Journal of Signal Processing Systems (2023) 95:305–318
https://doi.org/10.1007/s11265-022-01764-5

Closed-loop Feedback Network with Cross Back-Projection for Lightweight Image Super-Resolution

Beibei Wang[1] · Changjun Liu[1] · Seunggil Jeon[2] · Xiaomin Yang[1]

Received: 15 November 2021 / Revised: 14 March 2022 / Accepted: 24 April 2022 / Published online: 17 May 2022

Abstract
Since the development of deep learning, image super-resolution (SR) has made great progress, and become the focus of academic research. Because high-level features are more informative for the reconstruction, most SR networks have a large number of layers and parameters, which restrict their application in resource-constrained devices. Recently, lightweight networks got a lot attention for their broad application prospect. To improve the performance of lightweight networks by informative high-level features, we introduce feedback mechanism into our method, which can feed back high-level features to refine low-level ones. In this paper, we propose a closed-loop feedback network with cross back-projection for lightweight image super-resolution (CCFN), which uses feedback mechanism in three manners. First, based on error feedback, we propose a cross back-projection feedback block (CFB). CFB uses error feedback to correct the features of multi-scale fusion, which also can be viewed as two cross-learning back-projection units. Second, CFB works in a self-feedback manner, which feeds back the output high-level features to refine the low-level ones of the input. Third, we propose a global feedback, which feeds back the degradation results of SR to LR, to guide the learning of mapping functions from LR to HR. Finally, we use attention-based model as the basic block in CFB, and since our method works in an iterative manner, recursive concatenation is more suitable than multi-reconstruction. The final experimental results show that our CCFN has a competitive performance with few parameters.

Keywords Super-resolution · Cross back-projection · Closed-loop · Feedback · Lightweight

1 Introduction

The single image super-resolution (SISR) aims to reconstruct high-resolution (HR) image from the low-resolution (LR) image, which is an ill-posed problem, for the LR image can be generated from an infinite number of HR images. Many methods [1–8] have been proposed to solve this problem.

✉ Seunggil Jeon
simon.sgJeon@gmail.com

✉ Xiaomin Yang
arielyang@scu.edu.cn

Beibei Wang
690983790@qq.com

Changjun Liu
cjliu@scu.edu.cn

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

2 Samsung Electronics, 129, Samseong-ro Yeongtong-gu Suwon-si, Gyeonggi-do 16677, South Korea

SRCNN [2] proposed by Dong et al. introduced deep learning into image super-resolution for the first time. Then Dong et al. proposed FSRCNN [1], which learned LR features and used deconvolution on the last layer to reduce the amount of calculation. Sub-pixel convolution was proposed by Shi et al. in ESPCN [7], which upscaled the LR features by a periodic shuffling operator at the end of the network.

To improve the performance of SR networks, deep networks were proposed. Many literatures [9, 10] have proved that, the deeper the networks, the better the expressive ability. However, deep networks have two drawbacks. One drawback is that deep networks are very difficult to train, for they are easy to cause gradient vanishing/exploding problems. To solve this problem, residual learning was well used in ResNet [11]. Residual learning is used to fuse the low-level features into high-level ones to enhance the gradient flow. Then EDSR [12], DenseNet [13] and SRResNet [5] were proposed based on residual learning. Another drawback of deep networks is that they have too many parameters, resulting in a lot of memory footprint. Therefore, recursive convolutional networks [4, 14] were proposed, for they can reduce parameters of deep networks by parameter sharing between recursive blocks.

However, since innumrable HR images can be degraded to one LR image, the solution space for mapping functions of LR-HR is very large. The traditional networks are trained by paired LR-HR datasets and calculate the loss by the difference between SR and HR. Recently, DRN [15] introduced dual-regression scheme into SR, which calculated the primal loss and the dual-regression loss to constrain the space of possible mapping functions. The dual-regression scheme can directly learn LR images without the supervision of corresponding HR images. Inspired by DRN [15], we propose a global feedback to guide the learning of mapping functions from LR to HR.

The methods mentioned above are all feedforward, the latter layers of which are just a nonlinear mapping of the outputs from previous layers. However, the high-level features are more informative for the reconstruction, so the refinement of low-level features with high-level ones is very important. Many feedback networks [6, 16–18] were proposed to solve this problem. Recently feedback mechanism was introduced into SR. DBPN [19] proposed error feedback in up- and down-projection units to realize self-correction. SRFBN [6] used feedback mechanism in a manner similar to recurrent neural network (RNN), which learned the recurrent block in a feedback manner.

Lightweight networks got a lot attention for their broad application prospect recently. All the lightweight methods make an effort to achieve a better performance with fewer parameters. Since high-level features are more informative for reconstruction, the refinement of low-level features with high-level ones is very important. Although feedback mechanism was introduced into SR, which has not been fully exploited in lightweight SR methods. Therefore, to further improve the performance of lightweight SR, feedback mechanism should be introduced into lightweight SR and fully utilized. In this paper, we propose a closed-loop feedback network with cross back-projection for lightweight image super-resolution (CCFN), which is shown in Fig. 1. SRFBN [6] is the most relevant work to our CCFN, which is a deep network with only self-feedback. We use feedback mechanism in three manners: error feedback, self-feedback and global feedback. Based on error feedback and inspired by multi-scale fusion proposed in HRNet [20], we propose a motivation that, the error feedback can be used to correct the features of multi-scale fusion, which also can be viewed as two cross-learning back-projection units. Therefore, a cross back-projection feedback block (CFB) is proposed, which can enhance the representation ability of multi-scale fusion and back-projection units. The feedback block CFB works in a self-feedback manner in CCFN, which can feed back high-level features from ouput to its input, so that the new ouputs are more informative for reconstruction. Then we propose a motivation that, a global feedback can feed back the degradation results of SR to LR to guide the learning of mapping functions of LR-HR. Then we use RCAB [21] as the basic block in CFB, which integrated channel attention into residual blocks. Channel attention module can extract the channel statistic to enhance network discriminative ability. At last, since our CCFN is a feedback network, which works in an iterative manner, the previous iterations are less informative for the reconstruction. Therefore, we propose a motivation that concatenation reconstruction is more suitable for feedback networks than the multi-reconstruction used in existing feedback methods. With the help of the motivations we proposed, our method acheives an outstanding performance as a lightweight network, as shown in Fig. 2.

In summary, the contributions of the CCFN we proposed are as follows:

- We propose a lightweight network based on feedback mechanism (CCFN), which uses feedback mechanism in three manners: self-feedback, error feedback and global feedback. Since high-level features are more informative for reconstruction, which is proved in deep networks, the refinement of low-level features with high-level ones by feedback mechanism is very important.
- We propose a cross back-projection feedback block (CFB), which uses error feedback to correct the features of multi-scale fusion. CFB also can be viewed as two cross-learning back-projection units, which can get useful

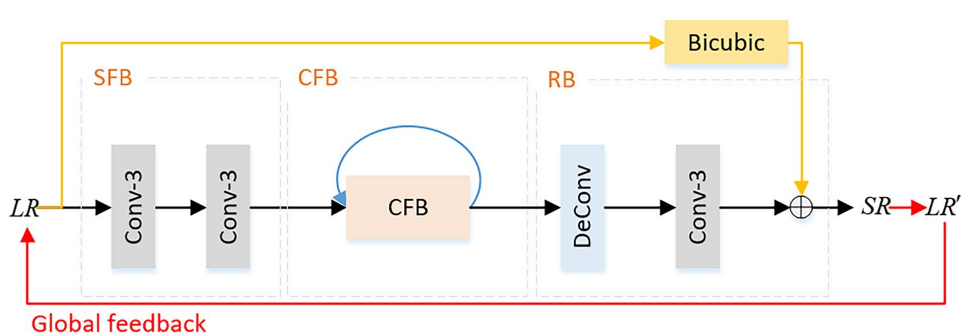
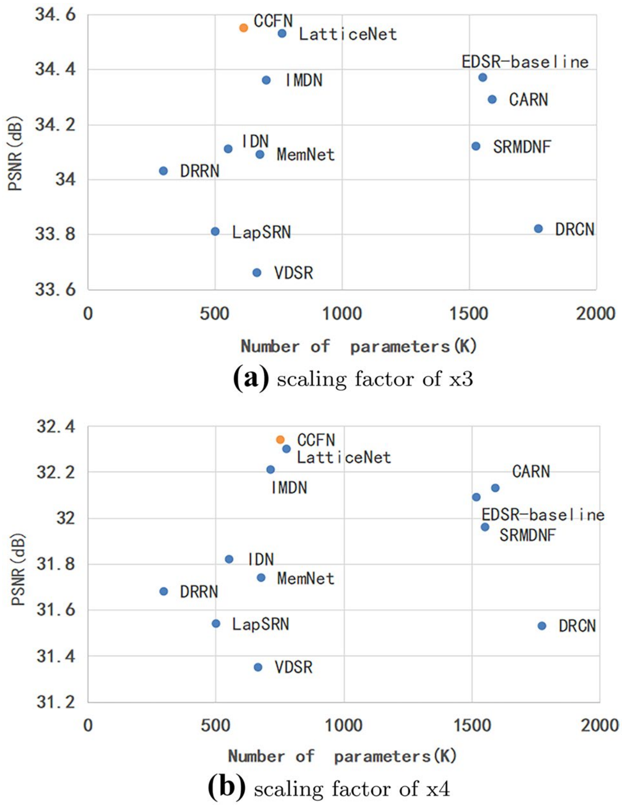


Figure 1 Closed-loop feedback network with cross back-projection for lightweight image super-resolution (CCFN). Blue arrows represent self-feedback. Red arrows represent the global feedback. Orange arrows represent bicubic upsampling.



(a) scaling factor of x3
(b) scaling factor of x4

Figure 2 PSNR vs. number of parameters on Set5 dataset.

information from each other. The CFB we proposed has a better performance than the existing back-projection units and multi-scale fusion. (more details see Sect. 3.2).

– We propose a global feedback to better guide the learning of mapping functions from LR to HR, which feeds back the degradation results of SR to calculate feedback-regression loss with the primal LR. The feedback-regression loss can supervise the train of the network together with primal regression loss. Our global feedback can be used in all scales of SR networks directly, which can obtain a better mapping function but introduces very few parameters.

2 Related Work

2.1 Lightweight Networks

Lightweight networks have gotten a lot of attention in recent years, for they are appliable for embedded devices with resource-constrained. Dong et al. proposed SRCNN [2], which introduced deep learning into image SR. Then they proposed FSRCNN [1], which learned LR features and used deconvolution at the end to reduce calculations. VDSR [3] proposed by Jiwon Kim et al. had 20 layers, which took an interpolated LR image as input. Then Jiwon Kim et al. proposed DRCN [4], which introduced recursive convolution into image SR. Therefore, the networks can be very deep without adding new parameters, for they share the same weights among recursive convolutional layers. In DRRN [14], a recursive block contained several residual units, and recursive-supervision was proposed to improve the performance. LapSRN [22] reconstructed the final SR image progressively by multiple intermediate SR predictions in the network. Based on residual learning and information distillation block, IDN [23] achieved a competitive results with less layers. CARN [24] used cascading residual to improve the network performance. IMDN [25] proposed information multi-distillation and contrast-aware attention module (CCA) to achieve outstanding performance. In the year of 2020, LatticeNet [26] was proposed, in which lattice block was used to expand the representation capabilities of the network significantly by the combination of residual blocks. Then SMSR [27] learned sparse masks to reduce the redundant computation of the network in the year of 2021.

The lightweight methods mentioned above are all feed-forward. Since high-level features are more informative for reconstruction, which is proved in deep networks, to improve the performance of lightweight networks, we propose a motivation that, the refinement of low-level features with high-level ones by feedback is very important. Experimental results indicate that, with the help of feedback mechanism, our CCFN achieves a better performance than the lightweight networks mentioned above.

2.2 Back-projection

Most of the networks [1–3, 23, 25, 26] upscaled LR features at the beginning or end of the networks only once. [4, 14] upscaled the LR features at each iterarion. LapSRN [22] upscaled the LR features to HR features progressively. The LR features of these networks are up-sampled to HR features by direct or progressive upsampling process.

Back-projection can minimize the reconstruction error by iterative up- and down-sampling procedure, which was proposed in an early SR method [28] originally for multiple LR inputs. Then bilateral back-projection was proposed for single LR input in [29]. NLIBP [30] proposed a non-local iterative back-projection method for image enlargement. [31] proved that, back-projection refinement can improve the performance of learning-based SISR. Recently, DBPN [19] proposed up- and down-projection units, which were learned in an iterative procedure to guide the reconstruction, then the authors improved the reconstruction performance by dense skip connections.

The back-projection mentioned above worked in an independent manner. Inspired by DBPN [19] and multi-scale fusion proposed in HRNet [20], we propose a motivation that the cross-learning between back-projection units can further enhance their representation ability. Experimental results indicate that cross back-projection we proposed has a better performance than the independent projection units used in existing methods.

2.3 Feedback Mechanism

Most of the neural networks are feedforward networks [1, 2, 22, 32–34]. Recursive networks [4, 14, 35] shared the same weights among recursive blocks in a feedforward manner. Residual networks [11–13, 36] fused different level information by sending shallow features to the latter layers, which were also feedforward networks. In feedforward networks, the inputs of latter layer are just a nonlinear mapping of the outputs from previous layers.

Feedback mechanism works in a way of top to down, feeding back high-level features to previous layers, so that previous layers can get useful information from the following layers and refine low-level features. Feedback mechanism has been used in various computer vision methods [16–18, 37], which was first introduced into SR by DBPN [19]. In DBPN [19], error feedback was proposed in up- and down-projection units to realize self-correction. Then SRFBN [6] used feedback mechanism in a manner similar to RNN, which repalced the recurrent block with feedback block.

Although feedback mechanism was introduced into SR, which has not been fully exploited in lightweight SR methods. Since high-level features are more informative for reconstruction, we propose a motivation that, the performace of lightweight SR networks can be further improved by making full use of feedback mechanism. We use error feedback and self-feedback manners in our CCFN. Furthermore, we propose a global feedback to guide the learning of mapping functions from LR to HR. Experimental results indicate that, all the feedback manners have a better performance than the correspongding feedforward manners.

3 Our Method

In this section, the network architecture of CCFN is first described. Then CFB as the basic feedback block in CCFN is described in detail. At last, the global feedback together with loss function are described.

3.1 Network Structure

Since CFB works in a self-feedback manner, the CCFN can be unfolded to T iterations. As [6] do, we set iteration T = 4, so iteration t is ordered from 1 to 4. Different from SRFBN [6], we recursively concatenate the upsampling results at the end of each iteration and reconstruct SR images at the last iteration, as shown in Fig. 3, which is proven to improve the network performance efficiently.

The CCFN contains three parts: shallow feature extraction block (SFB), cross back-projection feedback block (CFB) and reconstruction block (RB). In the first part, SFB extracts shallow features, then the shallow features are passed to CFB. In the second part, CFB is a feedback block, so the output of CFB is fed back to itself in next iteration, and also passed to RB of the current iteration. Finally, in RB, the concatenation results are used to reconstruct SR image at the last iteration by adding bicubic upsampling results. Then the degradation of SR results is fed back to calculate the feedback-regression loss.

We define L_{in} is the output of SFB, as well as one of the inputs of CFB, which can be obtained by:

$$L_{in} = f_{SFB}(LR), \tag{1}$$

where f_{SFB} is the operations of SFB, which consists of a conv(3,128) and a conv(128,32) to extract the shallow LR features.

In the first iteration, CFB takes L_{in} as input. While in the other iterations, CFB takes L_{in} and the output of CFB from last

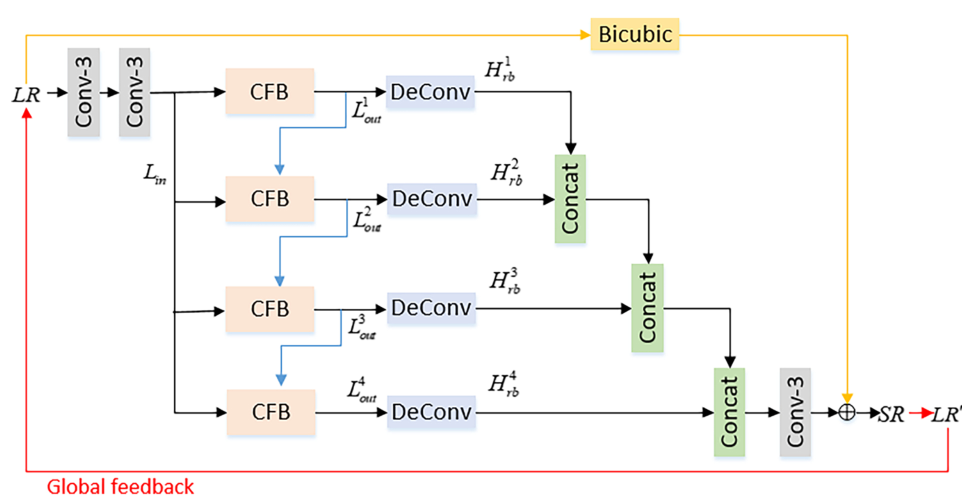


Figure 3 The unfolded CCFN.

iteration as inputs, which will be covered in detail in Sect. 3.2. Therefore, L^t_{out} as the output of CFB in the t-th iteration, which can be obtained by:

$$L^t_{out} = \begin{Bmatrix} f_{CFB}(L_{in}) & t = 1 \\ f_{CFB}([L_{in}, L^{t-1}_{out}]) & t \geq 2 \end{Bmatrix}, \tag{2}$$

where f_{CFB} is the operations of CFB.

In RB, we use deconvolutional layer on the output of CFB, and then concatenate them recursively to reconstruct SR image at the last iteration. We define the results after the deconvolutional upsampling in the t-th iteration as follows:

$$H^t_{rb} = f_{up}(L^t_{out}). \tag{3}$$

Because of the recursive concatenation and global residual learning, the SR image can be obtained by:

$$SR = f_{cm}([[[H^1_{rb}, H^2_{rb}], H^3_{rb}], H^4_{rb}]) + f_{BC}(LR), \tag{4}$$

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

Because of the global feedback, we generate LR' by downsampling operator f_{down}, which consists of conv(3,32) and conv(32,3). LR' is used to calculate the feedback-regression loss together with the primal LR, which will be covered in detail in Sect. 3.3.

$$LR' = f_{down}(SR), \tag{5}$$

3.2 Cross Back-projection Feedback Block (CFB)

CFB is the basic feedback block of CCFN, which works in a self-feedback manner. CFB has two inputs, the output of SFB and the output of CFB from last iteration. The output of CFB is passed to next part of the current iteration, and fed back to itself in next iteration.

Inspired by HRNet [20] and DBPN [19], we propose cross back-projection, as shown in Fig. 4. The first line is HR feature flow, and the second line is LR feature flow. We cross connect the two feature flows by upsampling and downsampling operations to exchange the HR and LR information densely. Different from the multi-scale fusion proposed in HRNet [20], we use error feedback mechanism at the end of the two feaure flows. Therefore, two cross-learning feature flows form two cross back-projection units. The green arrows are connected to form a up-projection unit. The orange arrows are connected to form a down-projection unit. The black arrows connect the up-projection unit and down-projection unit to exchange informations of the two units.

To further improve the performance of our cross back-projection, we use residual channel attention block (RCAB) [21] as each basic block, which integrated channel attention into residual blocks. Attention module can extract the channel statistic to enhance network discriminative ability, which improved the performance of CFB greatly.

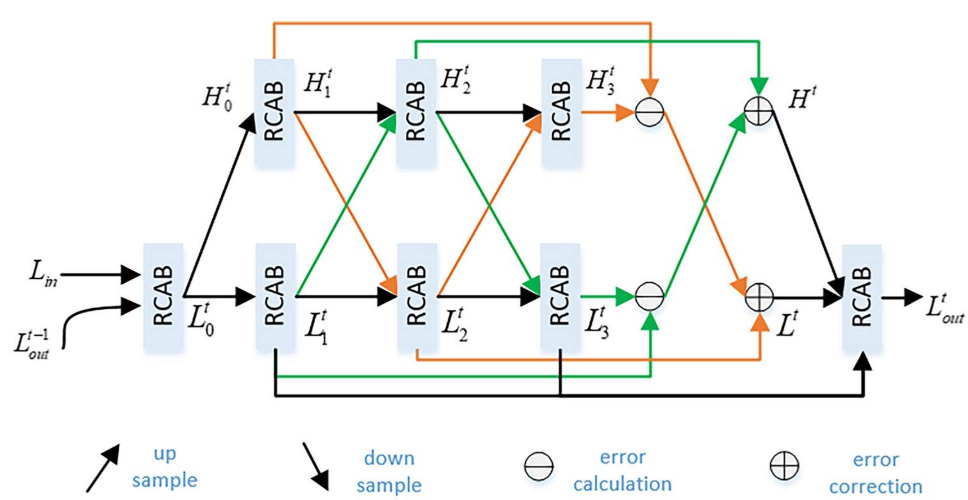
In CFB of the t-th iteration, we define the HR features in HR flow are $H^t_0, H^t_1, H^t_2, H^t_3$, respectively, and define the LR features in LR flow are $L^t_0, L^t_1, L^t_2, L^t_3$, respectively. The process of the CFB is as follows:

$$\begin{Bmatrix} L^t_0 = f_{RCAB}([L_{in}, L^{t-1}_{out}]) \\ H^t_0 = f_{up}(L^t_0) \end{Bmatrix}, \tag{6}$$

$$\begin{Bmatrix} L^t_1 = f_{RCAB}(L^t_0) \\ H^t_1 = f_{RCAB}(H^t_0) \end{Bmatrix}, \tag{7}$$

$$\begin{Bmatrix} L^t_2 = f_{RCAB}([L^t_1, f_{down}(H^t_1)]) \\ H^t_2 = f_{RCAB}([H^t_1, f_{up}(L^t_1)]) \end{Bmatrix}, \tag{8}$$

Figure 4 Cross back-projection feedback block (CFB). The green arrows are connected to form a up-projection unit. The orange arrows are connected to form a down-projection unit. The black arrows connect the up-projection unit and down-projection unit to exchange information of the two units.



$$\left\{ \begin{array}{l} L_3^t = f_{RCAB}([L_2^t, f_{down}(H_2^t)]) \\ H_3^t = f_{RCAB}([H_2^t, f_{up}(L_2^t)]) \end{array} \right\}, \tag{9}$$

where f_{RCAB} is the operation of the basic block in CFB. f_{up} and f_{down} are the deconvolutional upsampling operation and convolutional downsampling operation, respectively.

Then we use L_1^t and L_3^t to correct H_2^t, H_1^t and H_3^t to correct L_2^t:

$$\left\{ \begin{array}{l} L^t = L_2^t + f_{down}(H_3^t - H_1^t) \\ H^t = H_2^t + f_{up}(L_3^t - L_1^t) \end{array} \right\}. \tag{10}$$

Finally, we downscale the output of up-projection unit, and fuse it with the output of down-projection unit by the final basic block. So the output of CFB can be obtained by:

$$\left\{ L_{out}^t = f_{RCAB}([L^t, L_1^t, L_3^t, f_{down}(H^t)]) \right\}. \tag{11}$$

3.3 Loss Function with Global Feedback

We propose a global feedback in our network, which feeds back the degradation results of SR to LR. LR' as the degraded result of SR is compared with the primal LR to constrain the space of possible solutions for mapping functions. So the loss function contains two parts: the primal regression loss which can be calculated by SR and HR, the feedback-regression loss which can be calculated by LR' and LR. So the loss function is as follows:

$$Loss = L_1(SR, HR) + \theta L_1(LR', LR), \tag{12}$$

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

4 Experimental Details

4.1 Datasets

DIV2k dataset [38] contains 800 images with 2K resolution. We expand the number to 8000 by augmentation (rotation and cropping), and downscale them by bicubic downsampling to generate LR images to train our CCFN.

We test our network on five benchmark datasets: Set5, Set14, BSD100, Urban100 and Manga109 datasets. Set5 contains 5 test images, namely babies, birds, butterflies, heads and women. Set14 contains 14 test images, which are more diverse and larger than images in Set5. BSD100 contains 100 images, which covers a wide variety of real-life scenes. Urban100 consists of 100 urban environmental images with high self-similarity. Manga109 contains 109 Japanese comics, which is the latest test dataset. At last, we test our network on RealSR [39], which captured 234 scenes by digital cameras to generate 595 real-world HR-LR image pairs for different scales.

4.2 Implementation Details

We use adam optimizer, which was proposed by Jimmy Ba et al. in Adam [40], and was recommended as the default algorithm to use. Learning rate controls the updating speed of parameters of mapping functions to find the optimal solution. Therefore, we set the initial lr = 0.0005 for fast network convergence, and halve it per 200 epoches with total 1000 epoches to find the optimal solution. We set the kernel size of convolutional layers in our CCFN is 3×3, except for the 1×1 convolutional layers for compressing feature channels. Batch size is set to 16. The training of our network is supervised by L1 loss, which is more robust against outliers and guide the loss to achieve a better local minimum. The L1 function is shown in Eq. (13), which calculates the difference between the reconstructed image and primal real image.

$$L_1(I, \hat{I}) = \sum_i^n \left| I_i - \hat{I}_i \right|, \tag{13}$$

where $\hat{I}$ is the reconstructed image, and I is the primal real image.

At last, we calculate the signal-to-noise ratio (PSNR) and structural similarity index (SSIM) values to measure the reconstruction quality of CCFN, which are shown in Eqs. (14) and (15). All the results are obtained on GPU 3060 using Pytorch framework.

$$PSNR = 10 \times log_{10}\left(\frac{MAX_I^2}{MSE}\right), \tag{14}$$

where MSE is the mean square error of I and $\hat{I}$, MAX is the maximum possible pixel value.

$$SSIM(I, \hat{I}) = \frac{2\mu_I\mu_{\hat{I}} + k_1}{\mu_I^2 + \mu_{\hat{I}}^2 + k_1} \times \frac{\sigma_{I\hat{I}} + k_2}{\sigma_I^2 + \sigma_{\hat{I}}^2 + k_2}, \tag{15}$$

where μ_I and σ_I^2 are the mean and variance of I. $\sigma_{I\hat{I}}$ is the covariance between $\hat{I}$ and I. k_1 and k_2 are constant terms.

5 Experimental Results

To validate our motivation that feedback mechanism can improve the performance of SR networks, we compare our self-feedback, error feedback and global feedback with the existing feedforward methods, as shown in Sects. 5.1, 5.2 and 5.3. To validate our motivation that cross back-projection has a better performance than the existing independent back-projeciton, we use independent back-projeciton on our method, as shown in Sect. 5.4. Then,

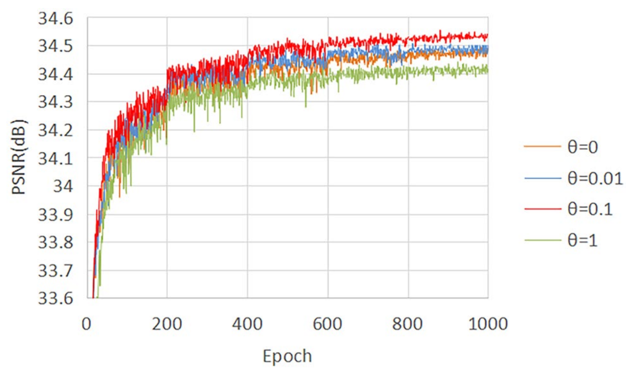


Figure 5 Effect of global feedback.

to validate our motivation that residual channel attention block as the basic block can improve the performance of CFB, we use the regular convolution used in existing mehtods instead, as shown in Sect. 5.5. At last, to validate our motivation that recursive concatenation is more suitable for feedback networks than multi-reconstruction, we use the existing multi-reconstruction on our method, as shown in Sect. 5.6. All the motivations we proposed improve the methodologies proposed in existing methods. In Sect. 5.7, we compare the performance of our method with the state-of-the-art lightweight methods on five widely used benchmark datasets and real-world dataset RealSR, our method has a better performance with fewer parameters.

5.1 Effect of Global Feedback

We propose a global feedback to feed back the degradation of SR results back to LR, which is used to calculate feedback-regression loss with the primal LR. Therefore, the global feedback can guide the learning of mapping functions from LR to HR. In Eq. (12), θ controls the weight of feedback-regression loss. We set $\theta = 0$ to prove the contribution of global feedback to our method. Then we change the value of θ from 0.01 to 1 to get an appropriate weight value for feedback-regression loss. From the results shown in Fig. 5, we can find that, the global feedback is beneficial to our CCFN, and the network has a best performance when $\theta = 0.1$. Therefore, we set $\theta = 0.1$ in the other experiments.

5.2 Effectiveness of Self-feedback

CFB works in a self-feedback manner, which feeds back the output to its input. To prove the effectiveness of self-feedback, we let CFB work in a feedforward manner, as shown in Fig. 6. Since CFB works in a feedforward manner, the feedback architecture becomes a recursive network, which is named CCFN-RNN. To make a fair comparison, CFB as a recursive block is applied 4 times. To avoid the problem of gradient vanishing/exploding, we use residual learning to concat the outputs of recursive blocks. The comparison results are shown in Table 1. We can find that, self-feedback manner has a better performance than the existing feedforward manner.

5.3 Effectiveness of Error Feedback

In CFB, we use error feedback at the end of multi-scale fusion, so the projection errors are used to correct the features in early layers. To prove the effectiveness of error feedback, we replace cross back-projection in CCFN with the multi-scale fusion proposed in HRNet [20], which is named CCFN-HRNet, as shown in Fig. 7. The comparison results are shown in Table 2.

Figure 6 The architecture of CCFN-RNN.

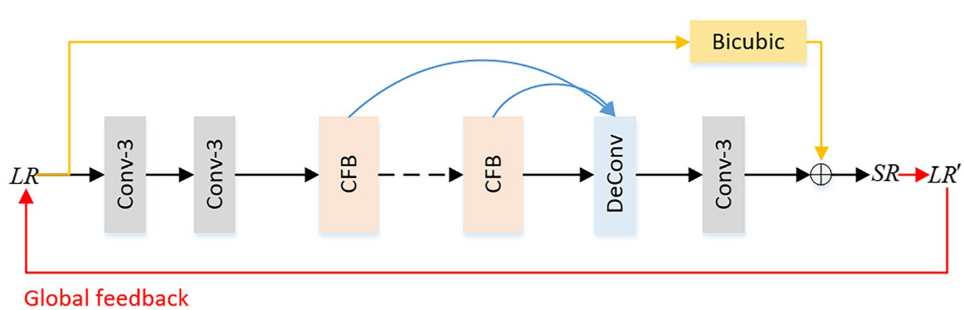


Table 1 Comparison of self-feedback and feedforward recursive manner.

Methods	Scale	Params	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
CCFN	×3	612K	34.55/0.9281	30.41/0.8435	29.15/0.8065	28.40/0.8572	33.80/0.9460
CCFN-RNN		613K	34.39/0.9271	30.34/0.8422	29.09/0.8048	28.17/0.8531	33.49/0.9440

Figure 7 The basic block of CCFN-HRNet.

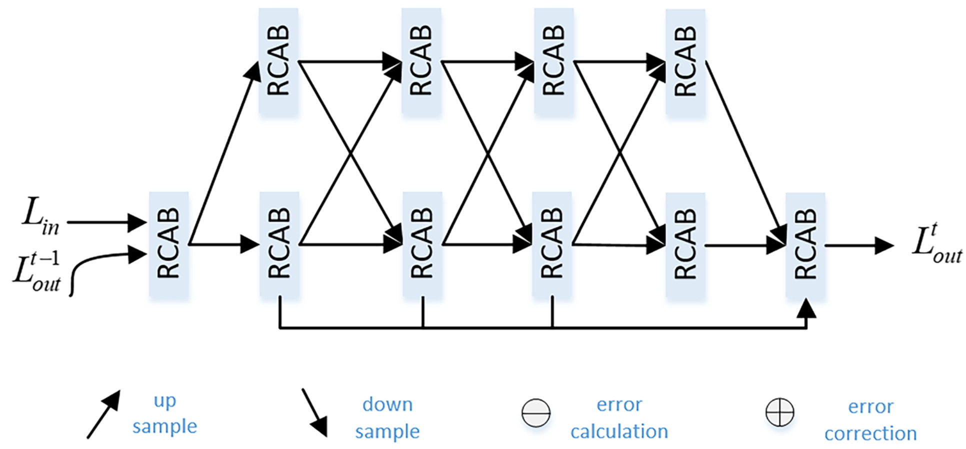


Table 2 Comparison of our cross back-projection and multi-scale fusion.

Methods	Scale	Params	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
CCFN	×3	612K	34.55/0.9281	30.41/0.8435	29.15/0.8065	28.40/0.8572	33.80/0.9460
CCFN-HRNet		656K	34.52/0.9277	30.42/0.8440	29.13/0.8064	28.40/0.8574	33.77/0.9457

We can find that, our cross back-projection has a better performance than cross-learning.

5.4 Comparison of our Cross Back-projection and Independent Back-projection

We propose a cross back-projection feedback block (CFB), in which the two back-projection units are cross-learned. Therefore, the two units can get useful information from each other to enhance the representation ability of features. To prove the effectiveness of cross back-projection, we replace cross back-projection in CCFN with independent up- and down-projection units proposed in DBPN [19], which is named CCFN-DBPN, as shown in Fig. 8. The comparison results are shown in Table 3.

Figure 8 The basic block of CCFN-DBPN.

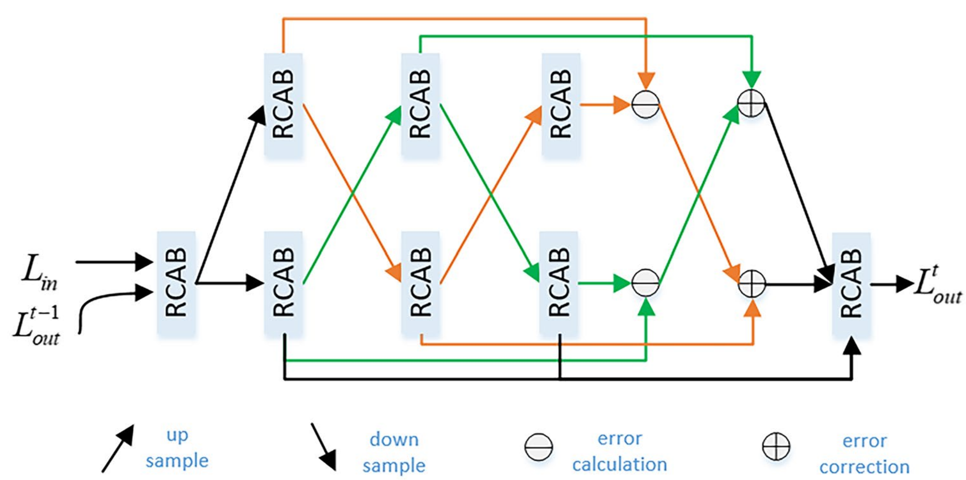


Table 3 Comparison of our cross back-projection and independent back-projection.

Methods	Scale	Params	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
CCFN	×3	612K	34.55/0.9281	30.41/0.8435	29.15/0.8065	28.40/0.8572	33.80/0.9460
CCFN-DBPN		609K	34.49/0.9279	30.40/0.8435	29.13/0.8063	28.42/0.8575	33.77/0.9458

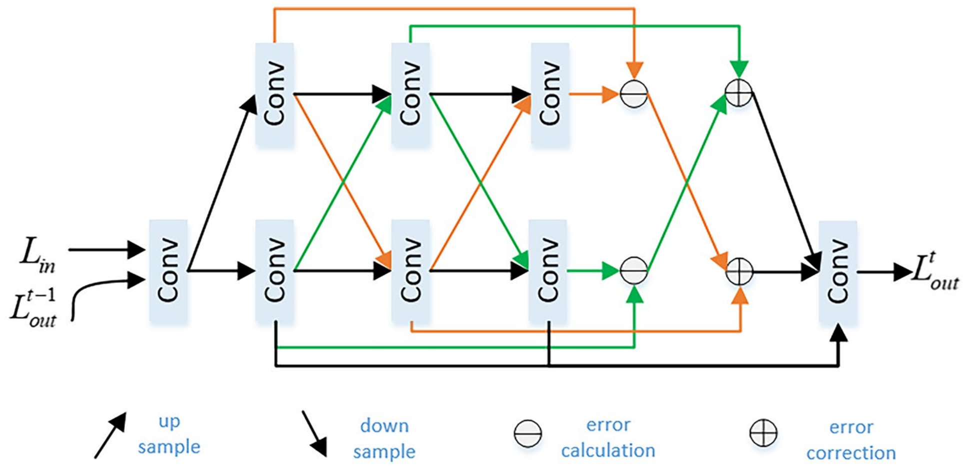


Figure 9 The basic block in CCFN-Conv.

Table 4 Comparison of the residual channel attention block and regular convolution as the basic block of CFB.

Methods	Scale	Params	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
CCFN	×3	612K	34.55/0.9281	30.41/0.8435	29.15/0.8065	28.40/0.8572	33.80/0.9460
CCFN-Conv		548K	34.42/0.9272	30.38/0.8430	29.10/0.8051	28.16/0.8522	33.49/0.9442

5.5 Effectiveness of the Residual Channel Attention Block

In CFB, we use RCAB [21] as the basic block, which integrated channel attention into residual block. Channel attention module can extract the channel statistic to enhance the discriminative ability of the network. To prove the effectiveness of the basic block, we replace it with regular convolution used in current methods (such as HRNet [20], DBPN [19], SRFBN [6] and so on), which is named CCFN-Conv, as shown in Fig. 9. The comparison results are shown in Table 4. From the comparison results, we can

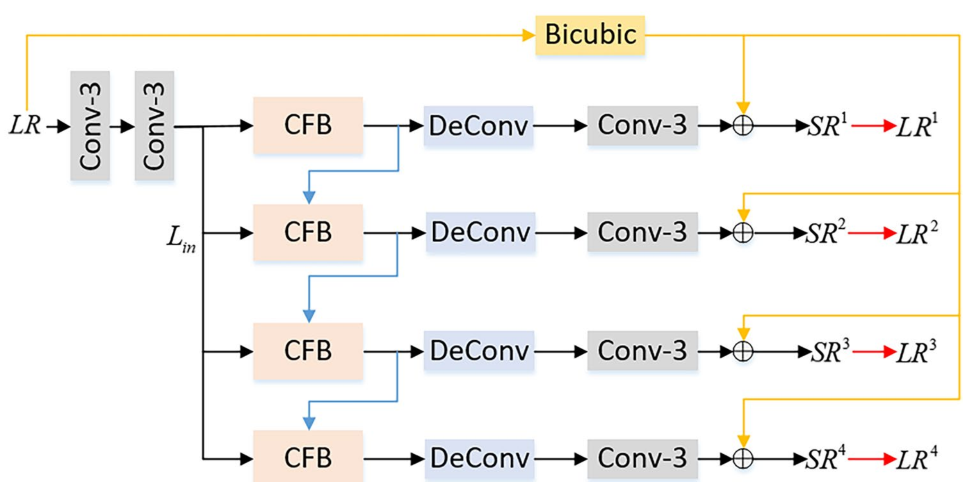


Figure 10 The architecture of CCFN-multi.

Table 5 Comparison of multi-reconstruction and our recursive concatenation reconstruction.

Methods	Scale	Params	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
CCFN	×3	612k	34.55/0.9281	30.41/0.8435	29.15/0.8065	28.40/0.8572	33.80/0.9460
CCFN-multi		613k	34.42/0.9273	30.38/0.8427	29.10/0.8051	28.29/0.8550	33.68/0.9447

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 red and blue, respectively.

Methods	Scale	Params	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 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 [2]		8K	36.66/0.9542	32.45/0.9067	31.36/0.8879	29.50/0.8946	35.60/0.9663
FSRCN [1]		13K	37.00/0.9558	32.63/0.9088	31.53/0.8920	29.88/0.9020	36.67/0.9710
VDSR [3]		666K	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140	37.22/0.9750
DRCN [4]		1774K	37.63/0.9588	33.04/0.9118	31.85/0.8942	30.75/0.9133	37.55/0.9732
LapSRN [22]		251K	37.52/0.9591	32.99/0.9124	31.80/0.8952	30.41/0.9103	37.27/0.9740
DRRN [14]		298K	37.74/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188	37.88/0.9749
MemNet [41]		678K	37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195	37.72/0.9740
SRFBN-S [6]		282K	37.78/0.9597	33.35/0.9156	32.00/0.8970	31.41/0.9207	38.06/0.9757
IDN [23]		553K	37.83/0.9600	33.30/0.9148	32.08/0.8985	31.27/0.9196	38.01/0.9749
EDSR-baseline [12]		1370K	37.99/0.9604	33.57/0.9175	32.16/0.8994	31.98/0.9272	38.54/0.9769
SRMDNF [42]		1511K	37.79/0.9601	33.32/0.9159	32.05/0.8985	31.33/0.9204	38.07/0.9761
CARN [24]		1592K	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256	38.36/0.9765
IMDN [25]		694K	38.00/0.9605	33.63/0.9177	32.19/0.8996	32.17/0.9283	38.88/0.9774
LatticeNet [26]		756K	38.15/0.9610	33.78/0.9193	32.25/0.9005	32.43/0.9302	-/-
SMSR [27]		985K	38.00/0.9601	33.64/0.9179	32.17/0.8990	32.19/0.9284	38.76/0.9771
CCFN(ours)		491K	38.00/0.9606	33.62/0.9186	32.16/0.8995	32.21/0.9289	38.60/0.9770
Bicubic	× 3	-	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349	26.95/0.8556
SRCNN [2]		8K	32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989	30.48/0.9117
FSRCN [1]		13K	33.18/0.9140	29.37/0.8240	28.53/0.7910	26.43/0.8080	31.10/0.9210
VDSR [3]		666K	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279	32.01/0.9340
DRCN [4]		1774K	33.82/0.9226	29.76/0.8311	28.80/0.7963	27.15/0.8276	32.24/0.9343
LapSRN [22]		502K	33.81/0.9220	29.79/0.8325	28.82/0.7980	27.07/0.8275	32.21/0.9350
DRRN [14]		298K	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378	32.71/0.9379
MemNet [41]		678K	34.09/0.9248	30.00/0.8350	28.96/0.8001	27.56/0.8376	32.51/0.9369
SRFBN-S [6]		375K	34.20/0.9255	30.10/0.8372	28.96/0.8010	27.66/0.8415	33.02/0.9404
IDN [23]		553K	34.11/0.9253	29.99/0.8354	28.95/0.8013	27.42/0.8359	32.71/0.9381
EDSR-baseline [12]		1555K	34.37/0.9270	30.28/0.8417	29.09/0.8052	28.15/0.8527	33.45/0.9439
SRMDNF [42]		1528K	34.12/0.9254	30.04/0.8382	28.97/0.8025	27.57/0.8398	33.00/0.9403
CARN [24]		1592K	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493	33.50/0.9440
IMDN [25]		703K	34.36/0.9270	30.32/0.8417	29.09/0.8046	28.17/0.8519	33.61/0.9445
LatticeNet [26]		765K	34.53/0.9281	30.39/0.8424	29.15/0.8059	28.33/0.8538	-/-
SMSR [27]		993K	34.40/0.9270	30.33/0.8412	29.10/0.8050	28.25/0.8536	33.68/0.9445
CCFN(ours)		612K	34.55/0.9281	30.41/0.8435	29.15/0.8065	28.40/0.8572	33.80/0.9460
Bicubic	× 4	-	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866
SRCNN [2]		8K	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221	27.58/0.8555
FSRCN [1]		13K	30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280	27.90/0.8610
VDSR [3]		666K	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524	28.83/0.8870
DRCN [4]		1774K	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510	28.93/0.8854
LapSRN [22]		502K	31.54/0.8852	28.09/0.7700	27.32/0.7275	25.21/0.7562	29.09/0.8900
DRRN [14]		298K	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638	29.45/0.8946
MemNet [41]		678K	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630	29.42/0.8942
SRFBN-S [6]		483K	31.98/0.8923	28.45/0.7779	27.44/0.7313	25.71/0.7719	29.91/0.9008
IDN [23]		553K	31.82/0.8903	28.25/0.7730	27.41/0.7297	25.41/0.7632	29.41/0.8942
EDSR-baseline [12]		1518K	32.09/0.8938	28.58/0.7813	27.57/0.7357	26.04/0.7849	30.35/0.9067
SRMDNF [42]		1552K	31.96/0.8925	28.35/0.7787	27.49/0.7337	25.68/0.7731	30.09/0.9024
CARN [24]		1592K	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	30.47/0.9084
IMDN [25]		715K	32.21/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838	30.45/0.9075

Table 6 (continued)

Methods	Scale	Params	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
LatticeNet [26]		777K	32.30/0.8962	28.68/0.7830	27.62/0.7367	26.25/0.7873	-/-
SMSR [27]		1006K	32.12/0.8932	28.55/0.7808	27.55/0.7351	26.11/0.7868	30.54/0.9085
CCFN(ours)		752K	32.34/0.8964	28.72/0.7847	27.63/0.7381	26.28/0.7919	30.72/0.9112

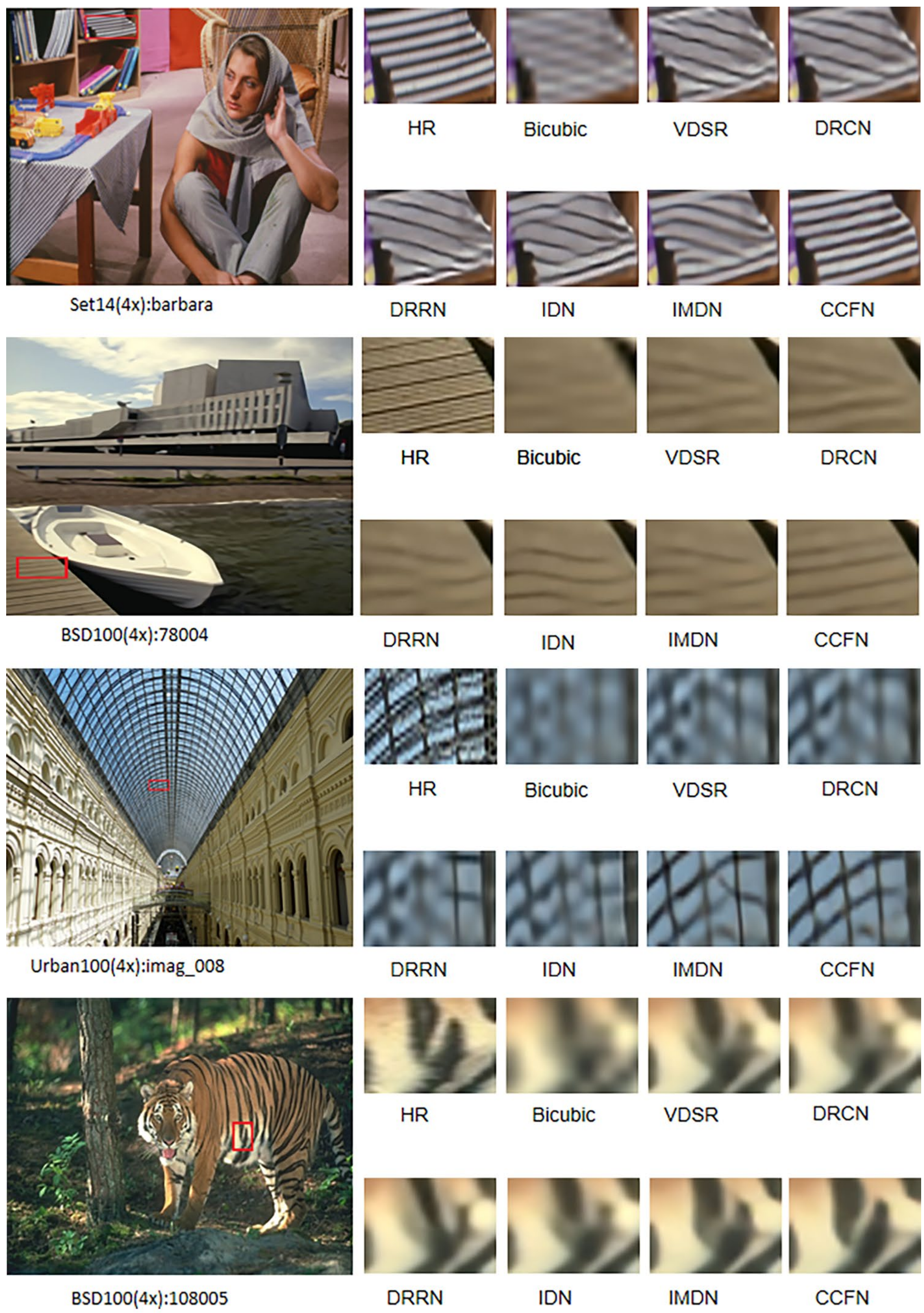


Figure 11 Visual comparisons of our CCFN with other SR methods on Set14, BSD100 and Urban100 datasets.

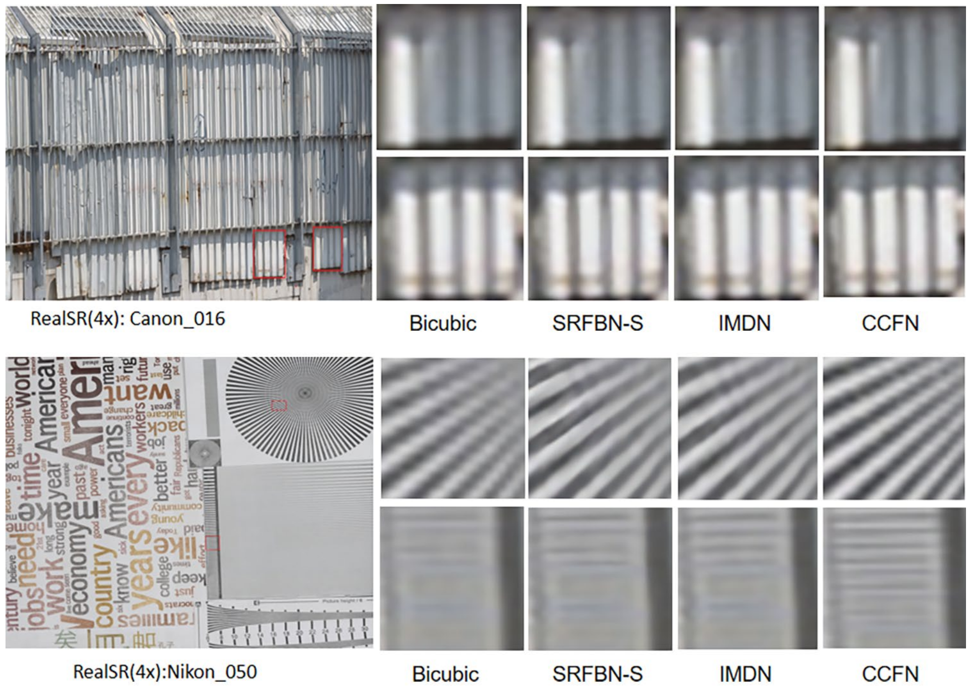


Figure 12 Visual comparisons of our CCFN with other SR methods on RealSR [39] datasets.

find that, residual channel attention block can improve the performance of our method significantly.

5.6 Comparison of Recursive Concatenation and the Existing Multi-reconstruction

In most of the existing iterative SR methods (such as SRFBN [6], DRCN [4]), SR image is reconstructed at each iteration. Since our CCFN is a feedback network, which works in an iterative manner to guide LR images to recover better SR images, the previous iterations are less informative for the reconstruction. Therefore, we use concatenation methods to replace the multi-reconstruction methods used in existing feedback methods, and reconstruct SR image at the last iteration (see Fig. 3). To validate this motivation, we use multi-reconstruction on our network, which is named CCFN-multi, as shown in Fig. 10, and the comparison results are shown in Table 5. From the results, we can find that, our concatenation method is more suitable for feedback networks than the multi-reconstruction method used in existing methods.

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

As a lightweight network, we compare our method with other lightweight state-of-the-art methods on classic simulated datasets Set5, Set 14, BSD100, Urban100 and Manga109. Other lightweight state-of-the-art methods are SRCNN [2], FSRCNN [1], VDSR [3], DRCN [4], LapSRN [22], DRRN [14], MemNet [41], IDN [23], EDSR-baseline [12], SRMDNF [42], CARN [24], IMDN [25], LatticeNet [26] proposed in 2020, and SMSR [27] proposed in 2021. We compare the PSNR and SSIM values, and the comparison results are shown in Table 6. We can find that our CCFN has less parameters, but better performance than other state-of-the-art lightweight networks with the scale factors of ×3 and ×4. When the scale factor is ×2, our method takes second place, but with many fewer parameters. Therefore, our method achieves an outstanding performance compared to the state-of-the-art methods.

The visual comparisons of the SR results on ×4 are shown in Fig. 11. From the comparison results, we can find that our method recovers the finer textures better than the others. All the comparison results prove the effectiveness of our method.

Furtherly, we test our method by the real-world datasets RealSR [39], which is paired real-world images captured by digital cameras with different focal length. The visual comparisons are shown in Fig. 12, we can find that, our method still has a better performance than other methods on real-world image reconstruction.

6 Conclusion

In this paper, we propose a closed-loop feedback network with cross back-projection for lightweight image super-resolution (CCFN). The CCFN is a lightweight SR network, which uses feedback mechanism in three manners: error feedback, self-feedback and global feedback. First, based on error feedback and multi-scale fusion, we propose a cross back-projetion feedback block (CFB). In CFB, we use error feedback to correct the features of multi-scale fusion, which also can be viewed as two cross-learning back-projection units. The cross back-projetion has a better performance than independent back-projection units and multi-scale fusion. Second, CFB works in a self-feedback manner, which feeds back the high-level features from output to refine the shallow ones as input. Third, we propose a global feedback to guide the learning of mapping functions from LR to HR by feeding back the degradation results of SR to LR. We use attention-based model as the basic block in CFB, which improved the discriminative ability of the network. Finally, since our feedback network works in an iterative manner and high-level features are more informative for reconstruction, recursive concatenation is more suitable than the multi-reconstruction used in existing literatures. All the methodologies we proposed are proven to improve the network performance. Further experimental results show that, the CCFN we proposed has an outstanding performance with few parameters. Our CCFN contains only one self-feedback block. We conjecture that complex feedback networks with double or more self-feedback blocks may have a better performance, the self-feedback blocks in which can work in a synchronous or asynchronous feedback manners. We will try more later.

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 Interests/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, C. C. L., & Tang, X. (2016). Accelerating the super-resolution convolutional neural network. In *Computer Vision - ECCV 2016* (pp. 391–407). https://doi.org/10.1007/978-3-319-46475-6_25
2. Chao Dong, K. H., Chen Change Loy, & Tang, X. (2014). Learning a deep convolutional network for image super-resolution. In *Computer Vision - ECCV 2014* (pp. 184–199). volume 8692. https://doi.org/10.1007/978-3-319-10593-2_13
3. 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). https://doi.org/10.1109/CVPR.2016.182
4. Kim, J., Lee, J. K., & Lee, K. M. (2016). Deeply-recursive convolutional network for image super-resolution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1637–1645). https://doi.org/10.1109/CVPR.2016.181
5. 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). https://doi.org/10.1109/CVPR.2017.19
6. Li, Z., Li, Q., Wu, W., Wu, Z., Lu, L., & Yang, X. (2020). Clustering based multiple branches deep networks for single image super-resolution. *Multimedia Tools and Applications*, *79*. https://doi.org/10.1007/s11042-018-7017-y
7. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., 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). https://doi.org/10.1109/CVPR.2016.207
8. Yang, X., Wu, W., Lu, L., Yan, B., Zhang, L., & Liu, K. (2020). Multiple regressions based image super-resolution. *Multimedia Tools and Applications*, *79*. https://doi.org/10.1007/s11042-019-7716-z
9. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
10. Szegedy, C., Wei Liu, Yangqing Jia, 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). https://doi.org/10.1109/CVPR.2015.7298594
11. 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). https://doi.org/10.1109/CVPR.2016.90
12. Lim, B., Son, S., Kim, H., Nah, S., & Lee, K. M. (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). https://doi.org/10.1109/CVPRW.2017.151
13. Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. (2017). Densely connected convolutional networks. https://doi.org/10.1109/CVPR.2017.243
14. Tai, Y., Yang, J., & Liu, X. (2017a). Image super-resolution via deep recursive residual network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2790–2798). https://doi.org/10.1109/CVPR.2017.298
15. Guo, Y., Chen, J., Wang, J., Chen, Q., Cao, J., Deng, Z., Xu, Y., & Tan, M. (2020). Closed-loop matters: Dual regression networks for single image super-resolution. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5406–5415). https://doi.org/10.1109/CVPR42600.2020.00545
16. Gilbert, C. D., & Sigman, M. (2007). Brain states: Top-down influences in sensory processing. In *Neuron* (p. 677–696). volume sps54. https://doi.org/10.1016/j.neuron.2007.05.019
17. Stollenga, M., Masci, J., Gomez, F., & Schmidhuber, J. (2014). Deep networks with internal selective attention through feedback connections. volume 4.
18. Zamir, A. R., Wu, T.-L., Sun, L., Shen, W., Shi, B. E., Malik, J., & Savarese, S. (2016). Feedback networks. In *Computer Science - Computer Vision and Pattern Recognition*.
19. 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). https://doi.org/10.1109/CVPR.2018.00179
20. Sun, K., Xiao, B., Liu, D., & Wang, J. (2019). Deep High-Resolution Representation Learning for Human Pose Estimation. *arXiv e-prints*, (p. arXiv:1902.09212).

21. 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* (pp. 294–310). volume 11211. https://doi.org/10.1007/978-3-030-01234-2_18
22. 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*. https://doi.org/10.1109/CVPR.2017.618
23. 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). https://doi.org/10.1145/3343031.3351084
24. Namhyuk Ahn, B. K., & Sohn, K.-A. (2018). Fast, accurate, and lightweight super-resolution with cascading residual network. In *Computer Vision - ECCV 2018* (pp. 256–272). volume 11214. https://doi.org/10.1007/978-3-030-01249-6_16
25. 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). https://doi.org/10.1109/CVPR.2018.00082
26. Xie, Y., Zhang, Y., Qu, Y., Li, C., & Fu, Y. (2020). Latticenet: Towards lightweight image super-resolution with lattice block. (pp. 272–289). https://doi.org/10.1007/978-3-030-58542-6_17
27. Wang, L., Dong, X., Wang, Y., Ying, X., Lin, Z., An, W., & Guo, Y. (2021). Exploring sparsity in image super-resolution for efficient inference. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4915–4924). https://doi.org/10.1109/CVPR46437.2021.00488
28. Irani, M., & Peleg, S. (2002). Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, *4*. https://doi.org/10.1006/jvci.1993.1030
29. Dai, S., Han, M., Wu, Y., & Gong, Y. (2007). Bilateral back-projection for single image super resolution. (pp. 1039 – 1042). https://doi.org/10.1109/ICME.2007.4284831
30. Dong, W., Zhang, L., Shi, G., & Wu, X. (2009). Nonlocal back-projection for adaptive image enlargement. In *2009 16th IEEE International Conference on Image Processing (ICIP)* (pp. 349–352). https://doi.org/10.1109/ICIP.2009.5414423
31. Timofte, R., Rothe, R., & Van Gool, L. (2016). Seven ways to improve example-based single image super resolution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1865–1873). https://doi.org/10.1109/CVPR.2016.206
32. Fu, S., Li, Z., Liu, K., Din, S., Imran, M., & Yang, X. (2020). Model compression for iot applications in industry 4.0 via multiscale knowledge transfer. *IEEE Transactions on Industrial Informatics*, *16*, 6013–6022. https://doi.org/10.1109/TII.2019.2953106
33. 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)*.
34. Shipeng Fu, H. L. Z. L. W. W. A. P. G. J. X. Y., Lu Lu (2019). A real-time super-resolution method based on convolutional neural networks. In *Circuits, Systems, and Signal Processing*. https://doi.org/10.1007/s00034-019-01283-y
35. Zhen Li, W. W. J. Y. Z. L. X. Y., Qilei Li (2020). Deep recursive up-down sampling networks for single image super-resolution. In *Neurocomputing* (pp. 377–388). volume 398. https://doi.org/10.1016/j.neucom.2019.04.004
36. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2018). Residual dense network for image super-resolution. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2472–2481). https://doi.org/10.1109/CVPR.2018.00262
37. 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). https://doi.org/10.1109/CVPR.2016.512
38. Agustsson, E., & Timofte, R. (2017). Ntire 2017 challenge on single image super-resolution: Dataset and study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 1122–1131). https://doi.org/10.1109/CVPRW.2017.150
39. Cai, J., Zeng, H., Yong, H., Cao, Z., & Zhang, L. (2019). Toward real-world single image super-resolution: A new benchmark and a new model. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 3086–3095). https://doi.org/10.1109/ICCV.2019.00318
40. Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations.*
41. Tai, Y., Yang, J., Liu, X., & Xu, C. (2017b). Memnet: A persistent memory network for image restoration. In *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 4549–4557). https://doi.org/10.1109/ICCV.2017.486
42. Zhang, K., Zuo, W., & Zhang, L. (2018). Learning a single convolutional super-resolution network for multiple degradations. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3262–3271). https://doi.org/10.1109/CVPR.2018.00344