Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 2, No. 1, 2022

Lightweight Multi-Attention Fusion Network for Image Super-Resolution

Xinyu Wang, Jing Zhang

School of Computer Science and Technology, Tiangong University, Tianjin, China

Abstract: Single image super-resolution reconstruction (SISR) is one of the important techniques in computer vision and image processing. Most of the existing SISR methods adopt equal processing for different spatial domains and channel domains, resulting in a large amount of computational resources wasted on unimportant features. In order to address these problems, a novel lightweight multi-attention fusion network (LMAFN) is proposed, in which the multiple attention fusion block allocates computational resources more efficiently by capturing the weight information implied by the channel domain and the spatial domain separately, thus effectively reducing the number of parameters. The synthetic channel attention block in the multiple attention fusion block makes full use of inter-channel correlation by introducing both global standard deviation pooling and maximum pooling. Global features are fused through residual linking to alleviate the problem of high frequency information loss. Experimental results on several benchmark datasets show that the proposed method effectively reduces the number of parameters and computational effort without excessive loss of reconstruction performance, and achieves better performance than the compared models.

Keywords: Attention mechanism; Image super-resolution; Lightweight neural network.

1. Introduction

Image super-resolution reconstruction is the process of restoring a low-resolution image to a high-resolution image by algorithms and is one of the key techniques in computer vision and image processing. The concept has been of great academic interest since it was proposed. In 2015, Dong et al. [1] proposed SRCNN by combining SISR with CNN. And then more complex architectures are proposed to improve the performance of SR methods, such as SRGAN [2]. However, it is difficult to directly apply them in portable mobile devices due to their computational and parameter constraints. The lightweight research starts with FSRCNN [3], which directly applies the SR network to the LR image with removing costly up sampling layers to reduce training time. DRCN [4] was the first to apply a recursive algorithm to SISR, and reduce parameters by reusing part of the parameter's multiple times. Later, DRRN [5] utilizes recurrent layers to reduce parameters while maintaining the depth of the network. EDSR [6] modifies the structure of the residual block by design and removes the BN layer. It could save about 40% of GPU memory space. LapSRN [7] uses a pyramidal framework to gradually increase the image size so that SR image can be performed efficiently at very low resolution. CBPN [8] is proposed to replace the previous up-down projection module with a pixel shuffle layer.

With the further development of SISR, many issues arise. Firstly, most available CNN based models are primarily use multiple stacked convolution operations and increase the convolution kernel to enhance the image reconstruction's quality, but at the same time bring a large amount of computation. Secondly, the majority of current CNN networks obtain features by successive convolution operations while using identical processing for feature of each channel and position, but the importance of the features is different from each other, and equal processing causes a waste of computational resources and consequently a huge memory consumption.

In view of the extant shortcomings mentioned above, inspired by [9] , we propose an improved lightweight network. Among them, the multi-attention fusion block (MAFB) focus on the significance of distinct channel and spatial position features to acquire the corresponding position's weight parameters. What's more, the global feature supervision is established via the long skip connections to speed up network convergence and enable features to be effectively utilized. For the reconstruction part, shallow and deep features are fused by residual learning and sub-pixel convolution techniques, which are complementary to the previous feature extraction and feature fusion parts.

The main contributions of this paper are as follows: (1) We proposes an improved lightweight multiple attention image super-resolution reconstruction network (LAFMN) which reduces the number of parameters and computational effort without losing too much reconstruction performance. (2) We propose MAFB, a module that allocates computational resources according to the importance of features by applying channel attention enhancement and spatial attention enhancement to the features respectively. The synthetic channel attention block (SCAB) introduces two different pooling, which makes full use of the correlation information between channels. At the same time, global residual connectivity is added to fuse low-level and high-level features more effectively. (3) Experimental results on several benchmark datasets show that the proposed method in this paper achieves better performance compared to other existing state-of-the-art models.

2. Related Work

2.1. Single Image Super-Resolution Reconstruction based on Deep Learning

With DONG et al. [1] combining CNN with SISR, more and more neural network-based SISR models have been proposed. SRCNN proposes a simple three-layer network which first up-samples the input low-resolution LR image

using a bicubic interpolation algorithm to obtain the target size image. The next step is to process the input LR image through a three-layer convolutional network to obtain a high-resolution SR image, with the goal of making it alike as possible to the original HR image. The first part of the network extracts multiple patches of the input LR image, each of which is represented as a multidimensional vector by the convolution operation, and all the feature vectors form an n1-dimensional feature mapping matrix, and the process formula is expressed as:

$$F_1(Y) = max(0, W_1 * Y + B_1) \quad (1)$$

In the second part, the n1-dimensional feature mapping matrix is non-linearly mapped by a convolution operation to obtain an n2-dimensional feature matrix, which is expressed by the following equation.

$$F_2(Y) = max(0, W_2 * F_1(Y) + B_2) \quad (2)$$

The last step of the reconstruction process is equivalent to the deconvolution, that is, the n2-dimensional characteristic matrix is restored to HR image. It can be expressed as follows:

$$F(Y) = W_3 * F_2(Y) + B_3 \quad (3)$$

2.2. Image Super-Resolution based on Attention Mechanism

The visual attention mechanism is a unique signal processing mechanism of the human brain. Specifically, by observing the global image, selecting some local focus areas, and then paying more attention to these areas to obtain more detailed information and suppress other useless information. The essence of it is to learn a weight distribution for image features which will be applied to the original features to provide different feature influences for image processing tasks such as image classification, image recognition, etc.

Considering that the feature importance extracted by different convolution kernels is different, SENet [10] network introduce the concept of channel attention to deep neural networks initially. It mainly enhanced the valuable features by learning the weight value of each channel, so as to enhance the learning capacity of the network algorithm effectively while the computing resources are limited. Simultaneously, it can be used for designing a lightweight network architecture. Non-local [11] proposed by Liu et al. aimed to generate a broad range of attention maps by calculating each spatial point's correlation matrix in the feature map, and then uses them to instruct the aggregation of intensive context information. Yet, due to its large calculation, it is difficult to apply in real life. RCAN [9] has significantly improved the reconstruction effect via the use of channel attention in the field of SISR.

3. The Proposed Method

3.1. Framework of the Proposed Model

In this part, we despict the details of the proposed model in Figure 1. First, a convolutional layer is used to extract the shallow features of the LR image, and then further feature extraction is performed by multiple stacked MAFBs. Finally, the HR image is reconstructed by an upsampling module. The convolution is fused and added with the shallow features to obtain the reconstructed HR image. Furthermore, we add the features of the first layer and the features of the last layer through residual connection, and fuse the features of the shallow layer and the deep layer, so as to maintain the influence of the features of shallow layer on the deep layer to the greatest extent.

As shown in Fig. 1, considering the lightweight design of the model, this part only consists of a simple 3×3 convolution. The process can be expressed as:

$$x_0 = f_{FE}(I_{LR}) \quad (4)$$

where $f_{FE}(\cdot)$ represents the 3×3 convolution operation for extracting features from the input LR image and x_0 is the output of this layer.

Then, we use a nonlinear mapping module consisting of 16 stacked MAFBs to generate new feature representations, denoted as:

$$x_n = f^n_{MAFB}(f^{n-1}_{MAFB}(\dots f^0_{MAFB}(x_0)\dots)) \quad (5)$$

where x_n represents the n-th MAFB's output.

Finally, the output feature map x_0 and x_n is used as the input of the reconstruction module, and further feature fusion is performed by a 3×3 convolution after the reconstruction module Additionally, we add a global residual connection in which performs bilinear interpolation on the input, adding its output to the output of the reconstruction module. HR maps that upsample the features into the target size. Finally, we will get:

$$I_{SR} = f_{Fusion}(S(x_0 + x_n)) + f_{UP}(I_{LR}) \quad (6)$$

where $S(\cdot)$ denotes the operation of the upsampling module, $f_{Fusion}(\cdot)$ denotes 3×3 convolution, $f_{UP}(\cdot)$denotes interpolated upsampling, and I_{SR} is the final output of the network.

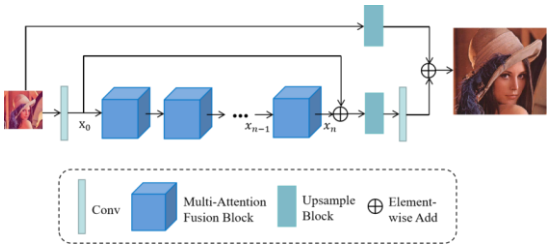

Fig.1 The Proposed Lightweight Multi-Attention Fusion Network's Structure

3.2. Multi-Attention Fusion Block

Inspired by SCNet [12], its structure is improved in this paper. Figure 2 shows that the MAFB consists of two parts: The upper layer performs higher-level feature operations, that is, adding weight distribution to the features. And another layer is used to remain the primitive information. We adopt two attention modules, namely SCAB and modified spatial attention block (MSAB), respectively, in the upper layer.

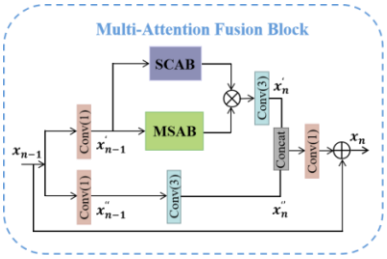

Fig. 2 Multi-Attention Fusion Block

The x_{n-1} and x_n are defined as the input and output of the n-th MAFB separately. Similar to SENet, the two branches of MAFB are first subjected to dimensionality reduction through a 1×1 convolutional layer which called $f_{D-reduction}(\cdot)$. The two pathways in the first layer correspond to the channel attention and spatial attention modules respectively. Given input features:

$$x'_{n-1} = f'_{D-reduction}(x_{n-1}) \quad (7)$$

$$x''_{n-1} = f''_{D-reduction}(x_{n-1}) \quad (8)$$

where the number of channels of x'_{n-1} and x''_{n-1} is only half of that of x_{n-1}.

Next, input the output of the first layer to calculate the channel and spatial position weights. Finally superimpose the learned weight value to each corresponding feature point position by multiplying the corresponding positions, it can be expressed as:

$$u = a \otimes b \quad (9)$$

where a and b represent the weight, values output by the channel attention block and the spatial attention block separately, $\otimes$ is the multiplication operation of the corresponding position weight information, u is the output of the previous layer in MAFB which will be passed through a 3×3 convolution and output as x'_n.

The operation of the lower layer is to convert x''_{n-1} to x''_n. Likewise, a 1×1 convolutional is used for dimensionality reduction, and then a 3×3 convolutional layer for generating x''_n to preserve the original information. Finally, concatenate the outputs x'_n and x''_n of the two layers. The x_n is then generated by the 1×1 convolution. To accelerate training, a shortcut is added to generate the output features of this part.

The main difference between this structure and [12] is that we use two attention mechanisms instead of the pooling and upsampling layers in it. It can be expressed as:

$$x_n = Concat(x'_n, x''_n) + x_{n-1} \quad (10)$$

where $\mathrm{Concat}(\cdot)$ is the concatenate operation, and x_{n-1} is the input (i.e., the output of the previous MAFB)?

3.2.1. Synthetic Channel Attention Block

Inspired by [13], SCAB first replaces global average pooling by global standard deviation pooling and maximum pooling, and performs feature concatenation on its output. The expression formula is:

$$z_c = H_{SAM}(x_c) = Concat(s_c, Max_c) \quad (11)$$

where z_c and Max_c stand for the output of the c-th element and max pooling separately.

To make the most of the detailed information contained in features, the two one-dimensional vectors output by the standard deviation pooling and the maximum pooling are concatenated into one two-dimensional matrix. Fig. 3 shows that the dimensionality reduction and enhancement are carried out respectively through two consecutives 1×1 convolution, and then the weights are normalized by a Sigmoid activation function to apply the weights to the input features. The final output channel is 1×1×C. The process is:

$$a = f(W_i(\sigma(W_s(z_c)))) \quad (12)$$

where a and z_c are the output of the SCAB and the result of two pooling and splicing separately, $\sigma(\cdot)$ and $f(\cdot)$ represent the ReLU function and the Sigmoid activation function; W_s and W_i are 1×1 convolution in which W_s represents the operation to be compressed into one dimension and W_i is used to increase the dimension. They represent compression and amplification of feature channels according to scale r respectively.

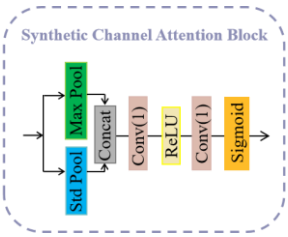

Fig. 3 Synthetic Channel Attention Block

The max pooling is to take the largest feature point in the neighborhood. It can learn the edge and texture structure of the image well. Global standard deviation pooling can provide more effective information for channel weight learning. Let $x'_{n-1} = [x_1, \ldots, x_c, \ldots, x_C]$ as the input, it has C feature maps of size H×W. The formula for standard deviation pooling is:

$$std_c = F_{std}(x_c) = \sqrt{\frac{1}{HW}\sum_{(i,j)\in x_c}\left(x_c^{i,j} - \frac{1}{HW}\sum_{(i,j)\in x_c} x_c^{i,j}\right)^2} \quad (13)$$

where std_c is the standard deviation of the c-th channel?

3.2.2. Modified Spatial Attention Block

Texture details vary at different spatial locations. Fig. 4 shows the structure of the proposed MSAB.Inspired by SENet [10], we design a structure for global average pooling and standard deviation pooling along the channel axis respectively when building a spatial attention mechanism. First, we perform a global average pooling operation for each channel of the input features. We assume that the dimension of it is C×H×W, and the formula of the c-th channel feature is:

$$avg_c = F_{GAP}(x_c) = \frac{1}{HW}\sum_{i-1}^{H}\sum_{j=1}^{W} x_c^{i,j} \quad (14)$$

where $x_c^{i,j}$ represents the pixel value of the position (i, j) in the c-th channel of the input feature map. $F_{GAP}(\cdot)$ represents the global average pooling operation, which is used to generate descriptors to describe the significance of distinct location features.

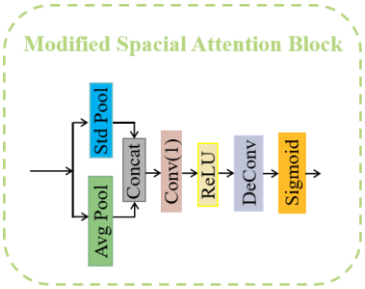

Fig. 4 Modified Spatial Attention Block

The pooled features are concatenated first, and the number of channels is compressed by 1×1 convolution. The non-linear mapping of spatial weight information is realized by deconvolution operation, which further reduces the volume of calculation and ensures that information of multiple spatial position are able to be strengthened. Ultimately, the calculated spatial weight feature maps are normalized by the Sigmoid activation function. The final output can be expressed as:

$$b = f(W_i(\sigma(W_D(g_c)))) \quad (15)$$

where b is the output feature of the MSAB, and g_c is the concatenating result of two pooling. W_i and W_D denote for 1×1 convolution and deconvolution respectively.

3.2.3. Up sample Block

Fig. 5 shows the up sample block which uses sub-pixel convolution [14] to transform the feature in low-frequency into features in high-frequency in reconstruction block. It mainly consists of Conv with Shuffle ×2 represents a 3×3 convolution with Shuffle in scale of 2, and Conv with Shuffle ×3 represents a 3×3 convolution with Shuffle in scale of 3. Conv with Shuffle ×2 and two Conv with Shuffle ×2 are used for ×2 and ×4 scales separately, and Conv with Shuffle ×3 is used for scale of 3.

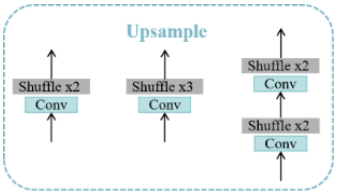

Fig. 5 Upsample Block

When a SR model is trained on a single scale, the upsampling channels of the corresponding scale can be selected separately. The output of the previous step is upsampled by sub-pixel convolution, and the features are further fused by a 3×3 convolution. It is then added to the result of upsampling by interpolation with the features of the input I_{LR}.

$$0_{RB} = S(x_0 + x_n) \quad (16)$$

where 0_{RB} and $S(\cdot)$ denote the output of the reconstruction module and the sub-pixel convolution operation respectively, while x_0 and x_n denote the shallow and deep features respectively.

4. Experiment

4.1. Datasets and Metrics

The training process uses the DIV2K [15] dataset, which contains 1000 high resolution images with rich scene, edge and texture detail. We selected 800 of them for training. The LR images were downsampled by bicubic to obtain low-resolution images at ×2, ×3 and ×4 scales. Five standard benchmark datasets, Set5 [16], Set14 [8], B100 [4], Urban100 [17] and Manga109 [18], were used as the test set for the testing process. Peak signal-to-noise ratio (PSNR) and structural similarity index [19] (SSIM) are used as evaluation metrics. In this paper, the reconstructed images are converted to YCbCr space and compared in the Y channel. We use parameters and multi-adds to measure the lightness of the model and compare it with the existing mainstream models.

4.2. Implementation Details

In order to avoid under-fitting phenomenon during the training process, the training datasets are augmented by random rotations of 90°, 180°, 270° and horizontal flipping to make it 8 times larger than the original. During the training of the model at three scales (×2, ×3 and ×4), 32 image blocks of sizes 128 × 128, 192 × 192 and 256 × 256 were randomly cropped for each batch as input. The L1 loss function [19] and Adam optimizer are chosen for the training. We adapt the cosine annealing learning scheme, and the initial maximum and minimum learning rates were set to 1e-3 and 1e-7, respectively. The cosine period is 250k iterations. Our model is based on the PyTorch framework, and an Nvidia Tesla V100 GPU with 32GB of memory was selected for training acceleration.

4.3. Ablation Experiments

4.3.1. The influence of standard deviation pooling and max pooling

To demonstrate the effectiveness of the designed structure, three sets of experiments were conducted as follows: (1) removing maximum pooling from SCAB; (2) removing global standard deviation pooling from SCAB; (3) the combination of global standard deviation pooling and maximum pooling. The results of the experiments are shown in Table 1.

Table 1. Comparison of Different Pooling Methods in SCAB in Set5 Dataset (×3)

Standard Deviation Pooling	Max Pooling	PSNR (dB)/SSIM
√	×	34.32/0.9254
×	√	34.28/0/9251
√	√	34.37/0.9256

Table 1 shows the comparison of the evaluation indicators of the Set5 dataset (×3). We can see that (1) with standard deviation pooling has a 0.04dB improvement in the PSNR and a 0.0003 improvement in the SSIM relative to (2) with maximum pooling. Whereas (3), which adaptively combines maximum pooling and global standard deviation pooling, has a 0.05dB improvement in PSNR and a 0.0002 improvement in SSIM relative to (1). It is easy to see that global standard deviation pooling has a more significant effect on the PSNR values of the model than maximum pooling. The experimental results show that the designed double pooling structure has better impact on learning channel weight, which proves the effectiveness of the structure.

4.3.2. Influence of Channel Attention Mechanism and Spatial Attention Mechanism

Table 2 shows the comparison of the evaluation indicators of the Set5 dataset (×2). The effectiveness of the proposed MAFB is verified by ablation experiments. We first remove SCAB and then keep the rest the same. The removed model is trained and tested with the same Set5 dataset (×2). We can find that after removing the SCAB, the PSNR decreased by 0.08dB, and the SSIM value also decreased, which prove the effectiveness of the module. Then the MSAB is removed for training, and both PSNR and SSIM values decrease. The comparison results are shown in Table 2. It can be seen that the effect of MAFB proposed in this paper is significantly better than that with only single attention. And the PSNR is improved by about 0.06~0.08dB, which also proves that MAFB has better effect on SISR.

Table 2. Comparison of Different Pooling Methods in SCAB in Set5 Dataset (×3)

SCAB	MSAB	PSNR (dB)/SSIM
√	×	37.87/0.9605
×	√	37.85/0.9604
√	√	37.93/0.9607

4.4. Comparison with state-of-the-art Methods

To prove the effectiveness of the proposed method for image super-resolution, we visualize the partially reconstructed images of B100 and Urban100 datasets. We select three groups of images for visual display at the scale of ×2, ×3, and ×4. As shown in Fig. 6, the blue tall building (img012) and the gray building (img001) in the Urban100 dataset are selected for visualization at ×2 and ×3 scale, and the bird (img_8023) in the B100 dataset is visualized at ×4 scale visualization. The reconstructed images are compared with existing models such as LapSRN [7], DRRN [5], MSRN [20]. Fig. 6 first shows img012 (×2) in Urban100 dataset. We can see that LMAFN can restore the detailed texture and edge information of the image very well. For the texture of the window in img012, most existing models will generate abnormal texture which is different from the original image, and the edge of the window is basically blurred. Taking LapSRN [7] and MemNet [27] as examples, the reconstructed

images generated by them have poor linear details for the brown-yellow window part, while MAFFSRN [28] and MSRN [20], which have relatively well, cannot fully recover most of the straight lines in the image (especially the blue window part). LMAFN achieves a better visual effect while restoring the straight-line texture of the whole image, which is closer to the GT image.

The second part of Fig. 6 shows img001 (×3) in Urban100 dataset, which mainly compares the details of the window frame in the middle. It can be seen that for the oblique frame in the middle, most of the existing models cannot recover the edge information of the details well, and the reconstructed part is very blurred. For DRRN [5] and MemNet [27], which can recover the oblique border, cannot recover the details of the double-striped border on the right side of the image very well. Compared with GT images, MAFFSRN [28] and MSRN [20] obtain more realistic results and recover more image details, but LMAFN can recover the edge information of the window frame well without introducing blur artifacts, and the results are clearer compared to several other models. These results verify that LMAFN has more powerful representation ability.

The last part of Fig. 6 shows img_8023 (×4) in B100 dataset. Focus on contrasting the details of the bird feather texture in the picture, among them, the reconstructed feather texture of LapSRN[7] is very fuzzy, DRCN[4] and DRRN[5] can retain some striped information, but the edge is very fuzzy, and several other models will produce some wrong mesh information. LMAFN can recover the texture information of bird feathers better while the edge is relatively clear. In contrast, LMAFN obtains sharper results and recovers more high contrast and sharp edges, and the reconstruction effect of the algorithm is improved to a certain extent.

To illustrate the validity of our proposed model, we compared the super-resolution (SR) reconstruction results of 11 advanced SR models based on deep learning, such as SRCNN [1], DRCN [4], LapSRN [7], MemNet [27], MAFFSRN [28], in different scales of five mainstream benchmark datasets. The experimental results are shown in Table 3.

The best results have been bolded and underlined. Compared with other methods, LMAFN maintains better accuracy while keeping the model lightweight, and has the best comprehensive performance. Both LMAFN and MAFFSRN [28] adopt channel attention to learn the interdependence among features, so that the network can focus on more important features, and thus obtaining similar and better results than other methods. However, the parameters and multi-adds of our LMAFN (×2) are reduced by about 211.9K and 34.3G respectively, which shows that our model is more lightweight. Compared with another lightweight model, CARN [21], the PSNR of our model is 0.08, 0.05 and 0.16 dB higher than that of CARN on datasets with more texture information (such as Set5, B100 and Urban100) × 3 scale respectively, Though, the results on edge-informed datasets (such as Manga109) are 0.11dB lower than CARN, LMAFN has about 13% of its parameters and multi-add about 20%. Texture information is a higher-order pattern with more complex statistical features, and edge information is a first-order pattern that are able to be extracted by a first-order gradient operator. Therefore, LMAFN has better reconstruction quality on images with more high-order information such as textures. In summary, compared with the ×3 and ×4 scale, LMAFN has more obvious advantages on the ×2 scale.

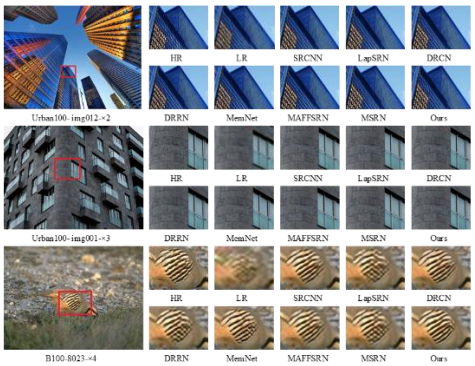

Fig .6 Visual comparison with other image super-resolution models on Urban100 and B100 datasets

Table 3. Comparison of Reconstruction Effects of Different Image Super-Resolution Models

Models	Scale	Params	Multi-adds	Set5	Set14	B100	Urban100	Manga109
				PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Bicubic	×2	-	-	33.67/0.9291	30.24/0.8677	29.56/0.8431	26.88/0.8423	30.80/0.9337
SRCNN		57K	52.7G	36.67/0.9540	32.38/0.9058	31.36/0.8876	29.51/0.8950	35.61/0.9663
FSRCNN		12K	6.0G	37.01/0.9556	32.68/0.9086	31.53/0.8919	29.88/0.9020	36.65/0.9710
VDSR		665K	612.6G	37.53/0.9585	33.02/0.9122	31.90/0.8960	30.77/0.9143	37.20/0/9746
DRCN		1.7M	17.6T	37.63/0.9588	33.05/0.9117	31.85/0.8942	30.75/0.9136	37.56/0.9723
DRRN		297K	6.6T	37.74/0.9590	33.23/0.9136	32.05/0.8973	31.23/0.9188	37.88/0.9749
LapSRN		251K	29.9G	37.52/0.9587	33.05/0.9127	31.87/0.8955	30.41/0.9103	37.27/0.9740
CARN		1.6M	222.8G	37.75/0.9585	33.52/0.9166	32.09/0.8978	31.92/0.9254	38.40/**0.9766**
MemNet		678K	2.6T	37.76/0.9589	33.28/0.9139	32.04/0.8976	31.31/0.9195	37.69/0.9741
MAFFSRN		402K	77.2G	37.89/0.9604	33.52/0.9170	**32.14**/0.8991	31.96/0.9268	-
AWSRN-S		397K	91.2G	37.75/0.9596	33.31/0.9151	32.00/0.8974	31.39/0.9207	37.90/0.9755
LMAFN		**190.1K**	**42.9G**	**37.93/0.9607**	**33.52/0.9175**	**32.10/0.9010**	**31.97/0.9267**	**38.45/0.9764**
Bicubic	×3	-	-	30.41/0.8645	27.64/0.7722	27.21/0.7342	24.46/0.7401	26.96/0.8545
SRCNN		57K	52.7G	32.75/0.9090	29.28/0.8210	28.41/0.7863	26.25/0.7989	30.54/0.9112
FSRCNN		13K	5.0G	33.18/0.9101	29.42/0.821	28.53/0.7910	26.30/0.8091	31.02/0.9210
VDSR		665K	612.6G	33.68/0.9208	29.80/0.8314	28.80/0.7976	27.14/0.8272	32.01/0.9340
DRCN		1774K	17.6T	32.82/0.9224	29.76/0.8312	28.80/0.7962	27.15/0.8276	32.27/0.9335
DRRN		298K	6.64T	34.03/0.9243	29.96/0.8349	28.95/0.8005	27.53/0.8377	32.72/0.9381
LapSRN		502K	38.9G	33.82/0.9218	29.82/0.8316	28.82/0.7968	27.07/0.8281	32.21/0.9334
CARN		1.6M	118.8G	34.29/0.9254	30.29/0.8407	29.06/0.8034	28.05/0.8493	**33.50**/0.9440
MemNet		678K	2.6T	34.09/0.9247	30.02/0.8350	28.96/0.8003	27.56/0.8374	32.50/0.9366
MAFFSRN		418K	34.2G	34.35/**0.9269**	30.35/0.8429	29.09/**0.8052**	28.13/**0.8521**	-
AWSRN-S		477K	48.6G	34.02/0.9240	30.09/0.8376	28.92/0.8009	27.57/0.8391	32.82/0.9393
LMAFN		**262.3K**	**27.0G**	**34.37/0.9256**	**30.31/0.8437**	**29.11/0.8043**	**28.21/0.8520**	**33.39/0.9442**
Bicubic	×4	-	-	28.42/0.8052	26.10/0.6976	25.96/0.6572	23.15/0.6589	24.89/0.7825
SRCNN		57K	52.7G	30.49/0.8628	27.54/0.7514	26.90/0.7105	24.53/0.7226	27.62/0.8530
FSRCNN		12K	4.6G	30.72/0.8657	27.63/0.7532	26.98/0.7148	24.16/0.7277	29.87/0.8563
VDSR		665K	612.6G	31.35/0.8829	28.03/0.7678	27.29/0.7239	25.18/0.7530	28.76/0.8765
DRCN		1774K	17.6T	31.54/0.8844	28.05/0.7663	27.23/0.7220	25.15/0.7510	28.96/0.8835
DRRN		297K	6.8T	31.69/0.8871	28.21/0.7720	27.38/0.7284	25.45/0.7638	29.45/0.8953
LapSRN		813K	149.4G	31.54/0.8846	28.14/0.7745	27.32/0.7279	25.21/0.7560	29.02/0.8873
CARN		1.6M	90.9G	32.13/0.8936	28.61/0.7806	27.57/0.7319	26.12/0.7837	**30.38**/0.9086
MemNet		677K	2.6T	31.73/0.8901	28.26/0.7723	27.41/0.7289	25.50/0.7627	29.02/0.8970

5. Conclusion

We propose a novel lightweight multi-attention fusion image super-resolution network model, LMAFN. The model effectively obtains the weight values of different features by integrating the channel attention and the spatial attention while designing them in a lightweight manner. For the SCAB, two kinds of pooling are introduced. Compared with single pooling, the connection between each channel is fully considered. It can extract richer high-level features. The experimental results imply that LMAFN improves the evaluation index, while having better results in the visual effect of the reconstructed image.

References

[1] Dong C, Loy C C, He K, et al. Image super-resolution using deep convolutional networks[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 38(2): 295-307.

[2] Dong C, Loy C C, He K, et al. Learning a deep convolutional network for image super-resolution[C]//European conference on computer vision. Springer, Cham, 2014: 184-199.

[3] Dong C, Loy C C, Tang X. Accelerating the super-resolution convolutional neural network[C]//European conference on computer vision. Springer, Cham, 2016: 391-407.

[4] Kim J, Lee J K, Lee K M. Deeply-recursive convolutional network for image super-resolution[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 1637-1645.

[5] Tai Y, Yang J, Liu X. Image super-resolution via deep recursive residual network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 3147-3155.

[6] Lim B, Son S, Kim H, et al. Enhanced deep residual networks for single image super-resolution[C]//Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2017: 136-144.

[7] Lai W S, Huang J B, Ahuja N, et al. Fast and accurate image super-resolution with deep laplacian pyramid networks[J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(11): 2599-2613.

[8] Zhao H, Kong X, He J, et al. Efficient image super-resolution using pixel attention[C]//European Conference on Computer Vision. Springer, Cham, 2020: 56-72.

[9] Zhang Y, Li K, Li K, et al. Image super-resolution using very deep residual channel attention networks[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 286-301.

[10] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.

[11] Liu D, Wen B, Fan Y, et al. Non-local recurrent network for image restoration[J]. Advances in neural information processing systems, 2018, 31.

[12] Liu J J, Hou Q, Cheng M M, et al. Improving convolutional networks with self-calibrated convolutions[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 10096-10105.

[13] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[14] Shi W, Caballero J, Huszár F, et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 1874-1883.

[15] Agustsson E, Timofte R. Ntire 2017 challenge on single image super-resolution: Dataset and study[C]//Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2017: 126-135.

[16] Bevilacqua M, Roumy A, Guillemot C, et al. Low-complexity single-image super-resolution based on nonnegative neighbor embedding[J]. 2012.

[17] Fan Y, Shi H, Yu J, et al. Balanced two-stage residual networks for image super-resolution[C]//Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2017: 161-168.

[18] Matsui Y, Ito K, Aramaki Y, et al. Sketch-based manga retrieval using manga109 dataset[J]. Multimedia Tools and Applications, 2017, 76(20): 21811-21838.

[19] Wang Z, Bovik A C, Sheikh H R, et al. Image quality assessment: from error visibility to structural similarity[J]. IEEE transactions on image processing, 2004, 13(4): 600-612.

[20] Li J, Fang F, Mei K, et al. Multi-scale residual network for image super-resolution[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 517-532.

[21] Ahn N, Kang B, Sohn K A. Fast, accurate, and lightweight super-resolution with cascading residual network[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 252-268.

[22] Lu Y, Zhou Y, Jiang Z, et al. Channel attention and multi-level features fusion for single image super-resolution[C]//2018 IEEE Visual Communications and Image Processing (VCIP). IEEE, 2018: 1-4.

[23] Hui Z, Gao X, Yang Y, et al. Lightweight image super-resolution with information multi-distillation network[C]//Proceedings of the 27th acm international conference on multimedia. 2019: 2024-2032. national conference on multimedia. 2019: 2024-2032.

[24] Hui Z, Wang X, Gao X. Fast and accurate single image super-resolution via information distillation network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 723-731.

[25] Wang C, Li Z, Shi J. Lightweight image super-resolution with adaptive weighted learning network[J]. arXiv preprint arXiv:1904.02358, 2019.

[26] Dai T, Cai J, Zhang Y, et al. Second-order attention network for single image super-resolution[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 11065-11074.

[27] Tai Y, Yang J, Liu X, et al. Memnet: A persistent memory network for image restoration[C]//Proceedings of the IEEE international conference on computer vision. 2017: 4539-4547.

[28] Muqeet A, Hwang J, Yang S, et al. Multi-attention based ultra lightweight image super-resolution[C]//European Conference on Computer Vision. Springer, Cham, 2020: 103-118.

[29] Zagoruyko S, Komodakis N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer[J]. arXiv preprint arXiv:1612.03928, 2016.

[30] Kim J, Lee J K, Lee K M. Accurate image super-resolution using very deep convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 1646-1654.