Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.049314

ARTICLE

PSMFNet: Lightweight Partial Separation and Multiscale Fusion Network for Image Super-Resolution

Shuai Cao[1,3], Jianan Liang[1,2,*], Yongjun Cao[1,2,3,4], Jinglun Huang[1,4] and Zhishu Yang[1,4]

[1]Institute of Intelligent Manufacturing, GDAS, Guangdong Key Laboratory of Modern Control Technology, Guangzhou, 510030, China

[2]School of Mechanical & Automotive Engineering, South China University of Technology, Guangzhou, 511442, China

[3]Faculty of Information Engineering and Automation, Kunming University of Science and Technology , Kunming, 650500, China

[4]School of Faculty of Intelligent Manufacturing, Wuyi University, Jiangmen, 529020, China

*Corresponding Author: Jianan Liang. Email: jn.liang@giim.ac.cn

Received: 03 January 2024 Accepted: 07 April 2024 Published: 15 October 2024

ABSTRACT

The employment of deep convolutional neural networks has recently contributed to significant progress in single image super-resolution (SISR) research. However, the high computational demands of most SR techniques hinder their applicability to edge devices, despite their satisfactory reconstruction performance. These methods commonly use standard convolutions, which increase the convolutional operation cost of the model. In this paper, a lightweight Partial Separation and Multiscale Fusion Network (PSMFNet) is proposed to alleviate this problem. Specifically, this paper introduces partial convolution (PConv), which reduces the redundant convolution operations throughout the model by separating some of the features of an image while retaining features useful for image reconstruction. Additionally, it is worth noting that the existing methods have not fully utilized the rich feature information, leading to information loss, which reduces the ability to learn feature representations. Inspired by self-attention, this paper develops a multiscale feature fusion block (MFFB), which can better utilize the non-local features of an image. MFFB can learn long-range dependencies from the spatial dimension and extract features from the channel dimension, thereby obtaining more comprehensive and rich feature information. As the role of the MFFB is to capture rich global features, this paper further introduces an efficient inverted residual block (EIRB) to supplement the local feature extraction ability of PSMFNet. A comprehensive analysis of the experimental results shows that PSMFNet maintains a better performance with fewer parameters than the state-of-the-art models.

KEYWORDS

Deep learning; single image super-resolution; lightweight network; multiscale fusion

1 Introduction

Single image super-resolution (SISR) seeks to generate a high-resolution (HR) image from its low-resolution (LR) counterpart by recovering lost information. The swift advancement of high-speed internet transmission has led to a surge in high-quality data, such as high-definition images and videos,

resulting in the extensive application of SISR across various domains [1]. Consequently, devising an efficient and potent SR technique is crucial for enhancing the visual experience.

The progress of deep learning (DL) has led to the emergence of various SISR approaches that exhibit outstanding performance. The SRCNN [2] achieved superior performance compared to traditional methods using only three convolutional layers. On the basis of residual learning, the VDSR [3] was developed to a depth of 20 layers, while RCAN [4] goes a step further to 400 layers. These networks have achieved impressive performance, but their most significant drawback is the high computational cost, which is not conducive to the practical needs of resource-limited devices. On the other hand, the introduction of Transformer architecture has further developed the field of image restoration. For example, SwinIR [5] achieved more advanced performance than CNN models at the time. Although these models require high computational costs, they have also demonstrated the importance of non-local feature interactions in image reconstruction.

To reduce model parameters and complexity, many lightweight SR networks have been proposed. These networks have employed various strategies to achieve high efficiency, including lightweight module design [1–3], neural network architecture search [4,5] structural reparameterization [6,7], knowledge distillation [8–10] and attention mechanisms [11–15]. They have implemented efficient architecture and modules to significantly reduce the parameters and complexity of the model, but there is still redundancy in the convolution operation. By reducing unnecessary calculations and developing more effective modules, we can construct a more efficient SR model.

Motivated by the aforementioned observations, this paper has proposed a novel lightweight SR network, called partial separation and multiscale fusion network (PSMFNet). By optimizing convolution operations and introducing multiscale feature modulation, it achieves a favorable balance between performance and complexity. Specifically, PSMFNet uses partial convolution (PConv) [16] to construct its basic modules, which reduces a significant amount of calculation redundancy while maintaining feature extraction capability. PConv is advantageous for efficient SR. Moreover, the implementation of long-range dependencies and attention mechanisms can effectively boost the performance of SR networks. In this paper, a multi-scale feature fusion block (MFFB) is proposed to achieve this goal. The MFFB combines a multi-scale spatial feature modulation mechanism and spatial attention enhancement group to deeply explore features in both spatial and channel directions, resulting in better image detail restoration. This paper also proposes an efficient inverse residual block (EIRB) to enhance the extraction of local contextual information.

The specific contributions of this paper are as follows:

Standard convolutions, including grouped convolutions, often involve redundant computations. In order to improve the utilization of convolutions and ensure reconstruction effectiveness, local convolutions are introduced to construct basic modules, demonstrating their effectiveness for super-resolution tasks;

Multiscale feature information is crucial for image reconstruction. In order to address the issue of single-feature extraction, a multiscale feature fusion block is proposed to capture more representative features in both spatial and channel directions. It is combined with effective inverse residual blocks to compensate for the weak local contextual information interaction capability of the network;

In this paper, EIRB and MFFB have been merged into a lightweight feature enhancement block (LFEB) and used to construct PSMFNet. Benchmark dataset evaluations indicate that our PSMFNet strikes an advantageous balance between its performance and the complexity of the model.

2 Related Work

2.1 Deep Learning-Based Image Super-Resolution

With the introduction of the groundbreaking SRCNN [17] network, deep learning has experienced substantial progress in the domain of super-resolution. For example, VDSR [18] has achieved better performance by deepening the network layer. EDSR [19] showed that batch normalization (BN) layers are unnecessary for SR tasks and remove them to enhance the expressive power of model. CARN [20] incorporated dense connections into the network to offset the loss resulting from recursive networks. Lately, image SR tasks have exhibited superior performance compared to CNN models when utilizing the ViT [21] architecture. SwinIR [22] introduces a baseline model based on the Swin Transformer and incorporates it as the feature extraction module in the composite model. The powerful feature extraction capability enables the model to achieve outstanding performance. The GRL [23] network architecture utilizes self-attention mechanism and integrates channel feature information to model image features at different levels of global, regional, and local scopes, leading to improved image restoration results. However, these methods [22,24–27] have brought expensive computational costs along with their excellent performance, making deployment on resource-constrained devices more challenging. This has also prompted developers to develop more efficient SR methods.

2.2 Efficient Image Super-Resolution

Aiming to lower the computational cost associated with the model, many effective SR methods utilizing CNN have been introduced [28,29]. FSRCNN [30] adopted a post-upsampling method to reduce complexity while maintaining performance. ESPCN [31] developed a sub-pixel convolution that directly transforms LR images into HR images at the end, thereby reducing time complexity. PAN [14] proposed a pixel attention approach that greatly reducing the parameters and achieving better SR performance. IMDN [9] developed an information distillation block to separate and refine features, thereby enhancing the restoration of image details. RFDN [10] reevaluated the network structure of IMDN and introduced a shallow residual block as the foundational module of RFDN. By incorporating feature distillation connections, RFDN achieved a lighter architecture compared to IMDN. ShuffleMixer [32] explored image feature extraction from a different perspective by employing large convolutions and channel-wise shuffle operations instead of stacking multiple small-kernel convolutions. It also introduces Fused-MBConvs to enhance local connectivity, effectively restoring image details. RLFN [33] simplified the feature aggregation operation of RFDN by employing three layers of standard convolution for residual connections, enhancing the learning of local features and significantly improving the model's runtime. FMEN [34] designed a high-frequency attention block to enhance image details, and applied structural reparameterization to reduce feature fusion and further accelerate network inference speed. BSRN [35] built the model based on blueprint separable convolution [36], reducing redundant operations in depthwise convolution (DWConv) but also increasing inference time. To incorporate the benefits of ViT, SAFMN [37] proposed a lightweight spatially adaptive feature modulation module to learn long-range dependencies from multiscale features. The above methods have made improvements to the model in different aspects, but there is still room for trade-offs.

3 Method

3.1 Network Architecture

The architecture of Partial Separation and Multiscale Fusion Network (PSMFNet) that has been proposed in this paper is shown in Fig. 1. The PSMFNet is composed of three primary components: Shallow feature extraction, multiple stacked lightweight feature enhancement modules (LFEB), and

an image reconstruction module. In the initial segment, a 3 × 3 convolutional layer is employed to extract shallow features from the input image. In this paper, the ILR has been represented as an input to PSMFNet, and the operation can be represented as:

$$F_0 = H_{sfe}(I_{LR}) \tag{1}$$

where $H_{sfe}(\cdot)$ denotes the module for extracting shallow features, and F_0 is the extracted shallow features. A stack of LFEBs is then utilized to extract deep features from F_0. This process can be denoted as:

$$F_n = H^n_{LFEB}(F_{n-1}) = H^n_{LFEB}((H^{n-1}_{LFEB}(\ldots H^0_{LFEB}(F_0))) \tag{2}$$

where $H^n_{LFEB}(\cdot)$ denotes the *n-th* LFEB function, while F_{n-1} and F_n represent the input and output features of the *n-th* LFEB, respectively.

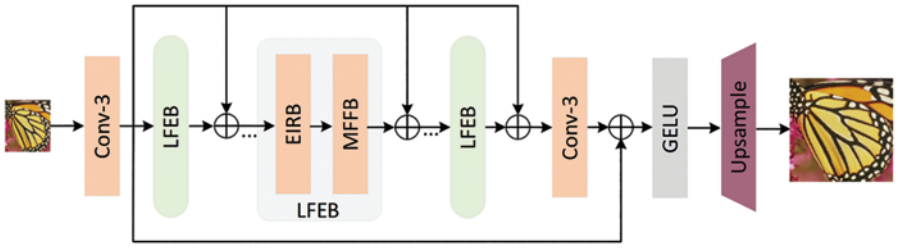


Figure 1: The architecture of partial separation and multiscale fusion network (PSMFNet)

Subsequently, in this paper, 3 × 3 convolutional layers have been used to smooth the extracted depth feature maps and introduce long jump connections before image reconstruction. Finally, the image reconstruction module is employed to produce the final output ISR, which can be depicted as:

$$I_{SR} = H_{rec}(H_{smooth}(F_n) + F_0) \tag{3}$$

where $H_{rec}(\cdot)$ denotes the image reconstruction module function, encompassing a 3 × 3 convolutional layer and a sub-pixel convolution [31] operation. $H_{smooth}(\cdot)$ represents the 3 × 3 convolutional operation. We optimized the model using the $L1$ loss function, which can be expressed as:

$$L1(I_{SR}, I_{HR}) = \sum_{i=1}^{m} |I_{SR} - I_{HR}| \tag{4}$$

where I_{HR} denotes the ground-truth image. The $L1$ loss calculates the sum of the absolute differences between the actual values and the target values. In image super-resolution tasks, the goal is to make the generated image as close as possible to the real high-resolution image. The $L1$ loss calculates the error between the values at corresponding pixel positions in the super-resolved and high-resolution images.

3.2 Efficient Inverted Residual Block

In previous work, MobileNetv2 [3] proposed a depthwise convolution-based inverted residual block. Although the introduction of depthwise convolution (DWConv) [2] reduces the computational complexity and parameters of the model, expanding the channels increases the time required for convolutional operations. Recently, a novel approach [16] has been developed, which utilizes an efficient module based on PConv. This module directly connects to the inverted residual block after

PConv, eliminating the DWConv in the residual block and reducing the computational burden of convolution, resulting in faster inference speed.

Given an input $I \in R^{H\times W\times C_1}$, convolution operation is performed on it using a kernel of size k × k, resulting in an output $O \in R^{H\times W\times C_2}$. In standard convolution, the number of floating point operations is:

$$FLOPs_{regular} = k^2 \times H \times W \times C_1 \times C_2 \tag{5}$$

From Fig. 2, it is evident that when r is 4, which means one-fourth of the input features undergo convolution, the FLOPs of local convolution is:

$$FLOPs_{PConv} = k^2 \times H \times W \times \frac{C_1}{4} \times \frac{C_2}{4} \tag{6}$$

In general, the number of output channels in convolution matches the number of input channels. In this scenario, the FLOPs of local convolution is only 1/16 of the standard convolution.

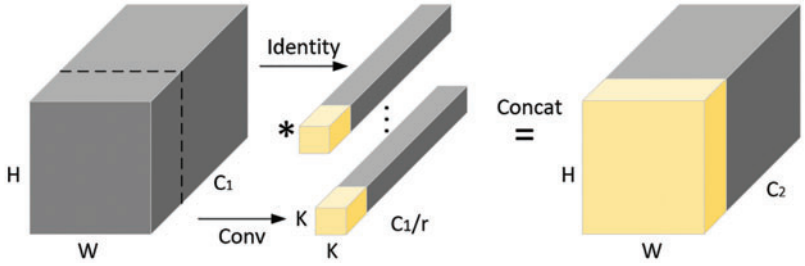


Figure 2: Partial convolution

This module has been further optimized in this paper according to specific SR tasks. The batch normalization layer has been shown to potentially cause unexpected artifacts in image reconstruction [19,38], so we removed it. Additionally, GELU [39] has become the preferred choice for recent SR methods [22,35,37]. As shown in Fig. 3, given the input feature Fin, the entire structure can be described by as:

$$F_{shortcut} = F_{in} \tag{7}$$

$$F_{EIRB} = H_{EIRB}(F_{in}) + F_{shortcut} \tag{8}$$

where $F_{shortcut}$ denotes the shortcut operation, $H_{EIRB}(\cdot)$ denotes the EIRB function, and F_{EIRB} represents the output feature of EIRB.

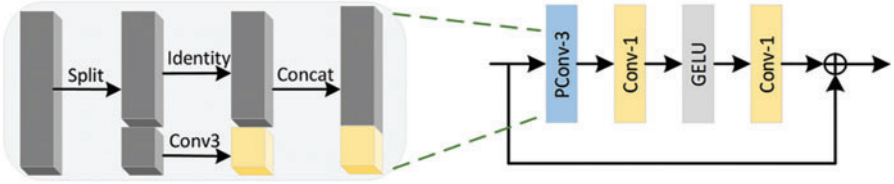


Figure 3: The structure of EIRB. Partial convolution (PConv) only extracts features through convolution on a portion of the input channels, without affecting the remaining channels

3.3 Multiscale Feature Fusion Block

Self-attention mechanisms [23–25] have the ability to capture long-range dependencies within neural networks and boost model performance. However, the adoption of self-attention mechanisms leads to a marked increase in parameters and computational complexity. Some researchers have proposed alternative approaches to self-attention mechanisms, such as the utilization of large convolutional kernels [40] or spatially adaptive feature modulation [37]. All of these methods share a common feature of utilizing DWConv, which results in a reduction of the feature extraction ability. Moreover, they frequently neglect the channel information of the image, which results in incomplete information for image reconstruction. Therefore, in this paper, we have proposed a lightweight multiscale feature fusion block (MFFB) that explores important features in the spatial and channel domains while learning long-range dependencies. As illustrated in the Fig. 4, MFFB focuses on a wider range of pixel information, ensuring the reconstruction of image details as much as possible.

(a) Spatial long-range feature extraction (b) Channel feature extraction

Figure 4: Multi-scale feature fusion

The MFFB primarily consists of multiscale spatial feature modulation (MSFM) block and channel attention enhancement group (CAEG). As shown in Fig. 5, the MFFB divides the features output by EIRB into three parts for processing. In the first part, spatial features are extracted from a long-range perspective through MSFM. Firstly, the features undergo channel split operations, and then each feature component is sent to different levels of spatial feature extraction channels. This procedure can be expressed as:

$$\begin{aligned} &[F_{\mathrm{d0}}, F_{\mathrm{d1}}, F_{\mathrm{d2}}, F_{\mathrm{d3}}] = Split\left(F_{EIRB}\right) \\ &F_0 = H_{pconv}\left(F_{\mathrm{d0}}\right) \\ &F_i = \uparrow_{2^i}\left(H_{pconv}\left(\downarrow_{\frac{1}{2^i}}\left(F_{\mathrm{d}i}\right)\right)\right), 1 \le i \le 3 \end{aligned} \tag{9}$$

where $Split(\cdot)$ represents channel split operation. $H_{pconv}(\cdot)$ represents the PConv operation. $\uparrow_{2^i}$ represents upsampling the feature map to the original input size using nearest neighbor interpolation. $\downarrow_{\frac{1}{2^i}}$ represents downsampling the input feature to a size of $\frac{1}{2^i}$. In this paper, we have concatenated features from different spatial levels, aggregated them using 1×1 convolution and activated them nonlinearly using GELU. This procedure can be expressed as:

$$F_{part1} = GELU\left(H_{pwconv}\left(Concat\left([F_{\mathrm{d0}}, F_{\mathrm{d1}}, F_{\mathrm{d2}}, F_{\mathrm{d3}}]\right)\right)\right) \tag{10}$$

where F_{part1} represents the output of MSFM, $H_{pwconv}(\cdot)$ and $Concat(\cdot)$ denote pointwise convolution (PWConv) and concatenation operations, respectively. In the second part, this paper has used Partial Convolutional Enhancement Group (PCEG) consisting of 3 × 3 PConv and PWConv to filter the input features and the process can be formulated as follows:

$$F_{part2} = H_{PCEG}(F_{EIRB}) \tag{11}$$

where F_{part2} denotes the output of PCEG, and $H_{PCG}(\cdot)$ denotes the PCEG function. In the final part, CAEG is used to prevent the loss of channel information. CAEG consists of two parts, and in order to obtain useful channel information from deeper layers, we use PCEG for feature enhancement before CA [11]. This procedure can be formulated as:

$$F_{part3} = H_{CA}(H_{PCEG}(F_{EIRB})) \tag{12}$$

where F_{part3} denotes the output of CAEG, and $H_{CA}(\cdot)$ denotes the CA function.

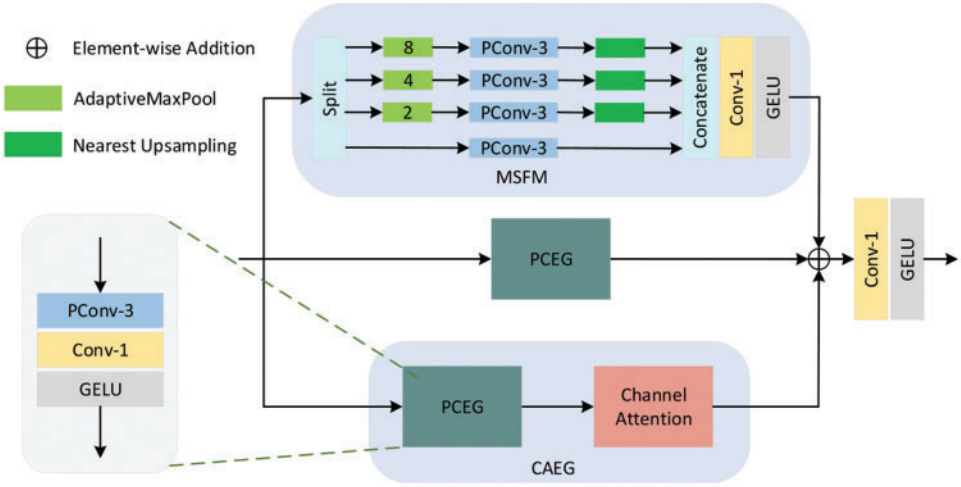


Figure 5: Global feature fusion block (MFFB)

After obtaining representative features, in this paper, they have been aggregated using 1 × 1 convolution method and normalized using GELU nonlinear function. The feature aggregation is formulated as:

$$F_{MFFB} = GELU\left(H_{pwconv}\left(F_{part1} + F_{part2} + F_{part3}\right)\right) \tag{13}$$

where F_{MFFB} denotes the output of MFFB.

3.4 Image Reconstruction Module

The high-resolution images obtained by networks like SRCNN through bicubic interpolation may lead to increased time complexity. Therefore, in this paper, the PixelShuffle operation is used to upsample images.

As shown in Fig. 6, After the feature extraction module, a convolutional layer is utilized to generate r × r channel feature maps, where r represents the upsampling factor. Subsequently, PixelShuffle is employed to reorganize the r × r channel feature maps into an upsampled image of size W × r, H × r, where W and H denote the width and height of the low-resolution image, respectively.

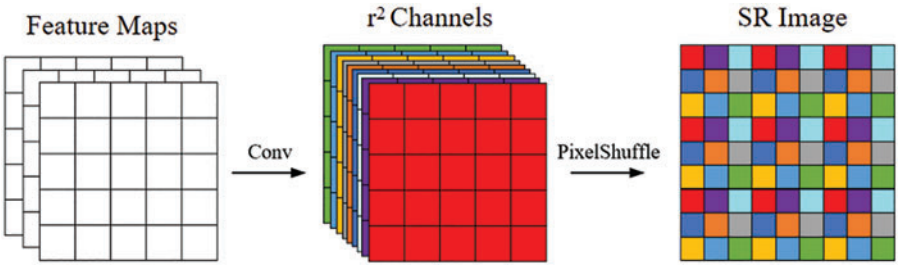


Figure 6: Image reconstruction module

4 Experiment

4.1 Datasets and Metrics

The training image collection consists of 800 images originating from DIV2K [41] and 2650 images derived from Flickr2K [19]. In this paper, our model has been evaluated using five commonly used benchmark datasets: Set5 [42], Set14 [43], BSD100 [44], Urban100 [45], and Manga109 [46].

Set5 and Set14 contain 5 and 14 test images, respectively, covering a variety of scenes and content. These images are used to comprehensively evaluate the performance of algorithms in different scenarios and settings. BSD100 contains 100 test images with higher complexity and diversity, designed to evaluate algorithms in real-world scenarios. Urban100 is a super-resolution reconstruction dataset tailored for urban landscapes, comprising 100 test images typically featuring buildings, streets, and cityscapes to assess algorithm performance in urban environments. Manga109 is a super-resolution reconstruction dataset specifically curated for manga images, which often exhibit unique styles and details, aimed at evaluating algorithm performance in handling manga-style images.

In this paper, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [47] have been used as evaluation metrics. All PSNR and SSIM values are computed on the Y channel of images converted to the YCbCr color space. Given the ground-true image I_{HR} and the super-resolution image I_{SR}, PSNR is defined as:

$$PSNR\left(I_{HR,}I_{SR}\right) = 10\log_{10}\frac{\max_{value}^{2}}{MSE} \tag{14}$$

where

$$MSE = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(I_{SR}\left(i,j\right) - I_{HR}\left(i,j\right)\right)^{2} \tag{15}$$

Max_{Value} represents the maximum pixel value, and M and N respectively denote the height and width. The definition of $SSIM$ is as follows:

$$SSIM\left(I_{HR}, I_{SR}\right) = \frac{\left(2\mu_{hr}\mu_{sr} + c_1\right)\left(2\sigma_{hr\cdot sr} + c_2\right)}{\left(\mu_{hr}^2 + \mu_{sr}^2 + c_1\right)\left(\sigma_{hr}^2 + \sigma_{sr}^2 + c_2\right)} \tag{16}$$

The variables μ_{hr} and μ_{sr} represent the mean grayscale value, σ_{hr} and σ_{sr} is the variance of the image, and $\sigma_{hr\cdot sr}$ is the covariance of the image. $c_1 = (K_1L)^2$ and $c_2 = (K_2L)^2$ are two constant terms, where L represents the range of pixel values.

4.2 Implementation Details

During the data augmentation process for training, this paper applied random rotations of 90, 180, and 270 to the images, as well as horizontal flipping. Furthermore, this paper randomly extracted 64 patches of 48 × 48 pixels from the LR images to serve as training inputs for the model. The model in this paper needs to balance the accuracy and complexity of the model. When the number of LFEB is 9, Param (K) is 435, FLOPs (G) is 24.5, and Acts (M) is 270, which is obviously higher than the current advanced SR method such as ShuffeMixer and does not meet the lightweight requirements. When LFEB is 7, Param (K) is 353, FLOPs (G) is 19.9, Acts (M) is 212, and the reconstruction quality cannot be guaranteed. Therefore, the number of LFEB selected in this paper is 8, which can improve the quality of image reconstruction as much as possible under the premise of lightweight. The proposed PSMFNet consists of 8 LFEBs with a channel number of 64, and PConv preserves three-quarters of the channels. This paper utilized the Adam [48] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to solve the proposed model. The iteration quantity is fixed at 1×10^6. The initial learning rate is configured at 5×10^{-4}, which is updated by the Cosine Annealing scheme [49]. All experiments are conducted using the Pytorch framework on a GeForce RTX 3090 GPU.

4.3 Ablation Study

This paper conducted extensive ablation experiments in this section to further evaluate the effectiveness of each component of PSMFNet. We trained all experiments based on ×4 PSMFNet with the same settings.

4.3.1 Effectiveness of the Partial Convolution

Unlike DWConv [2] and group convolution (GConv) [50], PConv only performs convolution on a portion of the input channels, while the remaining channels are preserved. Compared to standard convolution, PConv has fewer parameters and FLOPs, while also possessing superior spatial feature extraction capabilities compared to DWConv and GConv. As shown in Table 1, when PConv was replaced with DWConv in the backbone network, the PSNR decreased by 0.05 and 0.16 dB on the DIV2K-val and Urban100 datasets, respectively. Similarly, when GConv was used, the PSNR decreased by 0.03 and 0.13 dB on the same datasets after replacing PConv with DWConv in the backbone network. The experimental results demonstrate that PConv not only maximizes the feature extraction capabilities, but also positively contributes to subsequent modeling through the preserved features.

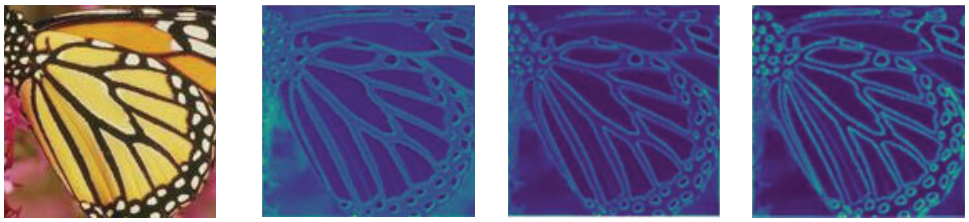
4.3.2 Effectiveness of the Lightweight Feature Enhancement Block

This paper visualized the feature maps in Fig. 7 to illustrate the effectiveness of LFEB. LFEB consists of two modules, MFFB and EIRB, which explore global and local features, respectively. To verify their importance, this paper conducted the following experiments: (1) Using only two MFFB blocks in LFEB, and (2) using only two EIRB blocks in LFEB. The reason for doing this is to ensure that the models have similar parameters and achieve a fairer comparison. In Table 1, w/o EIRB indicates the use of only two MFFBs in LFEB, while w/o MFFB indicates the use of only two EIRBs in LFEB. Table 1 shows that using only MFFB resulted in a decrease of 0.03 and 0.12 dB in PSNR on the DIV2K-val and Urban100 datasets, respectively, while using only EIRB resulted in a decrease of 0.11 and 0.3 dB in PSNR on the same datasets. As shown in Fig. 8, The decrease in performance indicates that using only GFFB or EIRB alone will result in the loss of some useful information, leading to a

decrease in image reconstruction quality. However, using both modules simultaneously can fuse rich features and improve model performance.

Table 1: Ablation for PSMFNet on DIV2K-val and Manga109 datasets. PSMFNet with a scaling factor of ×4 is utilized as the baseline for ablation studies. The PSNR/SSIM values on benchmarks are reported. "X→Y" is to replace X with Y. "FA" is an abbreviation for feature aggregation. The numbers of parameters, FLOPs, and Acts are counted by the fvcore library with a resolution of 320 × 180 pixels

Ablation	Variant	Param (K)	FLOPs (G)	Acts (M)	DIV2K-val PSNR/SSIM	Urban100 PSNR/SSIM
Baseline	–	394	22.2	241	30.53/0.8397	26.31/0.7918
Main module	PConv→DWconv	355	19.8	315	30.48/0.8387	26.15/0.7873
	PConv→GConv (16 groups)	396	22.2	315	30.50/0.8390	26.18/0.7886
	w/o MFFB	365	21.0	202	30.42/0.8371	26.01/0.7826
	w/o EIRB	423	23.4	28	30.50/0.8387	26.19/0.7885
MFFB	w/o MSFM	375	21.3	217	30.42/0.8370	26.13/0.7856
	w/o CA	389	22.2	241	30.52/0.8397	26.21/0.7895
	w/o PCEG	343	19.3	204	30.48/0.8386	26.15/0.7872
	w/o FA	361	20.4	212	30.46/0.8379	26.14/0.7868
EIRB	w/o BN	396	22.5	241	30.51/0.8390	26.27/0.7902
	EIRB→FasterNet Block	396	22.5	241	30.50/0.8389	26.24/0.7901
	EIRB→IRB	427	24.5	322	30.50/0.8389	26.28/0.7908



(a) LR input (b) F_{sf} (c) F_{global} (d) F_{LFEB}

Figure 7: Illustration of learned deep features from the LFEB ablation: (a) Input image; (b) Shallow features of input images; (c) The feature map after obtaining global information by MFFB; (d) The feature map after the supplementation of local information by EIRB

4.3.3 Effectiveness of the Efficient Inverse Residual Block

Compared to the FasterNet Block [16], the EIRB has made modifications by removing the BN layer [51]. Additionally, this paper have utilized the GELU [39] activation function to better suit the SR task. This paper will conduct a sequence of ablation experiments to show its ability to effectively extract local contextual information.

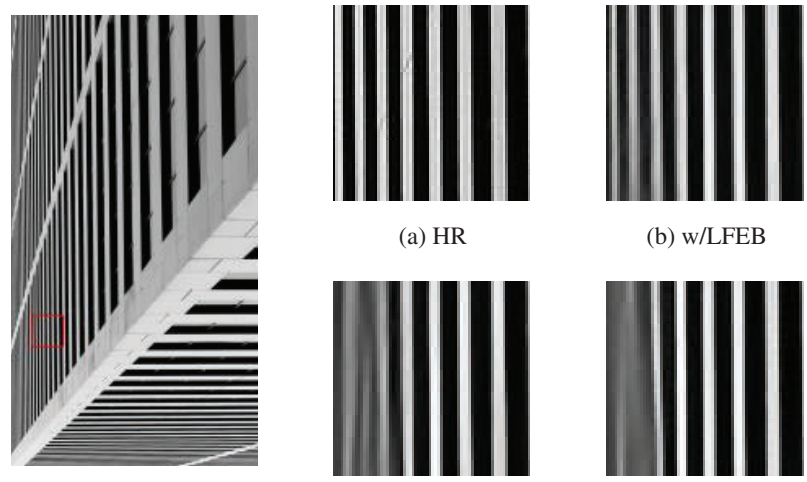


Figure 8: Effect of the MFFB and the EIRB in the LFEB for SISR

As shown in Table 1, this paper replaced EIRB with FasterNet Block, and the PSNR on the Urban100 dataset decreased by 0.07 dB. This performance decrease was due to the influence of the BN layer and activation function. When this paper replaced EIRB with IRB [3], although the performance only decreased by 0.03 dB after channel expansion through the inverse residual block, the corresponding parameter count and FLOPs increased by nearly 30 K and 2.0 G, respectively, and the activations (Acts) also increased by nearly 80 M. It can be observed that the expansion of channels in the inverse residual block leads to a significant convolutional computational burden. Additionally, a performance decrease of 0.04 dB was observed when a BN layer was added after the first PWConv. Therefore, the improved EIRB has a more efficient performance.

4.3.4 Effectiveness of the Multiscale Feature Fusion Block

The MFFB primarily consists of a multiscale spatial feature modulation block and channel attention enhancement group. As shown in Fig. 9, this module enables the model to integrate more diverse features. Here, this paper delved deeper into this module to uncover the reasons behind its effectiveness.

- Multiscale spatial feature modulation. Here, "w/o MSFM" in Table 1 indicates that we replaced the MSFM in the MFFB with a PConv of kernel size 3 × 3. Without modulation of spatial features, a decrease of 0.18 dB in PSNR values was observed on the Urban100 dataset. It is evident that the lack of learning spatial long-range dependency features has a significant impact on performance;
- Channel attention enhancement group. When only used CA [11] without PCEG to filter features, a decrease of 0.16 dB in PSNR values was observed on the Urban100 dataset. If only PCEG is left, MFFB lacks channel information, leading to a decrease in image modeling ability and a 0.1 dB decrease in PSNR value on the dataset;

Feature aggregation. This paper used 1 × 1 convolution to aggregate spatial and channel information. After feature aggregation, the PSNR on the Urban100 dataset increased by 0.17 dB, demonstrating the necessity of aggregating spatial and channel features.

Figure 9: Comparison of local attribution maps (LAMs) [52] and diffusion indices (DIs) [52] between PSMFNet and other efficient SR models. The LAM outcomes highlight the significance of each pixel in the input LR image when processing the patches denoted by red boxes in SR. The DI value reflects the range of pixels involved. A larger DI value corresponds to a broader attention range. The proposed method can utilize more feature information

4.4 Comparisons with State-of-the-Art Methods

To gauge the performance of PSMFNet, this paper conducted a comparison with multiple state-of-the-art lightweight image super-resolution approaches, including SRCNN [17], ESPCN [31], VDSR [18], LapSRN [53], CARN [20], IDN [8], IMDN [9], PAN [14], LAPAR-A [54], RFDN [10], ShuffleMixer [32], RLFN [33]. Fig. 10 shows that PSMFNet achieves comparable performance at lower complexity.

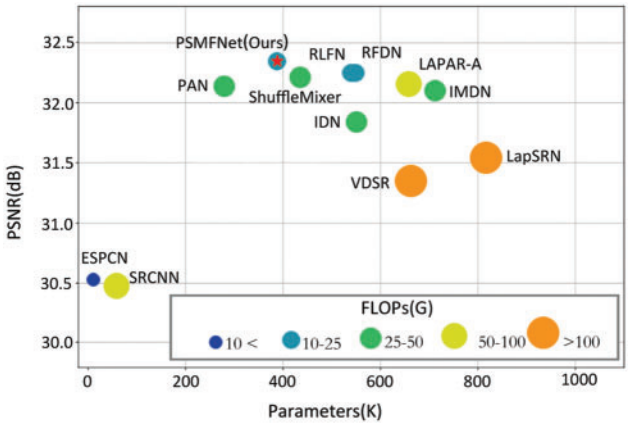


Figure 10: The complexity and performance of proposed PSMFNet model are compared with other lightweight methods on the Set5 dataset for $\times 4$ SR

4.4.1 Quantitative Comparisons 1

The quantitative comparison findings for various upscaling factors on five benchmark datasets are presented in Table 2. Along with the PSNR/SSIM indicators, this paper also included the number of parameters (Params) and floating-point operations (FLOPs). Benefiting from the simple yet efficient structure, the proposed PSMFNet achieved comparable performance with fewer parameters. Taking the example of $\times 4$ SR on the Urban100 dataset, PSMFNet has approximately 75% fewer parameters than CARN, 28% fewer parameters than RFDN, and 27% fewer parameters than RLFN. The results of quantitative comparison show that PSMFNet achieves the highest accuracy with fewer parameters.

Table 2: A quantitative analysis is conducted, comparing this paper method to state-of-the-art approaches on benchmark datasets. The optimal and suboptimal performances are denoted in red and blue, respectively. FLOPs are computed using a 1280 × 720 GT image

Method	Scale	Params	FLOPs	Set5	Set14	B100	Urban100	Manga109
		(K)	(G)	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Bicubic	×2	–	–	33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403	30.80/0.9339
SRCNN [17]	×2	57	52.7	36.66/0.9299	32.45/0.9607	31.36/0.8879	29.50/0.8946	35.60/0.9663
ESPCN [31]	×2	21	5	36.83/0.9564	32.40/0.9096	31.29/0.8917	29.48/0.8975	–
VDSR [18]	×2	666	612.6	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140	37.22/0.9750
LapSRN [53]	×2	251	29.9	37.52/0.9591	32.99/0.9124	31.80/0.8952	30.41/0.9103	37.27/0.9740
CARN [20]	×2	1952	222.8	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256	38.36/0.9765
IMDN [9]	×2	694	158.8	38.00/0.9605	33.63/0.9177	32.19/0.8996	32.17/0.9283	38.88/0.9774
PAN [14]	×2	261	70.5	38.00/0.9605	33.59/0.9181	32.18/0.8997	32.01/0.9273	38.70/0.9773
LAPAR-A [54]	×2	548	171.0	38.01/0.9605	33.62/0.9183	32.19/0.8999	32.10/0.9283	38.67/0.9772
RFDN [10]	×2	534	95.0	38.05/0.9606	33.68/0.9184	32.16/0.8994	32.12/0.9278	**38.88**/0.9773
ShuffleMixer [32]	×2	394	91.0	38.01/0.9606	33.63/0.9180	32.17/0.8995	31.89/0.9257	38.83/**0.9774**
RLFN [33]	×2	527	115.4	**38.07/0.9607**	**33.72/0.9187**	**32.22/0.9000**	**32.33/0.9299**	–
PSMFNet (ours)	×2	373	84.0	**38.12/0.9609**	**33.70/0.9185**	**32.24/0.9004**	**32.46/0.9307**	**39.13/0.9780**
Bicubic	×3	–	–	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349	26.95/0.8556
SRCNN [17]	×3	57	52.7	32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989	30.48/0.9117
VDSR [18]	×3	666	612.6	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279	32.01/0.9340
CARN [20]	×3	1592	118.8	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493	33.50/0.9440
IMDN [9]	×3	703	71.5	34.36/0.9270	30.32/0.8417	29.09/0.8046	28.17/0.8519	33.61/0.9445
PAN [14]	×3	261	39.0	34.40/0.9271	30.36/**0.8423**	29.11/0.8050	28.11/0.8511	33.61/0.9448
LAPAR-A [54]	×3	544	114.0	34.36/0.9267	30.34/0.8421	29.11/**0.8054**	28.15/0.8523	33.51/0.9441
RFDN [10]	×3	541	42.2	**34.41/0.9273**	30.34/0.8420	29.09/0.8050	**28.21/0.8525**	33.67/**0.9449**
ShuffleMixer [32]	×3	415	43.0	34.40/0.9272	**30.37/0.8423**	**29.12**/0.8051	28.08/0.8498	**33.69**/0.9448
PSMFNet (ours)	×3	382	38.1	**34.52/0.9281**	**30.42/0.8431**	**29.16/0.8063**	**28.36/0.8560**	**33.99/0.9496**
Bicubic	×4	–	–	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866
SRCNN [17]	×4	57	52.7	30.48/0.8628	27.49/0.7503	26.90/0.7101	24.52/0.7221	27.66/0.8505
ESPCN [31]	×4	25	1	30.52/0.8697	27.42/0.7606	26.87/0.7216	24.39/0.7241	–
VDSR [18]	×4	666	612.6	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524	28.83/0.8870
LapSRN [53]	×4	813	149.4	31.54/0.8852	28.09/0.7700	27.32/0.7275	25.21/0.7562	29.09/0.8900
CARN [20]	×4	1592	90.9	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	30.47/0.9084
IMDN [9]	×4	715	40.9	32.21/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838	30.45/0.9075
PAN [14]	×4	272	28.2	32.13/0.8948	28.61/0.7822	27.59/0.7363	26.11/0.7854	30.51/**0.9095**
LAPAR-A [54]	×4	659	94.0	32.15/0.8944	28.61/0.7818	**27.61/0.7366**	26.14/0.7871	30.42/0.9074
RFDN [10]	×4	550	23.9	**32.24**/0.8952	28.61/0.7819	27.57/0.7360	26.11/0.7858	30.58/0.9089
ShuffleMixer [32]	×4	411	28	32.21/**0.8953**	**28.66/0.7827**	**27.61/0.7366**	26.08/0.7835	**30.65**/0.9093
RLFN [33]	×4	543	29.8	**32.24**/0.8952	28.62/0.7813	27.60/0.7364	**26.17/0.7877**	–
PSMFNet (ours)	×4	394	22.1	**32.35/0.8970**	**28.73/0.7842**	**27.65/0.7382**	**26.31/0.7918**	**30.87/0.9124**

4.4.2 Quantitative Comparisons 2

In addition to quantitative evaluation, this paper conducted a qualitative analysis of proposed PSMFNet by comparing it with state-of-the-art methods through visual comparison. It can be observed in Fig. 11 that the images restored by PSMFNet exhibit superior performance in terms

of texture details. The results validate that the proposed PSMFNet, which utilizes multiscale feature fusion, can explore deeper and more effective features.

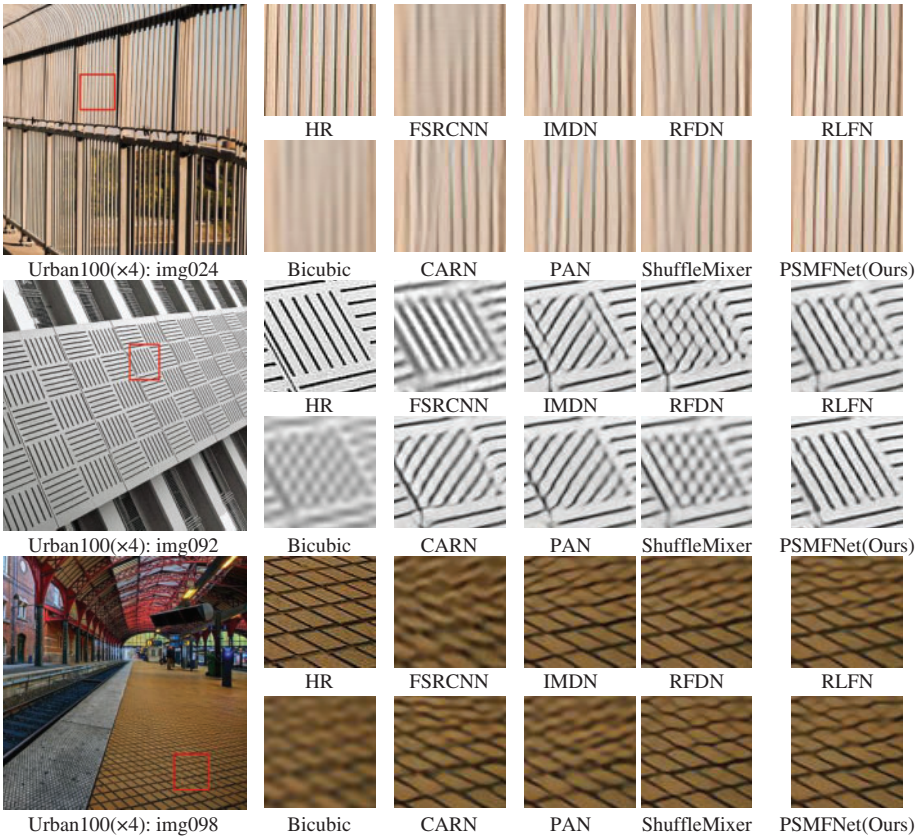


Figure 11: Visual comparisons for ×4 SR on the Urban100 dataset

Urban100 is a dataset primarily focused on urban landscapes. To validate the performance of the model in other scenarios, images from the Set14, Manga109, and B100 datasets were selected for visualization, as shown in Fig. 12.

4.4.3 Memory and Running Time Comparisons

To further validate the efficiency of PSMFNet, this paper compared it with five representative efficient SR methods, including CARN [21], IMDN [13], PAN [17], RFDN [14], and RLFN [51]. This paper conducted tests on the DIV2K-val ×4 dataset and recorded the maximum GPU memory consumption (GPU Mem) and average running time (Avg.Time) during the inference process to further validate the performance of PSMFNet. A comparison of memory consumption and runtime has been presented in Table 3, where PSMFNet's GPU consumption is only 30% of that of the CARN series and 37% of that of IMDN; Compared with PAN, our method has a similar running speed but significantly reduces GPU memory consumption. To fully leverage the advantages of Partial Convolution, this paper employed Pointwise Convolution to enhance the fusion of features. However, Pointwise Convolution is a computationally intensive operation in convolutional neural networks, which may impact the utilization of hardware resources and consequently affect the execution speed of tasks. This leads to certain drawbacks in terms of runtime compared to RFDN and RLFN. Tables 2

and 3 demonstrate that proposed PSMFNet achieves a favorable balance between model complexity and performance.

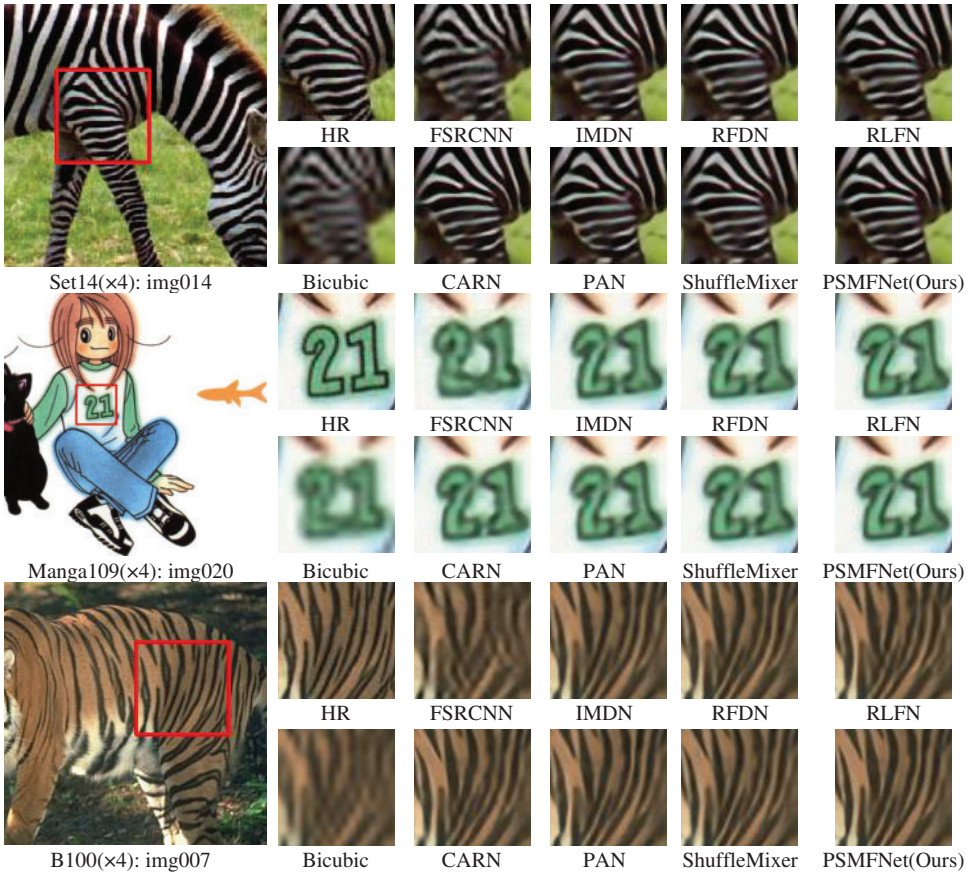


Figure 12: visual comparisons of ×4 SR on other SR datasets

Table 3: Memory and running time comparisons on DIV2K-val ×4. GPU Mem. denotes the maximum GPU memory consumption during the inference phase, and Avg.Time denotes the average running time. The testing method is based on the testing code of the NTIRE 2022 Challenge on Efficient Super-Resolution [55]

Methods	GPU Mem (M)	Avg.Time (ms)
CARN [20]	3058.11	51.26
IMDN [9]	2546.79	46.46
PAN [14]	1229.45	34.80
RFDN [10]	767.10	24.80
RLFN [33]	629.26	19.40
PSMFNet (ours)	957.02	35.96

5 Conclusion

In this paper, a simple and efficient model has been proposed to solve the problem of efficient image super-resolution, which is called Partial Separation and Multiscale Fusion Network (PSMFNet). The lightweight feature enhancement module (LFEB), based on partial convolution (PConv), is constructed as the basic module of PSMFNet. The efficient and lightweight architecture design effectively reduces redundant convolution operations. This module consists of a multiscale feature fusion block (MFFB) and an efficient inverse residual block (EIRB). This paper designed MFFB can aggregate spatial and channel features and learn long-range dependencies, while EIRB supplements the model with local contextual information extraction. By modeling the image at multiple levels, including local, global, channel, and spatial levels, PSMFNet fully leverages the rich feature information in the image. The wide-ranging experimental results indicate that our PSMFNet offers a more competitive performance using a smaller number of parameters relative to the state-of-the-art efficient SR approaches.

Acknowledgement: We would like to express our sincere gratitude to all those who have contributed to this research project in various ways.

Funding Statement: This research was funded by Guangdong Science and Technology Program under Grant No. 202206010052, Foshan Province R & D Key Project under Grant No. 2020001006827 and Guangdong Academy of Sciences Integrated Industry Technology Innovation Center Action Special Project under Grant No. 2022GDASZH-2022010108.

Author Contributions: The authors confirm contribution to the paper as follows: Study conception and design: Shuai Cao , Jianan Liang; data collection: Jinglun Huang; analysis and interpretation of results: Zhishu Yang, Yongjun Cao; draft manuscript preparation: Shuai Cao. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, Jianan Liang, upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflict of interest to report regarding the present study.

References

[1] J. Shu, S. Wang, S. Yu, and J. Zhang, "CFSA-Net: Efficient large-scale point cloud semantic segmentation based on cross-fusion self-attention," *Comput. Mater. Contin.*, vol. 77, no. 3, pp. 2677–2697, 2023. doi: 10.32604/cmc.2023.045818.

[2] C. Dong, C. C. G. Loy, K. M. He, and X. O. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. ECCV*, Zurich, Switzerland, 2014, pp. 184–199.

[3] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. CVPR*, Las Vegas, USA, 2016, pp. 1646–1654.

[4] Y. Zhang, K. Li, K. Li, L. Wang, and B. Zhong, "Image super-resolution using very deep residual channel attention networks," in *Proc. ECCV*, Munich, Germany, 2018, pp. 294–310.

[5] J. Y. Liang, J. Z. Cao, and G. L. Sun, "SwinIR: Image restoration using swin transformer," in *Proc. ICCV*, Montreal, Canada, 2021, pp. 1833–1844.

[6] J. Si and S. Kim, "PP-GAN: Style transfer from korean portraits to ID photos using landmark extractor with GAN," *Comput. Mater. Contin.*, vol. 77, no. 3, pp. 3119–3138, 2023. doi: 10.32604/cmc.2023.043797.

[7] Y. Harjoseputro, I. Yuda, and K. P. Danukusumo, "MobileNets: Efficient convolutional neural network for identification of protected birds," *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 10, no. 6, pp. 2290–2296, 2020.
[8] M. Sandler, A. Howard, M. Zhu, and A. Zhmoginov, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. CVPR*, Salt Lake, USA, 2018, pp. 4510–4520.
[9] D. H. Song, C. Xu, X. Jia, and Y. Y. Chen, "Efficient residual dense block search for image super-resolution," in *Proc. AAAI*, New York, USA, 2020, pp. 12007–12014.
[10] X. Chu, B. Zhang, H. Ma, R. Xu, and Q. Li, "Fast, accurate and lightweight super-resolution with neural architecture search," in *Proc. ICPR*, Milan, Italy, 2021, pp. 59–64.
[11] N. Ullah, J. A. Khan, S. Almakdi, M. S. Alshehri, and M. Al Qathrady, "A lightweight deep learning-based model for tomato leaf disease classification," *Comput. Mater. Contin.*, vol. 77, no. 3, pp. 3969–3992, 2023. doi: 10.32604/cmc.2023.041819.
[12] Y. Wang, "Edge-enhanced feature distillation network for efficient super-resolution," in *Proc. CVPRW*, New Orleans, LA, USA, 2022, pp. 776–784.
[13] Z. Hui, X. Wang, and X. Gao, "Fast and accurate single image super-resolution via information distillation network," in *Proc. CVPR*, Salt Lake, USA, 2018, pp. 723–731.
[14] Z. Hui, X. Gao, Y. Yang, and X. Wang, "Lightweight image super-resolution with information multi-distillation network," in *Proc. ACM*, Aizu Wakamatsu, Tokyo, Japan, 2019, pp. 2024–2032.
[15] J. Liu, J. Tang, and G. Wu, "Residual feature distillation network for lightweight image super-resolution," in *Proc. ECCV*, 2020, pp. 2359–2368.
[16] S. Balatti, S. Ambrogio, R. Carboni, and V. Milo, "Physical unbiased generation of random numbers with coupled resistive switching devices," *IEEE Trans. Electron Devices*, vol. 63, no. 5, pp. 2029–2035, 2016. doi: 10.1109/TED.2016.2537792.
[17] J. Liu, W. J. Zhang, Y. T. Tang, and J. Tang, "Residual feature aggregation network for image super-resolution," in *Proc. CVPR*, Seattle, USA, 2020, pp. 2356–2365.
[18] H. Zhao, X. Kong, J. He, Y. Qiao, and C. Dong, "Efficient image super-resolution using pixel attention," in *Proc. ECCV*, 2020, pp. 56–72.
[19] H. Zang, Y. Zhao, C. Niu, and H. Zhang, "Attention network with information distillation for super-resolution," *Entropy*, vol. 24, no. 9, 2022, Art. no. 1226. doi: 10.3390/e24091226.
[20] J. Chen, S. H. Kao, H. He, and W. Zhuo, "Run, don't walk: Chasing higher FLOPS for faster neural networks," arXiv:2303.03667, 2023, doi: 10.48550/arXiv.2303.03667.
[21] N. Ahn, B. Kang, and K. A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proc. ECCV*, Munich, Germany, 2018, pp. 256–272.
[22] F. Y. Kong, M. X. Li, and S. W. Liu, "Residual local feature network for efficient super-resolution," in *Proc. CVPRW*, New Orleans, USA, 2022, pp. 765–775.
[23] J. Gu and C. Dong, "Interpreting super-resolution networks with local attribution maps," in *Proc. CVPR*, 2021, pp. 9195–9204.
[24] B. Lim, S. Son, H. Kim, and S. Nah, "Enhanced deep residual networks for single image super-resolution," in *Proc. CVPRW*, Hawaii, USA, 2017, pp. 1132–1140.
[25] Z. Liu, Y. T. Lin, Y. Cao, and H. Hu, "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. ICCV*, Montreal, Canada, 2021, pp. 9992–10002.
[26] S. W. Zamir, A. Arora, S. Khan, and M. Hayat, "Restormer: Efficient transformer for high-resolution image restoration," in *Proc. CVPR*, New Orleans, USA, 2022, pp. 5718–5729.
[27] J. Zhang, Y. Zhang, J. Gu, and Y. Zhang, "Accurate image restoration with attention retractable transformer," arXiv:2210.01427, 2022, doi: 10.48550/arXiv.2210.01427.
[28] S. Shi, J. Gu, and L. Xie, "Rethinking alignment in video super-resolution transformers," in *Proc. NeurIPS*, New Orleans, LA, USA, 2022, pp. 36081–36093.
[29] X. Chen, X. Wang, and J. Zhou, "Activating more pixels in image super-resolution transformer," arXiv.2205.04437, 2022, doi: 10.48550/arXiv.2205.04437.
[30] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proc. CVPR*, Hawaii, USA, 2017, pp. 2790–2798.

[31] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A persistent memory network for image restoration," in *Proc. ICCV*, Venice, Italy, 2017, pp. 4549–4557.
[32] C. Dong, C. C. Loy, and X. O. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. ECCV*, Amsterdam, Netherlands, 2016, pp. 391–407.
[33] W. Shi, J. Caballero, F. Huszar, and J. Totz, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. CVPR*, Las Vegas, USA, 2016, pp. 1874–1883.
[34] Z. C. Du, D. Liu, J. Liu, and J. Tang, "Fast and memory-efficient network towards efficient image super-resolution," in *Proc. CVPRW*, New Orleans, LA, USA, 2022, pp. 852–861.
[35] Z. Y. Li, Y. Q. Liu, X. Y. Chen, and H. M. Cai, "Blueprint separable residual network for efficient image super-resolution," in *Proc. CVPRW*, New Orleans, LA, USA, 2022, pp. 832–842.
[36] D. Haase and M. Amthor, "Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved mobilenets," in *Proc. CVPR*, Seattle, SEA, USA, 2020, pp. 14588–14597.
[37] L. Sun, J. Dong, J. Tang, and J. Pan, "Spatially-adaptive feature modulation for efficient image super-resolution," arXiv:2302.13800, 2023, doi: 10.48550/arXiv.2302.13800.
[38] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu and C. Dong, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. ECCV*, 2019, pp. 63–79.
[39] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," arXiv:1606.08415, 2016, doi: 10.48550/arXiv.1606.08415.
[40] Y. Li *et al.*, "Efficient and explicit modelling of image hierarchies for image restoration," arXiv:2303.00748, 2023, doi: 10.48550/arXiv.2303.00748.
[41] M. H. Guo, C. Z. Lu, Z. N. Liu, and M. M. Cheng, "Visual attention network," arXiv:2202.09741, 2022, doi: 10.48550/arXiv.2202.09741.
[42] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. CVPRW*, New Orleans, LA, USA, 2017, pp. 1122–1131.
[43] M. Bevilacqua, A. Roumy, C. Guillemot, and A. Morel, "Low-complexity single image super-resolution based on nonnegative neighbor embedding," in *Proc. BMVC*, Surrey, UK, 2012, pp. 1–10.
[44] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. CS*, Avignon, France, 2010, pp. 711–730.
[45] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. ICCV*, Vancouver, Canada, 2001, pp. 416–423.
[46] J. B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. CVPR*, Boston, MA, USA, 2015, pp. 5197–5206.
[47] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, and T. Ogawa, "Sketch-based manga retrieval using manga109 dataset," *Multimed. Tools Appl.*, vol. 76, no. 1, pp. 21811–21838, 2016.
[48] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. on Image Process.*, vol. 13, no. 4, pp. 600–612, 2004. doi: 10.1109/TIP.2003.819861.
[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014, doi: 10.48550/arXiv.1412.6980.
[50] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," arXiv:1608.03983, 2016.
[51] Y. Ioannou, D. Robertson, R. Cipolla, and A. Criminisi, "Deep roots: Improving CNN efficiency with hierarchical filter groups," in *Proc. CVPR*, Hawaii, USA, 2017, pp. 5977–5986.
[52] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, Lille, France, 2015, pp. 448–456.
[53] W. S. Lai, J. B. Huang, N. Ahuja, and M. H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proc. CVPR*, Hawaii, HI, USA, 2017, pp. 5835–5843.

[54] W. Li, K. Zhou, L. Qi, N. Jiang, J. Lu and J. Jia, "LAPAR: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond," in *Proc. NeurIPS*, New Orleans, LA, USA, 2020, pp. 20343–20355.

[55] L. Sun, J. Pan, and J. Tang, "ShuffleMixer: An efficient convnet for image super-resolution," in *Proc. NeurIPS*, New Orleans, LA, USA, 2022, pp. 17314–17326.