of computations. When the scaling factor is $\times 4$, the calculations of SRFBN are more than 20 times of GFFRN, hence the SRFBN is inefficient in terms of computational complexity.

Figure 8 presents some reconstructed images obtained by using these methods with different scaling factors. For image ”img067”, we observe that most of compared methods can not restore the complete line of the building. In contrast, our GFFRN restores a complete line, which is closer to the original HR image. For image ”Belmondo”, all the compared methods restore the words with noticeable artifacts and blurred edges. While, our GFFRN can recover clearer words. The same conclusions can be obtained by image ”img034” and image ”14802”. This

[1] Comparison of the larger models can be found in our supplementary material.

Table 1. Quantitative comparisons of state-of-the-art methods. Red and blue represent the best and the second best result respectively.

Scale	Model	Params	Multi-Adds	Set5	Set14	B100	Urban100	Manga109
2	Bicubic	—	—	33.69/0.9284	30.34/0.8675	29.57/0.8434	26.88/0.8438	30.82/0.9332
	SRCNN [7]	0.02M	19G	36.66/0.9542	32.42/0.9063	31.36/0.8879	29.50/0.8946	35.60/0.9663
	VDSR [11]	0.67M	612G	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140	37.22/0.9750
	DRRN [18]	0.30M	6797G	37.74/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188	37.60/0.9736
	IDN [32]	0.59M	124G	37.83/0.9600	33.30/0.9148	32.08/0.8985	31.27/0.9196	-/-
	CARN [16]	1.59M	224G	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.51/0.9312	-/-
	SRFBN-S [19]	0.37M	653G	37.78/0.9597	33.35/0.9156	32.00/0.8970	31.41/0.9207	38.06/0.9757
	GFFRN-L(ours)	0.94M	215G	37.96/0.9603	33.51/0.9169	32.13/0.8992	31.91/0.9263	38.38/0.9766
	MSRN [23]	5.93M	1370G	38.07/0.9608	33.68/0.9184	32.22/0.9002	32.32/0.9304	38.64/0.9771
	SRFBN [19]	2.14M	5044G	38.11/0.9609	33.82/0.9196	32.29/0.9010	32.62/0.9328	39.08/0.9779
	GFFRN(ours)	5.32M	1226G	38.15/0.9610	33.84/0.9202	32.29/0.9010	32.57/0.9326	38.97/0.9777
3	Bicubic	—	—	30.41/0.8655	27.64/0.7722	27.21/0.7344	24.46/0.7411	26.95/0.8556
	SRCNN [7]	0.02M	19G	32.75/0.9090	29.28/0.8209	28.41/0.7863	26.24/0.7989	30.48/0.9117
	VDSR [11]	0.67M	612G	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279	32.01/0.9340
	DRRN [18]	0.30M	6797G	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378	32.42/0.9359
	IDN [32]	0.59M	55G	34.11/0.9253	29.99/0.8354	28.95/0.8013	27.42/0.8359	-/-
	CARN [16]	1.59M	119G	34.29/0.9255	30.29/0.8407	29.06/0.8034	27.38/0.8404	-/-
	SRFBN-S [19]	0.49M	788G	34.20/0.9255	30.10/0.8372	28.96/0.8010	27.66/0.8415	33.02/0.9404
	GFFRN-L(ours)	0.96M	98G	34.27/0.9263	30.29/0.8409	29.07/0.8039	28.03/0.8493	33.31/0.9429
	MSRN [23]	6.11M	627G	34.48/0.9276	30.40/0.8436	29.13/0.8061	28.31/0.8560	33.56/0.9451
	SRFBN [19]	2.83M	6024G	34.70/0.9292	30.51/0.8461	29.24/0.8084	28.73/0.8641	34.18/0.9481
	GFFRN(ours)	5.34M	546G	34.57/0.9286	30.46/0.8449	29.20/0.8077	28.54/0.8605	33.89/0.9470
4	Bicubic	—	—	28.43/0.8022	26.10/0.6936	25.97/0.6517	23.14/0.6599	24.89/0.7866
	SRCNN [7]	0.02M	19G	30.48/0.8628	27.49/0.7503	26.90/0.7101	24.52/0.7221	27.58/0.8555
	VDSR [11]	0.67M	612G	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524	28.83/0.8870
	DRRN [18]	0.30M	6797G	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638	29.18/0.8914
	IDN [32]	0.59M	31G	31.82/0.8903	28.25/0.7730	27.41/0.7297	25.41/0.7632	-/-
	CARN [16]	1.59M	91G	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	-/-
	SRFBN-S [19]	0.63M	983G	31.98/0.8923	28.45/0.7779	27.44/0.7313	25.71/0.7719	29.91/0.9008
	GFFRN-L(ours)	0.98M	56G	32.03/0.8934	28.54/0.7803	27.54/0.7347	25.94/0.7815	30.23/0.9050
	MSRN [23]	6.08M	377G	32.25/0.8958	28.63/0.7833	27.61/0.7377	26.22/0.7905	30.57/0.9103
	SRFBN [19]	3.63M	7466G	32.47/0.8983	28.81/0.7868	27.72/0.7409	26.60/0.8015	31.15/0.9160
	GFFRN(ours)	5.36M	309G	32.37/0.8974	28.70/0.7853	27.66/0.7394	26.38/0.7962	30.81/0.9132

is mainly because of the powerful local feature extraction ability of the well-designed GFFRB. [2]

4.4 Discussion

The most important contribution of this paper is the newly proposed group feature fusion residual block (GFFRB). Based on GFFRB, we design two efficient SR models GFFRN and GFFRN-L. To demonstrate the effectiveness of GFFRB, we have done a series of experiments on the two models. In addition, we also discuss the influence of the group feature fusion layer.

[2] More compared images can be found in our supplementary material.

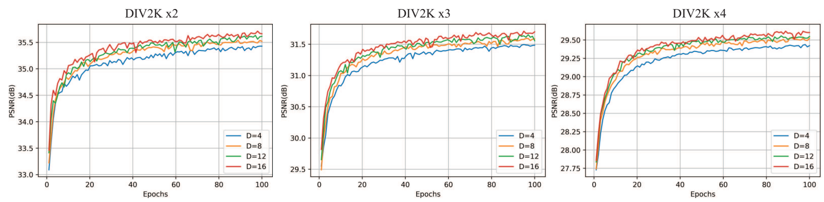

Fig. 5. Performance comparison of GFFRN with different number of GFFRBs.

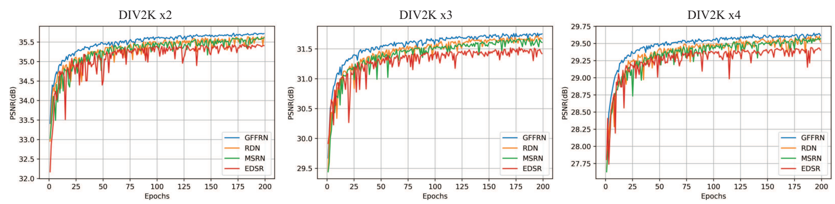

Fig. 6. Performance comparison of the models with different residual blocks.

Analysis of D As we all know, increasing network depth can improve reconstruction accuracy. In order to verify the impact of the number of GFFRB on network, we have done a series of experiments. Here we mainly discuss the different numbers of GFFRB in GFFRN ($D = 4, 8, 12, 16$ respectively). For a quick verification, all the models of this subsection are trained 100 epochs (1×10^5 iterations) in the same environment. As shown in Figure 5, the accuracy of GFFRN keeps raising with the increasing number of GFFRB. Although increasing the number of the GFFRB can improve the reconstruction accuracy, it also leads to a more complex network. By weighting the performance and the complexity of the network, we use 12 GFFRBs in the final model. Experiments shows that our final model outperforms MSRN [23] on all the benchmark test sets with fewer parameters and computations.

Efficiency Analysis of GFFRB The GFFRB is the key component to establish our network. In this subsection, we compare it with some other widely used residual blocks, including the residual blocks (RB) used in EDSR [12], the residual dense block (RDB) used in RDN [13] and the multi-scale residual block (MSRB) used in MSRN [23]. For a fair comparison, the number of RB in EDSR decreased from 32 to 5. The number of RDB in RDN decreased from 16 to 4. The number of MSRB in MSRN is set to 8, which is consistent with the original paper. All the models of this subsection are trained 200 epochs (2×10^5 iterations) in the same environment.

Figure 6 presents the convergence curves obtained by using these models under different scaling factors. From the figure, we can see that compared with other models, the proposed GFFRN achieves the best performance. We also compare these models by quantitative indicators, including the number of parameters,

Table 2. Quantitative comparisons of state-of-the-art methods. All the models are trained 200 epoches (2×10^5 iterations) in the same environment. Red represents the best result.

Scale	Model	Params	Multi-Adds	Set5	Set14	B100	Urban100	Manga109
2	EDSR [12]	8.87M	2050G	37.81/0.9599	33.44/0.9158	32.04/0.8978	31.49/0.9225	37.94/0.9757
	MSRN [23]	5.93M	1370G	37.98/0.9605	33.49/0.9168	32.13/0.8990	31.88/0.9261	38.38/0.9765
	RDN [13]	5.70M	1310G	37.92/0.9603	33.51/0.9173	32.13/0.8995	31.96/0.9273	38.17/0.9764
	GFFRN(ours)	5.32M	1226G	38.03/0.9607	33.59/0.9175	32.18/0.8997	32.09/0.9282	38.47/0.9768
3	EDSR [12]	11.8M	1210G	34.10/0.9250	30.14/0.8389	28.97/0.8020	27.69/0.8428	32.83/0.9400
	MSRN [23]	6.11M	627G	34.33/0.9266	30.32/0.8423	29.08/0.8049	28.08/0.8509	33.36/0.9435
	RDN [13]	5.88M	603G	34.40/0.9274	30.34/0.8423	29.09/0.8047	28.13/0.8520	33.44/0.9439
	GFFRN(ours)	5.34M	546G	34.43/0.9274	30.37/0.8432	29.12/0.8059	28.24/0.8547	33.52/0.9448
4	EDSR [12]	11.23M	1060G	31.88/0.8909	28.43/0.7781	27.46/0.7323	25.72/0.7742	29.86/0.9000
	MSRN [23]	6.08M	377G	32.04/0.8933	28.56/0.7809	27.56/0.7352	26.02/0.7831	30.29/0.9055
	RDN [13]	5.86M	364G	32.09/0.8940	28.58/0.7820	27.58/0.7363	26.06/0.7851	30.39/0.9073
	GFFRN(ours)	5.36M	309G	32.20/0.8950	28.61/0.7827	27.60/0.7372	26.19/0.7894	30.53/0.9093

the computational complexity and the performance on different benchmark test sets. As shown in table 2, compared with other state-of-the-art methods, the proposed GFFRN gets the highest accuracy on all the benchmark test sets with the fewest parameters and calculations of all the methods. All of these experiments fully demonstrate that the proposed GFFRB is more efficient than most current residual blocks.

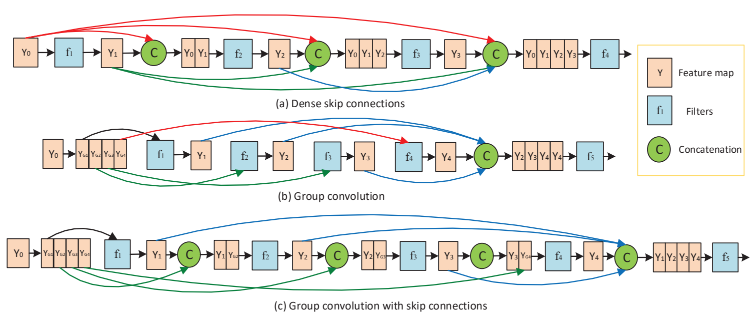

Fig. 7. Diagram the structural relations of Group convolution, Dense Skip Connections and Group feature fusion.

Ablation Studies on the group feature fusion layer The novelty of the proposed GFFRN is the skip connections among different groups. Here we will analyse the effectiveness of this structure. In Fig. 7, we present the structure of dense skip connection, the structure of group convolution and the structure of the group feature fusion layer. Notably, Fig. 7 (c) is the unfolded format of the group feature fusion layer in Fig. 3. From the figure, we can find that the structure of

Table 3. Quantitative comparisons of the models that use different modules. Red represent the best result.

Scale	Model	Params	Set5	Set14	B100	Urban100	Manga109
	Group Convolution	1.21M	38.00/0.9605	33.59/0.9177	32.14/0.8993	31.90/0.9264	38.46/0.9767
2	Dense Skip Connections	1.77M	38.02/0.9606	33.61/0.9180	32.20/0.9000	32.21/0.9293	38.59/0.9770
	Group Feature Fusion	1.49M	38.04/0.9606	33.64/0.9182	32.19/0.8999	32.19/0.9293	38.64/0.9771

group feature fusion layer is the combination of the group convolution and the dense skip connection.

Next we will do some experiments to compare the efficiency of the three structures. We use 10 GFFRBs to conduct experiments. We replace the group feature fusion layer with dense connection and group convolution. For a fair comparison, the number of dense connection layer is set to 4 and the number of the group in group convolutional layer is set to 4. In Table 3, we use PSNR/SSIM to measure the accuracy of the reconstruction result, and the number of the parameters to measure the storage efficiency. We can see that the model with group convolution has the least parameters, but gets the poorest performance. When we add skip connections on the group convolution, the performance improves effectively. Compared with dense skip connections, our group feature fusion layer has fewer parameters, but achieves a comparable performance. This fully demonstrates the effectiveness of the skip connections among different groups.

5 Conclusions and Future Works

In this paper, we propose a novel group feature fusion residual block (GFFRB), which combines the group convolution with skip connection to fully fuse abundant local features. Experiments show that the well-designed GFFRB outperforms most current residual blocks. Based on GFFRB, we propose a two-path group feature fusion residual network (GFFRN). Experiments show that the proposed GFFRN achieves higher efficiency compared with most state-of-the-art methods. We also design a lightweight group feature fusion residual network (GFFRN-L), which achieves the best performance among the models that have less than 1M parameters.

Future works can be mainly explored from the following two aspects: (1) In this paper, we specify that the number of groups is 4. Future works can discuss the number of groups g to further improve the efficiency of GFFRB. (2) It would be worthwhile to try to apply the well-designed GFFRB to other computer vision tasks, such as image denoising and deblurring.

6 Funding

This research was supported by the National Natural Science Foundation of China under Grant No. 61703278.

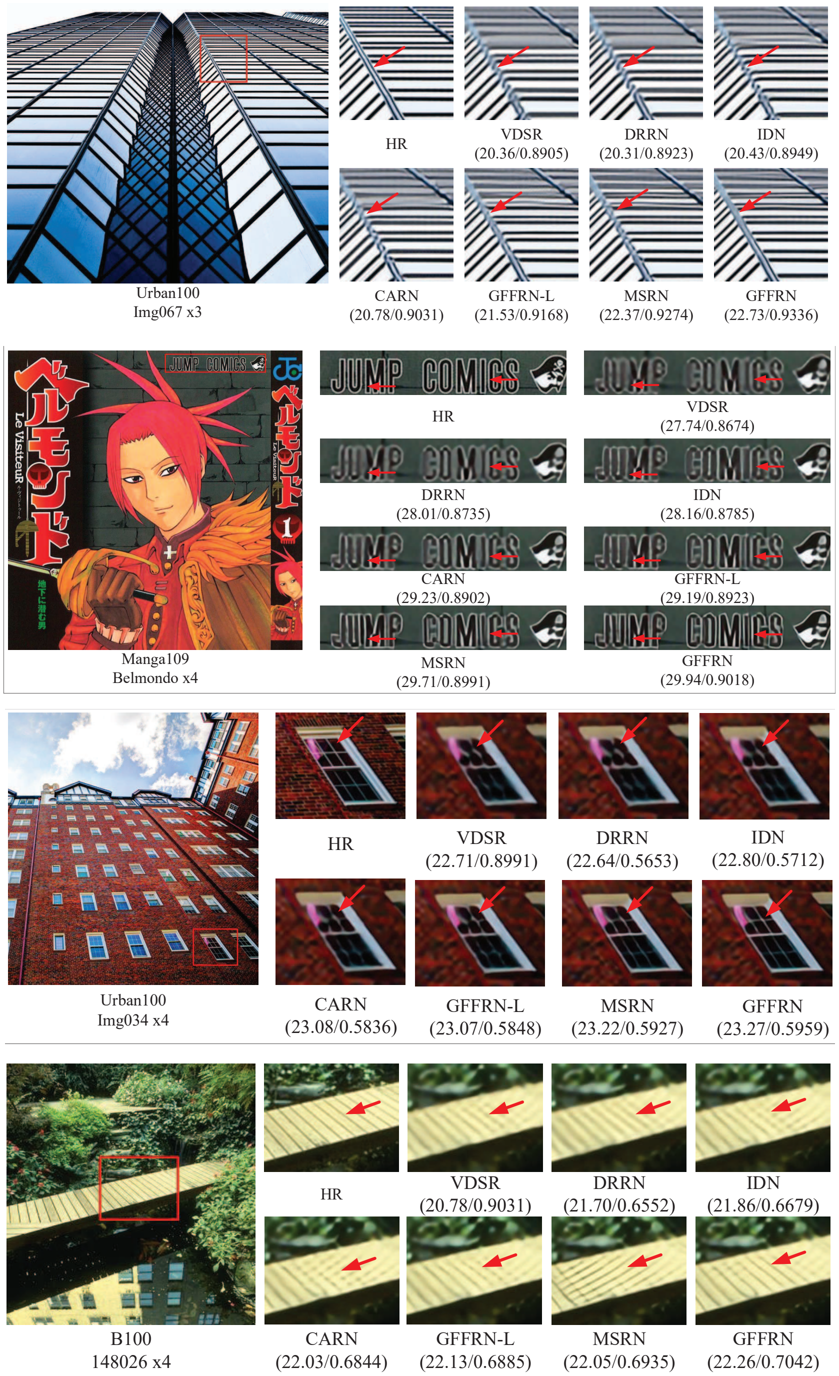

Fig. 8. Visual comparison of different methods on different images.

References

1. Bilgazyev, E., Efraty, B., Shah, S.K., Kakadiaris, I.A.: Improved face recognition using super-resolution (2011) 2011 International Joint Conference on Biometrics (IJCB).
2. Jian, M., Lam, K.M.: Simultaneous hallucination and recognition of low-resolution faces based on singular value decomposition. IEEE Transactions on Circuits and Systems for Video Technology **25** (2015) 1761–1772
3. Lei, Z., Xiaolin, W.: An edge-guided image interpolation algorithm via directional filtering and data fusion. IEEE Transactions on Image Processing **15** (2006) 2226–2238
4. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations (2012) International Conference on Curves and Surfaces.
5. Jianchao, Y., Zhaowen, W., Zhe, L., Scott, C., Thomas, H.: Coupled dictionary training for image super-resolution. IEEE Transactions on Image Processing **21** (2012) 3467–3478
6. Fang, B., Huang, Z., Yong, L., Yong, W.: v-support vector machine based on discriminant sparse neighborhood preserving embedding. Pattern Analysis and Applications **20** (2017) 1077–1089
7. Dong, C., C.C.L.K.H., Tang, X.: Learning a deep convolutional network for image super-resolution (2014) European Conference on Computer Vision (ECCV).
8. Dong, C., C.C.L., Tang, X.: Accelerating the super-resolution convolutional neural network (2016) Proceedings of the European Conference on Computer Vision (ECCV).
9. Shi, W., Caballero, J., Huszr, F., Totz, J., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. IEEE Conference on Computer Vision and Pattern Recognition (2016)
10. He K, Zhang X, R.S.e.a.: Deep residual learning for image recognition (2015) Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
11. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks (2016) 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
12. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution (2017) Proceedings of the IEEE Computer Vision and Pattern Recognition Workshops.
13. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. CoRR **abs/1802.08797** (2018)
14. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. European Conference on Computer Vision (ECCV) (2018)
15. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration (2017) 2017 IEEE International Conference on Computer Vision (ICCV).
16. Ahn, N., K.B., Sohn: Fast, accurate, and lightweight super-resolution with cascading residual network. European Conference on Computer Vision (ECCV) (2018)
17. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution (2016) 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
18. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network (2017) 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

19. Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., Wu, W.: Feedback network for image super-resolution. CoRR **abs/1903.09814** (2019)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Neural Information Processing Systems (2012) 1097–1105
21. Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications (2017)
22. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. arXiv: Computer Vision and Pattern Recognition (2016)
23. Li, J., Fang, F., Mei, K., Zhang, G.: Multi-scale residual network for image super-resolution (2018) 2018 European Conference on Computer Vision(ECCV).
24. Yu, J., Fan, Y., Yang, J., Xu, N., Wang, Z., Wang, X., Huang, T.S.: Wide activation for efficient and accurate image super-resolution. arXiv: Computer Vision and Pattern Recognition (2018)
25. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. (2017)
26. Bevilacqua, M., Roumy, A., Guillemot, C., Alberimorel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: 2012 British Machine Vision Conference. (2012) 1–10
27. Li, T., Chang, K., Mo, C., Zhang, X., Qin, T.: Single image super resolution using local and non-local priors. In: Advances in Multimedia Information Processing – PCM 2018, Cham, Springer International Publishing (2018) 264–273
28. Arbelaez, P., Maire, M., Fowlkes, C.C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **33** (2011) 898–916
29. Huang, J., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: IEEE Conference on Computer Vision and Pattern Recognition. (2015) 5197–5206
30. Matsui, Y., Ito, K., Aramaki, Y., Yamasaki, T., Aizawa, K.: Sketch-based manga retrieval using manga109 dataset. Multimedia Tools and Applications **76** (2017) 21811–21838
31. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. Computer Science (2014)
32. Hui, Z., Wang, X., Gao, X.: Fast and accurate single image super-resolution via information distillation network. IEEE Conference on Computer Vision and Pattern Recognition (2018)