Journal of Real-Time Image Processing (2022) 19:941–955
https://doi.org/10.1007/s11554-022-01235-x

ORIGINAL RESEARCH PAPER

BAM: a balanced attention mechanism to optimize single image super-resolution

Fanyi Wang[1] · Haotian Hu[1] · Cheng Shen[2] · Tianpeng Feng[3] · Yandong Guo[3]

Received: 25 April 2022 / Accepted: 22 June 2022 / Published online: 26 July 2022

Abstract

Recovering texture information from the aliasing regions has always been a major challenge for single image super-resolution (SISR) task. These regions are often submerged in noise so that we have to restore texture details while suppressing noise. To address this issue, we propose an efficient Balanced Attention Mechanism (BAM), which consists of Avgpool Channel Attention Module (ACAM) and Maxpool Spatial Attention Module (MSAM) in parallel. ACAM is designed to suppress extreme noise in the large-scale feature maps, while MSAM preserves high-frequency texture details. Thanks to the parallel structure, these two modules not only conduct self-optimization, but also mutual optimization to obtain the balance of noise reduction and high-frequency texture restoration during the back propagation process, and the parallel structure makes the inference faster. To verify the effectiveness and robustness of BAM, we applied it to 10 state-of-the-art SISR networks. The results demonstrate that BAM can efficiently improve the networks' performance, and for those originally with attention mechanism, the substitution with BAM further reduces the amount of parameters and increases the inference speed. Information multi-distillation network (IMDN), a representative lightweight SISR network with attention, when the input image size is 200×200, the FPS of proposed IMDN-BAM precedes IMDN $\{8.1\%, 8.7\%, 8.8\%\}$ under the three SR magnifications of $\times2, \times3, \times4$, respectively. Densely residual Laplacian network (DRLN), a representative heavyweight SISR network with attention, when the scale is 60×60, the proposed DRLN-BAM is $\{11.0\%, 8.8\%, 10.1\%\}$ faster than DRLN under $\times2, \times3, \times4$. Moreover, we present a dataset with rich texture aliasing regions in real scenes, named realSR7. Experiments prove that BAM achieves better super-resolution results on the aliasing area.

Keywords Single image super-resolution · Texture aliasing · Inference acceleration · Lightweight attention mechanism

1 Introduction

Single image super-resolution (SISR) is one of the popular computer vision research topics [1, 2], which aims to reconstruct a high-resolution (HR) image from a low-resolution (LR) image. With the success of deep learning prevailed in computer vision, many convolutional neural network (CNN)-based super-resolution (SR) methods have been proposed. According to their architectures, they can be categorized into linear [3–8], residual [9, 10], recursive [11–13], densely connected [14–16], multi-path [17], and adversarial [18] designs. To further improve the quality [19] of SR results while controlling parameter amounts, attention mechanisms [20, 21] were adopted in some SISR networks. At the same time, there exist quite a lot of excellent SISR networks [22–26] without the attention mechanism. One motivation of our work is to propose a plug-and-play attention mechanism for them so that their applications can be more extensive, and make it more fair to compare these networks with those with attention [27–30]. The attention mechanism [20, 21] was first applied to classification tasks. Due to its remarkable results in classification, great efforts have been made along this direction and expanded its application to SISR tasks. However, the SISR networks are so diverse that the attention module is usually designed solely for a specific network structure. These proposed attention mechanisms require a baseline to compare with in order to verify their effectiveness. Therefore, another motivation of

✉ Fanyi Wang
11730038@zju.edu.cn

[1] Zhejiang University, Hangzhou 310027, China

[2] California Institute of Technology, Pasadena, CA 91125, USA

[3] OPPO Research Institute, Shenzhen, China

our work is to propose a baseline of attention mechanism for SISR. Actually, our BAM is not only more efficient but also more lightweight than the attention mechanisms proposed in [27–30], which has been proved in our experiments. One major problem for the existing SISR networks is the information restoration in the texture aliasing area, so our biggest motivation is to overcome this problem. As shown in Fig. 1, IMDN-BAM has superior results in the texture aliasing area compared with IMDN.

The proposed BAM is plug-and-play for the majority of SISR networks. For those without attention, BAM can be easily inserted behind the basic block or before the upsampling layer. Only adding a few number of parameters, it can generally improve the SR results, validated by peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) [19] metrics. For those with attention, BAM can seamlessly replace their attention mechanism. Due to the simple structure and high efficiency of BAM, it can generally reduce the amount of parameters and improve the SR performance. We experimented on six networks without attention and four with attention to verify the effectiveness and robustness of BAM. Contributions are summarized as follows:

- We propose a lightweight and efficient attention mechanism, for the SISR task. BAM can restore high-frequency texture information as much as possible while suppressing the extreme noise in the large-scale feature maps. Furthermore, the parallel structure can improve the inference speed.
- We conduct comparative experiments on 10 state-of-the-art SISR networks [22–30]. The insertion or replacement of BAM generally improves the PSNR and SSIM values of final SR results and the visual quality, and for those [27–30] with attention, the replacement of BAM further reduces the amount of parameters and accelerates the inference speed. What is more, for lightweight SISR networks [23–25, 27, 31], the comparative experiments illustrate that BAM can generally improve their performance but barely increase or even decrease the parameters, which is significant for their deployment on terminals.
- We present a real-scene SISR dataset considering the practical texture aliasing issue. BAM can achieve better SR performance on this more realistic dataset.

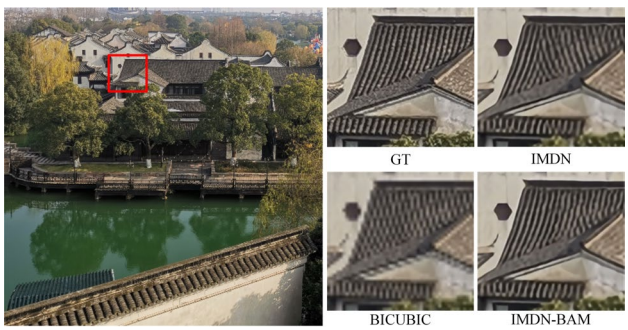


Fig. 1 Comparison of ×4 SR results of IMDN and IMDN-BAM on the realSR7 dataset. IMDN-BAM shows better super-resolution results on texture aliasing areas

2 Related works

In this section, 10 SISR networks used in control experiments will be introduced. The specific position where the BAM is inserted or replace the original attention module in each SISR network is shown in Fig. 2.

2.1 SISR networks without attention mechanism

Enhanced deep residual super-resolution network (EDSR) [26] as the champion of NTIRE 2017 Challenge on Single Image Super-Resolution, removes the BN layer and the last activation layer in the residual network, allowing the residual structure originally designed for high-level problems to make a significant breakthrough in the low-level SISR problem. Our BAM module is inserted before the upsampling layer and marked with a purple solid circle in Fig. 2. To achieve real-time performance, Namhyuk Ahn proposed cascading residual network (CARN) [25] in which the middle part is based on ResNet. In addition, the local and global cascade structures can integrate features from multiple layers, which enables learning multi-scale information of the feature maps. Its lightweight variant, CARN-M, compromises the performance for speed. For these two networks, BAM is inserted behind each block. multi-scale residual network (MSRN) [22] combines local multi-scale features with global features to fully exploit the LR image, which solves the issue of feature disappearance during propagation. BAM will be concatenated to the end of each MSRN block.

Super lightweight super-resolution network (s-LWSR) [23] is specifically designed for the deployment of real-time SISR task on mobile devices. It borrows the idea of U-Net [31] and is the first attempt to apply the encoder–decoder structure for the SISR problem. The encoder part employs a similar structure with MobileNetV2 [32] and residual block as the basic building blocks of the network. To adapt to different scenarios, three networks of different size, s-LWSR$_{16}$, s-LWSR$_{32}$ and s-LWSR$_{64}$, were proposed. Here, we choose the middle-size one, s-LWSR$_{32}$. For s-LWSR$_{32}$, the BAM will be inserted before the upsampling layer.

In recent years, many lightweight SR models have been proposed. Among them, adaptive weighted super-resolution network (AWSRN) [24] is a representative one. A novel local fusion block is designed in AWSRN for efficient residual

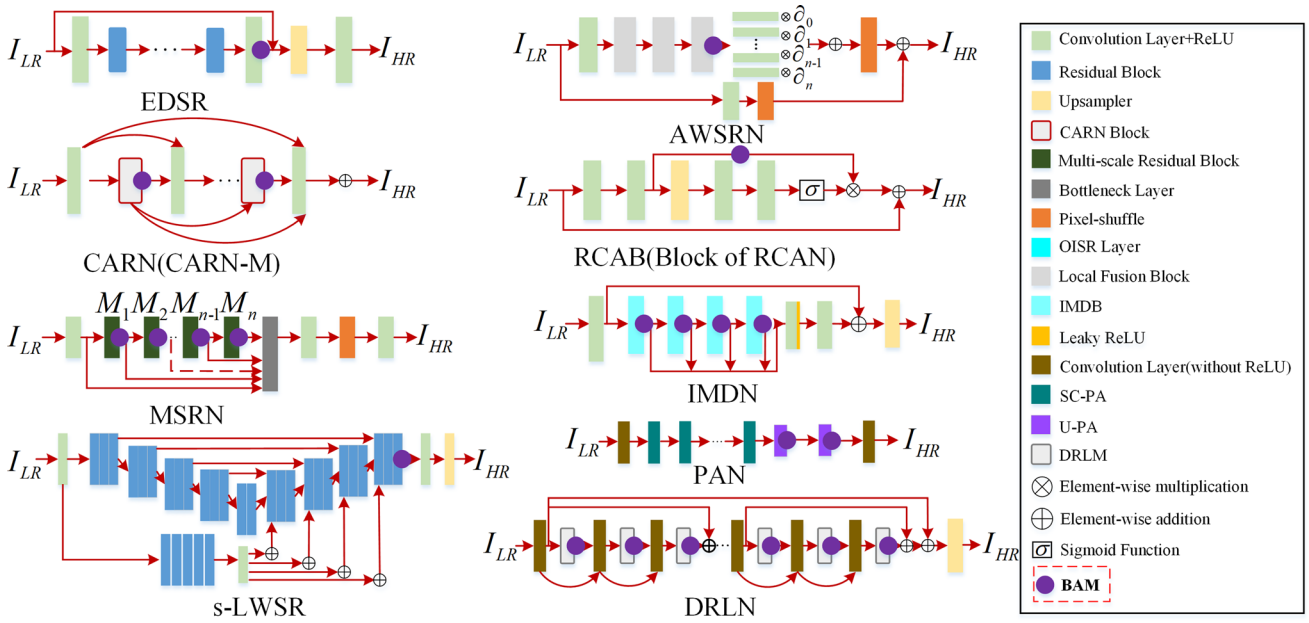


Fig. 2 Structure diagram of six SISR networks without attention (where CARN and CARN-m have the same network structure but different number of channels) and four SISR networks with attention. BAM module is represented by a purple solid circle

learning, which consists of stacked adaptive weighted residual units and a local residual fusion unit. It can achieve efficient flow and fusion of information and gradients. Moreover, an adaptive weighted multi-scale (AWMS) module is proposed to not only make full use of the features in reconstruction layer but also reduce the amount of parameters by analyzing the information redundancy between branches of different scales. Different from the aforementioned networks, BAM will be inserted before the AWMS module.

2.2 SISR networks with attention mechanism

In the SISR field, the study focused on attention mechanism is relatively less than the ones on the network structure. The common attention mechanisms applied to SR are mainly the soft ones, including channel attention, spatial attention, pixel attention, and non-local attention. We introduce four networks with their own attention mechanism here.

LR input images contain rich low-frequency information, which is usually treated equally with high-frequency information across channels. This will hamper the network's learning ability. In order to solve the problem, residual channel attention network (RCAN) [29] was proposed. It leads the SISR model performance in terms of PSNR and SSIM metrics, thus is often used as the baseline by the following works. RCAN utilized a residual-in-residual (RIR) structure to construct the whole network, which allows the rich low-frequency information to directly propagate to the rear part through multiple skip connections. Thus, the network can focus on learning high-frequency information. What is more, a channel attention (CA) mechanism was utilized to adaptively adjusts features by considering the interdependence between channels. In our experiments, CA will be replaced with BAM.

IMDN is a representative lightweight SISR network with attention mechanism. It is constructed by the cascaded information multi-distillation blocks (IMDB) consisting of distillation and selective fusion parts. The distillation module extracts hierarchical features step-by-step, and fusion module aggregates them according to the importance of candidate features, which is evaluated by the proposed contrast-aware channel attention (CCA) mechanism.

Pixel attention network (PAN) [27] is the winning solution of AIM2020 VTSR Challenge. Although its amount of parameters is only 272 K, its performance is comparable to SRResNet [5] and CARN. PAN newly proposed a pixel attention (PA) mechanism, similar to channel attention and spatial attention. The difference is that PA generates 3D attention maps, which allows the performance improvement with fewer parameters.

DRLN [30] employs cascading residual on the residual structure to allow the flow of low-frequency information so that the network can focus on learning high and mid-level features. Moreover, it proposes a Laplacian attention (LA) to model the crucial features to learn the inter-level and intra-level dependencies between the feature maps. In the comparative experiments, CA, CCA, PA, and LA will be replaced with BAM.

3 Proposed method

Some texture details in low-resolution images are often overwhelmed by extreme noises, which leads to a major difficulty to recover texture information from the texture aliasing area. To solve this problem, we proposed the BAM composed of ACAM and MSAM in parallel, where ACAM is dedicated to suppressing extreme noise in the large-scale feature maps and MSAM tries to pay more attention to the high-frequency texture details. Moreover, the parallel structure of BAM will allow not only self-optimization, but also mutual optimization of the channel and spatial attention during the gradient backpropagation process so as to achieve a balance between them. It can obtain the best noise reduction and high-frequency information recovery capabilities, and the parallel structure can speed up the inference process. The schematic of BAM is shown in Fig. 3. Since ACAM and MSAM generate vertical and horizontal attention weights for the input feature maps respectively, the dimension of their output is inconsistent. One is $N \times C \times 1 \times 1$ and the other is $N \times 1 \times H \times W$. Thus, we use broadcast multiplication to fuse them into an $N \times C \times H \times W$ weight tensor, and then multiply it with the input feature maps element-wisely. Here, N is the batch size ($N = 16$ in our experiments), C is the number of channels of the feature maps, H and W are the height and width of the feature maps. In ACAM, avgpool operation is used to obtain the average value of each feature map, while in MSAM, maxpool operation is used to get the max value among the C channels for each position on the feature map, and they can be expressed as

$$\text{Avgpool}(N, C, 1, 1) = \frac{1}{H \times W} \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} F(N, C, h, w), \tag{1}$$

$$\text{Maxpool}(N, 1, H, W) = \max\{F(N, c, H, W), \quad c \in [0, C-1]\}, \tag{2}$$

where $F \in \mathbb{R}^{N \times C \times H \times W}$ represents the input feature maps, max{} means to get the max value.

3.1 Avgpool channel attention module

Channel attention needs to find channels with more important information from the input feature maps and give them higher weights. It is highly likely for a channel with the dimension of $H \times W$ (in our experiments, $H = W \geq 64$) to contain some abnormal extrema. Maxpool will pick these extreme values as noise and get the wrong attention information, which will make the texture recovery more difficult. Therefore, we only use avgpool to extract channel information so that it complies with Occam's razor principle when suppressing extreme noise and then pass it through a multilayer perceptron (MLP) composed of two point-wise convolution layers. To increase the nonlinearity of MLP, PReLU [33] is used to activate the first convolution layer output. In

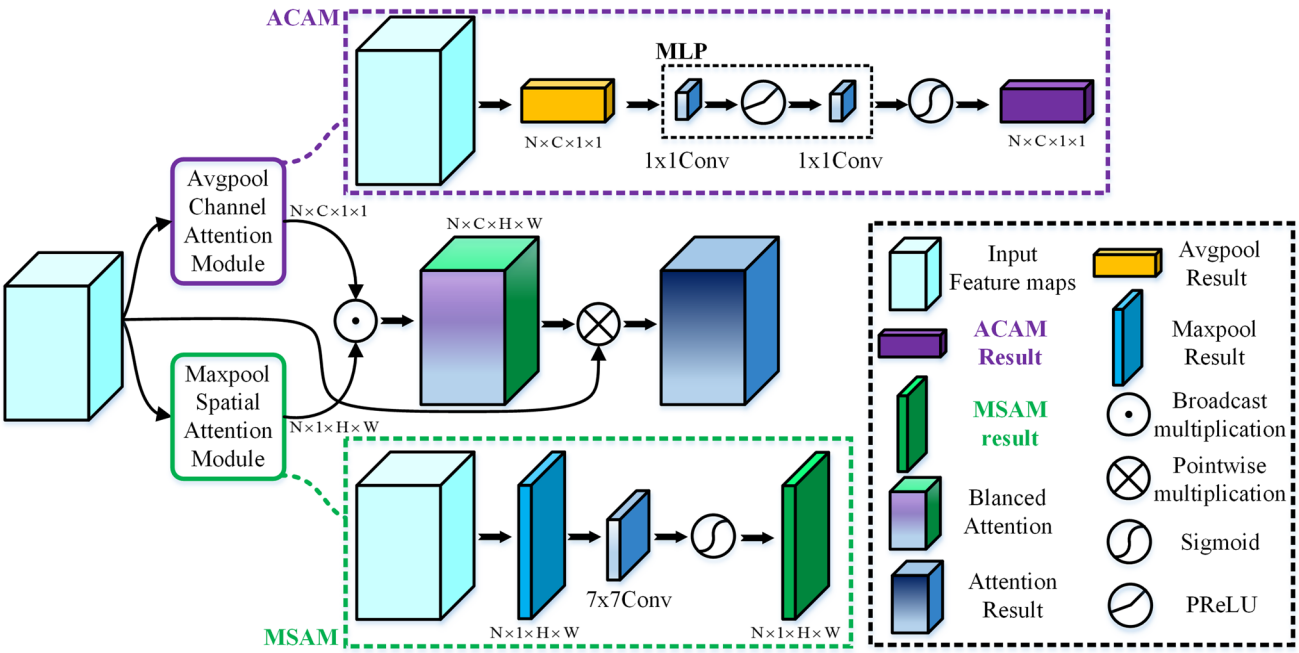


Fig. 3 BAM, consisting of ACAM and MSAM in parallel. The channel attention from ACAM and the spatial attention from MSAM will be fused by broadcast multiplication and then multiplied with the input feature maps element-wisely to obtain the final attention result. **a** ACAM. The channel attention information is extracted by avgpool. **b** MSAM. The spatial attention information is extracted by maxpool

addition, to reduce the parameter amount and computational complexity of ACAM, MLP adopts the bottleneck architecture [34]. The number of input channels is r times the number of output channels for the first convolution layer. After PReLU activation, the number of channels is restored by the second convolution layer. Finally, the channel weights are generated by a sigmoid activation function. The generation process of ACAM can be described by

$$\mathrm{ACAM}(F) = \mathrm{Sigmoid}[\mathcal{F}^{k\times k}_{n/r->n}(\mathrm{PReLU}(\mathcal{F}^{k\times k}_{n->n/r}(\mathrm{Avgpool}(F))))], \tag{3}$$

where $\mathcal{F}^{k\times k}_{n->n/r}$ represents the convolution layer with the kernel size of $k \times k$ (for Eq. 3, $k = 1$), the input channel number of n and the output channel number of n/r, r is set to 16 and n is determined by the channel numbers of the input feature maps in experiments. PReLU and Sigmoid are defined as

$$\mathrm{PReLU} = \begin{cases} x, & x > 0 \\ a \cdot x, & x \le 0,\ a = 1 \end{cases}, \tag{4}$$

$$\mathrm{Sigmoid} = \frac{1}{1 + \mathrm{e}^{-x}}. \tag{5}$$

In the ablation experiments, in order to verify the minimalism of AvgPool for channel attention, a comparative experiment is carried out by adding MaxPool to ACAM, which is named as ACAM$^+$, and the mathematical expression is

$$\begin{aligned}\mathrm{ACAM}^+(x) = \mathrm{Sigmoid}[&f^{1\times1}_{n/r->n}(\mathrm{PReLU}(f^{1\times1}_{n->n/r}(\mathrm{AvgPool}(x)))) \\ &+ f^{1\times1}_{n/r->n}(\mathrm{PReLU}(f^{1\times1}_{n->n/r}(\mathrm{MaxPool}(x))))].\end{aligned} \tag{6}$$

To reduce the parameter amount and the computational complexity, AvgPool and MaxPool share the MLP.

3.2 Maxpool spatial attention module

Spatial attention generates weights for the horizontal section of the input feature maps. Its goal is to find lateral areas which contribute most to the final HR reconstruction and give them higher weights. These areas usually contain high-frequency details in the form of extreme values in the channel. Thus, using maxpool operation for spatial attention is appropriate.

The output of maxpool passes a convolution layer with large receptive field of $k \times k$ (for Eq. 7, $k = 7$), and then gets activated by the sigmoid function to obtain the spatial attention weights. This design effectively controls the amount of parameters. It can be expressed by

$$\mathrm{MSAM}(x) = \mathrm{Sigmoid}[\mathcal{F}^{7\times7}_{1->1}(\mathrm{Maxpool}(x))]. \tag{7}$$

Similarly, to verify the minimalism of MSAM, AvgPool will be added to MSAM to form a new structure named as MSAM$^+$ in ablation experiments. It can be written as

$$\mathrm{MSAM}^+(x) = \mathrm{Sigmoid}[f^{7\times7}_{2->1}(\mathrm{MaxPool}(x);\mathrm{AvgPool}(x))]. \tag{8}$$

3.3 Balanced attention mechanism

There are two innovations in the design of BAM. One is that the ACAM tries to suppress the extreme noise and the MSAM tries to maintain the texture information. The other is the parallel structure, which makes the generation process of channel attention and spatial attention independent of each other and allows the mutual optimization of two attentions during the backpropagation. The combination of these two innovations enables BAM to recover as much high-frequency information as possible from the texture aliasing area. Ablation experiments prove that the current design of BAM can effectively control the parameter amount and obtain better performance than the original networks, evaluated by PSNR and SSIM metrics. The formula of BAM is

$$\mathrm{BAM}(F) = \mathrm{ACAM}(F) \otimes \mathrm{MSAM}(F) \odot F, \tag{9}$$

where $\otimes$ means broadcast multiplication and $\odot$ stands for Hadamard multiplication. Because the outputs of ACAM and MSAM have different dimensions, we utilize broadcast multiplication to fuse them and then element-wisely multiply it with the input feature maps to obtain the final attention results. ACAM and MSAM are self-optimized in their respective gradient backpropagation process. To reveal the mutual optimization of ACAM and MSAM in the gradient backpropagation process of BAM, we give the partial derivative of BAM concerning the input feature maps F as follows:

$$\begin{aligned}\frac{\partial \mathrm{BAM}(F)}{\partial F} = &\frac{\partial \mathrm{ACAM}(F)}{\partial F} \otimes \mathrm{MSAM}(F) \odot F + \mathrm{ACAM}(F) \otimes \\ &\frac{\partial \mathrm{MSAM}(F)}{\partial F} \odot F + \mathrm{ACAM}(F) \otimes \mathrm{MSAM}(F).\end{aligned} \tag{10}$$

As illustrated in Eq. 10, not only is ACAM and MSAM related to each other but also related to each other's first-order partial differentials (The gradient), which means ACAM and MSAM can optimize mutually in the gradient backpropagation process of BAM.

To show that BAM is minimally effective, we replace ACAM and MSAM with ACAM$^+$ and MSAM$^+$ to form a new structure BAM$^+$ in the ablation comparative experiments. BAM$^+$ can be expressed as

$$\mathrm{BAM}^+(F) = \mathrm{ACAM}^+(F) \otimes \mathrm{MSAM}^+(F) \odot F. \tag{11}$$

3.4 Parameter amount analysis

Moreover, to study the effect of BAM insertion on the original network parameter amount, we calculate the parameters of BAM, which depend on the parameters of ACAM and MSAM. First, we calculate the parameter amount of the convolutional layer without bias term using

$$\text{Param} = k \times k \times n_{\text{in}} \times n_{\text{out}}, \tag{12}$$

where k is the size of the convolution kernel, n_{in} and n_{out} is the number of input and output channels of the convolutional layer, respectively.

Based on Eq. 3 and Eq. 12, the parameter amount of ACAM can be obtained by

$$\text{Param}_{\text{ACAM}} = k \times k \times \left(n_{\text{in}} \times \frac{n_{\text{in}}}{r} + \frac{n_{\text{in}}}{r} \times n_{\text{in}} \right) + \frac{n_{\text{in}}}{r}, \tag{13}$$

where k is the kernel size which is equal to 1 in Eq. 13, r is the scale factor between the number of input and output channels (in our experiments, r is set to 16) and the last item is the parameter amount of PReLU. Based on Eq. 7 and Eq. 12, the parameter amount of MSAM is

$$\text{Param}_{\text{MSAM}} = k \times k \times n_{\text{in}} \times n_{\text{out}}. \tag{14}$$

In Eq. 14, $k = 7$, $n_{\text{in}} = n_{\text{out}} = 1$. And we can see that MSAM only has 49 parameters.

4 Experiments and discussions

To demonstrate the effectiveness and robustness of BAM, we select six existing SISR networks without attention [22–26] and four with attention [27–30] for control experiments. How BAM is inserted or replaces the original attention module has been elaborated in Sect. 2. Also, to further improve the effectiveness of BAM, IMDN is selected as the base model for the ablation experiments. Its CCA module is replaced with CA, SE, CBAM and BAM sequentially.

4.1 Datasets and metrics

As shown in Table 1, the training sets for different SISR networks are different, and for the deep learning task, the richer the data is, the better the results would be. Therefore, to fully verify the efficient performance of the proposed BAM, we choose the smallest training set for training. Following [24, 25], we use 800 high-quality (2 K resolution) images from DIV2K [35] as the training set, and evaluate on Set5 [36], Set14 [37], BSD100 [38], and Manga109 [39] with the PSNR and SSIM metrics under the upscaling factors of $\times 2, \times 3$, and $\times 4$, respectively, for ablation experiments we add Urban100 [40] for validation. In all the experiments, bicubic interpolation is utilized as the resizing method. Referring to [41], we calculate the metrics on the luminance channel (Y channel of the YCbCr channels converted from the RGB channels).

Table 1 Training sets for the original networks used in experiments

Training sets	Networks
DIV2K800 [35]	AWSRN, RCAN, IMDN
DIV2K1000 [35]	EDSR, MSRN, s-LWSR$_{32}$
DIVCK800 [35], Flickr2K [42]	PAN, DRLN
DIV2K1000 [34], 291 images [43], Berkeley Segmentation Dataset [44]	CARN, CARN-M

4.2 Implementation details

During the training, we use the RGB patches with size of 64×64 from the LR input together with its corresponding HR patches. We only apply data augmentation to the training data. Specifically, the 800 image pairs in training set are cropped into five pairs from the four corners and center of the original image so that the training set is expanded by five times to 4000 image pairs. In addition, we randomly rotate and flip them during the training process.

For optimization, Adam is used and its initial learning rate is set as 0.0001, which will be halved at every 200 epochs. The batch size is set as 16. We train for a total of 1000 epochs. The loss function for training is L1 loss function, which can be expressed as

$$L1\text{loss} = \frac{1}{3hw} \sum_{c=0}^{C-1} \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} \left\| \mathcal{L}\Theta^{\text{SR}}(I_{\text{LR}}(c,h,w)) - I_{\text{HR}}(c,h,w) \right\|_1, \tag{15}$$

where I_{LR} and I_{HR} are the input LR image and the target HR image respectively, $\mathcal{L}^{\text{SR}}$ represents the SISR network using the upsampling scale of SR, h, w and c are the height, width and channels of the HR image, respectively, and $||_1$ is the L1 norm.

We adopt pytorch 1.1.0 framework to implement experiments on the desktop computer with 3.4 GHz Intel Xeon-E5-2643-v3 CPU, 64G RAM, and two NVIDIA GTX 2080Ti GPUs.

4.3 Comparisons with original SISR networks

For the convenience of discussion, we refer to the original networks as the control group, the BAM versions as the experimental group and add the "BAM" suffix to the networks' original name. The control experiments' results of without and with attention networks are summarized in Tables 2 and 3, respectively. The networks are listed in

the order of their publication times, respectively [24]. It can be seen from Tables 2 and 3 that, except for RCAN-BAM and PAN-BAM at the ×3 and ×4 scaling factor on the Manga109 benchmark, all the other experimental groups outperform their corresponding control group on PSNR metric. Although the PSNR metric of RCAN-BAM is lower than the one of RCAN, its SSIM metric is still higher than that of RCAN. It reflects that BAM is more capable of restoring the fine structures than the color. In addition, for the three scale factors, the highest PSNR and SSIM metrics are all achieved by DRLN-BAM, and for ×4 upsampling scale, the PSNR/SSIM metrics improvements on four benchmarks are {0.03/0.0007, 0.17/0.0036, 0.95/0.0289, 0.55/0.0070} separately, meanwhile the reduction of the parameter amount is 266.7 K. Compared with the original attention mechanism of DRLN, BAM reduces the parameters, but obtains better performance.

Actually, some control experiments used additional data sets [35, 42–44] for training in their original papers, as shown in Table 1. In detail, CARN used extra [35, 43, 44]; PAN, DRLN used extra [42]; s-LWSR$_{32}$, EDSR and MSRN used all the images in DIV2K [35]. For deep learning tasks, there is a universally used law, the richer the amount of data, the better the effect. Although our experimental groups have the disadvantage of a smaller training set, but can generally achieve a better PSNR/SSIM results than the corresponding control groups. For lightweight networks such as PAN and IMDN, it is traditionally quite difficult to further improve their performance. The proposal of BAM makes it possible to enhance these lightweight SISR networks even with reduced parameters, which is of great significance for their deployment in realistic cases.

The results of the comparative experiments in Tables 2 and 3 show that for the networks without attention, the

Table 2 Control experiment results on 6 SISR networks without attention, the lightweight SISR networks are marked in bold black

Scale	Method	Param	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Manga109 PSNR/SSIM
×2	EDSR(CVPRW'17)[26]	40729.6K	38.11/0.9601	33.92/0.9195	32.32/0.9013	-
	EDSR-BAM	40737.9K	38.19/0.9613$_{\uparrow 0.08/\uparrow 0.0012}$	34.00/0.9213$_{\uparrow 0.08/\uparrow 0.0018}$	34.20/0.9273$_{\uparrow 1.88/\uparrow 0.0260}$	39.72/0.9806
	CARN(ECCV'18)[25]	1592.0K	37.76/0.9590	33.52/0.9166	32.09/0.8978	-
	CARN-BAM	1593.7K	37.84/0.9600$_{\uparrow 0.08/\uparrow 0.0010}$	33.55/0.9167$_{\uparrow 0.03/\uparrow 0.0001}$	33.90/0.9245$_{\uparrow 1.81/\uparrow 0.0267}$	38.68/0.9787
	CARN-M(ECCV'18)[25]	1161.3K	37.53/0.9583	33.26/0.9141	31.92/0.8960	-
	CARN-M-BAM	1163.0K	37.75/0.9597$_{\uparrow 0.22/\uparrow 0.0014}$	33.44/0.9158$_{\uparrow 0.18/\uparrow 0.0017}$	33.81/0.9237$_{\uparrow 1.89/\uparrow 0.0277}$	38.48/0.9783
	MSRN(ECCV'18)[22]	5930.3K	38.08/0.9605	33.74/0.9170	32.23/0.9013	38.64/0.9771
	MSRN-BAM	5934.9K	38.11/0.9610$_{\uparrow 0.03/\uparrow 0.0005}$	33.84/0.9192$_{\uparrow 0.10/\uparrow 0.0018}$	34.12/0.9265$_{\uparrow 1.89/\uparrow 0.0252}$	39.45/0.9801$_{\uparrow 0.81/\uparrow 0.0030}$
	s-LWSR$_{32}$(TIP'19)[23]	534.1K	-	-	-	-
	s-LWSR$_{32}$-BAM	534.3K	37.91/0.9603	33.63/0.9174	33.97/0.9252	38.82/0.9791
	AWSRN(CVPR'19)[24]	1396.9K	38.11/0.9608	33.78/0.9189	32.26/0.9006	38.87/0.9776
	AWSRN-BAM	1397.2K	38.14/0.9610$_{\uparrow 0.03/\uparrow 0.0002}$	33.91/0.9201$_{\uparrow 0.13/\uparrow 0.0012}$	34.15/0.9268$_{\uparrow 1.89/\uparrow 0.0262}$	39.41/0.9802$_{\uparrow 0.54/\uparrow 0.0026}$
×3	EDSR(CVPRW'17)[26]	43680.0K	34.65/0.9282	30.52/0.8462	29.25/0.8091	-
	EDSR-BAM	43688.3K	35.26/0.9417$_{\uparrow 0.61/\uparrow 0.0135}$	31.15/0.8607$_{\uparrow 0.63/\uparrow 0.0145}$	29.73/0.8212$_{\uparrow 0.48/\uparrow 0.0121}$	34.04/0.9495
	CARN(ECCV'18)[25]	1592.0K	34.29/0.9255	30.29/0.8407	29.06/0.8034	-
	CARN-BAM	1593.7K	34.93/0.9392$_{\uparrow 0.64/\uparrow 0.0137}$	30.93/0.8560$_{\uparrow 0.64/\uparrow 0.0153}$	29.57/0.8171$_{\uparrow 0.51/\uparrow 0.0137}$	33.52/0.9456
	CARN-M(ECCV'18)[25]	1161.3K	33.99/0.9236	30.08/0.8367	28.91/0.8000	-
	CARN-M-BAM	1163.0K	34.81/0.9383$_{\uparrow 0.82/\uparrow 0.0147}$	30.84/0.8540$_{\uparrow 0.76/\uparrow 0.0173}$	29.49/0.8150$_{\uparrow 0.58/\uparrow 0.0150}$	33.31/0.9438
	MSRN(ECCV'18)[22]	6115.0K	34.38/0.9262	30.34/0.8395	29.08/0.8041	33.44/0.9427
	MSRN-BAM	6119.5K	35.20/0.9412$_{\uparrow 0.82/\uparrow 0.0150}$	31.10/0.8590$_{\uparrow 0.76/\uparrow 0.0195}$	29.66/0.8195$_{\uparrow 0.58/\uparrow 0.0154}$	33.90/0.9483$_{\uparrow 0.46/\uparrow 0.0056}$
	s-LWSR$_{32}$(TIP'19)[23]	580.4K	-	-	-	-
	s-LWSR$_{32}$-BAM	580.6K	34.98/0.9395	30.94/0.8569	29.58/0.8175	33.50/0.9459
	AWSRN(CVPR'19)[24]	1476.1K	34.52/0.9281	30.38/0.8426	29.16/0.8069	33.85/0.9463
	AWSRN-BAM	1476.5K	35.13/0.9408$_{\uparrow 0.61/\uparrow 0.0127}$	31.09/0.8590$_{\uparrow 0.71/\uparrow 0.0164}$	29.65/0.8191$_{\uparrow 0.49/\uparrow 0.0132}$	33.82/0.9478$_{\uparrow 0.03/\uparrow 0.0015}$
×4	EDSR(CVPRW'17)[26]	43089.9K	32.46/0.8968	28.80/0.7876	27.71/0.7420	-
	EDSR-BAM	43098.2K	32.46/0.8986$_{\uparrow 0.00/\uparrow 0.0018}$	28.92/0.7901$_{\uparrow 0.12/\uparrow 0.0025}$	28.63/0.7688$_{\uparrow 0.92/\uparrow 0.0268}$	31.49/0.9219
	CARN(ECCV'18)[25]	1592.0K	32.13/0.8940	28.60/0.7810	27.58/0.7350	-
	CARN-BAM	1593.7K	32.17/0.8944$_{\uparrow 0.04/\uparrow 0.0004}$	28.72/0.7839$_{\uparrow 0.12/\uparrow 0.0029}$	28.46/0.7628$_{\uparrow 0.88/\uparrow 0.0278}$	30.81/0.9140
	CARN-M(ECCV'18)[25]	1161.3K	31.92/0.8900	28.42/0.7760	27.44/0.7300	-
	CARN-M-BAM	1163.0K	31.98/0.8915$_{\uparrow 0.06/\uparrow 0.0015}$	28.54/0.7792$_{\uparrow 0.08/\uparrow 0.0032}$	28.35/0.7593$_{\uparrow 0.91/\uparrow 0.0293}$	30.44/0.9091
	MSRN(ECCV'18)[22]	6082.6K	32.07/0.8903	28.60/0.7751	27.52/0.7273	30.17/0.9034
	MSRN-BAM	6078.0K	32.14/0.8940$_{\uparrow 0.07/\uparrow 0.0037}$	28.66/0.7830$_{\uparrow 0.06/\uparrow 0.0079}$	28.45/0.7626$_{\uparrow 0.93/\uparrow 0.0353}$	30.69/0.9122$_{\uparrow 0.52/\uparrow 0.0088}$
	s-LWSR$_{32}$(TIP'19)[23]	571.1K	32.04/0.8930	28.15/0.7760	27.52/0.7340	-
	s-LWSR$_{32}$-BAM	571.3K	32.07/0.8935$_{\uparrow 0.03/\uparrow 0.0005}$	28.70/0.7843$_{\uparrow 0.55/\uparrow 0.0083}$	28.48/0.7636$_{\uparrow 0.96/\uparrow 0.0296}$	30.82/0.9137
	AWSRN(CVPR2019)[24]	1587.1K	32.27/0.8960	28.69/0.7843	27.64/0.7385	30.72/0.9109
	AWSRN-BAM	1587.4K	32.29/0.8962$_{\uparrow 0.02/\uparrow 0.0002}$	28.80/0.7863$_{\uparrow 0.11/\uparrow 0.0020}$	28.54/0.7658$_{\uparrow 0.90/\uparrow 0.0273}$	31.12/0.9172$_{\uparrow 0.40/\uparrow 0.0063}$

The parameter amount is calculated based on a 240×360 RGB image. The growth or decline of PSNR/SSIM compared with the corresponding control group is indicated by↑ and ↓ respectively (**the higher the better**). The best two results are highlighted in red and blue colors, respectively

Table 3 Control experiment results on 4 SISR networks with attention, the lightweight SISR networks are marked in **bold black**

Scale	Method	Param	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Manga109 PSNR/SSIM
×2	RCAN(ECCV'18)[29]	15444.7K	38.27/0.9617	34.23/0.9225	32.46/0.9031	39.44/0.9786
	RCAN-BAM	15441.7K$_{\downarrow 3.0K}$	38.32/0.9618$_{\uparrow 0.05/\uparrow 0.0001}$	34.25/0.9230$_{\uparrow 0.02/\uparrow 0.0005}$	34.29/0.9282$_{\uparrow 1.83/\uparrow 0.0251}$	39.86/0.9806$_{\uparrow 0.42/\uparrow 0.0020}$
	IMDN(ACM MM'19)[28]	694.4K	38.00/0.9605	33.63/0.9177	32.19/0.8996	38.88/0.9774
	IMDN-BAM	694.3K$_{\downarrow 0.1K}$	38.03/0.9607$_{\uparrow 0.03/\uparrow 0.0002}$	33.73/0.9183$_{\uparrow 0.10/\uparrow 0.0006}$	34.05/0.9259$_{\uparrow 1.86/\uparrow 0.0263}$	39.33/0.9800$_{\uparrow 0.45/\uparrow 0.0026}$
	PAN(ECCVW'20)[27]	261.4K	38.00/0.9605	33.59/0.9181	32.18/0.8997	38.70/0.9773
	PAN-BAM	261.0K$_{\downarrow 0.4K}$	38.00/0.9606$_{\uparrow 0.00/\uparrow 0.0001}$	33.70/0.9181$_{\uparrow 0.11/\uparrow 0.0000}$	34.03/0.9255$_{\uparrow 1.85/\uparrow 0.0258}$	39.19/0.9797$_{\uparrow 0.31/\uparrow 0.0024}$
	DRLN(TPAMI'20)[30]	34430.2K	38.27/0.9616	34.28/0.9231	32.44/0.9028	39.58/0.9786
	DRLN-BAM	34163.4K$_{\downarrow 266.8K}$	38.32/0.9619$_{\uparrow 0.05/\uparrow 0.0003}$	34.42/0.9237$_{\uparrow 0.14/\uparrow 0.0006}$	34.33/0.9284$_{\uparrow 1.89/\uparrow 0.0256}$	40.41/0.9820$_{\uparrow 0.83/\uparrow 0.0034}$
×3	RCAN(ECCV2018)[29]	15629.3K	34.74/0.9299	30.65/0.8482	29.32/0.8111	34.44/0.9499
	RCAN-BAM	15626.3K$_{\downarrow 3.0K}$	35.36/0.9424$_{\uparrow 0.62/\uparrow 0.0125}$	31.22/0.8611$_{\uparrow 0.57/\uparrow 0.0129}$	29.75/0.8215$_{\uparrow 0.43/\uparrow 0.0104}$	34.07/0.9501$_{\downarrow 0.37/\uparrow 0.0002}$
	IMDN(ACM MM'19)[28]	703.1K	34.36/0.9270	30.32/0.8417	29.09/0.8046	33.61/0.9445
	IMDN-BAM	703.0K$_{\downarrow 0.1K}$	35.06/0.9405$_{\uparrow 0.70/\uparrow 0.0135}$	30.99/0.8568$_{\uparrow 0.67/\uparrow 0.0151}$	29.61/0.8181$_{\uparrow 0.52/\uparrow 0.0135}$	33.80/0.9474$_{\uparrow 0.19/\uparrow 0.0029}$
	PAN(ECCVW'20)[27]	261.4K	34.40/0.9271	30.36/0.8423	29.11/0.8050	33.61/0.9448
	PAN-BAM	261.0K$_{\downarrow 0.4K}$	34.77/0.9379$_{\uparrow 0.37/\uparrow 0.108}$	30.88/0.8545$_{\uparrow 0.52/\uparrow 0.0122}$	29.50/0.8145$_{\uparrow 0.39/\uparrow 0.0095}$	33.19/0.9435$_{\downarrow 0.42/\downarrow 0.0013}$
	DRLN(TPAMI'20)[30]	34614.8K	34.78/0.9303	30.73/0.8488	29.36/0.8117	34.71/0.9509
	DRLN-BAM	34348.1K$_{\downarrow 266.7K}$	35.42/0.9431$_{\uparrow 0.64/\uparrow 0.0128}$	31.32/0.8628$_{\uparrow 0.59/\uparrow 0.0140}$	29.81/0.8224$_{\uparrow 0.45/\uparrow 0.0107}$	34.73/0.9527$_{\uparrow 0.02/\uparrow 0.0018}$
×4	RCAN(ECCV'18)[29]	15592.4K	32.63/0.9002	28.87/0.7889	27.77/0.7436	31.22/0.9173
	RCAN-BAM	15589.4K$_{\downarrow 3.0K}$	32.64/0.9003$_{\uparrow 0.01/\uparrow 0.0001}$	29.00/0.7918$_{\uparrow 0.13/\uparrow 0.0029}$	28.69/0.7710$_{\uparrow 0.92/\uparrow 0.0274}$	31.09/0.9209$_{\uparrow 0.13/\uparrow 0.0036}$
	IMDN(ACM MM'19)[28]	715.2K	32.21/0.8948	28.58/0.7811	27.56/0.7353	30.47/0.9084
	IMDN-BAM	715.1K$_{\downarrow 0.1K}$	32.24/0.8955$_{\uparrow 0.03/\uparrow 0.0007}$	28.75/0.7847$_{\uparrow 0.17/\uparrow 0.0036}$	28.51/0.7642$_{\uparrow 0.95/\uparrow 0.0289}$	31.02/0.9154$_{\uparrow 0.55/\uparrow 0.0070}$
	PAN(ECCVW'20)[27]	272.4K	32.13/0.8948	28.61/0.7822	27.59/0.7363	30.51/0.9095
	PAN-BAM	271.6K$_{\downarrow 0.8K}$	32.14/0.8941$_{\uparrow 0.01/\downarrow 0.0007}$	28.69/0.7831$_{\uparrow 0.08/\uparrow 0.0009}$	28.46/0.7623$_{\uparrow 0.87/\uparrow 0.0260}$	30.79/0.9131$_{\uparrow 0.28/\uparrow 0.0036}$
	DRLN(TPAMI'20)[30]	34577.9K	32.63/0.9002	28.94/0.7900	27.83/0.7444	31.54/0.9196
	DRLN-BAM	34311.2K$_{\downarrow 266.7K}$	32.66/0.9005$_{\uparrow 0.03/\uparrow 0.0003}$	29.08/0.7925$_{\uparrow 0.06/\uparrow 0.0025}$	28.75/0.7714$_{\uparrow 0.92/\uparrow 0.0270}$	31.90/0.9257$_{\uparrow 0.36/\uparrow 0.0061}$

The parameter amount is calculated based on a 240×360 RGB image, and its growth or decline compared with the corresponding control group is indicated by ↑ and ↓ respectively (**the lower the better**). The growth or decline of PSNR/SSIM compared with the corresponding control group is indicated by ↑ and ↓ respectively (**the higher the better**). The best two results are highlighted in red and blue colors respectively

incorporation of BAM can further increase their performance indicated by PSNR and SSIM metrics by only adding a small number of parameters. As illustrated in Fig. 4, for the original networks with attention, BAM not only reduces the number of parameters but also improves the model performance. This thoroughly proves the efficiency and robustness of BAM. The calculation of Param Decrement and PSNR Increment in Fig. 4 are expressed as following:

$$\text{ParamDecrement} = (P_{\text{ori}} - P_{\text{BAM}})/P_{\text{ori}} \cdot 1000‰, \tag{16}$$

$$\text{PSNRIncrement} = \text{PSNR}_{\text{BAM}} - \text{PSNR}_{\text{ori}}, \tag{17}$$

where P_{ori} and P_{BAM} represent the parameter amounts of the control and experimental groups, respectively, PSNR_{ori} and PSNR_{BAM} stand for the PSNR results of the control and experimental groups separately.

Figure 5 displays the ×4 SR results of five groups of SISR networks with or without BAM on a representative image selected from the BSD100 dataset. For the three networks without attention, EDSR, CARN and AWSRN, their BAM version only increases a few parameters but greatly improves the metrics. Especially for EDSR-BAM, which achieves a very obvious visual improvement compared to the control group. For the two lightweight networks with attention, IMDN and PAN, the BAM replacement increases the SR quality while reducing the number of parameters.

Figure 6 displays the visual perception comparison between the ×4 SR results of the experimental group and the control group for IMDN and DRLN. IMDN and DRLN can stand for the current lightweight and heavyweight top-level networks, respectively. As can be seen, the experimental group is capable of recovering more detailed information and has a significant improvement on the aliased texture areas, such as alphabet letters, Chinese characters, cloth textures, hairs, and even facial wrinkles. Whether for a lightweight network such as IMDN or a heavyweight network such as DRLN, the BAM replacement can further improve the visual quality of SR results with the reduced parameters. IMDN-BAM and DRLN-BAM can be utilized as baselines for the follow-up researches. And these two sets of figures thoroughly validate the effectiveness of BAM. Figure 7 illustrates the metrics improvement on four lightweight SISR networks of ×3 SR results. The SR results of experimental groups all make a great improvement compared to the control group on PSNR/SSIM metrics. For the two networks without attention, AWSRN and CARN, their BAM versions only increase a few parameters but greatly improves the metrics; for the two lightweight networks with attention, IMDN and PAN, the BAM replacement increases the SR quality while reducing the number of parameters.

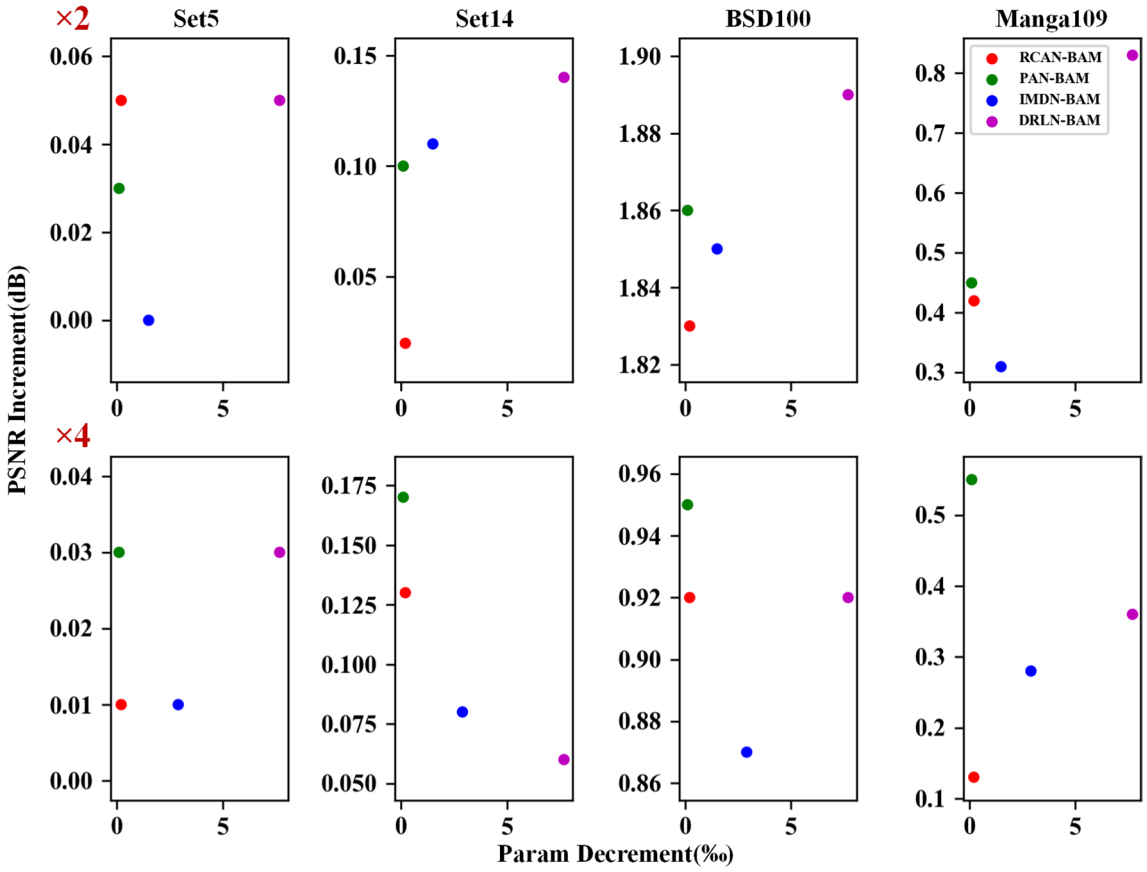


Fig. 4 Under the×2 and×4 upsampling scales, the relationship between the parameter decrement and the PSNR increment of the four SISR networks with attention

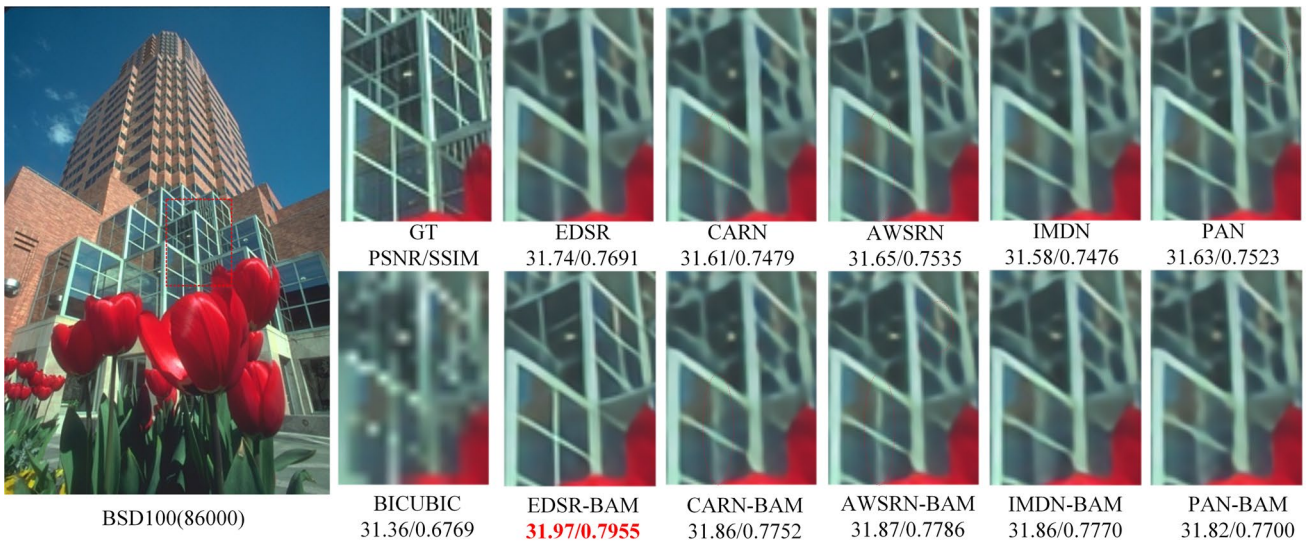


Fig. 5 Comparative experiments of five SISR networks under scaling factors of×4. The best two results are highlighted in red and blue colors, respectively. The red dashed ellipse is used to guide areas where the visual effect is not obvious improved. The improvement between EDSR-BAM and EDSR is significant

Fig. 6 Visual perception comparison of the SR results from IMDN versus IMDN-BAM and DRLN versus DRLN-BAM on the Manga109 dataset under the scale factor of ×4. Comparing their results, we can see that the SISR networks with BAM can generally better restore the fine structures, including cloth textures, alphabet letters, facial wrinkles, hairs and Chinese characters

4.4 Ablation experiments

4.4.1 Comparison with another four attention mechanisms

To verify the efficiency of BAM, we conduct ablation experiments on three scaling factors of ×2, ×3, and ×4 based on the IMDN. Its original attention module, CCA, is replaced with CA, SE, CBAM and BAM, respectively. We evaluate on the five benchmarks of Set5, Set14, BSD100, Urban100 and Manga109 with PSNR and SSIM metrics.

From the results of ablation experiments in Table 4, it can be found that under three scaling factors, all the networks using BAM obtain the highest SSIM and PSNR metrics on five benchmark datasets. Moreover, after replacing CCA with SE or CBAM, the performance of the model is worse than the original version, reflecting that the effective attention mechanism on classification tasks does not necessarily have the same effect on the SISR task. Moreover, Fig. 8 shows the ×4 SR results of five attention mechanisms used in Table 4, where we can see that BAM maintains a great balance between noise suppression and high-frequency texture detail recovery. BAM is the best one to recover the texture aliasing area among the five attention mechanisms.

4.4.2 Minimization verification

To verify the minimalism of BAM and its two basic modules, ACAM and MSAM, and the efficiency of the parallel structure of BAM, we conduct ablation experiments on three scaling factors of ×2, ×3, and ×4 based on the lightweight network IMDN. Its original attention module, CCA, is replaced with ACAM, ACAM$^+$, MSAM, MSAM$^+$, BAM, BAM$^+$, and CBAM, respectively. We evaluate on the four benchmarks of Set5, Set14, BSD100, and Manga109 with PSNR and SSIM metrics.

To show the minimalism of ACAM and MSAM, their results are compared with the ones of ACAM$^+$ and MSAM$^+$, respectively. As for BAM, we compare it with BAM$^+$. To verify that the parallel structure is more balanced than the series structure so as to generate attention more reasonably, BAM is compared with CBAM which cascades channel and spatial attentions. From the results of ablation experiments in Table 5, it can be found that under

Fig. 7 Metrics comparison of the ×3 SR results of 4 lightweight SISR networks, the best results are marked in bold red. The highest PSNR and SSIM scores are all achieved by AWSRN-BAM

Table 4 Ablation experiment results on Set5, Set14, BSD100, and Manga109 under three scaling factors of ×2, ×3 and ×4 for IMDN with another four attention mechanisms

Scale	Method	Param	GFLOPs	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
×2	IMDN(CCA)	694.4K	70.000	38.00/0.9605	33.63/0.9177	32.19/0.8996	32.17/0.9238	38.88/0.9774
	IMDN(CA)	694.4K	70.000	37.86/0.9602	33.62/0.9173	33.94/0.9250	31.64/0.9234	38.97/0.9793
	IMDN(SE)	694.0K	70.000	37.87/0.9602	33.60/0.9173	33.93/0.9249	31.69/0.9238	38.95/0.9792
	IMDN(CBAM)	694.6K	70.086	37.87/0.9602	33.54/0.9168	33.89/0.9244	31.64/0.9234	38.62/0.9786
	IMDN(BAM)	694.3K	70.027	38.03/0.9607	33.73/0.9183	34.05/0.9259	32.18/0.9283	39.33/0.9800
×3	IMDN(CCA)	703.1K	70.831	34.36/0.9270	30.32/0.8417	29.09/0.8046	28.17/0.8519	33.61/0.9445
	IMDN(CA)	703.1K	70.831	34.91/0.9392	30.91/0.8558	29.54/0.8168	28.92/0.8663	33.48/0.9456
	IMDN(SE)	702.7K	70.831	34.93/0.9396	30.92/0.8558	29.54/0.8170	28.94/0.8667	33.53/0.9456
	IMDN(CBAM)	703.2K	70.917	34.92/0.9393	30.91/0.8550	29.54/0.8164	28.82/0.8678	33.26/0.9444
	IMDN(BAM)	703.0K	70.858	35.06/0.9405	30.99/0.8568	29.61/0.8181	29.11/0.8698	33.80/0.9474
×4	IMDN(CCA)	715.2K	71.994	32.21/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838	30.47/0.9084
	IMDN(CA)	715.2K	71.994	32.01/0.8921	28.59/0.7815	28.39/0.7611	25.74/0.7749	30.63/0.9111
	IMDN(SE)	714.8K	71.994	32.07/0.8930	28.62/0.7822	28.41/0.7618	25.77/0.7760	30.70/0.9118
	IMDN(CBAM)	715.4K	72.080	32.18/0.8941	28.68/0.7829	28.45/0.7627	25.84/0.7788	30.59/0.9114
	IMDN(BAM)	715.1K	72.021	32.24/0.8955	28.75/0.7847	28.51/0.7642	26.08/0.7854	31.02/0.9154

The parameter amount and computational load are calculated based on an RGB image with the size of 240 × 360. The best two results are highlighted in red and blue colors, respectively

three scaling factors of ×2, ×3, and ×4, all the experiments using BAM obtain the highest SSIM and PSNR metrics on four benchmarks. Compared with ACAM$^+$ and MSAM$^+$, the networks with ACAM and MSAM are more lightweight but achieve higher PSNR and SSIM. This verifies that the use of only AvgPool to extract channel attention

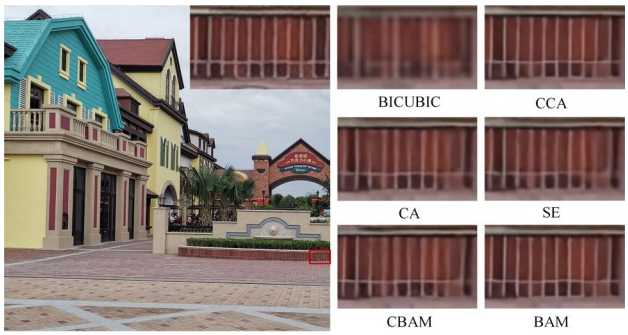


Fig. 8 Comparison of ×4 SR results by five attention mechanisms on the realSR7 dataset proposed in this paper

information and only MaxPool to extract spatial attention information is effective for the SISR task. The comparison between BAM and BAM$^+$ shows the minimalism of BAM.

4.4.3 Quantitative verification

To quantitatively verify that the insertion of BAM improves the network's SR performance, we conduct further experiments on IMDN-BAM. We randomly select one of the six BAMs in IMDN-BAM, extract its input and output feature maps, and then calculate the definition evaluation function SMD2 [45] values for each feature map in input and output feature maps separately, finally, obtain the average values of these two sets of data. The larger the SMD2 value, the richer the texture. The expression of SMD2 is as follows:

$$\mathrm{SMD2} = \sum_{h,w=1,1}^{H,W} \frac{|f(h,w)-f(h-1,w)| \cdot |f(h,w)-f(h,w-1)|}{255\mathrm{HW}}, \tag{18}$$

in which, H and W are the pixel height and width of each feature map, and $f(h,w)$ is the gray value of the feature map at pixel coordinate (h,w). Figure 9 shows the SMD2 values of each feature map in input and output feature maps in a certain BAM layer of IMDN-BAM, after the input feature maps are assigned attention by BAM, the average SMD2 indicators of the output feature maps are all improved, and the improvements are {0.0136, 0.0143, 0.0274,0.0189, 0.0244, 0.0242, 0.0128, 0.0364}, respectively. The experiment results reflect that BAM has indeed improved the clarity and texture richness of feature maps, and quantitatively verify the efficient performance of BAM.

4.5 Speed comparison

To further prove the minimalism and efficiency of BAM, we select IMDN and DRLN as the representatives of lightweight and heavyweight SISR networks respectively, and compare the FPS between the experimental group and the control group with multiple input scales. Under each input scale, we count the average inference time of 700 images to calculate FPS, and it can be expressed as following

$$\mathrm{FPS} = \mathrm{Frames}/\mathrm{Time}_{\mathrm{Frames}}, \tag{19}$$

where Frames is the number of images, and $\mathrm{Time}_{\mathrm{Frames}}$ is the total time utilized for inference.

Figure 10 shows the FPS curves of IMDN-BAM and IMDN, DRLN-BAM and DRLN under different input scales

Table 5 Minimization verification experiment results on Set5, Set14, BSD100, and Manga109 under scaling factors of ×2, ×3 and ×4 for IMDN

Scale	Method	Params	GFLOPs	Set5	Set14	BSD100	Manga109
				PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
×2	IMDN(ACAM)	694.0K	70.000	37.98/0.9606	33.69/0.9181	34.02/0.9256	39.22/0.9798
	IMDN(ACAM$^+$)	694.0K	70.033	37.97/0.9605	33.66/0.9177	33.97/0.9252	38.68/0.9787
	IMDN(MSAM)	691.2K	69.994	37.96/0.9605	33.68/0.9181	34.02/0.9257	38.93/0.9794
	IMDN(MSAM$^+$)	691.5K	70.019	37.93/0.9604	33.62/0.9173	33.97/0.9253	38.85/0.9792
	IMDN(BAM)	694.3K	70.027	38.03/0.9607	33.73/0.9183	34.05/0.9259	39.33/0.9800
	IMDN(BAM$^+$)	694.6K	70.086	37.94/0.9604	33.64/0.9176	33.98/0.9251	38.83/0.9788
×3	IMDN(ACAM)	702.7K	70.831	34.94/0.9394	30.94/0.8559	29.57/0.8171	33.55/0.9457
	IMDN(ACAM$^+$)	702.7K	70.864	34.92/0.9394	30.91/0.8555	29.56/0.8166	33.12/0.9438
	IMDN(MSAM)	699.9K	70.825	34.95/0.9395	30.95/0.8567	29.57/0.8173	33.50/0.9458
	IMDN(MSAM$^+$)	700.2K	70.850	34.94/0.9394	30.95/0.8566	29.56/0.8171	33.48/0.9456
	IMDN(BAM)	703.0K	70.858	35.06/0.9405	30.99/0.8568	29.61/0.8181	33.80/0.9474
	IMDN(BAM$^+$)	703.3K	70.917	35.00/0.9401	30.94/0.8565	29.57/0.8179	33.40/0.9458
×4	IMDN(ACAM)	714.8K	71.994	32.20/0.8946	28.71/0.7838	28.47/0.7633	30.85/0.9140
	IMDN(ACAM$^+$)	714.8K	72.027	32.12/0.8935	28.69/0.7828	28.46/0.7624	30.67/0.9115
	IMDN(MSAM)	712.0K	71.988	32.09/0.8937	28.67/0.7830	28.46/0.7628	30.76/0.9127
	IMDN(MSAM$^+$)	712.3K	72.013	32.08/0.8933	28.63/0.7824	28.43/0.7622	30.70/0.9116
	IMDN(BAM)	715.1K	72.021	32.24/0.8955	28.75/0.7847	28.51/0.7642	31.02/0.9154
	IMDN(BAM$^+$)	715.4K	72.080	32.21/0.8947	28.70/0.7834	28.48/0.7630	30.73/0.9122

The parameter amount and computational load are calculated based on a 240 × 360 RGB image. For each scaling factor group, the best and the second best results are **highlighted** In Red And Blue Colors, respectively

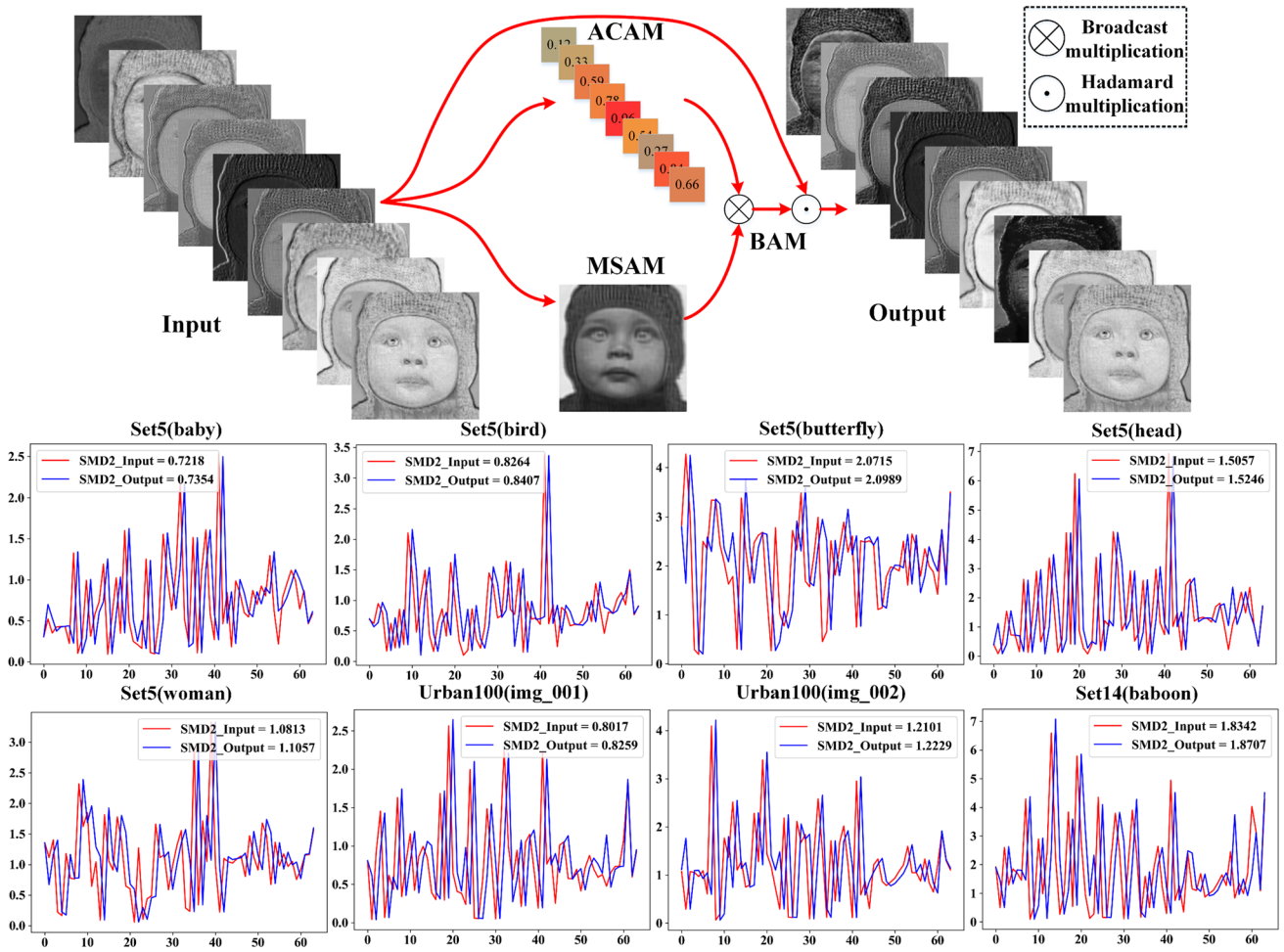


Fig. 9 Texture richness comparison between Input and Output feature maps of BAM based on IMDN-BAM, under×2 upsampling scales. We draw the SMD2 value curves of each feature map in Input (red) and Output (blue) feature maps of BAM, and use the average SMD2 value of each curve to measure their texture richness, the higher the better, the improvement of SMD2 index before and after BAM operation quantitatively illustrate the effectiveness of our proposed BAM

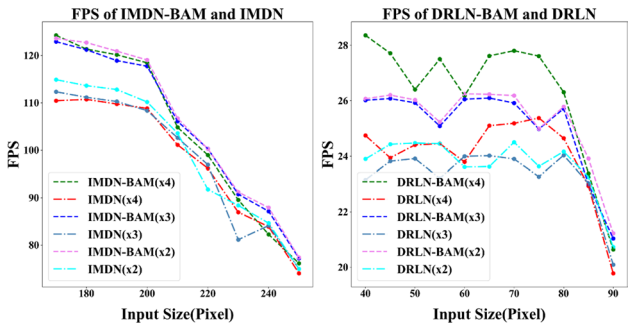


Fig. 10 Speed comparison between IMDN-BAM and IMDN, DRLN-BAM and DRLN on 2080Ti under×2,×3, and×4 upsampling scales. As shown in Fig. 9, experimental groups are faster than the control groups under each input scale and each upsampling factor

on 2080Ti. It can be seen that the experimental group has the advantage in inference speed as well, and the speed advantage gets more obvious when the scale of the input image is smaller. When the input image size is 200×200, the FPS of proposed IMDN-BAM exceeds IMDN {8.1%, 8.7%, 8.8%} under the three SR magnifications of×2,×3, and×4, respectively. When the input image scale is 60×60, the FPS of proposed DRLN-BAM exceeds DRLN {11.0%, 8.8%, 10.1%} under×2,×3, and×4. The above experimental results illustrate that BAM can accelerate the inference speed while improving network performance indicators, which has significant application value for the landing of lightweight networks on mobile terminals.

5 Conclusion

Aiming at the problem that textures are often overwhelmed by extreme noise in SISR tasks, we propose an attention mechanism BAM, consisting of ACAM and MSAM in parallel. ACAM can well suppress extreme noise in large-scale feature maps, while MSAM focuses more on high-frequency texture details. The overall parallel structure of BAM enables ACAM and MSAM to optimize each other during the back propagation process, so as to obtain an optimal balance between noise suppression and texture restoration. In addition, the parallel structure brings in a faster inference speed. BAM is a universal attention mechanism research for SISR tasks. This research can improve the performance of SISR networks without attention, and provide a strong baseline for the subsequent attention mechanism works for SISR. The control experimental results strongly prove that BAM can efficiently improve the performance of state-of-the-art SISR networks and further reduce the parameter amounts and improve the inference speed for those originally with attention. The ablation experimental results illustrate the efficiency of BAM. What's more, BAM demonstrates higher capability to restore the texture aliasing area in real scenes on the realSR7 dataset proposed in this paper.

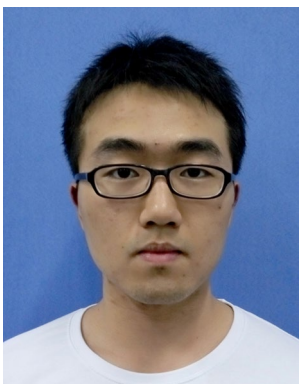
Acknowledgements The authors would like to thank the Associate Editor and the Reviewers for their constructive comments. This work is supported by OPPO Research Institute. And this work is pre-print at https://arxiv.org/abs/2104.07566

Availability of code and data The source code is released at https://github.com/dandingbudanding/BAM.

References

1. Wang, Z., Chen, J., Hoi, S.: Deep learning for image super-resolution: a Survey. IEEE Trans. Pattern Anal. Mach. Intell. **1–1**, 99 (2020)
2. Anwar, S., Khan, S., Barnes, N.: A deep journey into super-resolution: a survey. arXiv, preprint arXiv:1904.07523 (2019)
3. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans. Image Process. **26**, 3142–3155 (2017)
4. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep CNN denoiser prior for image restoration. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017)
5. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. **38**, 295–307 (2016)
6. Shi, W., Caballero, J., Huszár, F., Totz, J., Wang, Z.: Real-time single image and video super-resolution using an efficient subpixel convolutional neural network. In: CVPR (2016)
7. Chao, D., Chen, C.L., Tang, X.: Accelerating the super-resolution convolutional neural network. In: ECCV (2016)
8. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR (2016)
9. Jiao, J., Tu, W.C., Liu, D., He, S., Lau, R., Huang, T.S.: FormNet: formatted learning for image restoration. IEEE Trans. Image Process. **29**(99), 6302–6314 (2020)
10. Fan, Y., Shi, H., Yu, J., Ding, L., Huang, T.S.: Balanced two-stage residual networks for image super-resolution. In: CVPRW (2017)
11. Ying, T., Jian, Y., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR (2017)
12. Tai, Y., Yang, J., Liu, X., Xu, C.: MemNet: a persistent memory network for image restoration. In: IEEE Computer Society (2017)
13. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: CVPR (2016)
14. Abbass, M.Y.: Residual dense convolutional neural network for image super-resolution. Optik. (2020). https://doi.org/10.1016/j.ijleo.2020.165341
15. Haris, M., Shakhnarovich, G., Ukita, N.: Deep back-projection networks for super-resolution. In: CVPR (2018)
16. Tong, T., Li, G., Liu, X., Gao, Q.: Image super-resolution using dense skip connections. In: IEEE International Conference on Computer Vision (2017)
17. Park, S., Son, H., Cho, S., Hong, K., Lee, S.: SRFeat: single image super-resolution with feature discrimination, pp. 455–471. Springer International Publishing, Cham (2018)
18. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Loy, C.C., Qiao, Y., Tang, X.: ESRGAN: enhanced super-resolution generative adversarial networks. Springer, Cham (2018)
19. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**, 600–612 (2004)
20. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: CBAM: convolutional block attention module. In ECCV, pp. 3–19 (2018)
21. Jie, H., Li, S., Gang, S., Albanie, S.: Squeeze-and-excitation networks. IEEE Trans. Pattern Anal. 7132–7141 (2017)
22. Qin, J., Huang, Y., Wen, W.: Multi-scale feature fusion residual network for single image super-resolution. Neurocomputing **379**, 334–342 (2020)
23. Li, B., Wang, B., Liu, J., Qi, Z., Shi, Y.: s-LWSR: super lightweight super-resolution network. IEEE Trans. Image Process. **1**, 99 (2020)
24. Wang, C., Li, Z., Shi, J.: Lightweight image super-resolution with adaptive weighted learning network. arXiv preprint arXiv:1904.02358 (2019)
25. Ahn, N., Kang, B., Sohn, K.A.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: ECCV (2018)
26. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: CVPRW (2017)
27. Zhao, H., Kong, X., He, J., Qiao, Y., Dong, C.: Efficient image super-resolution using pixel attention. arXiv preprint arXiv:2010.01073 (2020)
28. Hui, Z., Gao, X., Yang, Y., Wang, X.: Lightweight image super-resolution with information multi-distillation network. In: ACM MM, pp. 2024–2032 (2019)
29. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: ECCV, pp. 286–301 (2018)
30. Anwar, S., Barnes, N.: Densely residual Laplacian super-resolution. IEEE Trans. Pattern Anal. Mach. Intell. 99 (2020)
31. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. Springer, Cham (2015)
32. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: inverted residuals and linear bottlenecks. In: CVPR (2018)

33. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: CVPR (2015)
34. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
35. Agustsson, E., Timofte, R.: NTIRE 2017 challenge on single image super-resolution: dataset and study. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2017)
36. Bevilacqua, M., Roumy, A., Guillemot, C., Morel, A.: Low-complexity single image super-resolution based on nonnegative neighbor embedding. In: BMVC (2012)
37. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: International Conference on Curves and Surfaces (2010)
38. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: IEEE International Conference on Computer Vision (2002)
39. Narita, R., Tsubota, K., Yamasaki, T., and Aizawa, K.: Sketch-based manga retrieval using deep features. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), pp. 49–53 (2017)
40. Huang J., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR, pp. 5197–5206 (2015)
41. Huang, G., Liu, Z., Laurens, V., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR, pp. 4700–4708 (2017)
42. Timofte, R., Agustsson, E., et al.: NTIRE 2017 challenge on single image super-resolution: methods and results. In: CVPRW, pp. 1110–1121 (2017)
43. Yang, J.W.J.H.: Image super-resolution via sparse representation. IEEE Trans. Image Process. **19**, 2861–2873 (2010)
44. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. **33**, 898–916 (2011)
45. Wang, F., Cao, P., Zhang, Y., Hu, H., Yang, Y.: A machine vision method for correction of eccentric error based on adaptive enhancement algorithm. IEEE Trans. Instrum. Meas. **70**, 1–11 (2020)



Fanyi Wang was born in China in 1995, He received the B.E. Degree in measurement and control technology and instrumentation from Harbin Institute of Technology, Harbin, China in 2017. He is currently pursuing Professional Ph.D. Degree in Zhejiang University, Zhejiang, China, majoring in optical engineering. His current research interests are machine vision and precision measurement.

Haotian Hu was born in China in 1996, He received the Bachelor Degree in Optoelectronic Information Science and Engineering from Chongqing Normal University, Chongqing, China in 2018. He is currently pursuing Master Degree in Zhejiang University, Zhejiang, China, majoring in optics. His current research interests is machine vision.

Cheng Shen was born in China in 1995. He received the B.E. and M.E. degree in Instrument Science and Technology from Harbin Institute of Technology, Harbin, China, in 2016 and 2018. He is currently pursuing the Ph.D. degree in Department of Electrical Engineering, California Institute of Technology, Pasadena, USA. His research interests include computational imaging, phase retrieval, computer vision and deep learning.

Tianpeng Feng researcher of OPPO Research Institute.

Yandong Guo researcher of OPPO Research Institute.