Abstract: Transformers have performed better than traditional convolutional neural networks (CNNs) for image super-resolution (SR) reconstruction in recent years. Currently, shifted window multi-head self-attention based on the Swin transformer is a typical method. Specifically, the multi-head self-attention is used to extract local features in each window, and then a shifted window strategy is used to discover information interaction between different windows. However, this information interaction method needs to be more efficient and include some global feature information, which limits the model’s performance to a certain extent. Furthermore, optimizing the utilization of shallow features, which exhibit significant energy reserves and invaluable low-frequency information, is critical for advancing the efficacy of super-resolution techniques. In order to solve the above issues, we propose the feature-enhanced fused attention (FE-FAIR) method for image super-resolution. Specifically, we design the multi-scale feature extraction module (MSFE) as a shallow feature extraction layer to extract rich low-frequency information from different scales. In addition, we propose the fused attention block (FAB), which introduces channel attention in the form of residual connection based on shifted window self-attention, effectively achieving the fusion of global and local features. Simultaneously, we also discuss other methods to enhance the performance of the FE-FAIR method, such as optimizing the loss function, increasing the window size, and using pre-training strategies. Compared with state-of-the-art SR methods, our proposed method demonstrates better performance. For instance, FE-FAIR outperforms SwinIR by over 0.9 dB when evaluated on the Urban100 (×4) dataset.

Keywords: transformer; super-resolution; feature-enhanced fused attention (FE-FAIR); multi-scale feature extraction module (MSFE); fused attention block (FAB)

1. Introduction

Image super-resolution reconstruction (SR) [1] refers to the reconstruction of low-resolution (LR) images into information-rich high-resolution (HR) images. This technique stands as a pivotal technology within computer vision, contributing significantly to various computational vision tasks like image denoising and target detection while simultaneously economizing on transmission and storage expenses.

Early image SR methods based on deep learning [2–4] mainly relied on simple convolutional neural network (CNN) structures for optimizing image reconstruction. In order to extract more image features, deeper network layers combined with more complex structures such as residual connections [5,6] and dense connections [7] were adopted. The attention mechanism enables the network to prioritize important information while disregarding irrelevant details. Various studies have demonstrated that using channel attention [8], layer attention [9], and high-order channel attention [10] can help the SR model recover more detailed features and improve the quality of the image. Nevertheless, limited by local convolution operations, the CNN method based on the attention mechanism exhibits a diminished ability to perceive long-range pixel relationships, thereby restricting enhancements in image quality.
Transformers [11] have attracted widespread attention in computer vision due to their excellent long-range dependency modeling capabilities. Many transformer models [12–15] have been proposed for low-level computational vision tasks. Subsequently, Liang et al. [16] combined the advantages of CNNs and transformers and proposed an image SR method based on the swin transformer, showing excellent performance across tasks such as image SR, image denoising, and image compression. This model, using a pre-training strategy and hybrid attention mechanism [17–19], effectively enhances the image reconstruction quality. The swin transformer, using shifted window technology for feature extraction, currently stands as a compelling structure for transformer-based image SR methods. However, although the swin transformer has excellent modeling capabilities for local features, its information interaction efficiency between different windows should be better and more effective for capturing global features. In addition, the classic super-resolution model consists of three parts: a shallow feature extraction layer, a deep feature extraction layer, and an upsampling layer. The low-frequency features obtained from the shallow feature extraction layer directly contribute to the network’s upsampling process. Additionally, the rich low-frequency features can provide more practical information for the subsequent deep feature extraction module. Hence, shallow features play a crucial role in SR tasks, and enhancing the effectiveness of the shallow extraction layer is crucial for improving image quality.

To address these challenges, we propose feature-enhanced fusion attention for image super-resolution (FE-FAIR). FE-FAIR mainly includes a shallow feature extraction layer, a deep feature extraction layer, and an image reconstruction layer. The shallow feature extraction layer uses the more effective multi-scale feature extraction (MSFE) module, which employs convolutional layers of varying depths combined with atrous convolutional layers to extract shallow features from multiple scales effectively, surpassing traditional single-layer $3 \times 3$ convolutional layers. MSFE also introduces richer low-frequency information for subsequent deep feature extraction layers. Inspired by the efficacy of channel attention in integrating global information and enhancing image reconstruction [8,20], we introduce the fused attention block (FAB). The FAB combines a multi-head self-attention mechanism with a channel attention mechanism, using residual connections to integrate global information into each self-attention layer. This enables the FAB to fuse information across local and global scales, proving highly very effective. Furthermore, we explore various methods to improve the performance of SR methods. These include enlarging the window size of the swin transformer for enhanced feature extraction, utilizing Smooth$L_1$ Loss for smoother model training, employing effective data augmentation techniques such as rotation and RGB channel shuffling during training to enhance robustness, implementing a pre-training strategy on ImageNet [21], and fine-tuning the model using the DF2K [22] dataset to further optimize performance. The comparison results between our proposed FE-FAIR and the state-of-the-art SR methods on the Manga109 and Urban100 benchmarks are shown in Figure 1. It demonstrates that FE-FAIR achieves state-of-the-art performance across all image super-resolution tasks and scales. In comparison to SwinIR, it exhibits a significant improvement of 0.84 dB to 0.96 dB on the Urban100 benchmark. In summary, our contributions can be summarized as follows:

- We propose a better transformer-based super-resolution reconstruction method called FE-FAIR. It combines a shallow feature enhancement module with a fused attention mechanism to achieve better model performance.
- We propose a more effective shallow feature extraction layer known as the multi-scale feature extraction (MSFE) module, aimed at enhancing the model’s capability to capture low-frequency information. By adjusting the depth and channel number of the convolutional layers of different branches and adding dilated convolutions, the receptive field is expanded and finer-grained shallow features are extracted.
- We analyze the characteristics of window self-attention and propose the fused attention block FAB. Based on moving window multi-head self-attention, we add channel
attention through the residual structure to achieve information fusion of global and local features.

- We explore several additional strategies aimed at enhancing the model’s performance. These include employing data augmentation techniques, implementing a smoother SmoothL1 Loss function, enlarging the window size of the swin-transformer, and adopting pre-training strategies.

Figure 1. Comparative performance evaluation between FE-FAIR and the state-of-the-art methods NLSN, SwinIR, and EDT. Quantitative analysis using PSNR (Y-channel only) on Urban100 and Manga109 Datasets at scale factors ×2, ×3, and ×4. “†” indicates we use a pre-training strategy on ImageNet.

The subsequent sections of this paper are organized as follows. Section 2 describes the research background for this research methodology. Section 3 describes the overall architecture of the FE-FAIR method. Section 4 introduces the evolution process of FE-FAIR and the experimental results on benchmark tests of performance in different tasks. Section 5 summarizes the contributions of this work.

2. Related Work

In this section, we briefly describe part of the evolution of the image SR method. First, we give an overview of attention mechanism methods and then analyze transformer-based methods.

2.1. Deep Network Methods for Image SR

Since SRCNN [2] first introduced convolutional neural networks into image SR, people have successively proposed a variety of deep network methods [3, 5–7] to improve the performance of the model, thereby improving the quality of model reconstruction images. For example, sub-pixel convolution techniques [4], deeper networks and residual blocks [5, 6], and more complex dense blocks [7] are used to improve the expressive ability of the model. In order to improve the visual quality after image reconstruction, Refs. [23, 24] used generative adversarial networks to generate more realistic images. Limited by CNN size effects and feature extraction mechanisms, enhancing the performance of deep convolutional neural networks (CNNs) in image SR tasks becomes increasingly challenging. To solve this problem, some studies integrate attention mechanisms into various layers of CNNs [9, 25], allowing for a more detailed understanding and analysis of images at different levels. In addition, researchers also explored techniques to introduce spatial and channel attention into these mechanisms [8, 10], aiming to improve model efficiency further. These pioneering
efforts provide valuable insights and motivate us to advocate deeper integration of attention mechanisms to effectively capture relevant information between different locations, thereby improving model performance.

2.2. Transformer-Based Methods for Image SR

In recent years, the great success of transformers in natural language processing (NLP) tasks has attracted attention in computer vision. Pure transformer-based methods perform excellently by handling long-distance dependencies well [13,15,26–30]. Some work has shown that combining convolutions and transformers can achieve more advanced results [31–33]. SwinIR [16] combines the advantages of convolutions and transformers and proposes a network for tasks such as image SR, which performs well in various image restoration tasks. EDT [17] explores the impact of pre-training mechanisms on transformer methods to enhance the performance of SR networks further. However, these works underestimate shallow feature importance and fail to combine global features with local features effectively. Therefore, our model focuses on more effective shallow feature extraction and feature fusion during the deep feature extraction process, effectively improving the model’s ability to depict image details.

3. Methodology

Inspired by the above work, we propose the FE-FAIR method in this section—the specific network structure shown in Figure 2.

Figure 2. The overall structure of FE-FAIR. Specifically, it mainly includes a multi-scale shallow feature extraction (MSFE) module, fused attention block (FAB), and residual connection FAB (RFAB).

3.1. Network Architecture

FE-FAIR mainly consists of three modules: the shallow feature, deep feature, and graphic reconstruction module. Specifically, the shallow feature module is mainly composed of multi-scale feature extraction layers, using different numbers of convolutional layers and atrous convolution combinations to extract shallow features from different scales. The deep feature extraction module mainly comprises the shifted window self-attention and channel attention mechanisms and introduces the residual structure. The image reconstruction module mainly consists of convolutional and pixel-shuffle layers.
Expressly, for the input low-resolution image $I_{LR} \in \omega^{C_0 \times H \times W}$, we initially utilize a multi-scale feature extraction (MSFE) module to extract shallow-layer features $F_0 \in \omega^{C \times H \times W}$ in different dimensions as follows:

$$F_0 = \phi_{MSFE}(I_{LR}) \quad (1)$$

where $H$ and $W$ represent the height and width of the input image, $C_0$ and $C$ represent the number of channels output by the input image and shallow feature extraction layer, respectively, and $\phi_{MSFE}$ represents the MSFE module. The intelligence of the MSFE module is to obtain rich low-frequency feature information from different perspectives and levels. Subsequently, the deep feature extraction module $\phi_{DF}(F_0)$ is utilized to obtain deep features $F_{DF} \in \omega^{C \times H \times W}$:

$$F_{DF} = \phi_{DF}(F_0) \quad (2)$$

where $\phi_{DF}$ consists of N residual fused attention block (RFAB) and a $3 \times 3$ convolutional layer $\phi_{Conv}$. This structure can extract deep features layer by layer, as follows:

$$F_i = \phi_{RFAB_i}(F_{i-1}), \quad i = 1, 2, \cdots, N \quad (3)$$

$$F_{DF} = \phi_{Conv}(F_i) \quad (4)$$

where $\phi_{RFAB_i}$ represents the $i$th RFAB block. Subsequently, a $3 \times 3$ convolutional layer is used after the deep feature extraction layer to aggregate features. Finally, the high-quality image reconstruction module $\phi_{HQ}$ is applied to reconstruct the high-quality image $I_{SR}$, as follows:

$$I_{SR} = \phi_{HQ}(F_0 + F_{DF}) \quad (5)$$

In order to enhance the stability of the model while effectively retaining the low-frequency and high-frequency information of the image, we use residual connections to transfer the low-frequency information to the image reconstruction block. In the image reconstruction block, we utilize the pixel-shuffle to upsample the reconstructed image.

### 3.2. Multi-Scale Feature Extraction (MSFE) Module

The shallow feature extraction block mainly maps input features from low latitudes to higher dimensions, usually containing low-frequency information. Convolutional layers perform well in the early processing of visual tasks, thereby facilitating better-optimized results [32]. Simultaneously, we concentrate on the correlation between the target pixel and surrounding pixels, noting that this correlation diminishes as the pixel distance increases. The atrous spatial pyramid pooling (ASPP) [34] uses multiple parallel atrous convolutional layers with different sampling rates to extract features from different scales. Inspired by ASPP, we design the MSFE module based on the concept of multi-scale convolution, with the primary process outlined as follows:

$$F_{in} = H_{Conv1}(I_{LR}) \quad (6)$$

$$F_i = H_{branch_i}(F_{in}), \quad i = 1, 2, 3, 4$$

$$F_{concat} = Concat(F_0, F_1, F_2, F_3)$$

$$F_{out} = H_{Conv2}(F_{concat} + F_{in})$$

where $F_{in}$ indicates the mapped high-dimensional features, and $branch_i$ represents dilated convolution modules of different scales. $Conv1$ implements the mapping of input features from low to high dimensions, and $Conv2$ indicates the aggregation of features from different branches.

As illustrated in Figure 3, for the input feature $I_{LR} \in \omega^{C_0 \times H \times W}$, a convolutional layer is employed to perform feature mapping from low to high dimensions. Subsequently, distinct branches are utilized to acquire features at various scales. Each branch ($branch_i$) comprises
varying numbers of convolutional layers and dilated convolutional layers. The dilation value of the dilated convolutional layer corresponds to the number of convolutional layers, where a higher number of convolutional layers entails a larger dilation value [35]. This design effectively expands the receptive field of the convolutional layer while enabling each branch to focus on the interrelation between the central feature and surrounding features across different scales. Ultimately, the information from different branches is concatenated, and a residual structure is employed to incorporate previous level information, thereby enhancing model stability.

![Multi-scale shallow feature extraction (MSFE) module.](image)

### 3.3. Fused Attention Block (FAB)

The window-based multi-head self-attention mechanism can extract high-frequency information and local features within the feature map. By adding global features, the model can effectively integrate the information from the entire feature map. Previous studies have demonstrated that convolutional layers can enhance the performance of transformers [36]. Channel attention, as proposed by Hu et al. [20], focuses on the importance and correlation of different feature channels. It assigns different weight characteristics to each channel, thereby enhancing model’s capability for global feature extraction. Consequently, the fusion of multi-head self-attention and channel attention serves to amalgamate features effectively. As illustrated in Figure 4, subsequent to passing through the LayerNorm layer, the channel attention block (CAB) and W-MSA operate as parallel structures to calculate the feature map across different dimensions, yielding a residual summation as output. To balance channel attention and W-MSA, we multiply the original input and output features of CAB by the adaptive weights of the sum, respectively. For an input feature $X$, the entire FAB processing process is as follows:

$$
X_{\text{norm}} = \text{LN}(X)
$$

$$
X_{\text{MSA}} = W - \text{MSA}(X_{\text{norm}}) + X
$$

$$
X_T = \text{MLP}(\text{LN}(X_{\text{MSA}})) + X_{\text{MSA}}
$$

$$
X_{\text{out}} = X_T + \alpha \text{CAB}(X_{\text{norm}}) + \beta X
$$

where $X_{\text{norm}}$ represents the layernorm (LN) layer, $X_{\text{MSA}}$ represents the intermediate result of multi-head self-attention calculation, $X_T$ represents the output feature of W-MSA branch, and $X_{\text{out}}$ represents the output feature of the FAB. MLP stands for multi-layer perceptron layer, and CAB stands for channel attention block.
The specific calculation process of the window self-attention mechanism is as follows: given an input feature of size $H \times W \times C$, divide the input feature into non-overlapping windows of size $M^2$, then the total number of windows is $\frac{HW}{M^2}$. The input features can be reshaped into $\frac{HW}{M^2} \times M^2 \times C$. Subsequently, self-attention is calculated within each window independently. For each window feature $\in \mathbb{R}^{M^2 \times C}$, the query, key, and value matrix are calculated as

$$Q = XM_Q \quad K = XM_K \quad V = XM_V$$

(8)

where $M_Q$, $M_K$, and $M_V$ represent the mapping matrices of query, key, and value, respectively. Then, the self-attention of the window can be expressed as

$$Attention(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d}} + B)V$$

(9)

where $d$ represents the dimension of query/key, and $B$ indicates the relative position encoding. In addition, in order to promote information exchange between adjacent windows, the shift window method is also utilized, with the shift size being set at half the window size.

The CAB module mainly consists of convolutional and standard channel attention (CA) layers. The specific structure is shown in Figure 2. Due to the large number of channels, a high computational cost will be incurred when the standard channel attention layer is combined with the transformer. To address this, channels are compressed to $C_\gamma$ while maintaining similar performance using a convolutional layer. The entire CAB calculation process is as follows:

$$X_{out} = CA(\text{Conv}(X_{in}))$$

(10)

where $X_{in}$, $X_{out}$, and $\text{Conv}$ represent input features, output features, and convolution layer.

As shown in Figure 2, each residual group fused attention block (RFAB) contains $N$ fused attention block (FAB) modules and a $3 \times 3$ convolutional layer. Precisely, for the $i$th RFAB, it can be calculated as

$$F_{i,0} = F_{i-1}, \quad i = 1, 2, \cdots, N$$

$$F_{i,j} = H_{FAB,i}(F_{i,j-1}), \quad j = 1, 2, \cdots, M$$

$$F_i = H_{\text{Conv}}(F_{i,M}) + F_{i,0}$$

(11)

where $F_{i,0}$ and $F_i$ represent the input features and output features of the $i$th RFAB, $H_{FAB,i}$ represents the $i$th FAB calculation block in the $i$th RFAB, and $H_{\text{Conv}}$ represents the convolution layer of the $i$th RFAB module. This design offers two notable advantages. Firstly, the incorporation of convolutional layers facilitates a more stable aggregation of information. Secondly, the utilization of residual connections not only stabilizes the model’s training but also enhances inter-layer relationships by incorporating information from different modules.
3.4. Loss Function

For previous image SR methods, $L_1$ pixel loss is generally used as the loss function to optimize the model. Under normal circumstances, $L_1$ Loss can obtain better model performance. However, the $L_1$ Loss suffers from nondifferentiability at specific points, which can impede loss optimization. Secondly, in the later stage of model training, the difference between the sum and the sum is slight, but its derivative is still a constant. In this way, the loss value will fluctuate around the stable value when the learning rate remains unchanged, making it challenging to achieve higher accuracy. Therefore, we use Smooth$L_1$ Loss [37] as the loss optimization function of the method. Smooth$L_1$ Loss is expressed explicitly as

$$L = \begin{cases} \frac{0.5(I_{SR} - I_{HQ})^2}{\theta} & \text{if } |I_{SR} - I_{HQ}| < \theta \\ |I_{SR} - I_{HQ}| - 0.5\theta & \text{otherwise} \end{cases}$$

(12)

where $\theta$ is set to 1 by default in our method. As indicated by the formula above, Smooth$L_1$ Loss employs the form of $L_1$ Loss when $|I_{SR} - I_{HQ}| \geq \theta$ and uses the form of $L_2$ loss when $|I_{SR} - I_{HQ}| < \theta$. This approach effectively addresses issues such as gradient explosion arising from significant losses and accuracy concerns when losses are small.

4. Experiments

In this section, the FE-FAIR method proposed in this paper is compared with other state-of-the-art methods, such as EDSR [6], RCAN [8], SAN [10], IGNN [38], HAN [9], NLSN [25], SwinIR [16], EDT [17], CARN [39], IMDD-A [41], LatticeNet [42], BM3D [43], WNNM [44], DnCNN [45], IRCNN [46], FFDNet [47], NLRN [48], FOCNet [49], MWCNN [50], DRUNet [51], DSNet [52], RPCNN [53], BRDNet [54], and IPT [15].

Simultaneously, the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [55] are utilized for evaluation. The calculation process of PSNR is demonstrated as follows:

$$\text{PSNR} = 10 \times \log_{10} \frac{\text{Max}^2}{\text{MSE}} = 20 \times \log_{10} \frac{\text{Max}}{\sqrt{\text{MSE}}}$$

where the default value of $\text{Max}$ is 255. The calculation expression of mean square error (MSE) is presented as follows:

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

where $H$ and $W$ are the number of pixels in the length and width of the image, respectively. The value of PSNR depends on MSE, so the smaller the MSE, the greater the PSNR value, which means the smaller the difference between the reconstructed image and the actual image. The SSIM is also used to measure the similarity between the reconstructed and authentic images from brightness, contrast, and structure. SSIM can be expressed as

$$\text{SSIM} = \frac{(2\mu_{I_{HR}} \times \mu_{I_{SR}} + C_1)(2\sigma_{I_{HR}} \times \sigma_{I_{SR}} + C_2)}{\left(\mu_{I_{HR}}^2 + \mu_{I_{SR}}^2 + C_1\right)\left(\sigma_{I_{HR}}^2 + \sigma_{I_{SR}}^2 + C_2\right)}$$

where $\mu_{I_{HR}}$, $\mu_{I_{SR}}$, $\sigma_{I_{HR}}$, $\sigma_{I_{SR}}$, $\zeta_{I_{HR}}$, and $\zeta_{I_{SR}}$ represent the mean, standard deviation, and covariance of $I_{HR}$ and $I_{SR}$, respectively. $C_1$ and $C_2$ are constants. The closer the value of SSIM is to 1, the higher the similarity between the two images.

All experiments are conducted using PyTorch version 2.0 on 4 NVIDIA Tesla V100 GPUs with CUDA version 12.2.
4.1. Experimental Setup

In classical image SR, the DF2K dataset (comprising DIV2K [22] with 900 images and Flickr2K [22] with 2650 images) containing 3550 images is utilized as the original training set. Bicubic downsampling with scaling factors of ×2, ×3, and ×4 are performed using MATLAB to generate low-resolution images. The test set includes popular super-resolution benchmark datasets such as Set5 [56], Set14 [57], BSD100 [58], Urban100 [59], and Manga109 [60]. Regarding the architecture of FE-FAIR, the parameters are configured as follows: the number of RFABs, FABs, channels, attention heads, and window size are set to 6, 6, 180, 6, and 16, respectively. In lightweight image SR tasks, these parameters are set to 4, 6, 60, 6, and 16, respectively. The channel compression parameter γ in the MSFE module is defaulted to 3 for classical tasks and 6 for lightweight tasks. The channel compression parameter γ in CAB is set to 5, and α and δ in FAB are treated as adaptive parameters.

For image denoising, the training set comprises DIV2K (900 images), Flickr2K (2650 images), WED [61] (4744 images), and BSD200 [58] (200 images). The test set includes Set12 [45], BSD68 [58], CBSD68 [58], Kodak24 [62], McMaster [63], and Urban100 datasets. The parameter configurations remain consistent with classic SR.

For classical SR, we set the batch size to 32 and the total training iterations to 500 k. The initial learning rate is $2 \times 10^{-4}$, and it is halved at iterations [250 k, 400 k, 450 k, 475 k, 500 k], respectively. For lightweight SR, the batch size is 64, and the total number of training iterations is also 500 k. For the image denoising task, the batch size is 8, and the total number of training iterations is 1500 k. The initial learning rate is $2 \times 10^{-4}$, halved at iterations [600 k, 1000 k, 1300 k, 1450 k], respectively. For the pre-training model, 1.2 million images from ImageNet [21] are used for 1000 k iterations. The initial learning rate is also $2 \times 10^{-4}$, halved at iterations [300 k, 500 k, 750 k, 900 k, 1000 k]. Subsequently, the DF2K dataset (DIV2K with 900 images + Flickr2K with 2650 images) is employed for fine-tuning with 250 k iterations. The learning rate is initialized to $2 \times 10^{-5}$ and halved at iterations [130 k, 200 k, 230 k, 245 k, 250 k].

Simultaneously, to determine the optimal number of iterations in our proposed method, experiments are conducted on the DF2K dataset, and results are shown in Figure 5. We can observe that after 500 k iterations, the method converges in both tasks. Specifically, compared with classic tasks, the lightweight SR methods can enter the convergence state faster due to the smaller number of parameters and minor computational cost, and the indicators are more stable during the training process. The classic task method has better performance. Therefore, we set 500 k iterations in this work as the total training times.

![Figure 5. PSNR (Y channel) and SSIM trends in training on classic tasks (FE-FAIR) and lightweight tasks (FE-FAIR-T).](image-url)
4.2. Ablation Experiment

In this part of the work, the impact of a series of methods proposed in this paper on image SR are separately verified.

4.2.1. Effectiveness of MSFE

Rich shallow features can help the model retain sufficient low-frequency information [64] while providing more effective feature information for the deep feature extraction module. The MSFE module combines dilated convolutions with different numbers of convolutional layers to achieve more detailed feature capture. The effectiveness of the proposed MSFE module is demonstrated through experimental setups in our work. Using the traditional single-layer convolutional layer as the baseline, this part of the work tests the method gain of three convolutional layers, ASPP, and MSFE modules as shallow feature extraction layers. The results are quantified on Set14, Urban100, and Manga109. As shown in Table 1, when using MSFE as the shallow feature extraction layer, the network achieves a performance gain of 0.06 to 0.11 dB compared to other methods, demonstrating a significant improvement over alternative methods. All results show that MSFE can effectively improve the performance of SR methods.

Table 1. The effects of different shallow feature extraction modules. Bold text and numbers indicate the method we used and the best results among all methods.

<table>
<thead>
<tr>
<th>Module</th>
<th>Scale</th>
<th>Set14 [57]</th>
<th>Urban100 [59]</th>
<th>Manga109 [60]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv</td>
<td>2</td>
<td>34.46/0.9250</td>
<td>33.81/0.9427</td>
<td>39.92/0.9797</td>
</tr>
<tr>
<td>3 × Conv</td>
<td>2</td>
<td>34.47/0.9252</td>
<td>33.82/0.9428</td>
<td>39.93/0.9796</td>
</tr>
<tr>
<td>ASPP [34]</td>
<td>2</td>
<td>34.48/0.9253</td>
<td>33.84/0.9431</td>
<td>39.94/0.9798</td>
</tr>
<tr>
<td>MSFE</td>
<td>2</td>
<td>34.52/0.9257</td>
<td>33.92/0.9435</td>
<td>39.98/0.9804</td>
</tr>
<tr>
<td>Conv</td>
<td>4</td>
<td>29.09/0.7950</td>
<td>27.45/0.8254</td>
<td>32.03/0.9260</td>
</tr>
<tr>
<td>3 × Conv</td>
<td>4</td>
<td>29.10/0.7949</td>
<td>27.47/0.8256</td>
<td>32.05/0.9259</td>
</tr>
<tr>
<td>ASPP [34]</td>
<td>4</td>
<td>29.12/0.7952</td>
<td>27.48/0.8259</td>
<td>32.07/0.9262</td>
</tr>
<tr>
<td>MSFE</td>
<td>4</td>
<td>29.15/0.7958</td>
<td>27.53/0.8271</td>
<td>32.11/0.9265</td>
</tr>
</tbody>
</table>

4.2.2. Effects of the FAB

The FAB module combines self-attention and channel attention mechanisms through residual connections, enabling the integration of global features into local features. We conducted experiments to demonstrate the effectiveness of the FAB module. Table 2 presents the quantitative performance on the Set14, Urban100, and Manga109 test sets for ×4 super-resolution. Compared to the baseline performance of the STL module in SwinIR, the FAB module brings a performance gain of 0.05 to 0.09 dB. Adaptive weights $\alpha$ and $\beta$ are set to avoid conflicts between channel attention and window self-attention. We further investigate the impact of these variable weights on model performance. Experiments show that without adding parameters, the inclusion of $\alpha$ and $\beta$ results in a performance gain of 0.03 dB based on the FAB module. This indicates that the adaptive parameters $\alpha$ and $\beta$ reduce negative impacts between different attention mechanisms, facilitating their fusion and enabling improved model performance.

Table 2. The effects of the FAB module on performance. Bold text and numbers indicate the method we used and the best results among all methods.

<table>
<thead>
<tr>
<th>Module</th>
<th>Set14 [57]</th>
<th>Urban100 [59]</th>
<th>Manga109 [60]</th>
</tr>
</thead>
<tbody>
<tr>
<td>STL</td>
<td>29.15/0.7958</td>
<td>27.53/0.8271</td>
<td>32.11/0.9265</td>
</tr>
<tr>
<td>FAB</td>
<td>29.17/0.7959</td>
<td>27.62/0.8277</td>
<td>32.17/0.9270</td>
</tr>
<tr>
<td>FAB + $\alpha + \beta$</td>
<td>29.19/0.7960</td>
<td>27.65/0.8282</td>
<td>32.20/0.9273</td>
</tr>
</tbody>
</table>
4.2.3. Effects of Smooth $L_1$

Loss Smooth $L_1$ Loss offers smoother convergence and better performance compared to traditional $L_1$ Loss. To demonstrate the superiority of Smooth $L_1$ Loss as a loss function, experiments were conducted. Table 3 presents the quantitative results for the super-resolution task with a scaling factor of $\times 4$ on the Urban100 dataset. Smooth $L_1$ Loss achieves a performance gain of 0.03 dB compared to $L_1$ Loss. These results indicate that Smooth $L_1$ Loss, when used as a loss function for image super-resolution tasks, enhances the model’s performance.

Table 3. The effects of using different loss functions on performance. Bold text and numbers indicate the method we used and the best results among all methods.

<table>
<thead>
<tr>
<th>Module</th>
<th>$L_1$ Loss</th>
<th>Smooth $L_1$ Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR/SSIM</td>
<td>27.65/0.8282</td>
<td>27.68/0.8289</td>
</tr>
</tbody>
</table>

4.2.4. Effects of Window Size

EDT explored the impact of window size on the performance of the window self-attention mechanism. It has been proved that increasing the window size is a direct method of improving the performance of the SR network. However, previous studies only explored windows up to $12 \times 12$ in size. Therefore, we also examined the impact of larger window sizes on network performance. Table 4 shows the quantitative test results when the amplification factor is 4 on the Set14, Urban100, and Manga109 test sets. We can find from the results that when the window size is 16, the model’s performance can be effectively improved, especially the PSNR improvement on Urban100, which reaches 0.24 dB. Therefore, in FE-FAIR, we directly set the window size to 16.

Table 4. The effects of window size. Bold text and numbers indicate the method we used and the best results among all methods.

<table>
<thead>
<tr>
<th>Window Size</th>
<th>Set14 [57]</th>
<th>Urban100 [59]</th>
<th>Manga109 [60]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR/SSIM</td>
<td>PSNR/SSIM</td>
<td>PSNR/SSIM</td>
</tr>
<tr>
<td>(8, 8)</td>
<td>29.20/0.7962</td>
<td>27.68/0.8289</td>
<td>32.23/0.9268</td>
</tr>
<tr>
<td>(12, 12)</td>
<td>29.22/0.7966</td>
<td>27.87/0.8348</td>
<td>32.28/0.9279</td>
</tr>
<tr>
<td>(16, 16)</td>
<td>29.21/0.7966</td>
<td>27.96/0.8377</td>
<td>32.32/0.9287</td>
</tr>
</tbody>
</table>

4.3. Comparison Result

4.3.1. Results on Classical Image Super-Resolution

Quantitative comparison. Table 5 shows the quantitative comparison results between FE-FAIR and other state-of-the-art methods: EDSR [6], RCAN [8], SAN [10], IGNN [38], HAN [9], NLSN [25], SwinIR [16], and EDT [17]. We can see that FE-FAIR exhibits the best performance across all test sets and magnifications. Specifically, FE-FAIR achieves a performance gain of 0.33–0.55 dB on Urban100 and 0.27–0.34 dB on Manga109. Thus, FE-FAIR demonstrates superior performance in image super-resolution. Additionally, we present quantitative comparison results between the pre-training strategy FE-FAIR and state-of-the-art models IPT† [15] and EDT† [17]. The pre-trained model exhibits a substantial performance improvement, notably surpassing the baseline (SwinIR) by 0.8 dB on Urban100, thereby affirming the effectiveness of the pre-training strategy.

Visual Comparison. We selected several pictures (img011, img048, image074, image092) from the benchmark to show the super-resolution reconstruction results of the model. It can be found from Figure 6 that our results have greatly improved texture details and authenticity.
Figure 6. Visual comparison with state-of-the-art methods (average PSNR/SSIM) for scale ×4. The compared parts are marked with red markers in the image.

Table 5. Quantitative comparison with state-of-the-art methods (average PSNR/SSIM) for classical image SR on benchmark datasets. The best and second-best performances are **bolded** and *underlined*, respectively. “†” indicates we use a pre-training strategy on ImageNet.
Table 5. Cont.

<table>
<thead>
<tr>
<th>Method</th>
<th>Scale</th>
<th>Training</th>
<th>PSNR/SSIM</th>
<th>PSNR/SSIM</th>
<th>PSNR/SSIM</th>
<th>PSNR/SSIM</th>
<th>PSNR/SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPT + [15]</td>
<td>×2</td>
<td>ImageNet</td>
<td>38.37/-</td>
<td>34.43/-</td>
<td>32.48/-</td>
<td>34.76/-</td>
<td>34.82/-</td>
</tr>
<tr>
<td>EDT + [17]</td>
<td>×2</td>
<td>DF2K</td>
<td>38.63/0.9632</td>
<td>34.80/0.9723</td>
<td>32.62/0.9052</td>
<td>34.27/0.9456</td>
<td>40.37/0.9811</td>
</tr>
<tr>
<td>FE-FAIR †</td>
<td>×2</td>
<td>DF2K</td>
<td>35.02/0.932</td>
<td>31.02/0.858</td>
<td>29.50/0.8162</td>
<td>30.22/0.8898</td>
<td>35.42/0.9547</td>
</tr>
<tr>
<td>EDSR [6]</td>
<td>×3</td>
<td>DIV2K</td>
<td>34.65/0.9280</td>
<td>30.52/0.8462</td>
<td>29.25/0.8093</td>
<td>28.80/0.8653</td>
<td>34.17/0.9476</td>
</tr>
<tr>
<td>RCAN [8]</td>
<td>×3</td>
<td>DIV2K</td>
<td>34.74/0.9299</td>
<td>30.65/0.8482</td>
<td>29.32/0.8111</td>
<td>29.09/0.8702</td>
<td>34.44/0.9494</td>
</tr>
<tr>
<td>SAN [10]</td>
<td>×3</td>
<td>DIV2K</td>
<td>34.75/0.9300</td>
<td>30.59/0.8476</td>
<td>29.33/0.8112</td>
<td>29.83/0.8671</td>
<td>34.30/0.9494</td>
</tr>
<tr>
<td>IGNN [38]</td>
<td>×3</td>
<td>DIV2K</td>
<td>34.72/0.9298</td>
<td>30.66/0.8484</td>
<td>29.31/0.8105</td>
<td>29.03/0.8696</td>
<td>34.39/0.9496</td>
</tr>
<tr>
<td>HAN [9]</td>
<td>×3</td>
<td>DIV2K</td>
<td>34.75/0.9299</td>
<td>30.67/0.8483</td>
<td>29.32/0.8110</td>
<td>29.10/0.8705</td>
<td>34.48/0.9500</td>
</tr>
<tr>
<td>NLSN [25]</td>
<td>×3</td>
<td>DIV2K</td>
<td>34.85/0.9306</td>
<td>30.70/0.8485</td>
<td>29.34/0.8117</td>
<td>29.25/0.8760</td>
<td>34.57/0.9508</td>
</tr>
<tr>
<td>SwinIR [16]</td>
<td>×3</td>
<td>DF2K</td>
<td>34.97/0.9318</td>
<td>30.93/0.8534</td>
<td>29.46/0.8145</td>
<td>29.75/0.8826</td>
<td>35.12/0.9537</td>
</tr>
<tr>
<td>EDT [17]</td>
<td>×3</td>
<td>DF2K</td>
<td>34.97/0.9316</td>
<td>30.89/0.8527</td>
<td>29.44/0.8142</td>
<td>29.72/0.8814</td>
<td>35.13/0.9534</td>
</tr>
<tr>
<td>FE-FAIR †</td>
<td>×3</td>
<td>DF2K</td>
<td>35.02/0.9326</td>
<td>31.02/0.8581</td>
<td>29.50/0.8162</td>
<td>30.22/0.8898</td>
<td>35.42/0.9547</td>
</tr>
</tbody>
</table>

4.3.2. Results on Lightweight Image Super-Resolution

Quantitative comparison. Table 6 shows the quantitative performance comparison results between lightweight FE-FAIR and state-of-the-art lightweight methods: CARN [39], IMDN [40], LAPAR-A [41], LatticeNet [42], SwinIR [16], and EDT [17]. The total number of parameters in our method (evaluated on 1280 × 720 images) is also provided. As shown in Table 6, the quantification performance of FE-FAIR is significantly better than other methods, especially in SSIM. It proves that our method is effective in lightweight image super-resolution tasks.

Table 6. Quantitative comparison (average PSNR/SSIM) with state-of-the-art methods for lightweight image SR on benchmark datasets. The best and second-best performances are bolded and underlined, respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>Scale</th>
<th># Params</th>
<th>PSNR/SSIM</th>
<th>PSNR/SSIM</th>
<th>PSNR/SSIM</th>
<th>PSNR/SSIM</th>
<th>PSNR/SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARN [39]</td>
<td>×2</td>
<td>1392 k</td>
<td>37.76/0.9590</td>
<td>35.52/0.9166</td>
<td>32.09/0.8978</td>
<td>31.92/0.9256</td>
<td>38.36/0.9765</td>
</tr>
<tr>
<td>IMDN [40]</td>
<td>×2</td>
<td>548 k</td>
<td>38.00/0.9605</td>
<td>33.63/0.9177</td>
<td>32.19/0.8996</td>
<td>32.17/0.9283</td>
<td>38.88/0.9774</td>
</tr>
<tr>
<td>LAPAR-A [41]</td>
<td>×2</td>
<td>548 k</td>
<td>38.01/0.9605</td>
<td>33.62/0.9183</td>
<td>32.19/0.8999</td>
<td>32.10/0.9283</td>
<td>38.67/0.9772</td>
</tr>
<tr>
<td>LatticeNet [42]</td>
<td>×2</td>
<td>756 k</td>
<td>38.15/0.9610</td>
<td>33.78/0.9193</td>
<td>32.25/0.9005</td>
<td>32.43/0.9302</td>
<td>38.86/0.9772</td>
</tr>
<tr>
<td>SwinIR [16]</td>
<td>×2</td>
<td>878 k</td>
<td>38.14/0.9611</td>
<td>33.86/0.9206</td>
<td>32.31/0.9012</td>
<td>32.76/0.9340</td>
<td>39.12/0.9783</td>
</tr>
<tr>
<td>EDT [17]</td>
<td>×2</td>
<td>917 k</td>
<td>38.23/0.9615</td>
<td>33.99/0.9209</td>
<td>32.37/0.9021</td>
<td>32.98/0.9362</td>
<td>39.45/0.9789</td>
</tr>
<tr>
<td>FE-FAIR †</td>
<td>×2</td>
<td>2291 k</td>
<td>38.30/0.9621</td>
<td>33.10/0.9214</td>
<td>32.41/0.9027</td>
<td>33.37/0.9417</td>
<td>39.56/0.9808</td>
</tr>
</tbody>
</table>
We further explore the performance of our method in image denoising. We show the comparison results of FE-FAIR on grayscale and color image denoising tasks with other state-of-the-art methods: BM3D, SWT, BM3D, WNNM, ICT and SwinIR. Tables 7 and 8 provide quantitative comparison results at noise levels of 15, 25, and 50. Specifically, our method outperforms the state-of-the-art method SwinIR by 12 dB on the Urban100 benchmark dataset.

### Table 8: Quantitative comparison (average PSNR) with state-of-the-art methods for grayscale image denoising on benchmark datasets. The best and second-best performances are bolded and underlined.

<table>
<thead>
<tr>
<th>Method</th>
<th>-scale</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM3D</td>
<td>4</td>
<td>23.10</td>
</tr>
<tr>
<td>SWT</td>
<td>4</td>
<td>22.90</td>
</tr>
<tr>
<td>BM3D</td>
<td>4</td>
<td>22.80</td>
</tr>
<tr>
<td>WNNM</td>
<td>4</td>
<td>22.70</td>
</tr>
<tr>
<td>ICT</td>
<td>4</td>
<td>22.60</td>
</tr>
<tr>
<td>SwinIR</td>
<td>4</td>
<td>22.50</td>
</tr>
<tr>
<td>BM3D</td>
<td>3</td>
<td>22.40</td>
</tr>
<tr>
<td>SWT</td>
<td>3</td>
<td>22.30</td>
</tr>
<tr>
<td>BM3D</td>
<td>3</td>
<td>22.20</td>
</tr>
<tr>
<td>WNNM</td>
<td>3</td>
<td>22.10</td>
</tr>
<tr>
<td>ICT</td>
<td>3</td>
<td>22.00</td>
</tr>
<tr>
<td>SwinIR</td>
<td>3</td>
<td>21.90</td>
</tr>
</tbody>
</table>

### Table 4: Quantitative comparison (average PSNR) with state-of-the-art methods for grayscale image denoising on benchmark datasets. The best and second-best performances are bolded and underlined.

<table>
<thead>
<tr>
<th>Method</th>
<th>-scale</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM3D</td>
<td>4</td>
<td>23.10</td>
</tr>
<tr>
<td>SWT</td>
<td>4</td>
<td>22.90</td>
</tr>
<tr>
<td>BM3D</td>
<td>4</td>
<td>22.80</td>
</tr>
<tr>
<td>WNNM</td>
<td>4</td>
<td>22.70</td>
</tr>
<tr>
<td>ICT</td>
<td>4</td>
<td>22.60</td>
</tr>
<tr>
<td>SwinIR</td>
<td>4</td>
<td>22.50</td>
</tr>
<tr>
<td>BM3D</td>
<td>3</td>
<td>22.40</td>
</tr>
<tr>
<td>SWT</td>
<td>3</td>
<td>22.30</td>
</tr>
<tr>
<td>BM3D</td>
<td>3</td>
<td>22.20</td>
</tr>
<tr>
<td>WNNM</td>
<td>3</td>
<td>22.10</td>
</tr>
<tr>
<td>ICT</td>
<td>3</td>
<td>22.00</td>
</tr>
<tr>
<td>SwinIR</td>
<td>3</td>
<td>21.90</td>
</tr>
</tbody>
</table>
5. Conclusions

In this paper, we re-explore the importance of shallow features and propose a multi-scale shallow feature extraction module MSFE to obtain more prosperous and influential low-frequency features. Concurrently, we integrate window self-attention and channel attention in the form of residual connection, proposing the fused attention module FAB. The FAB effectively achieves an effective combination of local feature information and global feature information. In addition, we also incorporate other techniques to improve the model’s performance, such as data augmentation and increasing the window size. Combining these methods, we propose an image super-resolution reconstruction method FE-FAIR. Comparative evaluations on benchmark datasets demonstrate FE-FAIR’s superior performance compared to other state-of-the-art image super-resolution methods. Additionally, our method exhibits better performance in image denoising tasks.

In the future, we will continue exploring further interactions between shallow and deep features to achieve more fine-grained shallow feature capture. Additionally, investigating the impact of various attention fusion methods on image super-resolution remains a promising avenue of research. Due to the enormous potential of the transformer architecture, we aim to further explore its applications across various tasks, including the field of image SR.

Author Contributions: A.G. and K.S. completed the methodology, experimental data, and manuscript writing of this work. J.L. completed the revision and checking of the manuscript, and A.G. and J.L. provided financial support and supervised this work. All authors have read and approved the published version of the manuscript.

Funding: This research was supported in part by the National Natural Science Foundation of China under Grant 62204044, in part by the State Key Laboratory of Integrated Chips and Systems, and in part by Shanghai Science and Technology Innovation Action under Grants 22xtcx00700 and 22511101002.

Data Availability Statement: The data that support the findings of this study are available online. Data download address: https://github.com/sk0625-hhh/FE-FAIR (accessed on 8 February 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.