Computers and Electrical Engineering 102 (2022) 108254

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Lightweight bidirectional feedback network for image super-resolution☆

Beibei Wang [a], Binyu Yan [a,*], Changjun Liu [a], Ryul Hwangbo [b], Gwanggil Jeon [c], Xiaomin Yang [a]

[a] College of Electronic and Information Engineering, Sichuan University, Chengdu, China
[b] ISPark Co., Ltd., Republic of Korea
[c] Department of Embedded Systems Engineering, Incheon National University, Incheon 22012, Republic of Korea

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
Super-resolution
Cross-learning
Regression loss
Bidirectional feedback
Lightweight

ABSTRACT

Super-resolution has attracted academic attention recently, for its capabilities of image restoration and image enhancement. To generate informative high-level features for a better reconstruction performance, most super-resolution networks have many parameters, which limit their application in resource-constrained devices. Feedback networks can generate informative high-level features with few parameters by feeding high-level features back to previous layers. In this paper, we propose a lightweight bidirectional feedback network for image super-resolution (LBFN), which consists of two feedback procedures connected in reverse. Bidirectional feedback architecture further improves the perceptual abilities of feedback networks by fusing different level features sufficiently. We propose a residual attention block to enhance the detail expression ability of feature maps, which are cross-learned in feedback block. Finally, we propose a SR regression loss to supervise the training of our network. Extensive experiments demonstrate that our method has an outstanding performance while taking up little computing resources.

1. Introduction

The purpose of single image super-resolution (SISR) network is to reconstruct high-resolution (HR) images from the corresponding low-resolution (LR) images. However, SISR is an ill-posed problem, for innumerable HR images can be degraded to the same LR image. To solve this problem, many classical SR methods have been proposed [1–9].

Since dong et al. introduced deep learning into super-resolution (SR) in SRCNN [1], learning-based SISR has entered a period of rapid development. However, SRCNN [1] took upsampled LR image as input, which has high computational complexity. Then dong et al. proposed FSRCNN [2], which learned LR features and used deconvolution on the last layer to reduce the amount of calculation. To get rid of the redundant of deconvolution, subpixel convolutional layer was proposed in ESPCN [10] by shi et al. which generated HR feature maps by rearranging the pixels of LR feature maps after channel expansion.

To further improve the performance of SR networks, deep neural networks were proposed, for high-level features are more informative to achieve a better reconstruction performance. VDSR [11] proposed by Kim et al. was a deep network with 20 layers, which took interpolated LR images with multiple scales as inputs and introduced residual learning into SR methods. Then

☆ This paper is for special section VSI-iotl. Reviews were processed by Guest Editor Dr. Shiping Wen and recommended for publication.

* Corresponding author.
E-mail addresses: 690983790@qq.com (B. Wang), Yanby@scu.edu.cn (B. Yan), cjliu@scu.edu.cn (C. Liu), hbyul@ispark.kr (R. Hwangbo), ggjeon@gmail.com (G. Jeon), arielyang@scu.edu.cn (X. Yang).

https://doi.org/10.1016/j.compeleceng.2022.108254
Received 27 April 2022; Received in revised form 11 July 2022; Accepted 18 July 2022
Available online 29 July 2022

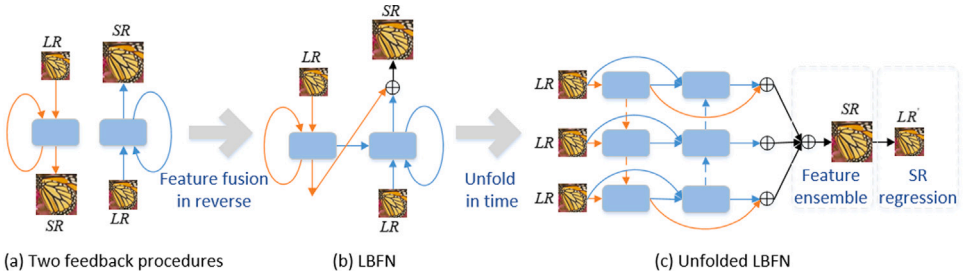

Fig. 1. The architecture of our LBFN. Orange arrows represent the forward feedback procedure, and blue arrows represent the reverse feedback procedure.

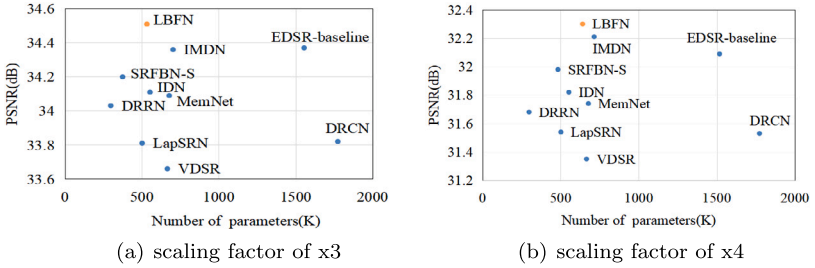

Fig. 2. PSNR vs. number of parameters on Set5 dataset.

they proposed DRCN [12], which was a deep recursive network with multiple supervision. The recursive mechanism reduced the parameters required for deep networks.

However, deep networks are difficult to train, and easy to cause gradient vanishing/exploding problems. Residual learning was fully utilized by He et al. in ResNet [13], which worked well in deep networks, for it can solve the problem of network degradation. Then SRDenseNet [14] was proposed, which sent the previous features to all subsequent layers. Tai et al. proposed DRRN [15], which was a deep recursive network with residual learning. EDSR [9] proposed by Lee et al. removed the usage of Batch Normalization (BN) in residual unit, for which is not suitable for SR tasks.

The networks mentioned above are all worked in a feedforward manner, which learn the non-linear mapping from low-level inputs to high-level outputs. Feedback mechanism makes it possible for the low-level inputs to get useful information from high-level outputs, which can improve the reconstruction performance. Hairs et al. first introduced feedback mechanism into SR in DBPN [6], which fed back the errors to realize self-correction. DBPN [6] proved the effectiveness of feedback mechanism in minimizing the reconstruction errors. Then SRFBN [4] was proposed by Li et al. which fed back the output of the feedback block to its input in next iteration. SRFBN [4] realized the refinement of high-level features to low-level ones with multi-reconstruction manner.

To achieve a satisfactory expressive ability, most SR networks have a large number of parameters, which restrict their application in embedded devices. Recently, lightweight networks have gained a lot attention for their wide application prospects. All the lightweight SR networks focus on getting a better performance with fewer parameters. Feedback mechanism can improve the performance of the networks without adding new parameters. Although feedback mechanism was introduced into SR, which has not been fully exploited in lightweight SR methods. Inspired by the good performance of bidirectional architecture in BRNN (Bidirectional Recurrent Neural Network), we propose a motivation that it can also do well in feedback networks. Bidirectional architecture solved the problem that the elements of previous sequence are not aware of the outputs of subsequent sequence in BRNN, and we think bidirectional feedback architecture can improve the performance of the feedback network by perceiving features of different levels, which has not been used in lightweight SR networks. In this paper, we propose a lightweight bidirectional feedback network for image super-resolution (LBFN). We use bidirectional architecture, so our LBFN consists of two feedback procedures. On one hand, the low-level features can be refined by hight-level ones in each feedback procedure. On the other hand, since the two feedback procedures are bidirectional, the shallow layers of the reverse procedure can get high-level information from the forward procedure, and the deep layers of the reverse procedure can get low-level information from the forward procedure. Different level features are fused sufficiently to improve the perceptual abilities of the feedback network. Furtherly, we use ensemble method to fuse the last high-level features of all iterations to reconstruct SR image, which achieves a better SR reconstruction performance. The structure of LBFN is shown in Fig. 1.

As a feedback network, LBFN can be unfolded in time. Then as a lightweight network, we study the number of iterations (denoted as T) and the base number of filters (denoted as m) to achieve a satisfying trade-off between performance and computational complexity. Furtherly, since attention module can enhance the discriminate capability of the network with little computational complexity, we integrate attention modules into residual blocks, which are cross-learned in feedback block. Our LBFN achieves an outstanding performance compared with other state-of-the-art lightweight networks, which is shown in Fig. 2.

The contributions of LBFN are as follows:

- To further improve the perceptual abilities of the feedback network, we propose a bidirectional feedback architecture. The forward feedback procedure is from top to bottom, and the reverse feedback procedure is from bottom to top. High-level features are fed back to correct low-level features in each feedback procedure, at the same time, the reverse feedback procedure can get useful information from the forward feedback procedure. Therefore the performance of the feedback network is improved.
- To achieve a powerful HR representation, we propose a cross-learning feedback block (CFB) as the basic feedback block. The HR flow and LR flow in CFB cross learn from each other to better learn the relationship between HR and LR features.
- To improve the detail expression ability of feature maps, we propose a residual attention block (RAB) as the basic block in CFB. RAB enhances the representation ability of CFB by integrating channels and pixels enhancement into residual blocks.
- To better learn the mapping function from LR to HR, we propose a SR regression loss, which degrades the SR results back to LR. The SR regression loss and the primal SR loss are used together to constrain the solution space of LR-HR mapping function, which can significantly improve the performance of lightweight networks but introduces very few parameters.

In the following sections, the related work is described at first in Section 2. Next, our mehtod is described in detail in Section 3. Then, the experimental details and results are shown in Section 4. At last, we make a conclusion of our method in Section 5.

2. Related work

2.1. Attention mechanism

Attention module was used in human perception earlier years [16,17]. Then RAN [18] was proposed for classification tasks. Squeeze-and-excitation (SE) module [19] was proposed to learn the relationship between channels, which assigned different weights to channels according to the amount of information they contained. Recently, residual channel attention blocks (RCABs) were proposed by Zhang et al. in RCAN [3], which integrated channel attention into residual blocks, and achieved remarkable performance improvements. In CBAM [20], the channel attention and spatial attention are used together for adaptive feature refinement. Hui et al. proposed the contrast-aware channel attention (CCA) module in IMDN [21], which used the summation of standard deviation and average pooling to represent the global information contained in the feature maps.

RCAB [3] integrated channel attention into residual blocks and achieved remarkable performance improvements. Inspired by RCAB [3], we propose a motivation that, spatial attention modules can enhance the discriminate ability in spatial dimension, which also should be integrated into residual blocks. Therefore, we propose a residual attention block (RAB), which integrates channel attention and spatial attention into residual blocks to achieve a better performance.

2.2. Recursive network

Recursive networks are the generalization of recurrent networks, and recurrent networks are the special type of recursive networks. DRCN [12] proposed by Kim et al. introduced recursive block into SR, which was a deep recursive network with multiple supervision. Recursive blocks shared the same weights to reduce the parameters usage. Then Tai et al. proposed DRRN [15], which is a deep recursive network with skip connections. Later, they proposed MemNet [22], which used recursive residual unit and dense residual learning to improve the reconstruction performance. Inspired by BRNN, we propose a bidirectional feedback network. BRNN solved the problem that the elements of previous sequence are not aware of the outputs of subsequent sequence. Our LBFN solves the problem that the low-level features of previous layers cannot be refined by high-level ones of subsequent layers.

2.3. Feedback mechanism

Most of the neural networks are feedforward networks [1,2,8], which send the shallow features to latter layers in direct or skip connection manners. Then feedback mechanism was proposed, which can send high-level features back to shallow layers to enhance the representation ability of shallow features. Feedback mechanism has been used in many computer vision methods. DBPN [6] first introduced feedback mechanism into SR, which proposed an error feedback to realize self-correction in up- and down-projection units. Then DSRN [23] was proposed, which used delayed feedback to exchange recurrent signals between LR and HR states. Li et al. proposed SRFBN [4], which is a feedback network similar to RNN (Recurrent Neural Network).

SRFBN [4] is the most relevant network to our LBFN, and we differ from it in three main ways. First, SRFBN [4] is a feedback network in one direction, but we cross connect two feedback procedures in reverse to further improve the performance of the feedback network by perceiving features of different levels. Second, SRFBN [4] reconstructed an SR image in each iteration and took the last one as final SR image, but we reconstruct the final SR image by feature ensemble method to make full use of the HR features of different levels, which ties the loss to all iterations so that the hidden states carry a notion of high-level features. Finally, the training of SRFBN [4] was supervised by primal SR loss, but we use primal SR loss and SR regression loss to supervise the learning of the mapping function from LR to HR. Further experimental results show that, our bidirectional feedback is better than the feedback architecture proposed in SRFBN [4].

3. Our method

In this section, the network architecture of LBFN is described at first. Next, CFB as the basic feedback block of LBFN is described. Then, RAB as the basic block in CFB is described in detail. At last, the loss function with SR regression loss is described.

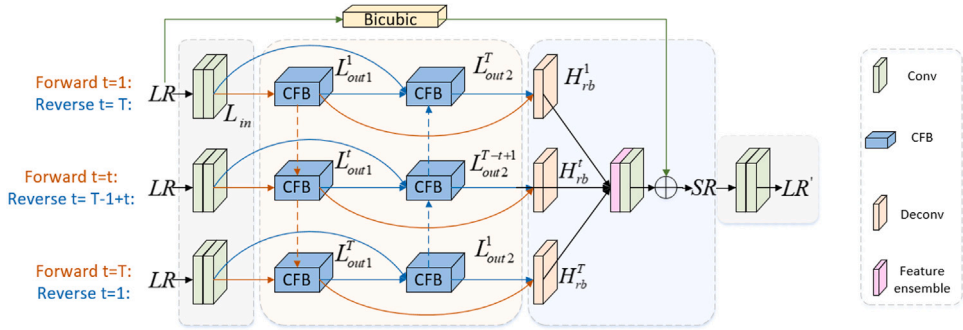

Fig. 3. The architecture of our LBFN. Orange arrows represent the forward feedback procedure, and blue arrows represent the reverse feedback procedure.

3.1. Network structure

Our LBFN consists of two feedback procedures connected in reverse, so it can be unfolded to T iterations, as shown in Fig. 3. Orange arrows represent the forward feedback procedure from top to bottom, blue arrows represent the reverse feedback procedure from bottom to top. The low-level features of the two feedback procedures can be refined by their own hight-level ones. Furtherly, since the two feedback procedures are bidirectional, the features of the two procedures are fused reversely. bidirectional architecture improved RNN greatly and broadened the application scope of RNN. As it did in RNN, bidirectional architecture also improves the performance of the feedback network by perceiving features of different levels.

LBFN contains four parts: shallow feature extraction part, bidirectional feedback part, reconstruction part and SR regression part. In the first part, shallow feature extraction part extracts shallow features, which are passed to CFB. In the second part, two feedback procedures are bidirectional, which use CFB as the basic feedback block. The CFB in forward feedback procedure has two inputs, the shallow features and the output of CFB from last iteration. The CFB in the reverse feedback procedure has three inputs, the shallow features, the output of CFB from last iteration and the output of CFB from the forward feedback procedure. Then, in reconstruction part, the corresponding results of the two feedback procedures are concatenated and upsampled to generate the final HR features in each iteration. Then the ensemble of final HR features from all iterations are used to reconstruct SR image by adding bicubic upsampling results. Finally, in SR regression part, the SR image degrades to LR image to calculate SR regression loss, so our LBFN can be supervised by both the SR regression loss and the primal SR loss.

We define L_{in} is the output of shallow feature extraction part, which can be obtained by:

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

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

In bidirectional feedback part, there are two bidirectional feedback procedures. In the forward feedback procedure, we define the output of CFB in the tth iteration is L^t_{out1}. In the reverse feedback procedure, we define the output of CFB in the tth iteration is L^t_{out2}. The functions is as follows:

$$L^t_{out1} = \left\{ \begin{array}{ll} f_{CFB}(L_{in}) & t = 1 \\ f_{CFB}([L_{in}, L^{t-1}_{out1}]) & t \geq 2 \end{array} \right\}, \tag{2}$$

$$L^t_{out2} = \left\{ \begin{array}{ll} f_{CFB}([L_{in}, L^{T}_{out1}]) & t = 1 \\ f_{CFB}([L_{in}, L^{t-1}_{out2}, L^{T-t+1}_{out1}]) & t \geq 2 \end{array} \right\}, \tag{3}$$

where f_{CFB} is the operations of CFB.

In reconstruction part, we concat the corresponding outputs of two feedback procedures, which are then upsampled by deconvolutional layer in each iteration. We define the result after deconvolutional upsampling in the forward tth iteration as H^t_{rb}, the ensemble result of which is used to reconstruct SR image by adding bicubic upsampling result. The process is as follows:

$$H^t_{rb} = f_{up}([L^t_{out1}, L^{T-t+1}_{out2}]). \tag{4}$$

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

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

In SR regression part, we generate LR' by downsampling operation f_{down}, which consists of a downsampling layer conv(3,32) and a channel compress layer conv(32,3). LR' and the primal LR are used to calculate the SR regression loss, which supervises the training of our LBFN together with the primal SR loss.

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

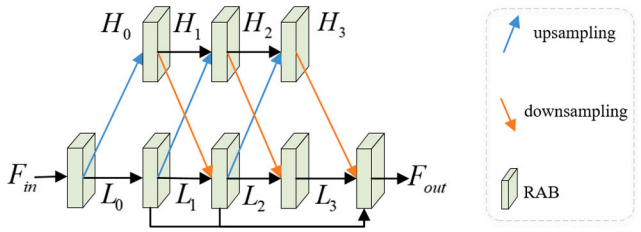

Fig. 4. Cross-learning feedback block (CFB).

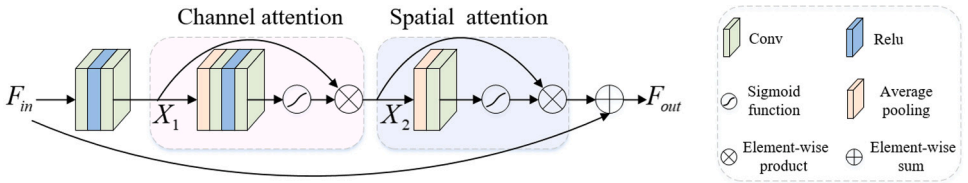

Fig. 5. Residual attention block (RAB).

3.2. Cross-learning feedback block (CFB)

In our feedback network, we propose a cross-learning feedback block (CFB) as the basic feedback block, 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. To further improve the performance of our CFB, we use residual attention block (RAB) as each basic block.

Since our LFBN has two feedback procedures connected in reverse, the input of CFB is different, as shown in Eqs. (2) and (3). The CFB in forward feedback procedure has two inputs, the shallow features and the output of CFB from last iteration. The CFB in the reverse feedback procedure has three inputs, the shallow features, the output of CFB from last iteration and the output of CFB from the forward feedback procedure.

In CFB, we use cross-learning twice. We define the HR features in HR flow are H_0, H_1, H_2, H_3, respectively, and define the LR features in LR flow are L_0, L_1, L_2, L_3, respectively. The process of CFB is as follows:

$$\left\{ \begin{array}{l} L_0 = f_{RAB}(F_{in}) \\ H_0 = f_{up}(L_0) \end{array} \right\}, \tag{7}$$

$$\left\{ \begin{array}{l} L_1 = f_{RAB}(L_0) \\ H_1 = f_{RAB}(H_0) \end{array} \right\}, \tag{8}$$

$$\left\{ \begin{array}{ll} L_g = f_{RAB}([L_{g-1}, f_{down}(H_{g-1})]) & 2 \le g \le 3 \\ H_g = f_{RAB}([H_{g-1}, f_{up}(L_{g-1})]) & 2 \le g \le 3 \end{array} \right\}, \tag{9}$$

where f_{RAB} is the operations of the basic block RAB. f_{up} and f_{down} are the deconvolutional upsampling operation and convolutional downsampling operation, respectively. [] is the concatenation operation.

Finally, we downscale the output of the HR feature flow, and fuse it with LR features from the LR feature flow by the final basic block. Therefore, the output of CFB can be obtained by:

$$F_{out} = f_{RAB}([L_1, L_2, L_3, f_{down}(H_3)]) \ . \tag{10}$$

3.3. Residual attention block (RAB)

To improve the performance of CFB, we propose residual attention block (RAB) as the basic block in CFB, which is inpired by RCAB [3]. RCAB [3] integrated channel attention into residual blocks. Feature channels with more information will be paid more attention for the differences between them are magnified. However, pixels in different spatial location are also different, which represent the information of edges, texture, and image content. The difference between pixels is very important for the restoration of image details. Therefore, we propose residual attention block (RAB) as the basic block in CFB, which is shown in Fig. 5. We use average-pooling to calculate channel attention (CA) and spatial attention (SA), which represents the average amount of information they contain in channel dimension and spatial dimension. CA pay different attention to channels, and SA pay different attention to pixels in different spatial locations. Then we integrate CA and SA into residual block as the basic block in CFB. The process of RAB is as follows:

$$X_1 = f_{c1}(\delta(f_{c2}(X_{in}))), \tag{11}$$

Table 1
Study of T with m = 32 for 3 x SR on Set5.

T	Runtime (s)	Set5
		PSNR/SSIM
1	0.0387	34.30/0.9264
2	0.0494	34.51/0.9282
3	0.0632	34.53/0.9283

Table 2
Study of m with T = 2 for 3 x SR on Set5.

m	Params	Flops	Set5
			PSNR/SSIM
16	136k	14.09G	34.08/0.9249
32	533k	54.08G	34.51/0.9282
48	1187k	120.07G	34.57/0.9285

where f_{c1} and f_{c2} are two stacked convolution operation with a kernel size of 3. δ represents the ReLU function.

$$X_2 = X_1 * \sigma f_{MLP}(f_{AvgPool}(X_1)), \tag{12}$$

where $f_{AvgPool}(X_1) \in R^{C*1*1}$. f_{MLP} is the operation of a multilayer perceptron with a hidden layer. σ represents the sigmoid function.

$$F_{out} = X_{in} + X_2 * \sigma(f^{7\times7}(f_{AvgPool}(X_2))), \tag{13}$$

where $f_{AvgPool}(X_2) \in R^{1*H*W}$. $f^{7\times7}$ is a convolution operation with a kernel size of 7.

3.4. Loss function

Since the primal LR image is degenerated from HR image, we proposed a motivation that, if the SR image is the optimal solution, it is also can be downsampled to obtain the primal LR image. Therefore, we propose a SR regression loss in our network. The loss of our LBFN contains two parts: the primal SR loss calculated by HR and SR images, and the SR regression loss calculated by LR and LR', which is the degradation result of SR. Then since SR regression loss is an assisted supervision, which should have a reasonable propotion in the loss function, we use θ to control the weight of SR regression loss. The loss function is as follows:

$$Loss = L_1(SR, HR) + \theta L_1(LR', LR), \tag{14}$$

where L_1 represents the L1 loss function.

4. Experimental results

4.1. Experimental details

The DIV2k datasets are used to train our LBFN, which contains 800 HR images. We double the number of images by 90 degree rotation, which are then expanded to 8000 by random cropping. Then we downscale them by bicubic interpolation to generate LR images. We use five images from Set5 dataset as the validation images. 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. At last, we perform test on 328 images from five benchmark datasets: Set5, Set14, BSD100, Urban100 and Manga109. All the experiments are performed on GPU with PyTorch framework.

4.2. Study of T and m

As a lightweight network, we study the number of iterations of each feedback procedure T and base number of filters m to get a best trade-off between performance and operations. Considering large T or m will increase the computational complexity, the base number of filters m is set to 32 at first, and the iterations T increases from 1 to 3, as shown in Table 1. LBFN with T = 1 has a poor performance, and LBFN with T = 3 has a better performance but more runtime, so that we set T = 2 in out LBFN. Then we set T = 2, and the base number of filters m increases from 16 to 48, as shown in Table 2. Flops is tested on an LR image of size 256 × 256. LBFN with m = 16 has a poor performance, and LBFN with m = 48 has a large number of operations and parameters, so that we set m = 32 in out LBFN. As a lightweight network, our LBFN has a best trade-off between performance and operations when T = 2 and m = 32. Therefore, we set T = 2 and m = 32 in our LBFN.

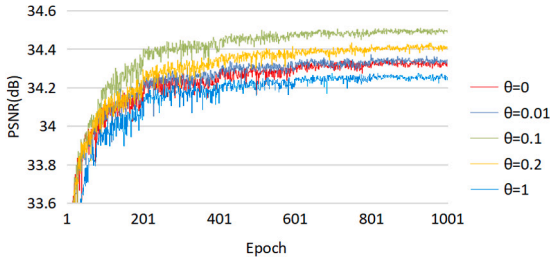

Fig. 6. Comparisons of different weights of the SR regression loss.

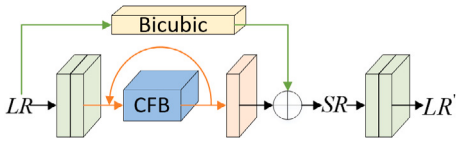

Fig. 7. Unidirectional feedback method used on our LBFN.

Table 3
Comparison of the traditional unidirectional feedback and our bidirectional on LBFN for 3 x SR.

Feedback methods	Params	Set5	Set14	BSD100	Urban100	Manga109
Unidirectional feedback	532k	34.49/0.9278	30.38/0.8434	29.11/0.8057	28.30/0.8552	33.66/0.9451
Bidirectional feedback	533k	34.51/0.9282	30.45/0.8442	29.13/0.8059	28.36/0.8564	33.80/0.9456

4.3. Effect of SR regression loss

We propose a SR regression loss to better learn the mapping function from LR to HR. Since SR regression loss is an assisted supervision, the propotion of it should not higher than the primal SR loss. We use θ to control the weight of SR regression loss, as shown in Eq. (14), which is changed from 0 to 1 to get a better trade-off. When $\theta = 0$, the results represent the ablation study of SR regression loss. From the comparison results in Fig. 6, we can find that, the network performance is getting better gradually when we increase θ from 0 to 0.1, but getting worse gradually when we increase θ from 0.1 to 1. LBFN has a best performance when $\theta = 0.1$, so we set $\theta = 0.1$ in our LBFN. After all, our SR regression loss improved the network performance when it occupies a reasonable propotion of the loss function.

4.4. Ablation study of bidirectional feedback

Inspired by the unidirectional feedback architecture proposed in SRFBN [4], we propose a bidirectional feedback architecture. Bidirectional architecture can improve the performance of feedback network by perceiving features of different levels. To verify our idea, we compare our bidirectional feedback with unidirectional feedback. In our LBFN, we set T = 2 with two bidirectional feedback procedures. To make a fair comparison, we set T = 4 in unidirectional feedback architecture, which is shown in Fig. 7. As shown in Table 3, our bidirectional feedback has a better performance for it enhances the perceptual abilities of feedback network. What is more, our bidirectional feedback only has half as many iterations as unidirectional feedback, which reduces the computational complexity. With more iterations, our bidirectional feedback can reduce computational complexity even more. Therefore, our bidirectional feedback is superior to unidirectional feedback.

4.5. Ablation study of spatial attention in RAB

Inspired by RCAB [3], we propose RAB as the basic block in CFB. RCAB [3] integrated channel attention into residual block, as shown in Fig. 8, which magnified the difference between channels. So that channels with more information will be paid more attention. However, RCAB [3] neglected the differences between pixels in different spatial locations, which represent the information about edges, texture, and image content. Therefore, we propose RAB, which integrates channel attention and spatial attention into residual block. We infer that RAB has a better performance than RCAB [3]. To prove the improvements of our RAB, we train our network with RCAB [3] as the basic block in CFB to make a comparison. From the comparison results shown in Table 4, we can find that, our RAB has a better performance than RCAB [3], which confirms our inference that the differences between pixels are very important for the restoration of image details.

Table 4
Comparison of RCAB [3] and our RAB as the basic block in CFB for 3 x SR.

Basic block	Params	Set5	Set14	BSD100	Urban100	Manga109
RCAB [3]	532k	34.33/0.9266	30.32/0.8417	29.05/0.8038	28.06/0.8502	33.40/0.9433
RAB	533k	34.51/0.9282	30.45/0.8442	29.13/0.8059	28.36/0.8564	33.80/0.9456

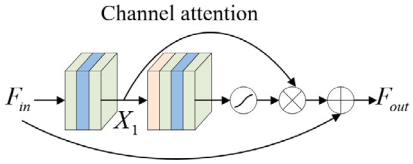

Fig. 8. Residual channel attention block (RCAB) [3].

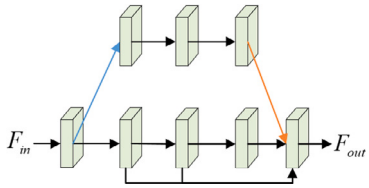

Fig. 9. The ablation of cross-learning in CFB.

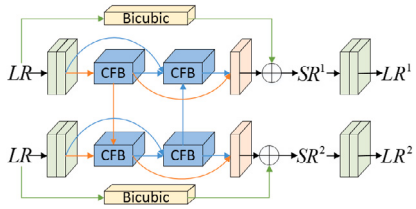

Fig. 10. Traditional multi-reconstruction method used on our LBFN.

Table 5
Ablation study of cross-learning in CFB for 3 x SR.

Cross-learning	Params	Set5	Set14	BSD100	Urban100	Manga109
✗	501k	34.29/0.9262	30.30/0.8416	29.04/0.8037	28.03/0.8493	33.36/0.9430
✓	533k	34.51/0.9282	30.45/0.8442	29.13/0.8059	28.36/0.8564	33.80/0.9456

4.6. Ablation study of cross-learning

In CFB, we use cross-learning between the HR feature flow and the LR feature flow. Cross-learning can learn the relationship between HR and LR features to guide the learning of mapping functions from LR to HR. To prove the effectiveness of cross-learning, we do ablation experiment of cross-learning, as shown in Fig. 9. Since our base number of filters m = 32, to make a fair comparison, we set m = 40 in the ablation model, which achieves almost the same number of parameters with LBFN. From the comparison results shown in Table 5, we can find that, LBFN with cross-learning has a better PSNR and SSIM than LBFN without cross-learning, which confirms our motivation that cross-learning can guide the learning of mapping functions from LR to HR. Therefore, our cross-learning is efficient.

4.7. Comparison of multi-reconstruction method and our ensemble method to reconstruct SR image

In most of the multi-branch networks [4,8,12], SR image is reconstructed by multi-reconstruction method. SRFBN [4] train the network by multi-reconstruction method, and took the SR image of the last iteration as the final SR result. However, in feedback networks, previous iterations are too shallow to achieve a valuable reconstruction. Therefore, in our LBFN, we fuse the HR features of all iterations by ensemble method to reconstruct SR image. We use multi-reconstruction method on LBFN to make a comparison, which is shown in Fig. 10, and the comparison results are shown in Table 6. LBFN with ensemble method get a better PSNR/SSIM

Table 6
Comparison of traditional multi-reconstruction method and our ensemble method to reconstruct SR image on LBFN for 3 x SR.

Reconstruction methods	Params	Set5	Set14	BSD100	Urban100	Manga109
Multi-reconstruction	532k	34.36/0.9270	30.30/0.8412	29.09/0.8048	28.19/0.8528	33.47/0.9434
Ensemble method	533k	34.51/0.9282	30.45/0.8442	29.13/0.8059	28.36/0.8564	33.80/0.9456

Table 7
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	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	36.66/0.9542	32.45/0.9067	31.36/0.8879	29.50/0.8946	35.60/0.9663
FSRCN [2]		13K	37.00/0.9558	32.63/0.9088	31.53/0.8920	29.88/0.9020	36.67/0.9710
VDSR [11]		666K	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140	37.22/0.9750
LapSRN [8]		251K	37.52/0.9591	32.99/0.9124	31.80/0.8952	30.41/0.9103	37.27/0.9740
DRRN [15]		298K	37.74/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188	37.88/0.9749
IDN [5]		553K	37.83/0.9600	33.30/0.9148	32.08/0.8985	31.27/0.9196	38.01/0.9749
EDSR-baseline [9]		1370K	37.99/0.9604	33.57/0.9175	32.16/0.8994	31.98/0.9272	38.54/0.9769
IMDN [21]		694K	38.00/0.9605	33.63/0.9177	32.19/0.8996	32.17/0.9283	38.88/0.9774
RFDN-L [24]		626K	38.08/0.9606	33.67/0.9190	32.18/0.8996	32.24/0.9290	38.95/0.9773
SMSR [25]		985K	38.00/0.9601	33.64/0.9179	32.17/0.8990	32.19/0.9284	38.76/0.9771
LBFN (ours)		438K	37.91/0.9602	33.50/0.9173	32.10/0.8988	31.96/0.9265	38.45/0.9766
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	32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989	30.48/0.9117
FSRCN [2]		13K	33.18/0.9140	29.37/0.8240	28.53/0.7910	26.43/0.8080	31.10/0.9210
VDSR [11]		666K	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279	32.01/0.9340
LapSRN [8]		502K	33.81/0.9220	29.79/0.8325	28.82/0.7980	27.07/0.8275	32.21/0.9350
DRRN [15]		298K	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378	32.71/0.9379
IDN [5]		553K	34.11/0.9253	29.99/0.8354	28.95/0.8013	27.42/0.8359	32.71/0.9381
EDSR-baseline [9]		1555K	34.37/0.9270	30.28/0.8417	29.09/0.8052	28.15/0.8527	33.45/0.9439
IMDN [21]		703K	34.36/0.9270	30.32/0.8417	29.09/0.8046	28.17/0.8519	33.61/0.9445
RFDN-L [24]		633K	34.47/0.9280	30.35/0.8421	29.11/0.8053	28.32/0.8547	33.78/0.9458
SMSR [25]		993K	34.40/0.9270	30.33/0.8412	29.10/0.8050	28.25/0.8536	33.68/0.9445
LBFN (ours)		533K	34.51/0.9282	30.45/0.8442	29.13/0.8059	28.36/0.8564	33.80/0.9456
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	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221	27.58/0.8555
FSRCN [2]		13K	30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280	27.90/0.8610
VDSR [11]		666K	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524	28.83/0.8870
LapSRN [8]		502K	31.54/0.8852	28.09/0.7700	27.32/0.7275	25.21/0.7562	29.09/0.8900
DRRN [15]		298K	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638	29.45/0.8946
IDN [5]		553K	31.82/0.8903	28.25/0.7730	27.41/0.7297	25.41/0.7632	29.41/0.8942
EDSR-baseline [9]		1518K	32.09/0.8938	28.58/0.7813	27.57/0.7357	26.04/0.7849	30.35/0.9067
IMDN [21]		715K	32.21/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838	30.45/0.9075
RFDN-L [24]		643K	32.28/0.8957	28.61/0.7818	27.58/0.7363	26.20/0.7883	30.61/0.9096
SMSR [25]		1006K	32.12/0.8932	28.55/0.7808	27.55/0.7351	26.11/0.7868	30.54/0.9085
LBFN (ours)		641K	32.30/0.8958	28.70/0.7843	27.62/0.7377	26.30/0.7918	30.71/0.9108

than LBFN with multi-reconstruction method, which confirms our motivation that ensemble method is more suitable for feedback networks than multi-reconstruction method.

4.8. Comparison with the state-of-the-art methods

As a lightweight network, we compare our LBFN with other lightweight state-of-the-art methods, such as SRCNN [1], FSRCNN [2], VDSR [11], LapSRN [8], DRRN [15], IDN [5], EDSR-baseline [9], IMDN [21], RFDN [24] and SMSR [25]. The PSNR and SSIM values are compared, and the comparison results are shown in Table 7. We can find that, our LBFN has little parameters, but outstanding performance, especially at the scales of ×3 and ×4.

At last, we provide visual comparisons of the SR images on ×4, which are shown in Fig. 11. From the comparison results, we can find that, our method recover the textures and details better than the others, which demonstrates the improvements of our LBFN.

5. Conclusion

Our LBFN achieved an outstanding performance with few parameters, which proves that feedback mechanism is ideal for lightweight networks, for it fully utilizes high-level features with few parameters. The bidirectional feedback architecture we proposed furtherly improved the performance of feedback network by perceiving the features of different levels. In the study of

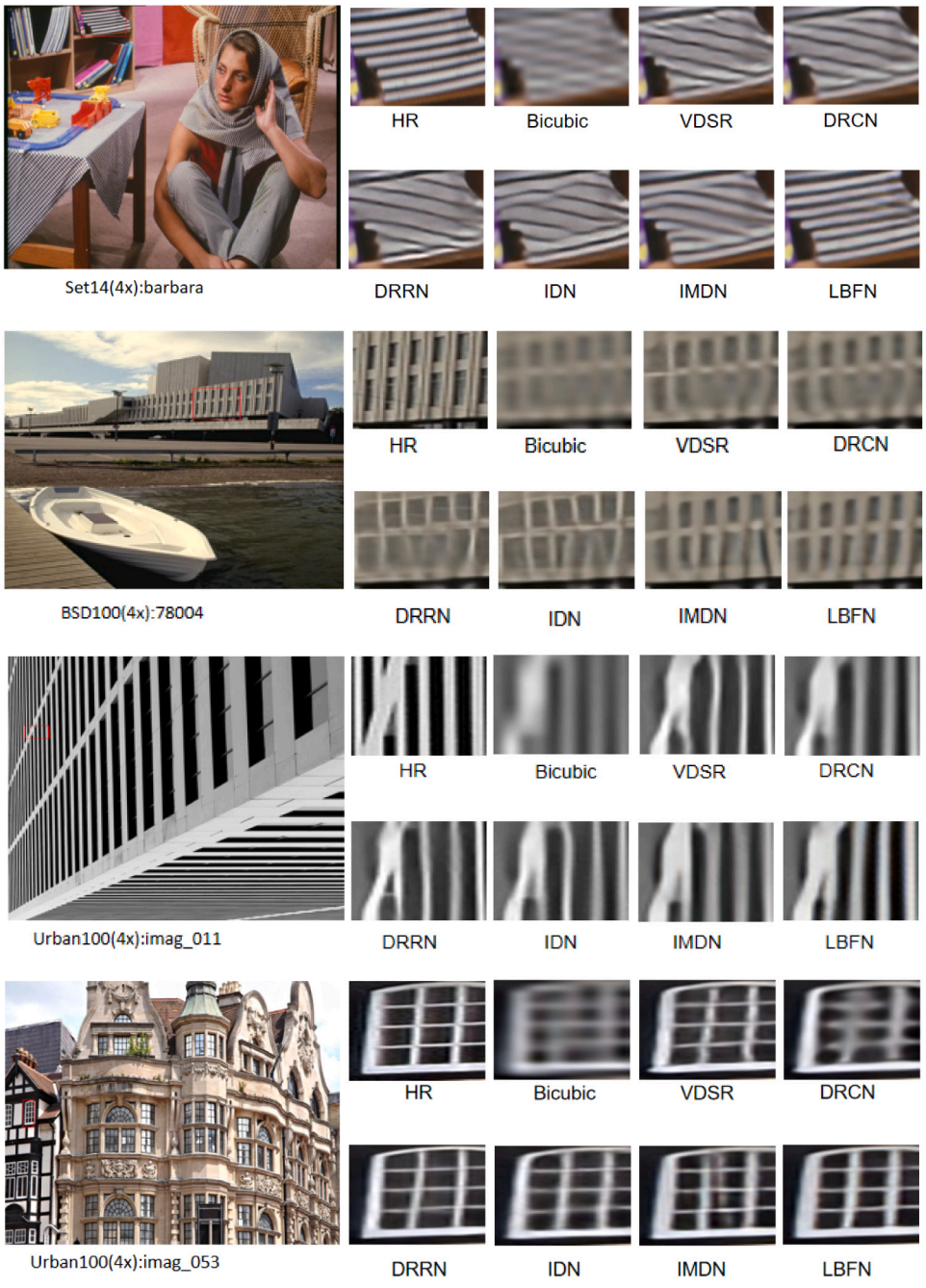

Fig. 11. Visual comparisons of our LBFN with other SR methods on Set14, BSD100 and Urban100 datasets.

SR regression loss, our SR regression loss improved the network performance when it occupied a reasonable propotion of the loss function. Through the ablation experiments of spatial attention and cross-learning, we respectively proved that residual blocks with channel and spatial attention are helpful for the restoration of image details, and cross-learning between HR and LR flows can generate powerful high-level representations. Our ensemble method significantly improved the reconstruction performance of our feedback network, which confirms our motivation that ensemble method is more suitable for feedback networks. All the motivations we proposed are proven contributing to our network performance. The final comparison results on benchmark datasets show that, the LBFN we proposed has an outstanding performance as a lightweight SR network.

Declaration of competing interest

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.

Data availability

Data will be made available on request.

Funding

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

References

[1] Dong KHC, Loy CC, Tang X. Learning a deep convolutional network for image super-resolution. In: Computer vision – ECCV 2014, vol. 8692. 2014, p. 184–99.
[2] Chao Dong CCL, Tang X. Accelerating the super-resolution convolutional neural network. In: Computer vision – ECCV 2016. 2016, p. 391–407.
[3] Zhang Y, Li k, Li K, Wang L, Zhong B, Fu Y. Image super-resolution using very deep residual channel attention networks. In: European conference on computer vision, vol. 11211. 2018, p. 294–310.
[4] Li Z, Yang J, Liu Z, Yang X, Jeon G, Wu W. Feedback network for image super-resolution. In: 2019 IEEE/CVF Conference on computer vision and pattern recognition. CVPR, 2019, p. 3862–71.
[5] Hui Z, Wang X, Gao X. Fast and accurate single image super-resolution via information distillation network. In: 2018 IEEE/CVF Conference on computer vision and pattern recognition. 2018, p. 723–31.
[6] Haris M, Shakhnarovich G, Ukita N. Deep back-projection networks for super-resolution. In: 2018 IEEE/CVF Conference on computer vision and pattern recognition. 2018, p. 1664–73.
[7] Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, et al. Photo-realistic single image super-resolution using a generative adversarial network. In: 2017 IEEE Conference on computer vision and pattern recognition. 2017, p. 105–14.
[8] Lai W-S, Huang J-B, Ahuja N, Yang M-H. Deep Laplacian pyramid networks for fast and accurate super-resolution. In: IEEE Conference on computer vision and pattern recognition. 2017, p. 5835–43.
[9] Lim B, Son S, Kim H, Nah S, Lee KM. Enhanced deep residual networks for single image super-resolution. In: 2017 IEEE Conference on computer vision and pattern recognition workshops. 2017, p. 1132–40.
[10] Shi W, Caballero J, Huszár F, Totz J, Aitken AP, Bishop R, et al. 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. 2016, p. 1874–83.
[11] Kim J, Lee J, Lee K. Accurate image super-resolution using very deep convolutional networks. In: 2016 IEEE Conference on computer vision and pattern recognition. 2016, p. 1646–54.
[12] Kim J, Lee JK, Lee KM. Deeply-recursive convolutional network for image super-resolution. In: 2016 IEEE Conference on computer vision and pattern recognition. 2016, p. 1637–45.
[13] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on computer vision and pattern recognition. 2016, p. 770–8.
[14] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: IEEE Conference on computer vision and pattern recognition. 2017, p. 2261–9.
[15] Tai Y, Yang J, Liu X. Image super-resolution via deep recursive residual network. In: 2017 IEEE Conference on computer vision and pattern recognition. 2017, p. 2790–8.
[16] Rensink RA. The dynamic representation of scenes. Vis Cogn 2000;7(1–3):17–42.
[17] Corbetta M, Shulman G. Control of goal-directed and stimulus-driven attention in the brain. Nat Rev Neurosci 2002;3:201–15.
[18] Wang F, Jiang M, Qian C, Yang S, Li C, Zhang H, et al. Residual attention network for image classification. In: 2017 IEEE Conference on computer vision and pattern recognition. 2017, p. 6450–8.
[19] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: 2018 IEEE/CVF Conference on computer vision and pattern recognition. 2018, p. 7132–41.
[20] Woo S, Park J, Lee J-Y, Kweon I. CBAM: Convolutional block attention module: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII. 2018, p. 3–19.
[21] Hui Z, Gao X, Yang Y, Wang X. Lightweight image super-resolution with information multi-distillation network. In: Proceedings of the 27th ACM International conference on multimedia. 2019, p. 2024–32.
[22] Tai Y, Yang J, Liu X, Xu C. Memnet: A persistent memory network for image restoration. In: 2017 IEEE International conference on computer vision. 2017, p. 4549–57.
[23] Han W, Chang S, Liu D, Yu M, Witbrock M, Huang TS. Image super-resolution via dual-state recurrent networks. In: 2018 IEEE/CVF Conference on computer vision and pattern recognition. 2018, p. 1654–63.
[24] Liu J, Tang J, Wu G. Residual feature distillation network for lightweight image super-resolution. In: Bartoli A, Fusiello A, editors. Computer vision - ECCV 2020 Wwrkshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part III, vol. 12537. 2020, p. 41–55.
[25] Wang L, Dong X, Wang Y, Ying X, Lin Z, An W, et al. Exploring sparsity in image super-resolution for efficient inference. In: 2021 IEEE/CVF Conference on computer vision and pattern recognition. 2021, p. 4915–24.

Beibei Wang is a Ph.D. student in College of Electronics and Information Engineering, Sichuan University. She is interested in image process and pattern recognition. Her current research focuses on image super-resolution using deep learning. 690983790@qq.com.

Binyu Yan is currently an associate professor in College of Electronics and Information Engineering, Sichuan University. His research interests are image process and pattern recognition. Yanby@scu.edu.cn.

Changjun Liu is currently a professor in College of Electronics and Information Engineering, Sichuan University. His research interests are microwave circuits, microwave wireless energy transmission, numerical simulation and other directions. cjliu@scu.edu.cn.

Ryul Hwangbo is the executive director of ISPark Co., Ltd. hbyul@ispark.kr.

Gwanggil Jeon is currently a professor at Xidian University and Incheon National University. His research interests are image processing and computational intelligence. He was the recipient of the IEEE Chester Sall Award in 2007 and the 2008 ETRI Journal Paper Award. ggjeon@gmail.com.

Xiaomin Yang held a postdoctoral position with the University of Adelaide. She is currently a Professor with the College of Electronics and Information Engineering, Sichuan University. Her research interests are image processing and pattern recognition. arielyang@scu.edu.cn.