Lightweight Parallel Feedback Network for Image Super-Resolution

Article *in* Neural Processing Letters · August 2022

DOI: 10.1007/s11063-022-11007-0

4 authors, including:

Changjun Liu
Sichuan University
196 PUBLICATIONS **2,503** CITATIONS
SEE PROFILE

Xiaomin Yang
Sichuan University
189 PUBLICATIONS **3,389** CITATIONS
SEE PROFILE

Neural Processing Letters
https://doi.org/10.1007/s11063-022-11007-0

Lightweight Parallel Feedback Network for Image Super-Resolution

Beibei Wang[1] · Changjun Liu[1] · Binyu Yan[1] · Xiaomin Yang[1]

Accepted: 9 August 2022

Abstract
Since deep learning was introduced into super-resolution (SR), SR has achieved remarkable performance improvements. Since high-level features are more informative for reconstruction, most of SR methods have a lage number of parameters, which restrict their application in resource-constrained devices. Feedback mechanism makes it possible to get informative high-level features with few parameters, for it can feed high-level features back to refine low-level ones, which is very suitable for lightweight networks. However, most feedback networks work in a single feedback manner, which refined low-level features just once in each iteration or each unit. In this paper, we propose a lightweight parallel feedback network for image super-resolution (LPFN), which enhances the refinement ability of the feedback network. In our method, all the feedback blocks feed back their outputs to previous layers in a parallel feedback manner. Based on parallel feedback and residual learning, a local-mirror architecture is proposed. Then, we propose a dispersion-aware attention residual block (DARB) as the basic block in feedback block, which calculates the dispersion of pixels along channel and spatial dimensions. We use ensemble method to reconstruct SR image. Finally, we propose a global feedback, which feeds back the degradation results of SR to primal LR image, supervising the learning of LR-HR mapping function. Further experimental results demonstrate that LPFN has an outstanding performance while taking up few computing resources.

Keywords Super-resolution · Global feedback · Residual learning · Parallel feedback · Lightweight

✉ Binyu Yan
Yanby@scu.edu.cn

Beibei Wang
690983790@qq.com

Changjun Liu
cjliu@scu.edu.cn

Xiaomin Yang
arielyang@scu.edu.cn

[1] College of Electronics and Information Engineering, Sichuan University, Chengdu, Sichuan, China

Published online: 20 August 2022

1 Introduction

Super-resolution (SR) technology has shown broad application prospects in image restoration and image enhancement, especially single image super-resolution (SISR), which has been paid more and more attention in recent years. SISR aims to recover image details from the limited information contained in low-resolution (LR) images, which is a challenging problem, for an infinite number of high-resolution (HR) images can degrade to the same LR image. To slove this problem, many classical SR methods were proposed [1–5, 5–12].

SRCNN [1] introduced deep learning into SR, and first applied Convolutional Neural Network (CNN) architecture for image SR. Then deconvolutional layer was introduced into SR by FSRCNN [2], which upsampled LR feature maps at the last layer, and greatly reduced the amount of computation and space complexity. Shi et al. introduced subpixel convolutional layer into SR in ESPCN [3], which rearranged the pixels of LR features with channel expansion to generate HR features, and got rid of the redundant of deconvolution layer. LapSRN [11] proposed charbonnier penalty function as the loss function and upsampled the LR feature maps progressively. In VDSR [4], interpolated LR images with multiple scales served as inputs, and were added to the end layer directly to realize residual learning. Then DRCN [5] introduced recursive architecture into SR and deepened the network, which reduced the parameters of deep networks.

Deep networks improved SR performance significantly, for high-level features are more informative for reconstruction. However, deep networks are difficult to train and easy to cause the problems of gradient vanishing/exploding. Residual learning was introduced into deep-learning-based SR methods by VDSR [4] and worked well in ResNet [6], which can solve the degradation problem of deep networks. Then SRResNet [7] was proposed, which contained 16 residual units. Inspired by the methods mentioned above, DRRN [8] was proposed, which combined local residual learning, global residual learning and multi-weight recursive learning. SRDenseNet [13] sent all the feature maps to latter layers with concatenation in dense block. EDSR [9] achieved a better performance with expanded model size by removing the usage of Batch Nomalization (BN) in SRResNet [7], for it is not suitable for SR tasks.

Most of the SR methods worked in a feedforward manner, in which the low-level feature maps are sent to latter layers directly or by skip connection. Feedback mechanism was proposed earlier in the year, and was introduced into different computer vision methods [14–17] recently. Then, feedback mechanism was introduced into SR by Hairs et al. in DBPN [18], which calculated the errors between up- and down-sampling and fed back the errors for self-correction. Li et al. proposed SRFBN [19], the feedback mechanism of which worked similar to Recurrent Neural Network (RNN) with constraints. Feedback mechanism can feed high-level features back to refine low-level ones, and further achieve powerful high-level representations.

To achieve a good performance, most of SR methods have a large number of parameters, which restrict their application in resource-constrained devices. Therefore, lightweight networks attract academic attention for their wide application prospect. Although feedback mechanism has been introduced into SR, which has not been widely used in lightweight networks. Since feedback mechanism can obtain powerful high-level representations with few parameters, which is very suitable for lightweight networks. In this paper, we propose a lightweight parallel feedback network for image super-resolution (LPFN). The structure of LPFN is shown in Fig. 1, which combines residual learning and feedback learning. The feedback learning refines low-level features by high-level ones in a synchronous parallel manner. The residual learning sends low-level features to latter layers by skip connection for difference

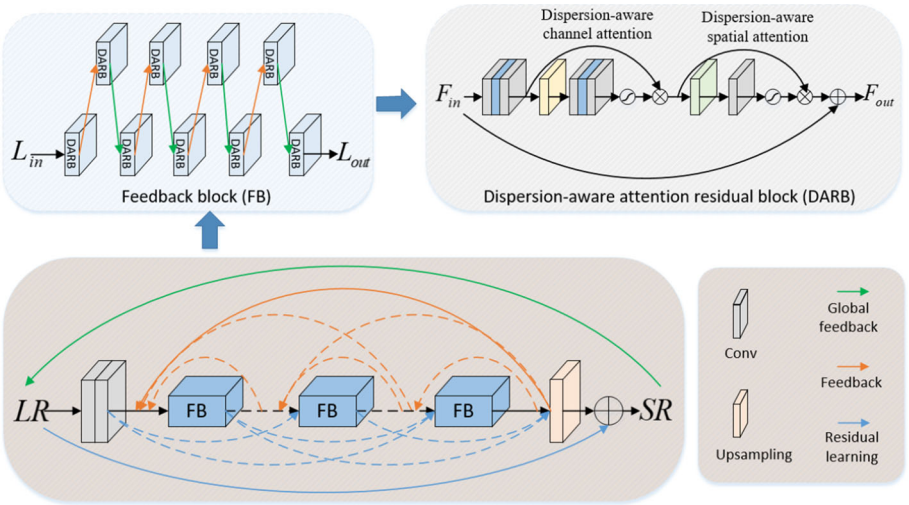
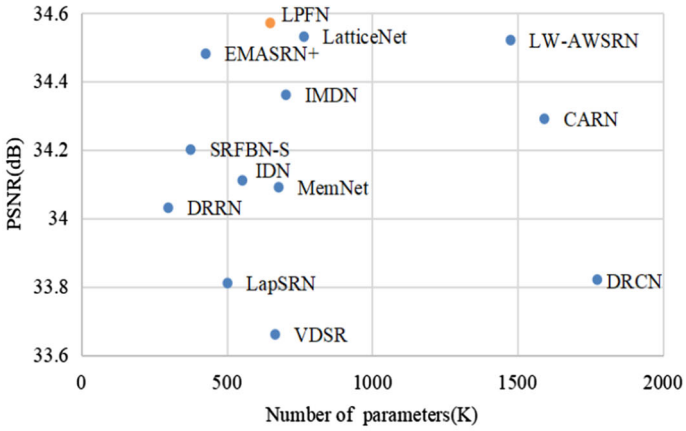


Fig. 1 The structure of LPFN. The blue lines and orange lines are mirror images except one solid blue line and one solid orange line. (Colour figure online)

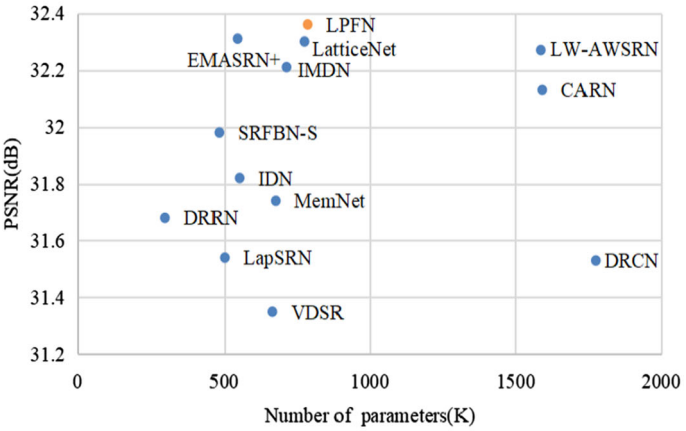
learning. Their combination forms a local-mirror architecture. At the same time, we believe that the features of previous iterations are too shallow to obtain valuable reconstruction results, so we use feature ensemble method to reconstruct the final SR image. The feature ensemble method ties the loss to all iterations, so that the hidden states carry a notion of high-level features. The feature ensemble method is proved better than multi-reconstruction method in experimental results. Our LPFN performs better than other state-of-the-art lightweight networks, as shown in Fig. 2.

The contributions of LPFN are as follows:

- To refine low-level feature maps sufficiently, we propose a parallel feedback scheme. All the outputs of feedback blocks are fed back to previous layers to refine low-level features in a synchronous parallel manner.
- To better learn the relationship between high-level features and low-level ones, we propose a local-mirror architecture. The feedback learning refines low-level features by high-level ones, and the residual learning sends low-level features to latter layers by skip connection for difference learning. The combination of them helps to achieve a better high-level representation.
- To enhance the discriminate ability of feature maps, we propose a dispersion-aware attention residual block (DARB) as the basic block in feedback blocks. The pixels with high dispersion must contain important information in some locations. Therefore, DARB integrates standard deviation into channel and spatial attention modules, which learns the dispersion of pixels along channel dimension and spatial dimensions.
- To better guide the learning of LR-HR mapping function, we propose a global feedback. We calculate feedback-regression loss by comparing the degradation result of SR and primal LR image, which is used to supervise the training of our network. Our global feedback can be added to other SR methods as a module, and can be used for all scale factors in one step, not necessarily multiple scales. Our global feedback can significantly improve the performance of the network but introduces very few parameters.



(a) scaling factor of x3



(b) scaling factor of x4

Fig. 2 PSNR vs. number of parameters on Set5 dataset. Orange points represent our method. (Colour figure online)

2 Related work

2.1 Lightweight Networks

Many literatures [20, 21] have shown that deeper networks have stronger representation power, but deep networks have a large number of parameters and are hard to train, which have high requirements on hardware resources. Recently, lightweight networks are getting more and more attention, for they can be applied to resource-limited devices. All lightweight networks strive for better performance with fewer parameters and little computational complexity. Dong et al. introduced deep learning into image SR in SRCNN [1]. Then they introduced deconvolution layer to reduce calculations in FSRCNN [2]. VDSR [4] introduced residual learning to avoid gradient explosion of deep networks. Then DRCN [5] introduced recursive convolution into image SR, so that the networks can be deep with few parameters.

IMDN [22] proposed information multi-distillation to achieve outstanding performance with little computational complexity. Recently, LatticeNet [23] achieves an outstanding performance by the combination of residual blocks. LW-AWSRN [24] has little computational complexity but outstanding performance.

Most of lightweight methods are feedforward. Since high-level features are more informative for reconstruction, we believe that, feedback mechanism is very suitable for lightweight networks. We propose parallel feedback mechanism to further enhance the refinement ability of the feedback network. Our LPFN achieves a better performance than the lightweight networks mentioned above.

2.2 Attention Mechanism

Networks with attention-based model can pay more attention to important features. The team of google mind [25] used attention mechanism in RNN for image classification tasks. Non-Local module [26] learned the relationship between pixels within a certain distance by capturing long-range dependencies. Squeeze-and-excitation (SE) module [27] was proposed to improve the accuracy of image recognition, which learned the relationship between channels to enhance the important features. Then, residual channel attention blocks (RCABs [28]) was proposed to improve the performance of SR, which integrated channel attention into residual blocks. In CBAM [29], the channel attention and spatial attention are used together to enhance the details in feature maps adaptively. Recently, the contrast-aware channel attention (CCA) was proposed in IMDN [22], which introduced standard deviation into channel attention to improve the representation ability of attention module. CVCnet [30] proposed cascaded spatial perception module to redistribute pixels in feature maps according to their weights.

Inspired by the above attention models, we propose dispersion-aware attention residual block (DARB) as the basic block in feedback blocks. We believe that if the dispersion of pixels is high along channel or spatial dimensions, the pixels must contain important information in some locations. Therefore, we integrates standard deviation into channel and spatial attention modules.

2.3 Feedback Mechanism

Most SR methods worked in a feedforward manner, in which the low-level features are sent to latter layers directly or by skip connection. Feedback mechanism makes it possible that, the previous layers can get useful information from high-level features. Feedback mechanism was proposed earlier in the year, and has been introduced into different computer vision methods [14–17] recently.

Feedback mechanism can feed back high-level information to refine low-level ones, which was introduced into SR by Hairs et al. in DBPN [18]. DBPN [18] proposed up- and down-projection units, which realized self-correcting procedure in each unit based on feedback mechanism. Then DSRN [31] was proposed, which used delayed feedback to exchange recurrent signals between LR and HR states. Li et al. proposed SRFBN [19], in which the feedback mechanism worked similar to RNN with constraints. With the help of feedback mechanism, EMASRN+ [32] achieved an outstanding performance with very few parameters.

SRFBN [19] is a single feedback network worked in a serial manner, inspired by which, we propose a parallel feedback network. In our LPFN, multiple feedback blocks work in a synchronous parallel manner, which has a better performance than the serial single feedback

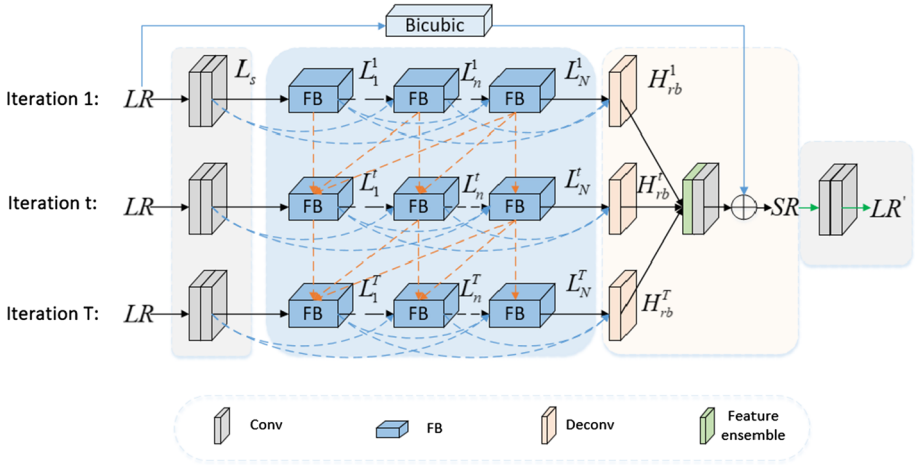


Fig. 3 The unfolded LPFN. Since LPFN is a feedback network, which can be unfolded in time. Orange arrows represent the parallel feedback procedure. Blue arrows represent the residual learning procedure. Then, we use feature ensemble method to reconstruct SR image. The final green arrows represent global feedback procedure. (Colour figure online)

manner used in SRFBN [19]. Furtherly, we believe the features of previous iterations are too shallow to obtain valuable reconstruction results, so we use feature ensemble method to reconstruct SR image, which ties the loss to all iterations. Our ensemble method has a better performance than the multi-reconstruction method used in SRFBN [19].

3 Our Method

In this section, the network architecture of LPFN is described at first. Next, the feedback block (FB) is described. Then the dispersion-aware attention residual block (DARB) as the basic block of FB is described. At last, the loss function with global feedback is described.

3.1 Network Structure

Since LPFN is a parallel feedback network, which can be unfolded as Fig. 3. Orange arrows represent the parallel feedback procedure, blue arrows represent the residual learning procedure. The ensemble of upsampling results from all iterations are used to calculate the final SR result.

LPFN contains four parts: shallow feature extraction part, parallel feedback part, reconstruction part and the global feedback part. In the first part, the stacked convolutional layers extract shallow features, which are passed to parallel feedback part. In the second part, the outputs of FBs are fed back to previous layers in a synchronous parallel manner. In reconstruction part, the outputs of FBs are concatenated and upsampled to generate HR features. Then we use feature ensemble on the HR features of all iterations. Because of the bicubic residual learning, the ensemble of HR features from all iterations was used to reconstuct SR image by adding bicubic upsampling results. Finally, in global feedback part, the SR image degrades to LR image to calculate feedback-regression loss with primal LR image.

We define L_s is the output of shallow feature extraction part, which can be obtained by:

$$L_s = f_c(LR), \tag{1}$$

where f_c consists of conv(3,128) and conv(128,32) to extract shallow LR features.

In parallel feedback part, we set the number of FB n from 1 to N, and the number of iteration t from 1 to T. Therefore, the output of the n-th FB in the t-th iteration is defined as L_n^t. Because of the local-mirror architecture, the input of FBs is different. In the first iteration, the input of the first FB is only shallow features L_s, and the input of the other FBs is shallow features L_s and all the outputs of previous FBs $L_1^1 \ldots . L_{n-1}^1$ because of the residual learning, which is shown in Eqn (2). In the t-th iteration, the input of the first FB is shallow features L_s and all the feedback from last iteration $L_1^{t-1} \ldots L_N^{t-1}$. The input of the other FBs is shallow features L_s, all the outputs of previous FBs from current iteration $L_1^t \ldots . L_{n-1}^t$ and the feedback of the following FBs from last iteration $L_n^{t-1} \ldots L_N^{t-1}$, which is shown in Eqn (3).

$$L_n^1 = \left\{ \begin{array}{ll} f_{FB}(L_s) & n = 1 \\ f_{FB}([L_s, L_1^1 \ldots . L_{n-1}^1]) & n \geq 2 \end{array} \right\}, \tag{2}$$

$$L_n^t = \left\{ \begin{array}{ll} f_{FB}([L_s, L_1^{t-1} \ldots L_N^{t-1}]) & n = 1 \\ f_{FB}([L_s, L_1^t \ldots . L_{n-1}^t, L_n^{t-1} \ldots L_N^{t-1}]) & n \geq 2 \end{array} \right\}, \tag{3}$$

where f_{FB} is the operations of feedback block. [] is the concatenation operation.

In reconstruction part, we concat the outputs of FBs and use deconvolutional layer to upsample them. We define the result after deconvolutional upsampling in the t-th iteration as follows:

$$H_{rb}^t = f_{up}([L_1^t, L_2^t \ldots . L_N^t]) \quad . \tag{4}$$

Then the ensemble of upsampling results from all iterations are used to reconstuct SR image by adding bicubic upsampling result, so the SR image can be obtained by:

$$SR = f_{cm}([H_{rb}^1, H_{rb}^2, \ldots, H_{rb}^T]) + f_{BC}(LR), \tag{5}$$

where f_{cm} is the convolutional layer used to compress feature channels, and f_{BC} is the bicubic upsampling operation.

In global feedback part, we generate LR' by downsampling operator f_{down}, which consists of a downsampling convolutional layer and a channel-transform convolutional layer. LR' is compared with the primal LR to supervise the training of LPFN.

$$LR' = f_{down}(SR), \tag{6}$$

3.2 Feedback Block (FB)

We use iterative projection in our feedback block, as shown in Fig. 4. The first line is HR feature flow, and the second line is LR feature flow. The two flows are learned iteratively by upsampling and downsampling operations. The LR features are projected to HR features, which are learned and projected back to LR flow to learn the relationship between LR and HR features. The iterative up- and down-sampling manner is helpful in improving the reconstruction performance. To further improve the performance of our FB, we use dispersion-aware attention residual block (DARB) as each basic block, which will be covered in detail in Sect. 3.3.

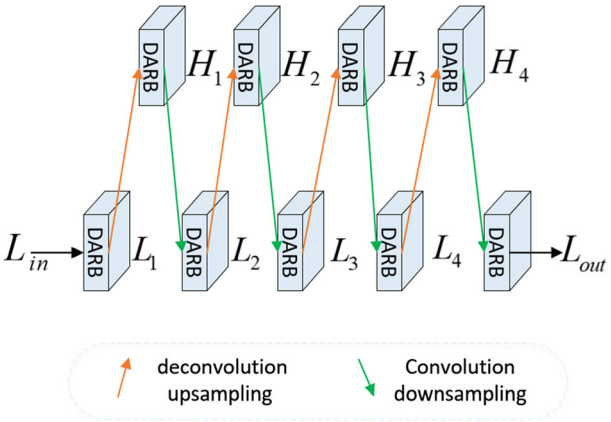


Fig. 4 Feedback block (FB) in LPFN, which learns dispersion-aware attention residual blocks (DARB) by iterative up-/down-projection

Since our LPFN has N FBs and can be unfolded to T iterations, the output of the n-th FB in the t-th iteration is defined as L_n^t. Because of the feedback learning and residual learning, the input of FBs is different, which is described in detail in sect. 3.1. In conclusion, the input of FBs can be obtained by:

$$L_{in} = \begin{cases} L_s & n = 1, t = 1 \\ f_{cm}([L_s, L_1^1 \ldots . L_{n-1}^1]) & n \geq 2, t = 1 \\ f_{cm}([L_s, L_1^{t-1} \ldots L_N^{t-1}]) & n = 1, t \geq 2 \\ f_{cm}([L_s, L_1^t \ldots . L_{n-1}^t, L_n^{t-1} \ldots L_N^{t-1}]) & n \geq 2, t \geq 2 \end{cases}, \tag{7}$$

In FB, we set the number of projection groups g=4, as shown in Fig. 4. The HR features in HR flow are defined as H_1, H_2, H_3, H_4, respectively, and the LR features in LR flow are defined as L_{in}, L_1, L_2, L_3, L_4, L_{out}, respectively. The process of FB is as follows:

$$L_g = \begin{cases} f_{DARB}(L_{in}) & g = 1 \\ f_{DARB}(f_{down}(H_{g-1})) & 2 \leq g \leq 4 \end{cases}, \tag{8}$$

$$H_g = f_{DARB}(f_{up}(L_g)) \quad 1 \leq g \leq 4, \tag{9}$$

$$L_{out} = f_{DARB}(f_{down}(H_4)), \tag{10}$$

where f_{DARB} is the operations of the basic block DARB in LR and HR feature flows. f_{up} and f_{down} are the deconvolutional upsampling operation and convolutional downsampling operation, respectively.

3.3 Dispersion-aware Attention Residual Block (DARB)

To improve the performance of FB, we propose a dispersion-aware attention residual block as each basic block, as shown in Fig. 5. We integrate dispersion-aware channel attention and dispersion-aware spatial attention into residual blocks. Dispersion-aware channel attention pays more attention to important channels, and dispersion-aware spatial attention pays more attention to important pixels. With the help of attention modules, the detail representation ability of feature maps are enhanced.

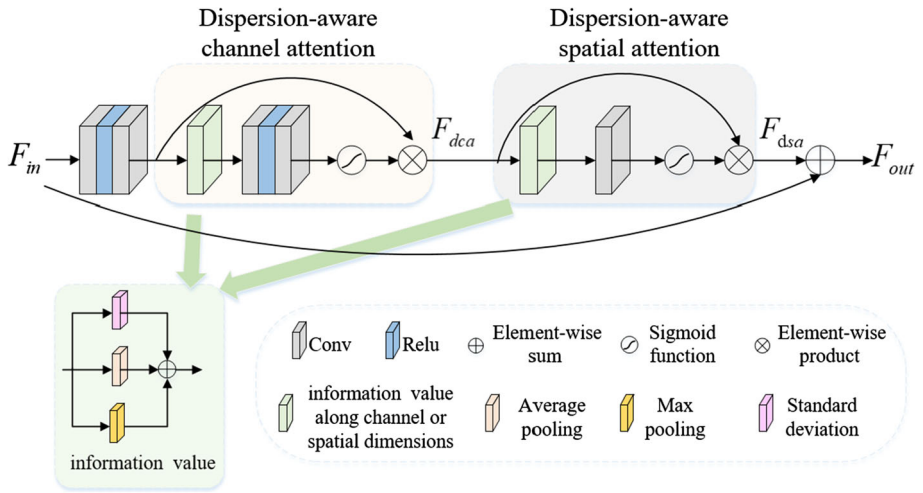


Fig. 5 Dispersion-aware attention residual block (DARB), which integrates dispersion-aware channle attention and dispersion-aware spatial attention into residual block. The dispersion-aware attention is calculated by standard deviation, average pooling and maximum pooling

Dispersion-aware channel attention We use standard deviation, average pooling and maximum pooling together to generate three different descriptions of channel context. Then we concat them to enhance the channel discriminative ability, as shown in Fig. 5. Average pooling can enhance discriminative ability according to the amount of information contained in channels. Max-pooling captures important information about features of distinctive object. Both of them are used together in CBAM [29], but standard deviation has not been used together with them. Standard deviation calculates the dispersion of pixels in feature maps, which represents the information about structures, textures, and edges. Therefore, we propose a dispersion-aware channel attention, which fuses three different descriptions of channel context and can improve the discriminate ability of feature maps greatly. We define the channel number of feature maps is c, which is from 1 to C. The information value of each channel can be calculated by:

$$V_c = \sqrt{\frac{1}{HW}\sum_{(i,j)\in x_c}(x_c^{i,j} - \frac{1}{HW}\sum_{(i,j)\in x_c} x_c^{i,j})^2} + \frac{1}{HW}\sum_{(i,j)\in x_c} x_c^{i,j} + \underset{(i,j)\in x_c}{MAX}(x_c^{i,j}). \tag{11}$$

Then the channel values in 1D channel attention map $R^{C\times 1\times 1}$ are learned by a multilayer perceptron with a hidden layer, and is normalized by the sigmoid function. At last, it multiples with the input F_{in} by element-wise multiplication along the spatial dimension. The output of dispersion-aware channel attention is as follows:

$$F_{dca} = F_{in} * \sigma f_{mlp}(V_c). \tag{12}$$

Dispersion-aware spatial attention We use standard deviation, average pooling and maximum pooling together to generate three different descriptions of pixels along spatial dimension, as shown in Fig. 5. Average pooling and maximum pooling indicate average

and maximum information of pixels at the same spatial location of all channels, which are used together in CBAM [29]. We believe that if the dispersion of pixels at the same spatial location of all channels is high, the pixels must contain important information in some channels. Therefore, pixels with high dispersion also should be paid more attention. Standard deviation value indicates the pixel-dispersion. Therefore, we propose dispersion-aware spatial attention, which fuses three different description of pixel context. The information value of pixels at the same spatial location of all channels can be calculated by:

$$V_{i,j} = \sqrt{\frac{1}{C}\sum_{c=1}^{C}(x_c^{i,j} - \frac{1}{C}\sum_{c=1}^{C}x_c^{i,j})^2} + \frac{1}{C}\sum_{c=1}^{C}x_c^{i,j} + \underset{1\le c\le C}{MAX}(x_c^{i,j}). \tag{13}$$

Then the pixel values in 2D spatial attention map $R^{1\times H\times W}$ are learned by a convolution operation with 7x7 sized kernel, for a large receptive field helps to decide important spatial regions. Then the output is normalized by the sigmoid function. At last, it multiples with the input F_{dca} by element-wise multiplication along the channel dimension. The output of dispersion-aware spatial attention is as follows:

$$F_{dsa} = F_{dca} * \sigma(f^{7\times 7}(V_{i,j})), \tag{14}$$

Finally, since DARB is a residual block (see Fig. 5), F_{out}, as the output of DARB, can be obtained by:

$$F_{out} = F_{dsa} + F_{in}. \tag{15}$$

3.4 Loss Function with Global Feedback

We propose a global feedback in our network, which feeds back the degradation result of SR to LR image to guide the learning of LR-HR mapping function. Therefore, the loss of our LPFN contains two parts: the primal regression loss calculated by HR and SR, the feedback-regression loss calculated by LR and LR'. The loss can be obtained by:

$$Loss = L_1(SR, HR) + \theta L_1(LR', LR), \tag{16}$$

where θ controls the weight of feedback-regression loss. L_1 represents the L1 loss function.

Our global feedback is used at the end of the network, so it can be added to other SR methods as a module. Since our global feedback uses convolutional downsampling to generate LR' in one step, it can be used for all scale factors, not necessarily multiple scales. Since our global feedback consists only of a downsampling convolutional layer and a channel-transform convolutional layer, which introduces very few parameters.

4 Experimental Results

4.1 Experimental Details

Datasets The DIV2k datasets are used to train our LPFN. We expand the datasets to 8000 by rotation and cropping augmentation to get HR images. Then we generate LR images by downsampling the HR images under bicubic downsampling. At last, we perform test on five benchmark datasets: Set5, Set14, BSD100, Urban100 and Manga109 datasets.

Table 1 Comparisons of different weights of the feedback-regression loss on LPFN

Weight	Scale	Params	Set5	Set14	BSD100	Urban100	Manga109
			PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
$\theta = 0$	$\times 3$	649k	34.53/0.9280	30.44/0.8441	29.10/0.8054	28.28/0.8558	33.68/0.9452
$\theta = 0.01$			34.54/0.9282	30.43/0.8443	29.12/0.8060	28.37/0.8567	33.70/0.9456
$\theta = 0.1$			34.57/0.9285	30.50/0.8453	29.15/0.8065	28.42/0.8578	33.92/0.9465
$\theta = 0.2$			34.55/0.9284	30.47/0.8450	29.14/0.8065	28.41/0.8575	33.89/0.9463
$\theta = 1$			34.45/0.9274	30.33/0.8419	29.10/0.8054	28.18/0.8531	33.62/0.9444

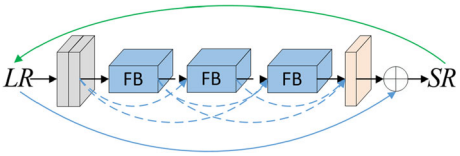


Fig. 6 LPFN-RNN. LBFN degenerates into a feedforward RNN network after the ablation of feedback

Implementation details We use adam optimizer and train our network with L1 loss. We set the initial learning rate to 0.0005, and halve it every 200 epoches for a total of 1000 epoches. To be lightweighted, we set the number of feedback blocks N=2, and the two feedback blocks shared the same parameters. At the same time, we set the number of iterations T=2, and the base filter numble is set to 32. All the experiments are performed on GPU with PyTorch framework.

4.2 Effect of Global Feedback

In this paper, we propose a global feedback to guide the learning of LR-HR mapping function. Therefore, our loss function contains two parts, as shown in Eqn.(16), which uses θ to control the weight of feedbak-regression loss. We change the value of θ from 0 to 1 to get the best trade-off. When $\theta = 0$, the results represent the ablation study of global feedback. From the results shown in Table 1, we can find that, the performance of our method gets better with θ increasing from 0 to 0.1, and then the performance gets worse with θ increasing from 0.1 to 1. Our network has a best performance when the weight of feedback-regression loss is set to 0.1, which demonstrates that, the loss function can better guide the learning of LR-HR mapping function with the help of global feedback. Therefore, we set $\theta = 0.1$ in Eqn.(16) to train our LPFN.

4.3 Ablation Study of Feedback

To prove the effectiveness of feedback architecture, we do ablation experiment of feedback. The ablation study of feedback is a feedforward network similar to RNN, as shown in Fig. 6, which is named LPFN-RNN. Since FB is applied twice in each iteration and iterations T=2, to make a fair comparison, FB is applied 4 times as a recursive block. The comparison results are shown in Table 2. From the comparison results, we can find that, LPFN has a better performance than LPFN-RNN with roughly the same number of parameters. Therefore, the introduction of feedback mechanism improves the performance of SR networks, which

Table 2 Comparisons of our feedback and feedforward on LPFN

architecture	Scale	Params	Set5	Set14	BSD100	Urban100	Manga109
			PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
LPFN	×3	649k	34.57/0.9285	30.50/0.8453	29.15/0.8065	28.42/0.8578	33.92/0.9465
LPFN-RNN		641k	34.52/0.9279	30.44/0.8447	29.13/0.8060	28.38/0.8570	33.79/0.9458

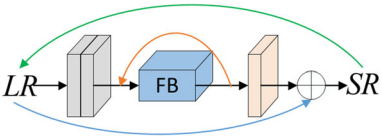


Fig. 7 LPFN-Serial. LBFN degenerates into a serail feedback network after the ablation of parallel feedback

Table 3 Comparison of our parallel feedback and serial feedback on LPFN

feedback manners	Scale	Params	Set5	Set14	BSD100	Urban100	Manga109
			PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
LPFN	×3	649k	34.57/0.9285	30.50/0.8453	29.15/0.8065	28.42/0.8578	33.92/0.9465
LPFN-Serial		640k	34.54/0.9281	30.45/0.8445	29.13/0.8062	28.34/0.8565	33.77/0.9457

demonstrates that feedback mechanism can generate powerful high-level representations without adding new parameters.

4.4 Ablation Study of Parallel Feedback

To prove the effectiveness of the parallel feedback we designed, we compare it with the serial feedback architecture proposed in SRFBN [19]. In our LPFN, we set T=2, and two parallel feedback blocks. To make a fair comparison, we set T=4 in the serial feedback architecture with one feedback block, as shown in Fig. 7, which is named LPFN-Serial. From the comparison results shown in Table 3, we can find that, the parallel feedback we designed has a better performance than the serial feedback architecture proposed in SRFBN [19]. Feedback architecture improved the performance of feedforward architecture (proved in Sect. 4.3), and our parallel feedback furtherly improved the performance of existing serial feedback architecture by refining low-level feature maps more sufficiently.

4.5 Improvement of DARB

In FB, we propose a dispersion-aware attention residual block (DARB) as the basic block, which integrates dispersion-aware channel attention and dispersion-aware spatial attention into residual blocks. To prove the effectiveness of DARB, we compare our DARB with CBAM [29] and IMDN [22]. We use the attention model CBAM [29] on our method, named LPFN-CBAM, which consisted of channel attetnion and spatial attention calculated by both average-pooling and max-pooling. We use contrast-aware channel attention (CCA) proposed in IMDN [22] on our method, named LPFN-IMDN, which is a channel attention model calculated by average-pool and standard deviation. The structure of attention modules mentioned above are shown in Fig. 8, and the comparison results are shown in Table 4. IMDN [22] only has

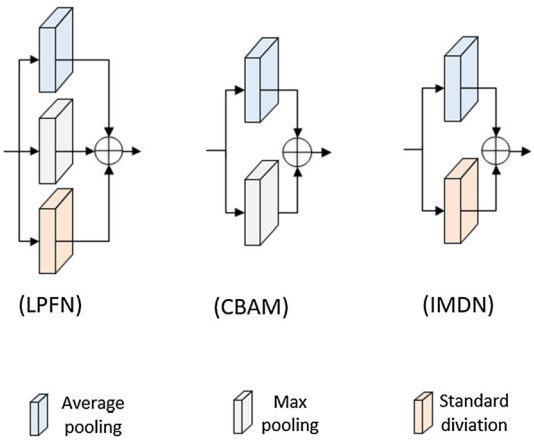


Fig. 8 Attention modules. LPFN is along channel and spatial dimensions using average pooling, max-pooling and standard deviation operations. CBAM [29] is along channel and spatial dimensions using both average pooling and max-pooling operations. IMDN [22] is along channel dimension using both average pooling and standard deviation operations

Table 4 Comparison of our DARB and existing attention models on LPFN

Attention module	Scale	Params	Set5	Set14	BSD100	Urban100	Manga109
			PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
LPFN	×3	649k	34.57/0.9285	30.50/0.8453	29.15/0.8065	28.42/0.8578	33.92/0.9465
LPFN-CBAM		648k	34.47/0.9279	30.46/0.8446	29.14/0.8066	28.37/0.8572	33.79/0.9459
LPFN-IMDN		647k	34.54/0.9281	30.45/0.8438	29.15/0.8063	28.43/0.8576	33.88/0.9462

channel attention, but has a better performance than CBAM [29] because of the standard deviation included in its CCA block. Our DARB has a better performance than CBAM [29] for we included standard deviation in DARB. Our DARB also has a better performance than IMDN [22], for we included channel attention and spatial attention in DARB. Therefore, standard deviation should be calculated in attention module, which indicates the dispersion of pixels along channel or spatial dimensions. The results demonstrate that our DARB is efficient to improve the performance of the network.

4.6 Ablation Study of Ensemble Method to Reconstruct SR Image

Most of the multi-branch networks reconstruct SR image by multi-reconstruction, such as MemNet [10], DRCN [5], LapSRN [11] and SRFBN [19]. SRFBN [19] was the most relevant method to our LPFN, which reconstructed SR image in each iteration to train the network and use the last SR image as the final SR result. textcolorredWe think the features of previous iterations are too shallow to obtain valuable reconstruction results, so we use ensemble method to reconstruct SR image. To prove the improvement of our ensemble method, we use multi-reconstruction method on our LPFN, which is shown in Fig. 9. From the comparison results shown in Table 5, we can find that, PSNR value of LBFN is 0.15 higher than LBFN with multi-reconstruction. The results demonstrate that, the features of previous iterations are too shallow to obtain valuable reconstruction results and ensemble method is more suitable for feedback networks than multi-reconstruction method.

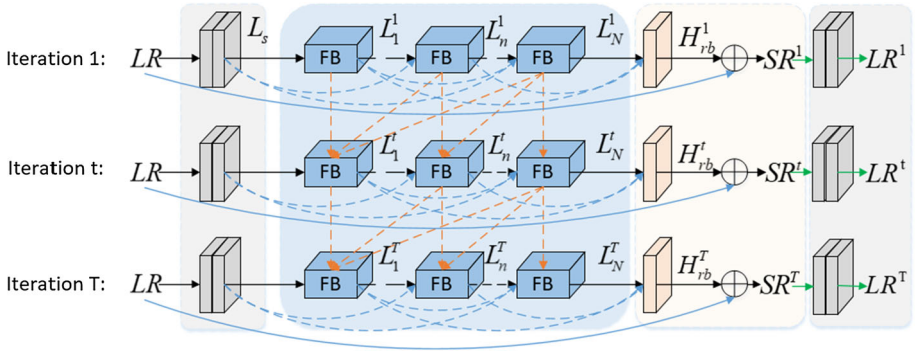


Fig. 9 Multi-reconstruction method used on our LPFN. To verify the improvement of our ensemble method, we use multi-reconstruction method on our LPFN to make a comparison

Table 5 Comparison of our ensemble method and multi-reconstruction method to reconstruct SR image on LPFN

Reconstruction methods	Scale	Params	Set5	Set14	BSD100	Urban100	Manga109
			PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Ensemble method	×3	649k	34.57/0.9285	30.50/0.8453	29.15/0.8065	28.42/0.8578	33.92/0.9465
Multi-reconstruction		649k	34.43/0.9274	30.40/0.8431	29.10/0.8052	28.30/0.8556	33.54/0.9436

4.7 Comparison with the State-of-the-art Methods

As a lightweight network, we compare our LPFN with other lightweight state-of-the-art methods, such as SRCNN [1], FSRCNN [2], VDSR [4], DRCN [5], LapSRN [11], DRRN [8], MemNet [10], SRFBN-S [19], LW-AWSRN [24], CARN [33], IMDN [22] and LatticeNet [23]. The PSNR and SSIM values of them are compared, as shown in Table 6. We can find that, our LPFN has less parameters, but better performance than other lightweight state-of-the-art methods. At the same time, we compare the multi-adds value with other methods by assuming the output image size to be 1280 × 720. Our network uses a lot of concatenation operations for our parral feedback architecture, which increases our multi-adds value. On the whole, however, our performance remains outstanding.

At last, we provide visual comparisons of the SR images on ×4 with other lightweight state-of-the-art methods, such as DRCN [5], DRRN [8], SRFBN-S [19], IDN [34], IMDN [22], as shown in Fig. 10. From the comparison results, we can find that, our method recovers textures and detials better than the others, which demonstrates the improvements of our LPFN.

5 Conclusion

In this paper, we proposed a lightweight parallel feedback network for image super-resolution (LPFN). All the feedback blocks in LPFN fed back their high-level features to refine low-level ones in a synchronous parallel manner. The parallel feedback and residual learning formed a local-mirror architecture, which learns more about the relationship between low-level features

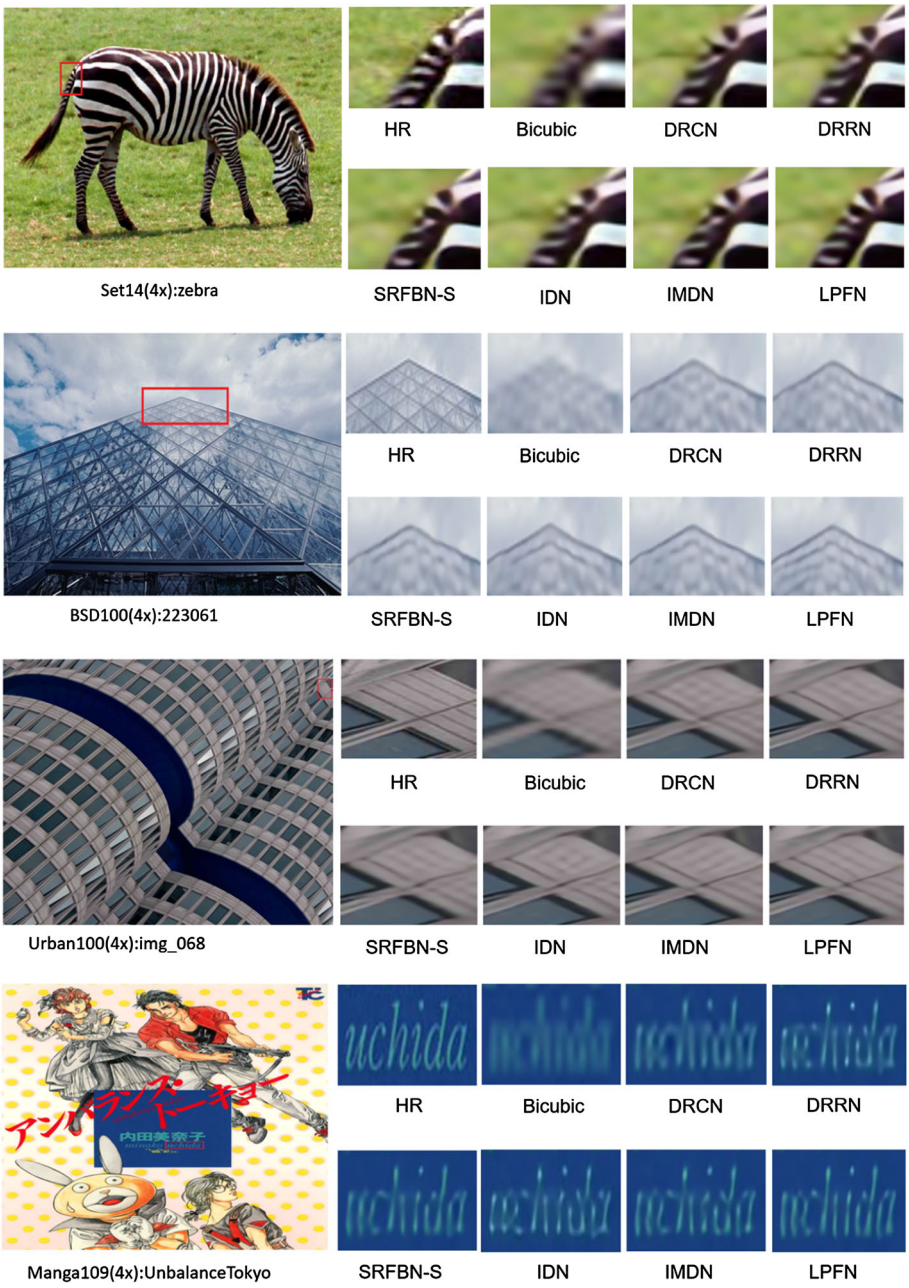


Fig. 10 Visual comparisons of our LPFN with other SR methods on Set14, BSD100 and Urban100 datasets

Table 6 Comparison of the average PSNRs/SSIMs for scale factors of ×2, ×3 and ×4 on the Set5, Set14, BSD100, Urban100, and Manga109 datasets. The best and the second-best results are highlighted in bold and italic, respectively. We calculate Muti-Adds with the output image of size 1280 × 720

Methods	Scale	Params	Multi-Adds	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 [1]		8K	52.7G	36.66/0.9542	32.45/0.9067	31.36/0.8879	29.50/0.8946	35.60/0.9663
FSRCN [2]		13K	6.0G	37.00/0.9558	32.63/0.9088	31.53/0.8920	29.88/0.9020	36.67/0.9710
VDSR [4]		666K	612.6G	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140	37.22/0.9750
DRCN [5]		1774K	9788.7G	37.63/0.9588	33.04/0.9118	31.85/0.8942	30.75/0.9133	37.55/0.9732
LapSRN [11]		813K	29.9G	37.52/0.9591	32.99/0.9124	31.80/0.8952	30.41/0.9103	37.27/0.9740
DRRN [8]		298K	6796.9G	/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188	37.88/0.9749
MemNet [10]		678K	623.9G	37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195	37.72/0.9740
SRFBN-S [19]		282K	138.2	37.78/0.9597	33.35/0.9156	32.00/0.8970	31.41/0.9207	38.06/0.9757
CARN [33]		1592K	222.8G	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256	38.36/0.9765
IMDN [22]		694K	158.8G	38.00/0.9605	33.63/0.9177	32.19/0.8996	32.17/0.9283	**38.88**/*0.9774*
LatticeNet [23]		756K	169.5G	**38.15/0.9610**	**33.78/0.9193**	*32.25/0.9005*	*32.43/0.9302*	–/–
LW-AWSRN [24]		1397K	320.5G	*38.11/0.9608*	**33.78**/*0.9189*	**32.26/0.9006**	**32.49/0.9316**	*38.87*/**0.9776**
LPFN(ours)		527K	348.7	38.04/*0.9608*	*33.70/0.9189*	32.18/0.8998	32.30/0.9298	38.78/0.9773
Bicubic	×3	–	–	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349	26.95/0.8556
SRCNN [1]		8K	52.7G	32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989	30.48/0.9117
FSRCN [2]		13K	5.0G	33.18/0.9140	29.37/0.8240	28.53/0.7910	26.43/0.8080	31.10/0.9210
VDSR [4]		666K	612.6	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279	32.01/0.9340
DRCN [5]		1774K	9788.7G	33.82/0.9226	29.76/0.8311	28.80/0.7963	27.15/0.8276	32.24/0.9343
DRRN [8]		298K	6796.9G	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378	32.71/0.9379
MemNet [10]		678K	623.9G	34.09/0.9248	30.00/0.8350	28.96/0.8001	27.56/0.8376	32.51/0.9369
SRFBN-S [19]		375K	86.7G	34.20/0.9255	30.10/0.8372	28.96/0.8010	27.66/0.8415	33.02/0.9404
CARN [33]		1592K	118.8G	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493	33.50/0.9440

Table 6 continued

Methods	Scale	Params	Multi-Adds	Set5	Set14	BSD100	Urban100	Manga109
			PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	
IMDN [22]		703K	71.5G	34.36/0.9270	30.32/0.8417	29.09/0.8046	28.17/0.8519	33.61/0.9445
LatticeNet [23]		765K	76.3G	*34.53/0.9281*	*30.39*/0.8424	*29.15*/0.8059	*28.33*/0.8538	–/–
LW-AWSRN [24]		1476K	150.6G	34.52/*0.9281*	30.38/*0.8426*	**29.16/0.8069**	**28.42/0.8580**	*33.85/0.9463*
LPFN(ours)		649K	259.07	**34.57/0.9285**	**30.50/0.8453**	*29.15/0.8065*	**28.42**/*0.8578*	**33.92/0.9465**
Bicubic	×4	–	–	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866
SRCNN [1]		8K	52.7G	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221	27.58/0.8555
FSRCN [2]		13K	4.6G	30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280	27.90/0.8610
VDSR [4]		666K	612.6G	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524	28.83/0.8870
DRCN [5]		1774K	9788.7G	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510	28.93/0.8854
LapSRN [11]		502K	149.4G	31.54/0.8852	28.09/0.7700	27.32/0.7275	25.21/0.7562	29.09/0.8900
DRRN [8]		298K	6796.9G	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638	29.45/0.8946
MemNet [10]		678K	623.9G	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630	29.42/0.8942
SRFBN-S [19]		483K	66G	31.98/0.8923	28.45/0.7779	27.44/0.7313	25.71/0.7719	29.91/0.9008
CARN [33]		1592K	90.9G	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	30.47/0.9084
IMDN [22]		715K	40.9G	32.21/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838	30.45/0.9075
LatticeNet [23]		777K	43.6G	*32.30/0.8962*	*28.68*/0.7830	27.62/0.7367	26.25/0.7873	–/–
LW-AWSRN [24]		1587K	91.1G	32.27/0.8960	**28.69/0.7843**	**27.64/0.7385**	*26.29*/**0.7930**	*30.72/0.9109*
LPFN(ours)		788K	224G	**32.36/0.8966**	*28.68/0.7840*	*27.63/0.7380*	**26.32**/*0.7922*	**30.79/0.9122**

and high-level ones. Then, we proposed a dispersion-aware attention residual block as the basic blocks in feedback blocks, which improved the discriminate ability of feature maps and enhanced image details. We used ensemble method to reconstruct SR image, which has a better performance than multi-reconstruction method. At last, we proposed a global feedback, which fed back the degradation results of SR to LR, and calculated the feedback-regression loss with primal LR. Feedback-regression loss helps to better learn the mapping function from LR to HR. Further experiments based on benchmark datasets show that, the LPFN we proposed has an outstanding performance as a lightweight SR network.

Author Contributions All authors contributed to the study conception and design. Software and experiments were performed by beibei wang. Material preparation, data collection and analysis were performed by beibei wang, changjun liu and xiaomin yang. The first draft of the manuscript was written by beibei wang and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This work was supported by the funding from Science Foundation of Sichuan Science and Technology Department 2021YFH0119 and the funding from Sichuan University under grant 2020SCUNG205.

Declarations

Conflicts of interest/Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Chao Dong KH, Change Loy Chen, Tang X (2014) Learning a deep convolutional network for image super-resolution In: Computer Vision – ECCV 2014, Vol 8692, 2014, pp 184–199
2. Chao Dong CCL, Tang X (2016) Accelerating the super-resolution convolutional neural network In: Computer Vision – ECCV 2016, pp 391–407
3. Shi W, Caballero J, Huszár F, Totz J, Aitken AP, Bishop R, Rueckert D, Wang Z (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1874–1883
4. Kim J, Lee J, Lee K (2016) Accurate image super-resolution using very deep convolutional networks In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1646–1654
5. Kim J, Lee JK, Lee KM (2016) Deeply-recursive convolutional network for image super-resolution In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1637–1645
6. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 770–778
7. Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, Shi W (2017) Photo-realistic single image super-resolution using a generative adversarial network In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 105–114
8. Tai Y, Yang J, Liu X (2017) Image super-resolution via deep recursive residual network In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2790–2798
9. Lim B, Son S, Kim H, Nah S, Lee KM (2017) Enhanced deep residual networks for single image super-resolution In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp 1132–1140
10. Tai Y, Yang J, Liu X, Xu C (2017) Memnet: A persistent memory network for image restoration In: 2017 IEEE International Conference on Computer Vision (ICCV), pp 4549–4557
11. Lai W-S, Huang J-B, Ahuja N, Yang M-H (2017) Deep laplacian pyramid networks for fast and accurate super-resolution In: IEEE Conference on Computer Vision and Pattern Recognition
12. H. L. Z. L. W. W. A. P. G. J. X. Y. Shipeng Fu, Lu Lu (2019) A real-time super-resolution method based on convolutional neural networks In: Circuits,Systems,andSignalProcessing
13. Huang G, Liu Z, van der Maaten L, Weinberger K (2017) Densely connected convolutional networks
14. Gilbert CD, Sigman M (2007) Brain states: Top-down influences in sensory processing. Neuron 54:677–696

15. Stollenga M, Masci J, Gomez F, Schmidhuber J (2014) Deep networks with internal selective attention through feedback connections, Vol 4
16. Zamir AR, Wu T-L, Sun L, Shen W, Shi BE, Malik J, Savarese S (2016) Feedback networks In: Computer Science - Computer Vision and Pattern Recognition
17. Carreira J, Agrawal P, Fragkiadaki K, Malik J (2016) Human pose estimation with iterative error feedback In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp 4733–4742
18. Haris M, Shakhnarovich G, Ukita N (2018) Deep back-projection networks for super-resolution In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 1664–1673
19. Li Z, Yang J, Liu Z, Yang X, Jeon G, Wu W (2019) Feedback network for image super-resolution In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
20. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition In: Conference on Computer Vision and Pattern Recognition (CVPR)
21. Szegedy C, Liu Wei, Jia Yangqing, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1–9
22. Hui Z, Gao X, Yang Y, Wang X (2019) Lightweight image super-resolution with information multi-distillation network In: Proceedings of the 27th ACM International Conference on Multimedia, pp 2024–2032
23. Xie Y, Zhang Y, Qu Y, Li C, Fu Y (2020) LatticeNet: Towards Lightweight Image Super-Resolution with Lattice Block, pp 272–289
24. Li Z, Wang C, Wang J, Ying S, Shi J (2021) Lightweight adaptive weighted network for single image super-resolution. Computer Vision and Image Understanding 211:103254
25. G. A. Mnih V, Heess N (2014) Recurrent models of visual attention In: Neural Information Processing Systems, pp 2204–2212
26. Wang X, Girshick R, Gupta A, He K (2018) Non-local neural networks In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 7794–7803
27. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 7132–7141
28. Zhang Y, kunpeng Li, Li K, Wang L, Zhong B, Fu Y (2018) Image super-resolution using very deep residual channel attention networks In: European Conference on Computer Vision, Vol 11211, pp 294–310
29. Woo S, Park J, Lee J-Y, Kweon I (2018) CBAM: Convolutional Block Attention Module: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII, pp 3–19
30. Zhu X, Guo K, Fang H, Chen L, Ren S, Hu B (2022) Cross view capture for stereo image super-resolution. IEEE Transactions on Multimedia 24:3074–3086
31. Han W, Chang S, Liu D, Yu M, Witbrock M, Huang TS (2018) Image super-resolution via dual-state recurrent networks In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 1654–1663
32. Zhu X, Guo K, Ren S, Hu B, Hu M, Fang H (2022) Lightweight image super-resolution with expectation-maximization attention mechanism. IEEE Transactions on Circuits and Systems for Video Technology 32(3):1273–1284
33. Namhyuk Ahn BK, Sohn K-A (2018) Fast, accurate, and lightweight super-resolution with cascading residual network In: Computer Vision – ECCV 2018, Vol 11214, pp 256–272
34. Hui Z, Wang X, Gao X (2018) Fast and accurate single image super-resolution via information distillation network In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 723–731