The Visual Computer
https://doi.org/10.1007/s00371-024-03610-0

ORIGINAL ARTICLE

HASN: hybrid attention separable network for efficient image super-resolution

Weifeng Cao[1] · Xiaoyan Lei[1] · Jun Shi[1] · Wanyong Liang[1] · Jie Liu[1] · Zongfei Bai[1]

Accepted: 12 August 2024

Abstract

Recently, lightweight methods for single-image super-resolution have gained significant popularity and achieved impressive performance due to limited hardware resources. These methods demonstrate that adopting residual feature distillation is an effective way to enhance performance. However, we find that using residual connections after each block increases the model's storage and computational cost. Therefore, to simplify the network structure and learn higher-level features and relationships between features, we use depth-wise separable convolutions, fully connected layers, and activation functions as the basic feature extraction modules. This significantly reduces computational load and the number of parameters while maintaining strong feature extraction capabilities. To further enhance model performance, we propose the hybrid attention separable block, which combines channel attention and spatial attention, thus making use of their complementary advantages. Additionally, we use depth-wise separable convolutions instead of standard convolutions, significantly reducing the computational load and the number of parameters while maintaining strong feature extraction capabilities. During the training phase, we also adopt a warm-start retraining strategy to exploit the potential of the model further. Extensive experiments demonstrate the effectiveness of our approach. Our method achieves a smaller model size and reduced computational complexity without compromising performance. Code can be available at https://github.com/nathan66666/HASN.git

Keywords Efficient super-resolution · Channel attention · Spatial attention · Hybrid attention separable block

1 Introduction

As the application scenarios of virtual reality technology continue to expand, so too does the demand for image quality. High-quality images can provide users with a more immersive experience. In this context, events such as the CGI and CASA conferences are dedicated to advancing various fields within computer graphics and virtual reality, making significant contributions to the progress of these technologies. The successful application of image super-resolution techniques will undoubtedly further promote the development of this field. Particularly, the emergence of efficient image super-resolution technology has made it easier to deploy this technology on edge devices, thereby broadening its application.

Image super-resolution (SR) is a typical branch of low-level vision methods, reconstructing high-resolution (HR) images from low-resolution (LR) inputs. Traditional SISR methods use interpolation techniques to recover corresponding HR images from LR ones. While simple and effective, these methods struggle to restore some of the details and textures in images. Since SRCNN [1] first introduced convolutional neural networks to the field of image super-resolution, deep learning (DL) has achieved remarkable performance and realistic visual effects due to its learnable feature representations. These SR networks [2–11] have significantly

✉ Xiaoyan Lei
xyan_lei@163.com

Weifeng Cao
weifeng_cao@163.com

Jun Shi
shijunzz@gmail.com

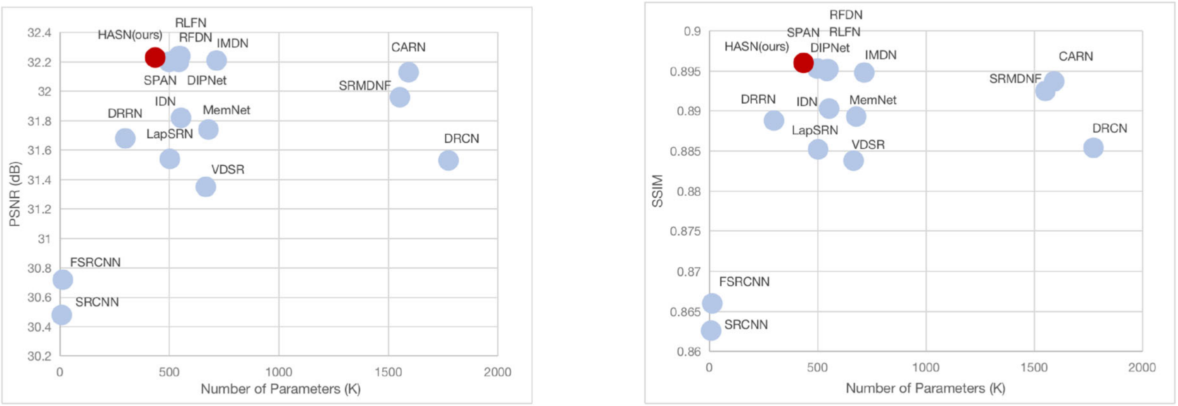
Wanyong Liang
lwy307@126.com

Jie Liu
ljie35206609@163.com

Zongfei Bai
zongfeibai@163.com

[1] The School of Electrical and Information Engineering, Zhengzhou University of Light Industry, No.5 Dongfeng Road, Zhengzhou 450002, Henan, China

Published online: 03 September 2024



(a) PSNR results v.s the total number of parameters of different methods for image SR on Set5.

(b) SSIM results v.s the total number of parameters of different methods for image SR on Set5.

Fig. 1 Comparison with other SOTA methods for image SR on Set5. The red dots represent the method proposed in this paper

improved the quality of reconstructed images. Their success can be partially attributed to their larger model capacity and intensive computational power. However, this makes them difficult to deploy on resource-constrained devices in real-world applications. Therefore, it is necessary to design lightweight models to improve the efficiency of SISR models, achieving a good balance between image quality and inference time.

Many prior works [1, 12–25] have been proposed to develop efficient image super-resolution models. They use different strategies to achieve high efficiency, including parameter sharing strategy [26], cascading network with grouped convolution [27], information or feature distillation mechanisms [21–23] and attention mechanisms [2, 3, 22]. Although they have improved efficiency using these strategies, redundancy still exists in convolution operations.

In this paper, to make the network more lightweight, we propose a new lightweight SR network, which consists of several stacked hybrid attention separable blocks. This structure is capable of extracting higher-level image features and includes more edge features and texture details. We only use a few necessary residual connections to prevent the vanishing gradient problem while integrating low-level features. Additionally, we use depth-wise separable convolutions instead of standard convolutions in convolutional blocks, significantly reducing the computational load and the number of parameters while maintaining strong feature extraction capabilities. To fully maximize the model's capabilities, we propose a warm-start retraining strategy to further learn the image distribution and use the geometric self-ensemble strategy during the inference phase. Specifically, our contributions are as follows:

- We propose a hybrid attention separable network for efficient image super-resolution, which can extract higher-level image features and include more edge features and texture details without additional residual connections.
- We propose a warm-start retraining strategy, which helps in learning the distribution of high-resolution images, effectively enhancing network performance.
- Extensive experiments demonstrate that our proposed method surpasses existing state-of-the-art (SOTA) methods in terms of parameters (Fig. 1) and FLOPs, while maintaining comparable performance in PSNR and SSIM metrics.

2 Related work

2.1 Classical SISR methods

SRCNN [1] is the first work that introduces deep convolutional neural networks (CNNs) to the image SR task. They use a three-layer convolutional neural network to jointly optimize feature extraction, nonlinear mapping, and image reconstruction in an end-to-end manner, achieving performance superior to traditional SR methods. Subsequent methods adopt more complex convolutional module designs, such as residual blocks [22, 28, 29] and dense blocks [30], to enhance the model's representational capacity. As networks become larger and deeper, the introduction of various attention mechanisms [2, 31] has become a new trend in image super-resolution research. For example, RCAN [32] employs channel attention, while PAN [33] uses pixel attention. Additionally, self-attention mechanisms have shown significant performance in image reconstruction. SwinIR [2] leverages the swin transformer architecture [34], multi-scale feature representation [35], hybrid attention mechanisms, and local–global feature interaction. HAT [31] further expands the window size and uses channel attention to better activate available pixels. PCCFormer [36] uses parallel attention transformer and adaptive convolution residual block to improve feature expression ability of the model. Recently, some emerging attention mechanisms have also achieved great success in imaging [37, 38]. Image super-resolution techniques have been applied in the medical field, making significant contributions to the diagnosis of brain diseases and morphometric studies [39].

2.2 Lightweight SISR methods

To meet the requirements of edge devices, it is crucial to develop lightweight and efficient SR models. The SR network SRCNN [1] achieves impressive results but also faces issues such as high computational demands. FSRCNN [12] addresses these issues by removing the interpolation upsampling, introducing transposed convolution at the end of the network, and using smaller but more numerous convolutional kernels, achieving approximately 17 times the acceleration compared to SRCNN. DRCN [14] employs recursive calls to the feature extraction layers, while DRRN [16] improves upon DRCN by combining recursive and residual networks to achieve better performance with fewer parameters. LapSRN [15] uses transposed convolution for upsampling, leveraging convolutional layers to learn the residuals between high-resolution images and upsampled feature maps, achieving multi-scale reconstruction through progressive upsampling. IDN [18] effectively extracts local long-path and short-path features through an information distillation module, achieving relatively fast inference speed. IMDN [21] constructed a cable information multi-distillation block (IMDB) consisting of distillation and selective fusion. The distillation module gradually extracts features, while the fusion module determines the importance of candidate features based on an attention mechanism and fuses them accordingly.

Recently, researchers have been optimizing convolution methods to develop lighter and more efficient SR models. For example, ECBSR [40] and RepVGG [41] effectively extract edge and texture information, while FMEN [42] and BSRN [29] further accelerate network inference and reduce the number of network parameters, achieving efficient super-resolution.

3 Methodology

3.1 Overall network architecture

For the overall network structure of HASN, we adopt a coarse-to-fine strategy to learn representative features from LR images. As shown in Fig. 2, HASN consists of three main stages: an initial feature extraction, a multi-stage feature extraction, and a high-resolution reconstruction. Here, I_{LR} represents the original image input, $I_{LR} \in \mathbb{R}^{H \times W \times C_{in}}$((H, W, and C are the image height, width and input channel number, respectively). A 3×3 convolutional layer $H_{IF}(\cdot)$ is used to extract initial feature. This process can be expressed as:

$$F_0 = H_{IF}(I_{LQ}), \tag{1}$$

The convolutional layer effectively captures local features of an image, providing feature maps for subsequent deep feature extraction. Next, F_0 extracts multi-stage features using HASBs. We extract deep feature as:

$$\begin{aligned} &F_i = H_{HASB_i}(F_0), i = 1, 2, \ldots, K, \\ &F_{DF} = H_{\mathrm{Conv}}(F_K), \end{aligned} \tag{2}$$

where $H_{HASB_i}(\cdot)$ denotes the i-th HASB. A 3×3 convolutional layer is used after several HASBs to further process and refine the feature representations, enhancing the feature learning capability.

$$I_{RHQ} = H_{REC}(F_{DF} + F_0), \tag{3}$$

where $H_{REC}(\cdot)$ is the function of the reconstruction module. It consists of a 3×3 convolutional layer and a sub-pixel layer. The 3×3 convolutional layer reduces the dimensionality of the high-dimensional feature maps while preserving important information, preparing them for the sub-pixel layer. The entire training process is divided into two stages. The $\mathcal{L}_1$ loss function is exploited to optimize the model in the first stage, which can be formulated as follows:

$$\mathcal{L}_1 = \|I_{SR} - I_{HR}\|_1 , \tag{4}$$

The loss function for the second stage($\mathcal{L}_{s2}$) is defined as follows:

$$\begin{aligned} &\mathcal{L}_{s2} = \alpha \mathcal{L}_1 + \beta \mathcal{L}_{D_{KL}}, \\ &\mathcal{L}_{D_{KL}} = \textstyle\sum_i P_{I_{HR}}(i) log \frac{P_{I_{HR}}(i)}{P_{I_{SR}}(i)}, \end{aligned} \tag{5}$$

where $\mathcal{L}_{D_{KL}}$ is KL divergence loss, which is used to measure the difference between the probability distributions of the actual high-resolution image and the predicted super-resolution image. $P_{I_{HR}}(i)$ represents the probability distribution of the i-th pixel in the high-resolution image, and $P_{I_{SR}}(i)$ represents the probability distribution of the i-th pixel in the super-resolution image.α and β are two different constants, which we set to 1 in this context.

3.2 Hybrid attention separable block

As shown in Fig. 3, our proposed HASB consists of two depth-wise separable convolutions, several fully connected

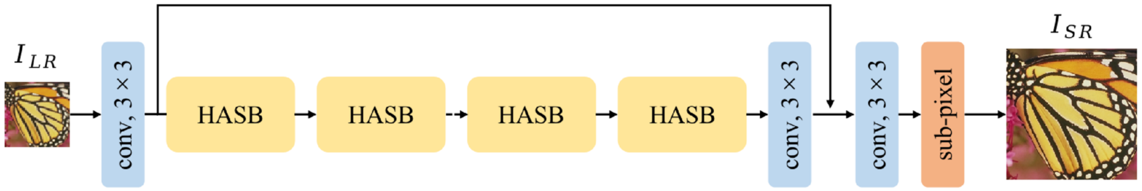
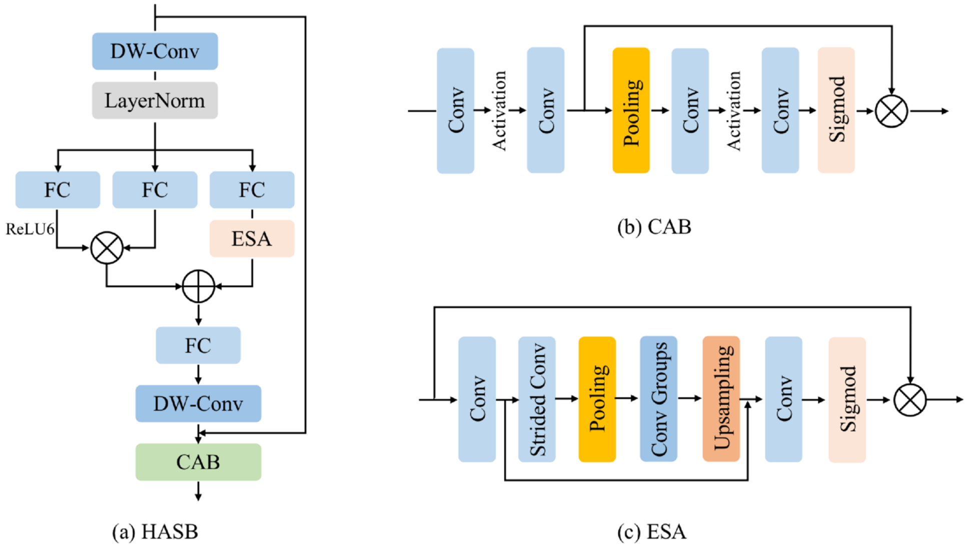


Fig. 2 Overall network architecture of our HASN

Fig. 3 **a** Architecture of hybrid attention separable block (HASB). **b** Architecture of channel attention block (CAB). **c** Architecture of enhanced spatial attention (ESA)



(a) HASB

(b) CAB

(c) ESA

layers, a channel attention block, and enhanced spatial attention. First, a 7 × 7 depth-wise separable convolution operation is applied to the input features F_{in} to extract local features. Then, the convolved features are subjected to layer normalization, resulting in the normalized features F_o. The normalized features F_o are passed to three parallel fully connected layers. The output of the first fully connected layer is passed through a ReLU6 activation function. The output of the second fully connected layer is used directly. The output of the third fully connected layer is processed through the enhanced spatial attention module. The output of the first fully connected layer is multiplied element-wise with the output of the second fully connected layer. The result of this multiplication is added element-wise to the output of the third fully connected layer (features processed by the ESA) to obtain the fused features. The fused features are passed to a fully connected layer for further processing. The features processed by the fully connected layer are passed through another depth-wise separable convolution layer to extract additional features. Finally, the features are processed through the channel attention block module to obtain the final output features. The input feature F_{in} is added directly to the features before the final depth-wise separable convolution layer (DW-Conv) through a residual connection. This design helps alleviate the vanishing gradient problem and enhances feature learning. The whole structure is described as

$$\begin{aligned}
&F_o = LN(DWConv_{7\times7}(F_{in})),\\
&F_{d_1}, F_{d_2}, F_{d_3} = FC(F_o), FC(F_o), FC(F_o),\\
&F_d = ReLU6(F_{d_1}) \otimes F_{d_2} + ESA(F_{d_3}),\\
&F_d = DWConv_{7\times7}(FC(F_d)) + F_{in}),\\
&F_{out} = CAB(F_d)
\end{aligned} \tag{6}$$

where $DWConv_{7\times7}$ represents a depth-wise separable convolution with a 7 × 7 kernel, $LN(\cdot)$ denotes the LayerNorm layer, and FC refers to the fully connected layer.

3.3 Warm-start retraining strategy

We propose a novel warm-start retraining strategy. Different from some previous works that use the 2× model as a pre-trained network instead of training from scratch, we train HASN for 4× from scratch in the first stage. In the second stage, we load the model weights from the first stage, which are not fully converged, and further expand the dataset (adding Flickr2K). We further learn the distribution of high-resolution images by minimizing the KL divergence loss and L1 loss, as formulated in Eq. 5. The other training settings remain consistent with the first stage.

4 Experiments

4.1 Datasets and metrics

In this paper, the entire training process is divided into two stages. In the first stage, we use the DIV2K [43] dataset, and in the second stage, we use the DF2K dataset (DIV2K + Flickr2K) [43] to further improve the network performance. DIV2K [43] is a high-quality (2K resolution) image dataset containing 800 training images. Flickr2K is an image dataset with 2K resolution containing 2650 images. Additionally, the low-resolution images of DIV2K and Flickr2K are generated from the ground truth images by the "bicubic" downsampling in MATLAB. For testing, we use five widely used benchmark datasets: Set5 [44], Set14 [45], BSD100 [46], Urban100 [47], and Manga109 [48]. We evaluate all the SR results using the PSNR and SSIM metrics on the Y channel of the YCbCr color space.

4.2 Implementation details

The proposed HASN consists of 6 HASBs, and the number of channels is set to 52. The kernel size of all depth-wise convolutions is set to 7. During training, we set the input patch size to 192 × 192 and use random rotation and horizontal flipping for data augmentation. The batch size is set to 128, and the total number of iterations is 500k. The initial learning rate is set to 2×10^{-4}. We adopt a multi-step learning rate strategy, where the learning rate will be halved when the iteration reaches 250,000, 400,000, 450,000, and 475,000, respectively. The model is trained by Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. In the second stage of training, we chose the model weights from the 100k-th iteration of the first stage as the starting point, and the total number of iterations is set to 1000k. Additionally, we use $\mathcal{L}_{s2}$ as the loss function for the second stage. Other training settings remain consistent with the first stage. To maximize the potential performance of the HASN proposed in this paper, we use geometric self-ensemble [7] in the experiment, which is applied during inference without additional training. The networks are implemented by using PyTorch framework with a NVIDIA 3090 GPU.

4.3 Comparison with state-of-the-arts

We compare our models with several advanced efficient super-resolution models with scale factor of 4. The comparison methods include SRCNN [1], FSRCNN [12], VDSR [13], DRCN [14], LapSRN [15], DRRN [16], MemNet [17], IDN [18], SRMDNF [19], CARN [20], IMDN [21], RFDN [22], RLFN [23], DIPNet [24], SPAN [25]. Firstly, in terms of model performance, we use PSNR and SSIM as evaluation metrics. In terms of model efficiency, we use Parameters and FLOPs to measure the model size and computational complexity. The quantitative performance comparison on five benchmark datasets is shown in Table 1. Compared with other state-of-the-art models, it can be seen that HASN achieves better performance on Set5, Set14, and BSD100. Its performance on the remaining two datasets is comparable. Overall, HASN achieves performance comparable to other networks with fewer parameters and computational complexity, achieving a better balance in performance and efficiency.

5 Ablation study

In this section, we conduct a set of ablation experiments to evaluate the performance of each proposed module.

5.1 The choice of multiplication and addition in convolution block

Many previous efficient image SR methods [22, 25, 49] benefit from residual connections, which extract features from each block up to the upsampling layer. Some methods [21–23] also perform feature distillation within each block. However, these approaches often make the network structure redundant. We want to design an efficient and compact network. Inspired by [50], element-wise multiplication seems to provide greater gains in a narrower network compared to addition. This finding is beneficial for our task, as we need to minimize network size while achieving equal or better performance compared to previous methods. Therefore, we design some simple experiments to validate this conclusion. As shown in Fig. 4, (a) presents the structure of the CB module. (b) illustrates the fitting curves of four different configurations. It is evident that when activation function is not used, element-wise multiplication performs significantly better than addition, despite some instability during training. When activation function is included, both addition and multiplication configurations exhibit smooth fitting curves, and the PSNR on the test set shows that the network using multiplication slightly outperforms the one using addition. As shown in Table 2, we set up networks with three different embedding dimensions. We find that in Urban100, the PSNR gain between element-wise multiplication and addition decreases as the dimension increases, from 0.08 dB to 0.07 dB, and finally to 0.01 dB. On other test sets, the changes do not seem to follow a consistent pattern. However, across various dimensions, using element-wise multiplication generally yields better performance.

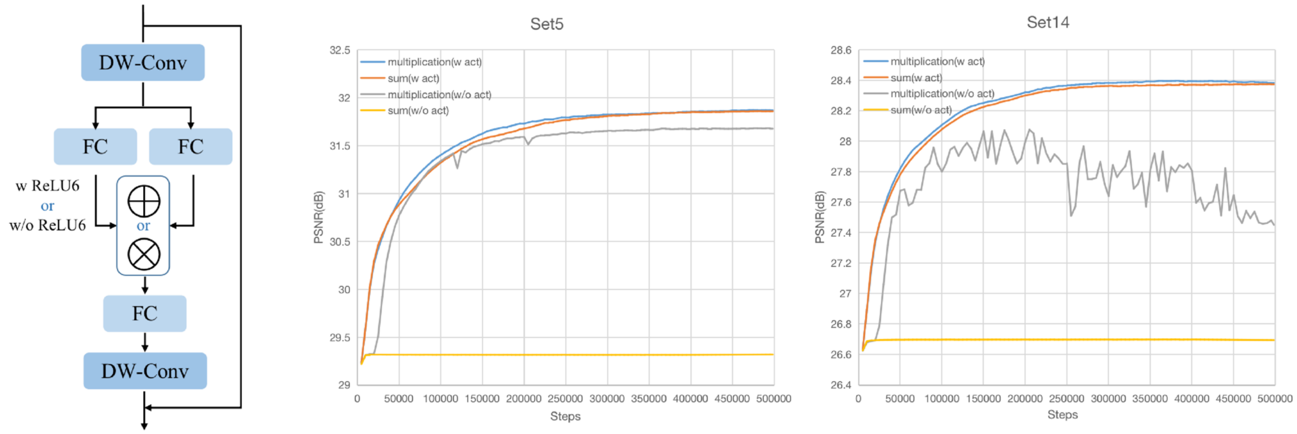
5.2 Study on HASB number

From Fig. 5, we can observe that with the increase in the number of HASBs, the PSNR shows an upward trend when the

Table 1 Average PSNR/SSIM for scale factor 4 on datasets Set5, Set14, BSD100, Urban100, and Manga109

Method	Params	FLOPs(G)	Set5 PSNR/SSIM	Set14 PSNR/SSIM	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM	Manga109 PSNR/SSIM
Bicubic	–	–	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866
SRCNN [1]	8K	52.7	30.48/0.8626	27.50/0.7513	26.90/0.7101	24.52/0.7221	27.58/0.8555
FSRCNN [12]	13K	4.6	30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280	27.90/0.8610
VDSR [13]	666K	612.6	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524	28.83/0.8870
DRCN [14]	1774K	17,974.0	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510	28.93/0.8854
LapSRN [15]	502K	149.4	31.54/0.8852	28.09/0.7700	27.32/0.7275	25.21/0.7562	29.09/0.8900
DRRN [16]	298K	6,796.9	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638	29.45/0.8946
MemNet [17]	678K	2662.4	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630	29.42/0.8942
IDN [18]	553K	81.8	31.82/0.8903	28.25/0.7730	27.41/0.7297	25.41/0.7632	29.41/0.8942
SRMDNF [19]	1552K	89.3	31.96/0.8925	28.35/0.7787	27.49/0.7337	25.68/0.7731	30.09/0.9024
CARN [20]	1592K	90.9	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	30.47/0.9084
IMDN [21]	715K	40.9	32.21/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838	30.45/0.9075
RFDN [22]	550K	*31.6*	32.24/0.8952	28.61/0.7819	27.57/0.7360	26.11/0.7858	*30.58/0.9089*
RLFN [23]	543K	33.9	**32.24**/0.8952	*28.62*/0.7813	*27.60*/0.7364	*26.17/0.7877*	–/–
DIPNet [24]	543K	–	32.20/0.8950	28.58/0.7811	27.59/0.7364	26.16/**0.7879**	30.53/0.9087
SPAN [25]	498K	–	32.20/*0.8953*	28.66/**0.7834**	27.62/*0.7374*	**26.18**/0.7879	**30.66/0.9103**
HASN (Ours)	435K	**26.6**	*32.23*/**0.8960**	**28.66**/*0.7830*	**27.62/0.7387**	26.13/0.7869	30.50/0.9077

The best and second best results are highlighted in bold and italic, respectively



(a) Design of convolutional block

(b) The convergence curves of the four combinations

Fig. 4 Design of convolutional block and convergence curves of different combinations

Table 2 Quantitative comparison (average PSNR/SSIM) of element-wise multiplication and addition across different embedding dimensions on benchmark datasets

Sum	Multiplication	Dim	Param	FLOPs(G)	Set5		Set14		B100		Urban100		Manga109	
					PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
✓	✗	30	90k	5.77	31.45	0.8847	28.13	0.7704	27.27	0.7272	25.15	0.7539	29.03	0.8868
✗	✓	30	90k	5.77	**31.52**	**0.8858**	**28.18**	**0.7716**	**27.30**	**0.7280**	**25.23**	**0.7560**	**29.04**	**0.8871**
✓	✗	52	227k	14.73	31.85	0.8908	28.36	0.7764	27.42	0.7328	25.52	0.7678	**29.67**	0.8979
✗	✓	52	227k	14.73	**31.87**	**0.8915**	**28.38**	**0.7770**	**27.45**	**0.7338**	**25.59**	**0.7700**	29.53	**0.8981**
✓	✗	90	610k	39.62	32.01	0.8933	28.47	0.7799	27.52	0.7362	25.86	0.7795	**30.08**	**0.9039**
✗	✓	90	610k	39.62	**32.07**	**0.8940**	**28.51**	**0.7809**	**27.54**	**0.7368**	**25.87**	**0.7800**	29.91	0.9031

Bold indicates the best result when the “Dim” in the table is the same

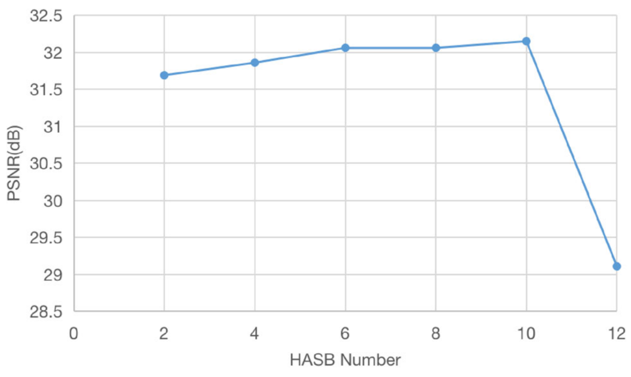


Fig. 5 PSNR of different numbers of HASB on Set5

Table 3 Quantitative comparison (average PSNR/SSIM) of different HASB number on benchmark datasets

HASB number	Param	FLOPs	Set5		Set14	
			PSNR	SSIM	PSNR	SSIM
2	177k	10.98	31.69	0.8878	28.26	0.7742
4	306k	18.81	31.86	0.8909	28.40	0.7776
6	435k	26.63	32.06	0.8938	28.52	0.7803
8	564k	34.46	32.06	0.8933	28.50	0.7785
10	693k	42.28	32.15	0.8948	28.56	0.7808
12	822k	50.11	29.11	0.8268	26.58	0.7277

HASB number is less than or equal to 10. However, when the HASB number is set to 12, there is a sharp decline in PSNR for Set5. This phenomenon indicates that while increasing the number of HASB modules can enhance the model's feature extraction capability to some extent, excessively increasing them may lead to overfitting the training data. Due to the complexity of the attention mechanism and fully connected layers within the HASB modules, the model may capture noise and details from the training data, resulting in a reduced generalization ability on the test data. As shown in Table 3, with the increase in the number of HASBs, the model's parameter count and computational complexity also increase. Setting the HASB number to 6 balances the model size and performance.

5.3 Study on kernel size of depth-wise convolution

To explore the impact of convolution kernel size on network performance, we set the kernel sizes of all depth-wise convolutions to 3, 5, 7, and 9, respectively. As shown in Table 4, we observed that performance improves with larger kernel sizes across the five benchmark datasets. However, as the kernel size increases, the number of network parameters and FLOPs also increase. From the table, the best results are seen between kernel sizes 7 and 9. To balance computational complexity and the number of parameters, choosing a kernel size of 7 is appropriate.

5.4 Study on residual connection

To explore the role of residual connections in image super-resolution, we use intermediate feature visualization to observe the changes in the network's intermediate features, as shown in Fig. 6. (d) and (f) show feature map visualizations without and with residual connections, respectively. From left to right, the features progress from lower to higher layers, gradually shifting from capturing detailed information (such as edges and textures) to more abstract information (such as shapes and overall contours). The lower layer feature maps focus more on local features, while the information in the feature maps becomes more abstract and global as the layers deepen.

Comparing (d) and (f), we observe that the feature maps in (d) capture more information at each layer, retaining more edge and texture details. In contrast, the feature maps in (f) lose detail information more quickly and shift to more abstract representations. This suggests that in our method, CBs [50] may be sufficient to learn important features, while using excessive residual connections could introduce noise. The quantitative performance comparison on several benchmark datasets is shown in Table 5. The PSNR on Set5, Set14, B100, Urban100, and Manga109 improved by 0.13dB, 0.07dB, 0.03dB, 0.08dB, and 0.02dB, respectively.

5.5 Effectiveness of HASB architecture

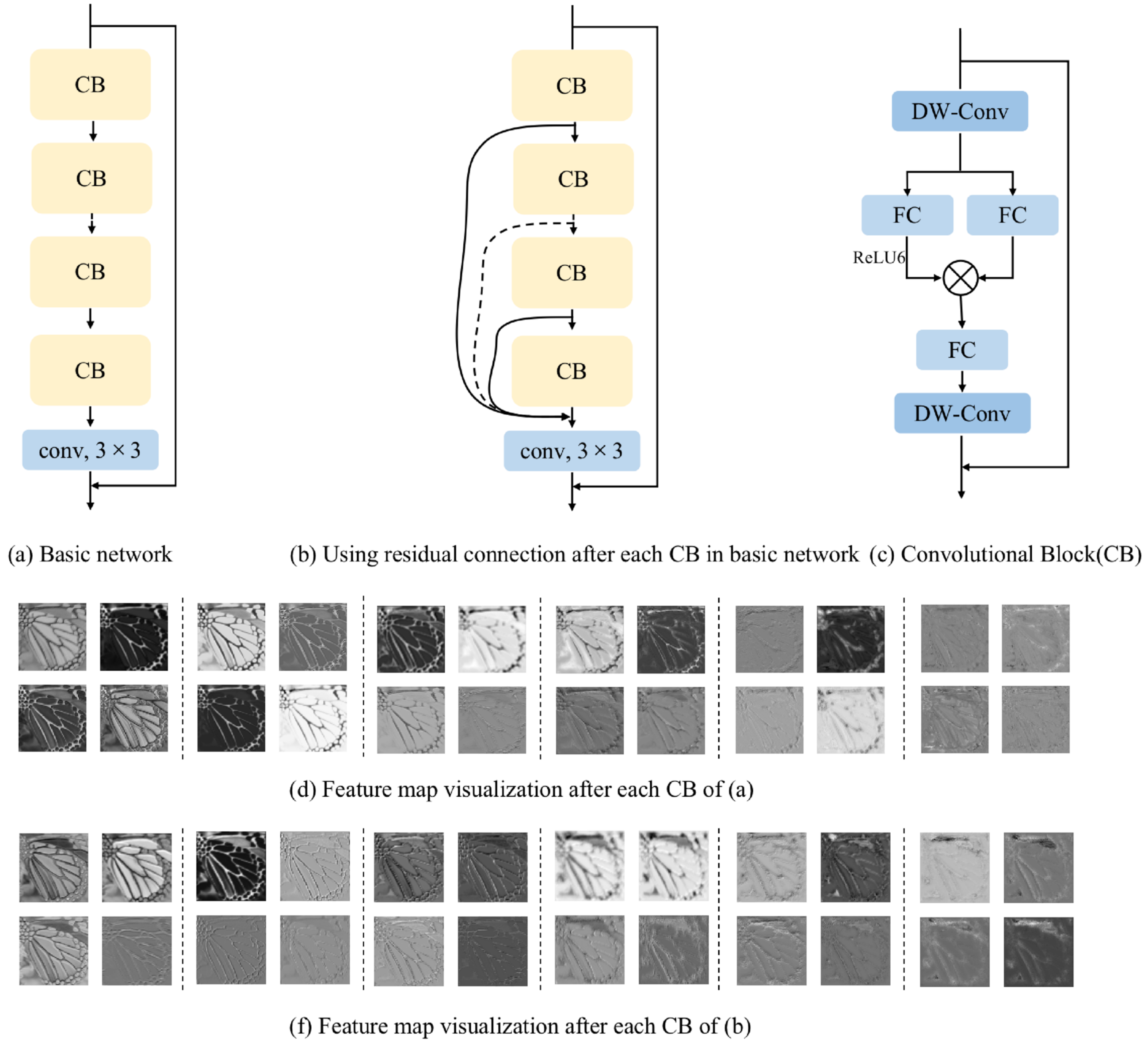
To investigate the impact of different configurations of individual modules in HASB on network performance, we conduct a set of comparative experiments, as shown in Table 6. For example, on Set5, adding CAB to CB increases the PSNR by 0.09dB and the SSIM by 0.0009. Adding ESA to CB increases the PSNR by 0.14dB and the SSIM by 0.0013. When both modules are added, the PSNR and SSIM increase by 0.2dB and 0.0022, respectively. Compared to the remaining five benchmark datasets, our network achieves the best performance when combining CB with the other two attention modules.

To explore the reason behind this phenomenon, we visualize the output features of the last two layers for these four different network structures, as shown in Fig. 7. We can observe that when these two attention modules are not added, the last two layers of the network extract high-level features that focus on local features with fewer details near the output. In contrast, with the addition of these two attention modules, edges and textures near the network input gradually increase. In low-level vision tasks, low-level features are beneficial for improving network performance.

Table 4 Quantitative comparison of different kernel sizes. We use the average PSNR/SSIM on the datasets Set5, Set14, BSD100, Urban100, and Manga109 as the metric

Kernel size	Param	FLOPs	Set5		Set14		B100		Urban100		Manga109	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
3 × 3	202k	13.09	31.74	0.8900	28.32	0.7759	27.39	0.7315	25.50	0.7656	29.49	0.8961
5 × 5	212k	13.75	31.79	0.8903	28.36	0.7768	27.43	0.7329	25.56	0.7687	29.65	0.8979
7 × 7	227k	14.73	31.87	**0.8915**	28.38	0.7770	**27.45**	**0.7338**	**25.59**	0.7700	29.53	**0.8981**
9 × 9	247k	16.04	**31.89**	0.8915	**28.41**	**0.7774**	27.45	0.7336	25.58	**0.7705**	**29.61**	0.8979

The best results are in bold



(a) Basic network (b) Using residual connection after each CB in basic network (c) Convolutional Block(CB)

(d) Feature map visualization after each CB of (a)

(f) Feature map visualization after each CB of (b)

Fig. 6 **a** Basic network consists of several CBs and a 3 × 3 convolutional layer. **b** Based on (**a**), a residual connection is used after each CB. **c** Network structure of the convolutional block. **d** Feature map visualization of the intermediate layers in (**a**) and (**b**)

Table 5 Quantitative comparison of networks with and without residual connections

Method	Set5		Set14		B100		Urban100		Manga109	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
(a)	**31.87**	**0.8915**	**28.38**	**0.7770**	**27.45**	**0.7338**	**25.59**	**0.7700**	**29.53**	**0.8981**
(b)	31.74	0.8897	28.31	0.7756	27.41	0.7324	25.51	0.7669	29.51	0.8955

(a) represents the network without residual connections, and (b) represents the network with residual connections. We use the average PSNR/SSIM on the datasets Set5, Set14, BSD100, Urban100, and Manga109 as the metric. The best results are in bold

Table 6 Quantitative results of the state-of-the-art models on five benchmark datasets

Method	ESA	CAB	Set5		Set14		B100		Urban100		Manga109	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CB	✗	✗	31.87	0.8915	28.38	0.7770	27.45	0.7338	25.59	0.7700	29.53	0.8981
CB	✗	✓	31.96	0.8924	28.44	0.7787	27.48	0.7347	25.70	0.7742	29.90	0.9005
CB	✓	✗	32.01	0.8928	28.42	0.7784	27.47	0.7346	25.75	0.7757	29.86	0.9010
CB	✓	✓	**32.07**	**0.8937**	**28.52**	**0.7802**	**27.52**	**0.7360**	**25.88**	**0.7798**	**30.12**	**0.9031**

The best result is marked with bold. "CB" is convolutional block, which is shown in Fig. 6c

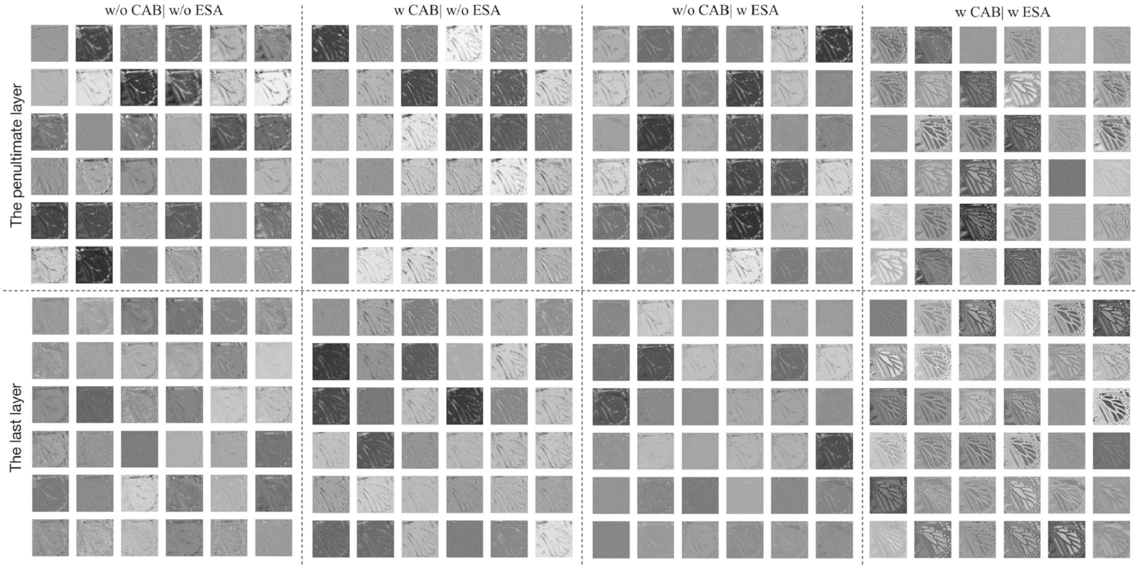


Fig. 7 Visualization analysis of the impact of CAB and ESA on network feature extraction

Table 7 Quantitative comparison of SPAB and HASB

Method	Param	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SPAB	836k	31.91	0.8922	28.39	0.7776	27.45	0.7335	25.66	0.7719	29.88	0.9004
HASB	435k	**32.06**	**0.8937**	**28.52**	**0.7802**	**27.52**	**0.7360**	**25.88**	**0.7798**	**30.12**	**0.9031**

The best results are in bold

Table 8 Quantitative comparison of different activation functions

Method	Set5		Set14		B100		Urban100		Manga109	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
ReLU	31.81	0.8907	**28.40**	**0.7775**	27.45	0.7337	**25.60**	**0.7702**	**29.67**	**0.8985**
LeakyReLU	31.80	0.8909	28.36	0.7768	27.45	0.7336	25.60	0.7702	29.60	0.8983
ReLU6	**31.87**	**0.8915**	28.38	0.7770	**27.45**	**0.7338**	25.59	0.7700	29.53	0.8981

The best result is marked with bold

Table 9 Quantitative comparison of models with and without the warm-start retraining strategy

Method	Dataset	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
w/o	DIV2k	32.06	0.8937	28.52	0.7802	27.52	0.7360	25.88	0.7798	30.12	0.9031
w	DIV2k	32.08	0.8940	28.55	0.7806	27.53	0.7365	25.89	0.7805	30.16	0.9039
w	DF2k	**32.17**	**0.8953**	**28.59**	**0.7817**	**27.58**	**0.7377**	**26.03**	**0.7846**	**30.29**	**0.9055**

"w" indicates the use of the warm-start retraining strategy, while "w/o" indicates the absence of the warm-start retraining strategy. The best result is marked with bold

Additionally, we aim to investigate the characteristics of HASB in advanced feature extraction and low-level feature retention. Therefore, we select SPAB [25], which leverages a parameter-free attention mechanism to achieve feature extraction from shallow to deep layers while maintaining low model complexity and parameter count. We replace HASB with SPAB, keeping all other experimental settings the same. As shown in Table 7, the parameter count of HASB is almost half that of SPAB, but it achieves significant improvements in both PSNR and SSIM across five benchmark datasets.

5.6 Exploration of different activation functions

Most of the previous SR networks adopt ReLU [51] or LeakyReLU [52] as the activation function. ReLU6 [53] is a variant of the ReLU activation function that constrains the output between 0 and 6. It is widely used in mobile and embedded devices because it can provide stable performance in low-precision computing environments. The results in Table 8 show that different activation functions can obviously affect the performance of the model. Among these activation functions, ReLU and ReLU6 perform comparably. In our experiments, we chose ReLU6 as the activation function.

5.7 Effectiveness of warm-start retraining strategy

To demonstrate the effectiveness of our proposed warm-start retraining strategy, we use HASN trained from scratch with DIV2K as the baseline. As shown in Table 9, when not expanding the training set, our model shows a slight performance improvement with the warm-start retraining strategy. When further expanding the training set, our model achieves PSNR improvements of 0.11dB, 0.07dB, 0.06dB, 0.15dB, and 0.17dB on the five benchmark datasets.

6 Conclusion

In this paper, we propose a hybrid attention separable network for efficient image super-resolution (HASN). To make the network more efficient, we use only a few necessary residual connections to avoid gradient vanishing. We design a simple CB module to extract high-level features from the input image and used two essential attention modules (ESA, CAB) to enhance edges and textures near the network input. We conduct extensive feature visualizations to comprehensively analyze the effectiveness of the network structure. Additionally, we propose a warm-start retraining strategy to further exploit the network's performance. Extensive experiments have shown that the proposed method achieves a better balance in performance and lightweight design compared to other networks.

Acknowledgements This work is supported in part by Graduate Education Reform Project of Henan Province (2023SJGLX037Y), National Natural Science Foundation of China (62076223), Key Science and Technology Program of Henan Province (232102211018), and Key Research Project of Henan Province Universities (24ZX005).

Author Contributions Conceptualization was performed by [Xiaoyan Lei], [Weifeng Cao]; methodology by [Xiaoyan Lei], [Weifeng Cao], [Jun Shi], [Wanyong Liang]; formal analysis and investigation by [Xiaoyan Lei], [Jie Liu], [Zongfei Bai]; writing—original draft preparation—by [Xiaoyan Lei]; writing— review and editing—by [Weifeng Cao], [Xiaoyan Lei], [Jun Shi], [Wanyong Liang], [Jie Liu], [Zongfei Bai]; supervision by [Xiaoyan Lei], [Jun Shi], [Wanyong Liang], [Jie Liu], [Zongfei Bai].

Funding No funds, grants, or other support was received.

Data availability All original codes have been deposited at Zenodo (https://doi.org/10.5281/zenodo.12730191) [54].

Declarations

Conflict of interest All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors declare no competing interests.

References

1. Dong, C., Loy, C.C., He, K., et al.: Learning a deep convolutional network for image super-resolution. In: ECCV (4), Lecture Notes in Computer Science, vol 8692. Springer, pp 184–199 (2014)
2. Liang, J., Cao, J., Sun, G., et al.: Swinir: Image restoration using swin transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 1833–1844 (2021)
3. Chen, H., Gu, J., Zhang, Z.: Attention in Attention Network for Image Super-Resolution. arXiv preprint arXiv:2104.09497 (2021)
4. Dong, C., Loy, C.C., He, K., et al.: Learning a deep convolutional network for image super-resolution. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13, Springer, pp 184–199 (2014)
5. Zhang, K., Zuo, W., Gu, S., et al.: Learning deep cnn denoiser prior for image restoration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3929–3938 (2017)
6. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1646–1654 (2016)
7. Lim, B., Son, S., Kim, H., et al.: Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 136–144 (2017)
8. Zhang, Y., Li, K., Li, K., et al.: Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 286–301 (2018)
9. Niu, B., Wen, W., Ren, W., et al.: Single image super-resolution via a holistic attention network. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16, Springer, pp 191–207 (2020)
10. Zhang, Y., Li, K., Li, K., et al.: Residual Non-local Attention Networks for Image Restoration. arXiv preprint arXiv:1903.10082 (2019)
11. Wang, X., Yu, K., Wu, S., et al.: Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European Conference on computer Vision (ECCV) Workshops, pp 0–0 (2018)
12. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: ECCV (2), Lecture Notes in Computer Science, vol 9906. Springer, pp 391–407 (2016)
13. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR. IEEE Computer Society, pp 1646–1654 (2016a)
14. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: CVPR. IEEE Computer Society, pp 1637–1645 (2016b)
15. Lai, W., Huang, J., Ahuja, N., et al.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR. IEEE Computer Society, pp 5835–5843 (2017)
16. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR. IEEE Computer Society, pp 2790–2798 (2017a)
17. Tai, Y., Yang, J., Liu, X., et al.: Memnet: A persistent memory network for image restoration. In: ICCV. IEEE Computer Society, pp 4549–4557 (2017b)
18. Hui, Z., Wang, X., Gao, X.: Fast and accurate single image super-resolution via information distillation network. In: CVPR. IEEE Computer Society, pp 723–731 (2018)
19. Zhang, K., Zuo, W., Zhang, L.: Learning a single convolutional super-resolution network for multiple degradations. In: CVPR. IEEE Computer Society, pp 3262–3271 (2018)
20. Ahn, N., Kang, B., Sohn, K.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: ECCV (10), Lecture Notes in Computer Science, vol 11214. Springer, pp 256–272 (2018)
21. Hui, Z., Gao, X., Yang, Y., et al.: Lightweight image super-resolution with information multi-distillation network. In: ACM Multimedia. ACM, pp 2024–2032 (2019)
22. Liu, J., Tang, J., Wu, G.: Residual feature distillation network for lightweight image super-resolution. In: Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, Springer, pp 41–55 (2020)
23. Kong, F., Li, M., Liu, S., et al.: Residual local feature network for efficient super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 766–776 (2022)
24. Yu, L., Li, X., Li, Y., et al.: Dipnet: efficiency distillation and iterative pruning for image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 1692–1701 (2023)
25. Wan, C., Yu, H., Li, Z., et al.: Swift parameter-free attention network for efficient super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 6246–6256 (2024)
26. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1637–1645 (2016)
27. Ahn, N., Kang, B., Sohn, K.A.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 252–268 (2018)
28. Liu, J., Zhang, W., Tang, Y., et al.: Residual feature aggregation network for image super-resolution. In: CVPR. IEEE, pp 2356–2365 (2020)
29. Li, Z., Liu, Y., Chen, X., et al.: Blueprint separable residual network for efficient image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 833–843 (2022)
30. Zhang, Y., Tian, Y., Kong, Y., et al.: Residual dense network for image super-resolution. In: CVPR. IEEE Computer Society, pp 2472–2481 (2018)
31. Chen, X., Wang, X., Zhou, J., et al.: Activating more pixels in image super-resolution transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 22367–22377 (2023)
32. Zhang, Y., Li, K., Li, K., et al.: Image super-resolution using very deep residual channel attention networks. In: ECCV (7), Lecture Notes in Computer Science, vol 11211. Springer, pp 294–310 (2018)
33. Zhao, H., Kong, X., He, J., et al.: Efficient image super-resolution using pixel attention. In: Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, Springer, pp 56–72 (2020)

34. Liu, Z., Lin, Y., Cao, Y., et al.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 10012–10022 (2021)
35. Deng, W., Yuan, H., Deng, L., et al.: Reparameterized residual feature network for lightweight image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 1712–1721 (2023)
36. Hou, B., Li, G.: Pccformer: Parallel Coupled Convolutional Transformer for Image Super-Resolution. The Visual Computer pp 1–12 (2024)
37. Lin, X., Sun, S., Huang, W., et al.: Eapt: efficient attention pyramid transformer for image processing. IEEE Trans. Multimedia **25**, 50–61 (2021)
38. Zhou, Y., Chen, Z., Li, P., et al.: Fsad-net: feedback spatial attention dehazing network. IEEE Transact. Neural Netw. Learn. Syst. **34**(10), 7719–7733 (2022)
39. Huang, S., Liu, X., Tan, T., et al.: Transmrsr: transformer-based self-distilled generative prior for brain mri super-resolution. Vis. Comput. **39**(8), 3647–3659 (2023)
40. Zhang, X., Zeng, H., Zhang, L.: Edge-oriented convolution block for real-time super resolution on mobile devices. In: Proceedings of the 29th ACM International Conference on Multimedia, pp 4034–4043 (2021)
41. Ding, X., Zhang, X., Ma, N., et al.: Repvgg: making vgg-style convnets great again. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 13733–13742 (2021)
42. Du Zongcai, L.D., Jie, L., Jie, T., et al.: Fast and memory-efficient network towards efficient image super-resolution. In: NTIRE (CVPR Workshop) (2022)
43. Timofte, R., Agustsson, E., Van Gool, L., et al.: Ntire 2017 challenge on single image super-resolution: methods and results. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 114–125 (2017)
44. Bevilacqua, M., Roumy, A., Guillemot, C., et al.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: BMVC. BMVA Press, pp 1–10 (2012)
45. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Curves and Surfaces, Lecture Notes in Computer Science, vol 6920. Springer, pp 711–730 (2010)
46. Martin, D.R., Fowlkes, C.C., Tal, D., et al.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV, pp 416–425 (2001)
47. Huang, J., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR. IEEE Computer Society, pp 5197–5206 (2015)
48. Matsui, Y., Ito, K., Aramaki, Y., et al.: Sketch-based manga retrieval using manga109 dataset. Multimedia Tools Appl. **76**(20), 21811–21838 (2017)
49. Wang, Y., Zhang, T.: Osffnet: Omni-stage feature fusion network for lightweight image super-resolution. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 5660–5668 (2024)
50. Ma, X., Dai, X., Bai, Y., et al.: Rewrite the stars. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 5694–5703 (2024)
51. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp 807–814 (2010)
52. Maas, A.L., Hannun, A.Y., Ng, A.Y., et al.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml, Atlanta, GA, p 3 (2013)
53. Sandler, M., Howard, A., Zhu, M., et al.: Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4510–4520 (2018)
54. nathan66666 (2024) Hasn: v1.0.1. Zenodo, https://doi.org/10.5281/zenodo.12730191

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Weifeng Cao received the M.S. degree from Guizhou University, Guiyang, China, in 2006. He is currently a professor with the School of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou, China. He has published over 20 international journal articles in the areas of information processing and intelligent control. His research interests include artificial intelligence, robotics, and computer vision.

Xiaoyan Lei received the B.E. degree in communication engineering from Zhengzhou University of Aeronautics, Zhengzhou, China, in 2021. She is currently pursuing the M.S. degree with the School of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou, China. Her research interests include image reconstruction and deep learning.

Jun Shi received the M.S. degree from Zhengzhou University of Light Industry, Henan, China, in 2006. He is currently a lecturer with the School of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou, China. His research interests include semiconductor automation and computer vision.

Wanyong Liang received the B.E. degree in industrial automation from Zhengzhou University in 2002 and the M.S. degree in measurement technology and instruments from Zhengzhou University in 2005. Since 2005, he has been working at the School of Electrical and Information Engineering, Zhengzhou University of Light Industry, as an associate professor. His research interests include embedded system design and development, artificial intelligence, and industrial robots.

Jie Liu received dual degrees in electrical engineering and automation, and English from Shandong University of Science and Technology. She is currently pursuing a master's degree at Zhengzhou University of Light Industry, China. Her research interests include neural networks and super-resolution.

Zongfei Bai received the B.E. degree from Zhengzhou University of Light Industry, Zhengzhou, China, in 2016. He is currently an external part-time supervising instructor with the College of Electrical and Information Engineering at Zhengzhou University of Light Industry. His research interests include intelligent control, image reconstruction, and deep learning.