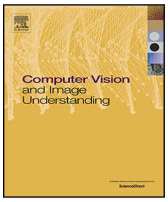
Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu



An efficient feature reuse distillation network for lightweight image super-resolution

Chunying Liu [a,b,c], Guangwei Gao [a,b,c,*], Fei Wu [a], Zhenhua Guo [d], Yi Yu [e]

[a] *Institute of Advanced Technology, Nanjing University of Posts and Telecommunications, Nanjing, China*
[b] *Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai, China*
[c] *Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, China*
[d] *Tianyijiaotong Technology Ltd., Suzhou, China*
[e] *Graduate School of Advanced Science and Engineering, Hiroshima University, Hiroshima, Japan*

ARTICLE INFO

Keywords:
Single-image super-resolution
Lightweight network
Feature reuse
Transformer

ABSTRACT

In recent research, single-image super-resolution (SISR) using deep Convolutional Neural Networks (CNN) has seen significant advancements. While previous methods excelled at learning complex mappings between low-resolution (LR) and high-resolution (HR) images, they often required substantial computational and memory resources. We propose the Efficient Feature Reuse Distillation Network (EFRDN) to alleviate these challenges. EFRDN primarily comprises Asymmetric Convolutional Distillation Modules (ACDM), incorporating the Multiple Self-Calibrating Convolution (MSCC) units for spatial and channel feature extraction. It includes an Asymmetric Convolution Residual Block (ACRB) to enhance the skeleton information of the square convolution kernel and a Feature Fusion Lattice Block (FFLB) to convert low-order input signals into higher-order representations. Introducing a Transformer module for global features, we enhance feature reuse and gradient flow, improving model performance and efficiency. Extensive experimental results demonstrate that EFRDN outperforms existing methods in performance while conserving computing and memory resources.

1. Introduction

single-image super-resolution (SISR) technology enhances the quality of captured photos by increasing their resolution and sharpness, resulting in clearer and more detailed images (Li et al., 2024b). Moreover, SISR is utilized in medical imaging equipment (Georgescu et al., 2023), surveillance systems (Jiang et al., 2022), and satellite imaging (Xiao et al., 2023), allowing for enhanced image clarity and precision in various applications (Li et al., 2024a; Gao et al., 2023). Overall, SISR significantly impacts computer vision technologies by improving image quality, optimizing visual experiences, and facilitating diverse applications across various devices and industries.

The development of single-image super-resolution (SISR) tasks has progressed significantly with the emergence of deep neural networks and residual learning. Dong et al. (2015) introduced the first image super-resolution neural network, SRCNN, surpassing traditional methods based on sparse representation and optimization. Subsequently, various neural networks have been proposed for image super-resolution, showing superior performance (Lim et al., 2017). Residual learning in deep networks (Zhang et al., 2018) enhances gradient problem-solving capabilities. This has led to the design of larger, deeper network architectures for improved performance, such as EDSR (Lim et al., 2017) and RCAN (Zhang et al., 2018).

However, the trend towards larger and deeper CNN-based models for improved performance has led to challenges in deploying these models on mobile devices due to the large number of parameters. To address the mentioned challenges, lightweight image super-resolution network models are proposed to achieve efficiency and reduce parameters and computations. This includes constructing shallow networks with single paths (Lai et al., 2017), recursive operations (Kim et al., 2016b), information distillation mechanisms (Hui et al., 2019), and neural structure search (NAS) (Hui et al., 2019). While traditional CNNs can only extract local context information, Transformer models have shown significant progress in computer vision. Transformer-based methods, like SwinIR (Liang et al., 2021), utilized global information extraction and sliding window mechanisms to address edge uncorrelation in SISR. Integrating CNN and Transformer, as seen in Gao et al. (2022b), combined local and global information for enhanced performance not achievable by pure CNN or Transformer models. Kim et al. (2024a) introduced a Transformer model designed to improve image resolution while ensuring computational efficiency. Zhang et al.

[*] Corresponding author at: Institute of Advanced Technology, Nanjing University of Posts and Telecommunications, Nanjing, China.
E-mail address: csggao@gmail.com (G. Gao).

https://doi.org/10.1016/j.cviu.2024.104178
Received 30 June 2024; Received in revised form 27 August 2024; Accepted 10 September 2024
Available online 27 September 2024

(2024a) developed a real-time Transformer framework optimized for practical applications, focusing on both speed and effectiveness. Liu et al. (2024b) advanced the field further with an attention-based Transformer that strikes a balance between efficiency and performance, offering more accessible and faster image enhancement solutions.

Existing methods that rely solely on CNN networks often struggle with context modeling, making it difficult to achieve high-quality image reconstruction. While some recent approaches have integrated Transformers to address this, they tend to introduce excessive parameters and computational complexity. To overcome these challenges, we propose the Efficient Feature Reuse Distillation Network (EFRDN), which strategically combines CNN and Transformer in a series configuration, effectively blending local and global information. To mitigate the increase in parameters and computational demands, we employ intermediate feature knowledge distillation, group convolution, and other techniques to ensure model efficiency. Our EFRDN achieves a good balance between image quality and model efficiency, using 100k to 200k fewer parameters and reducing computation by 10G to 20G compared to existing methods, while also delivering a 0.1 to 0.2 dB improvement in PSNR performance. Additionally, to enhance information flow across layers and leverage network potential through feature reuse, we employ skip connections to exploit information from multiple layers. By enhancing feature reuse and gradient flow with dense connections, we enhance model performance and generalization. The Asymmetric Convolution Distillation Block (ACDB) module within our CNN part includes the Feature Fusion Lattice Block (FFLB), Asymmetric Convolutional Residual Block (ACRB), and Multiple Self-Calibrated Convolution (MSCC). The FFLB module adjusts structure and connects branches adaptively, while the ACRB module reinforces the central skeleton of the square convolution kernel and extracts features in different directions, thus enhancing the ability of the model to express complex patterns, significantly boosting performance. The MSCC module, utilizing channel/spatial double-attention self-calibrated convolutions (SCCA/SCSA), extracts valuable spatial and channel features to improve network efficiency. In this article, our main contributions can be summarized as follows:

- We introduce a Self-calibrating Convolutional with Channel/ Spatial Attention (SCCA/SCSA) unit to enhance the discriminative ability of CNNs by adaptively attending to relevant spatial and channel information through self-calibrating operations. Additionally, we design Group Convolution Residual Blocks (GCRB) utilizing depth-separable convolution to improve model efficiency.
- We propose the Asymmetric Convolutional Distillation Module (ACDM) within the CNN, consisting of Feature Fusion Lattice Block (FFLB), Asymmetric Convolutional Residual Block (ACRB), and Multiple Self-Calibrated Convolution (MSCC). Through residual characteristic distillation, ACDM achieves superior results with fewer parameters.
- We introduce an Efficient Transformer to enhance the global features of the EFRDN model, employing a concatenated CNN-Transformer structure. Furthermore, skip connections are utilized to ensure feature reuse and address gradient-related challenges.

2. Related work

2.1. Deep SR model

SRCNN (Dong et al., 2015) pioneered CNN-based super-resolution, utilizing a 3-layer architecture for nonlinear mapping, outperforming traditional methods. VDSR (Kim et al., 2016a) increased network depth to 20 layers for a broader receptive field, enhancing performance. However, deeper models may face convergence challenges, impacting inference efficiency. Residual learning, exemplified by EDSR (Lim et al., 2017), overcomes this by stacking more layers to learn residuals. Models with larger receptive fields improve reconstruction quality but entail more parameters and computational load. Recursive learning, as seen in DRCN (Kim et al., 2016b), accelerates convergence and reduces model size using shared weights and skip connections. Attention mechanisms, like RCAN (Zhang et al., 2018), enhance feature extraction but may not suit lightweight applications due to their complexity. Zhang et al. (2024b) introduced a contrastive learning framework aimed at capturing high-frequency details and enhancing perceptual quality, demonstrating the effectiveness of contrastive methods in super-resolution (SR). Kim et al. (2024b) explored a transformer-based model featuring an adaptive attention mechanism that excels at managing long-range dependencies and refining high-resolution details. Zhang et al. (2023) proposed a hybrid model that integrates dynamic convolution with attention mechanisms, resulting in improved efficiency and performance in SR tasks.

2.2. Lightweight SR models

To enable SISR on mobile devices, researchers have focused on lightweight SISR models (Li et al., 2021). Current methods can be categorized into efficient model structure design (Gao et al., 2022a) pruning or quantization techniques and knowledge distillation (Lee et al., 2020). Weight sharing and channel grouping reduce the model size and facilitate structural design in many models. Strategies like channel splitting and layered distillation in IDN (Hui et al., 2018) and IMDN (Hui et al., 2019) enhance feature extraction. FALSR (Chu et al., 2021) applied neural architecture search (NAS) for compact SISR networks, setting a new direction for structural design. Knowledge transfer-based model compression (Lee et al., 2020) enhances student model performance by distilling knowledge from pre-trained teacher models. Pruning methods like Jiang et al. (2021) reduce model size with minimal accuracy loss. Despite extensive exploration, unresolved thematic issues in lightweight SISR models warrant further research.

2.3. Visual transformer

Transformers excel in advanced visual tasks (Li et al., 2022). To enhance transformer efficiency and effectiveness in complex tasks, various transformer-based methods have emerged. Swin Transformer (Liu et al., 2021) employed a local window and shift operation to control focus range and enhance window interaction. DaViT (Ding et al., 2022) introduced dual self-attention for global context capture with linear complexity. Leveraging the success of Transformers, researchers have explored their utility in low-level vision tasks. SwinIR (Liang et al., 2021) implemented Swin Transformers with spatial window self-attention and shift operations. Restorer utilized self-attention along the channel dimension and integrated the UNet architecture. These Transformer-based methods outperform CNN approaches, highlighting the significance of both spatial and channel information for performance. Liu et al. (2024a) proposed the Top-K Token Selective Transformer, which enhances high-resolution image reconstruction by focusing on the most informative tokens, thus increasing both efficiency and accuracy.

3. Proposed method

This section outlines the structure of our proposed Efficient Feature Reuse Distillation Network (EFRDN), detailing the sequential and dense connections between the CNN component and the Transformer backbone. Next, the Asymmetric Convolution Distillation Module (ACDM) within the CNN block is described, comprising the Feature Fusion Lattice Block (FFLB), Asymmetric Convolution Residual Block (ACRB), and Multiple Self-Calibrated Convolution (MSCC). Finally, the Efficient Transformer's specifics are presented.

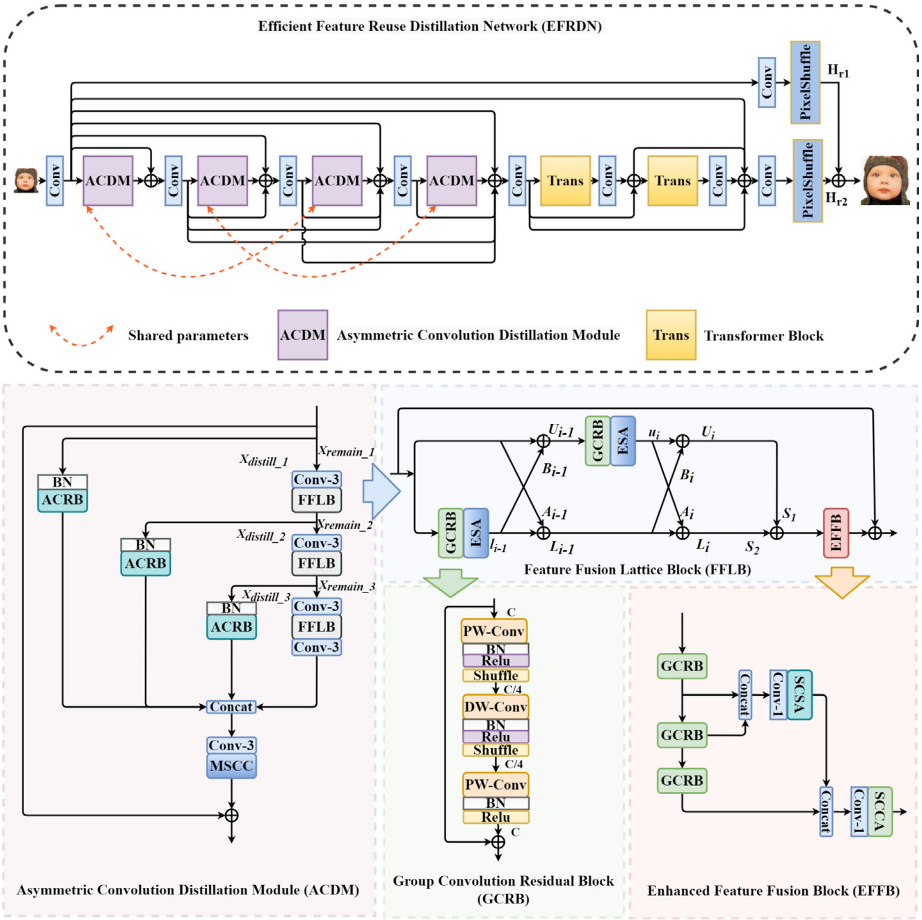

Efficient Feature Reuse Distillation Network (EFRDN)
Conv
ACDM
Trans
PixelShuffle
Hr1
Hr2
Shared parameters
ACDM Asymmetric Convolution Distillation Module
Trans Transformer Block
BN
ACRB
Xdistill_1
Xremain_1
Conv-3
FFLB
Xdistill_2
Xremain_2
Xdistill_3
Xremain_3
Concat
MSCC
Asymmetric Convolution Distillation Module (ACDM)
Ui-1
GCRB
ESA
ui
Ui
Bi-1
Bi
Ai-1
Ai
li-1
Li-1
Li
S1
S2
EFFB
Feature Fusion Lattice Block (FFLB)
C
PW-Conv
Relu
Shuffle
C/4
DW-Conv
Group Convolution Residual Block (GCRB)
Concat
Conv-1
SCCA
Enhanced Feature Fusion Block (EFFB)

Fig. 1. The architecture of the proposed Efficient Feature Reuse Distillation Network (EFRDN). In the FFLB, A_i and B_i denote the Combination Coefficient (CC) learning, which is elaborated in Fig. 4.

3.1. Network framework

As depicted in Fig. 1, EFRDN comprises three main sections: shallow feature extraction, deep feature extraction, and image reconstruction. The deep feature extraction component involves a sequence of CNN and Transformer mechanisms. I_{LR} and I_{SR} denote the input low-resolution image and super-resolution image, respectively. Initially, shallow features are extracted using a 3 × 3 convolution, depicted as

$$F_S = G_S\left(I_{LR}\right), \tag{1}$$

where F_S represents the shallow feature, and G_S represents the function for shallow feature extraction. These features are then forwarded to the CNN and Transformer modules sequentially for deep feature extraction. In the CNN segment, four ACDM units are utilized, with shared parameters between the first and third modules and between the second and fourth modules. In deep learning algorithms, the parameter-sharing strategy enables multiple features to use the same parameters, effectively reducing the overall parameter count. This reduction increases computational efficiency. EFRDN leverages parameter sharing to lower model complexity, accelerate training, and enhance the model's generalization capabilities. To enhance information flow across layers, a beneficial connection pattern is introduced between each ACDM. Direct connections from any layer to all subsequent layers are established, ensuring each layer receives feature maps from all preceding layers. The output of each layer is denoted as F_{Ai}, and the process can be described as:

$$F_{A\_i} = \begin{cases} G_{conv\_1}\left(G_{ACDM\_1}\left(F_S\right) + F_S\right) (i=1) \\ G_{convi\_}\left(G_{ACDM\_i}\left(F_{A\_i-1}\right) + \sum_{n=1}^{i-1} F_{A\_i-1} + F_S\right) \end{cases}, \tag{2}$$

where G_{ACDM_i-1} denotes the function of the ith ACDM in the ith Layer, and F_{A_i} signifies the output of the ith Layer. G_{conv_i} refers to the ith convolution operation post-concatenation. The output of the CNN segment can be represented as

$$C_{out} = \sum_{i=1}^{4} F_{A\_i} + F_S, \tag{3}$$

where C_{out} represents the output of the CNN part. Following this, the features derived from the CNN part are fed into the Transformer part. Similar to the CNN part, a dense connection operation is integrated into the Transformer part. In this case, two efficient trans modules are used in succession, with each layer incorporating all features from the preceding layer in the output. Finally, the shallow feature F_S is added to mitigate gradient vanishing due to excessive network depth. This process is articulated as follows:

$$T_{out\_1} = G_{conv\_5}\left(G_{trans\_1}\left(C_{out}\right)\right), \tag{4}$$

$$T_{add\_1} = T_{out\_1} + C_{out}, \tag{5}$$

$$T_{out\_2} = G_{conv\_6}\left(G_{trans\_2}\left(T_{add\_1}\right)\right), \tag{6}$$

$$T_{add\_2} = T_{out\_1} + C_{out} + T_{out\_2} + F_S, \tag{7}$$

where G_{conv_i} represents the ith convolution operation, and G_{trans_i} denotes the ith Transformer operation. T_{out_i} represents the output after the ith trans module operation, while T_{add_i} signifies the output following the ith addition operation.

Following shallow and deep feature extraction, the features are directed to the reconstruction module. To address gradient vanishing due to excessive network depth, we integrate the extraction of shallow and deep features concurrently in the image super-resolution reconstruction module. The operational process is as follows:

$$I_{SR} = H_{r\_1}\left(T_{add\_2}\right) + H_{r\_2}\left(F_S\right), \tag{8}$$

$$= H_{EFRDN}\left(I_{LR}\right). \tag{9}$$

The function $H_{r\_i}(\cdot)$ represents the image reconstruction operation. This process involves a 3 × 3 convolution and a pixel shuffle operation. H_{EFRDN} denotes the EFRDN model introduced in this paper.

3.2. Asymmetric convolution distillation module

As depicted in Fig. 1, the Asymmetric Convolution Distillation Module (ACDM) consists of three components: the Feature Fusion Lattice Block (FFLB), the Asymmetric Convolution Residual Block (ACRB), and Multiple self-calibration Convolutions (MSCC). Among these, two fundamental modules are the Group Convolution Residual Block (GCRB) and the Self-calibrated Convolution with Channel/Spatial Attention (SCCA/SCSA) unit.

Feature Fusion Lattice Block (FFLB): Building upon the benefits of lattice blocks (Luo et al., 2023), as illustrated in Fig. 1, we leverage this design to integrate the Group Convolution Residual Block (GCRB), Enhanced Spatial Attention (ESA), and Enhanced Feature Fusion Block (EFFB). We have made significant improvements to the original lattice blocks. In the original butterfly structure, two 3 × 3 convolutions were used to extract features from the upper and lower branches. However, in our proposed FFLB module, we introduced the GCRB module in combination with the ESA module to extract features more effectively. The GCRB module utilizes grouped convolution, which allows for more efficient information extraction without increasing the number of parameters or computational load. Additionally, while the original butterfly structure used only a 1 × 1 convolution for simple channel matching after merging the outputs from the upper and lower branches, our FFLB module incorporates an EFFB information fusion module. This further extracts and fuses valuable information, reducing information loss during network transmission and enhancing the effectiveness and richness of high-level features. Meanwhile, through the butterfly structure, we can output basic units in various combinations. The Feature Fusion Lattice Block (FFLB) contains two branches, with two butterfly structures combining features from these branches using the Combination Coefficient (CC) structure (Luo et al., 2023), detailed in Fig. 4. Unlike traditional channel attention that solely utilizes average pooling, CC incorporates a standard differential branch to enhance the visual impact of the image.

In each branch, we introduce the Group Convolution Residual Block (GCRB), as depicted in Fig. 1. The GCRB module comprises three components. Initially, features are extracted through point convolution, followed by normalization, activation, and a fourfold reduction in channel count before progressing to the second part. The second part involves a 3 × 3 depth-separable convolution, accompanied by normalization and activation operations. Subsequently, the feature transitions to the third part, which mirrors the first part but with a fourfold increase in channel count to restore the initial channels. The module concludes by adding the original input to the output, extracting comprehensive features without escalating parameter count through operations like point convolution, depth-separable convolution, and channel adjustments.

Following GCRB, we implemented the Enhanced Spatial Attention (ESA) unit to extract significant spatial features. More precisely, for the input feature E_{in} directed to the upper and lower branches, we define F_{GCRB_1} as the initial GCRB operation, and F_{ESA_1} as the first ESA unit operation. The operation in the lower branches can be expressed as:

$$l_{i-1} = F_{ESA\_1}\left(F_{GCRB\_1}\left(E_{in}\right)\right), \tag{10}$$

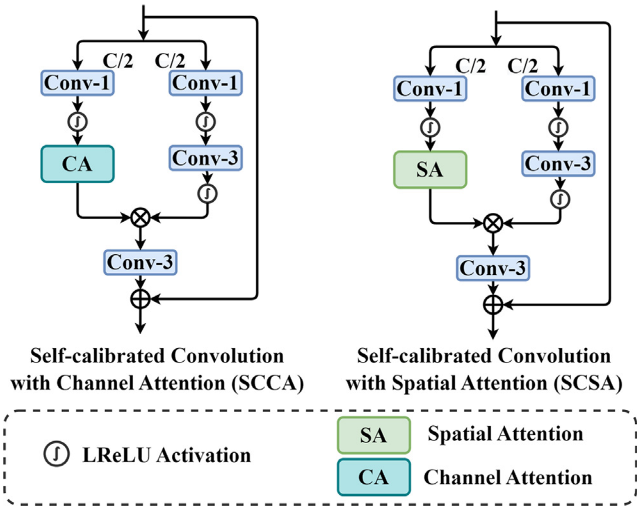


Fig. 2. The architecture of Self-calibrated Convolution with Channel Attention (SCCA) and Self-calibrated Convolution with Spatial Attention (SCSA) units.

where l_{i-1} indicates the output of the initial lower branch. Subsequently, the upper and lower branches are linked through the first butterfly mechanism, described as

$$U_{i-1} = X_{in} + B_{i-1}\left(l_{i-1}\right), \tag{11}$$

$$L_{i-1} = l_{i-1} + A_{i-1}\left(E_{in}\right), \tag{12}$$

where $B_{i-1}(\cdot)$ and $A_{i-1}(\cdot)$ denote the combination coefficient values for upper and lower branch features. U_{i-1} and L_{i-1} represent the output of the initial combination of the upper and lower branches. Subsequently, U_{i-1} and L_{i-1} are fed into a second butterfly structure mirroring the first. In this structure, F_{GCRB_2} represents the second GCRB operation, and $F_{ESA2\_}$ signifies the second ESA unit. The input to the branch in the second composite structure is described as

$$u_i = F_{ESA\_2}\left(F_{GCRB\_2}\left(U_{i-1}\right)\right), \tag{13}$$

where u_i represents the input on the branch of the second combination. Simultaneously, the operation of the second butterfly combination can be articulated as

$$U_i = u_i + B_i\left(L_{i-1}\right), \tag{14}$$

$$L_i = L_{i-1} + A_i\left(u_i\right), \tag{15}$$

where U_i and L_i denote the combined output of the second upper and lower branches respectively. Bi (·) and Ai (·) denote the combination coefficient values for upper and lower branch features. The results from the upper and lower branches are subsequently weighted, added, and then output through the EFFB module.

In the Enhanced Feature Fusion Block (EFFB) as shown in Fig. 1, we initially utilize the GCRB module to extract features. Subsequently, we merge the extracted outputs from the first and second GCRB modules through a 3 × 3 convolution module, followed by the SCSA module we designed. This resulting fusion output is then combined with the output from the third GCRB module, undergoes a 3 × 3 convolution, and is directed into the SCCA module we designed.

In the SCSA/SCCA module, as depicted in Fig. 2, self-calibrating convolution is employed. Initially, the input is divided into left and right branches based on channels. The left branch undergoes self-calibration by incorporating spatial or channel attention to enhance effective feature extraction, while the right branch maintains the original spatial context through a basic convolution operation. Upon merging the two intermediate outputs, a 3 × 3 convolution is conducted,

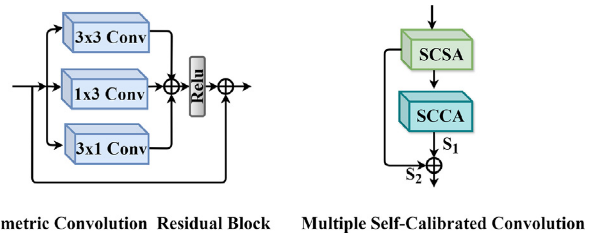

3x3 Conv
1x3 Conv
3x1 Conv
Relu
SCSA
SCCA
S1
S2
Asymmetric Convolution Residual Block
Multiple Self-Calibrated Convolution

Fig. 3. The architecture of Asymmetric Convolution Residual Block (ACRB) and Multiple Self-Calibrated Convolution (MSCC) units.

followed by the addition of the initial input. Ultimately, the output, along with the original input E_{in}, serves as the overall EFFB output. This operation within the EFFB module is represented as shown in the following formula:

$$F_{out} = E_{in} + F_{EFFB}\left(S_1\left(U_i\right) + S_2\left(L_i\right)\right), \tag{16}$$

where F_{out} represents the output of the FFLB module, F_{EFFB} denotes the function of the EFFB unit, while S_1, S_2 denote the weight coefficients of each branch.

Asymmetric Convolution Distillation Module (ACDM): Building upon IMDN (Hui et al., 2019) and RFDN (Liu et al., 2020), we have enhanced and introduced to create a unique characteristic distillation module. As illustrated in Fig. 1, following channel splits, the module branches into two paths. One path includes an Asymmetric Convolution Residual Block (ACRB), generating the distillation feature. In the ACRB module, the original 3 × 3 convolution is replaced by three parallel convolution: 1 × 3, 3 × 3, 3 × 1, as depicted in Fig. 3. This operation trims parameter count while enhancing the square convolution kernel's skeleton information. Additionally, the results from the three channels' convolutions are summed, passed through an activation function, and added to the original input to enrich image feature details. The other path refines the coarse features in progress, labeled the refinement layer. In this branch, a 3 × 3 convolution is initially processed through our proposed FFLB module for advanced feature extraction, followed by additional distillation of the features. The complete distillation process unfolds as

$$X_{remain\_1}, X_{distill\_1} = F_{split}\left(X_{in}\right), \tag{17}$$

$$X_{distill\_1} = F_{ACRB\_1}\left(X_{distill\_1}\right), \tag{18}$$

$$X_{remain\_1} = F_{FFLB\_1}\left(G_{conv\_1}\left(X_{remain\_1}\right)\right). \tag{19}$$

$$X_{remain\_2}, X_{distill\_2} = F_{split}\left(X_{remain\_1}\right), \tag{20}$$

$$X_{distill\_2} = F_{ACRB\_2}\left(X_{distill\_2}\right), \tag{21}$$

$$X_{remain\_2} = F_{FFLB\_2}\left(G_{conv\_2}\left(X_{remain\_2}\right)\right). \tag{22}$$

$$X_{remain\_3}, X_{distill\_3} = F_{split}\left(X_{remain\_2}\right), \tag{23}$$

$$X_{distill\_3} = F_{ACRB\_3}\left(X_{distill\_3}\right), \tag{24}$$

$$X_{remain\_3} = G_{conv\_4}\left(F_{FFLB\_3}\left(G_{conv\_3}\left(X_{remain\_3}\right)\right)\right). \tag{25}$$

Here, X_{remain_i} (i = 1, 2, 3) represents the remaining features, $X_{distill\_i}$ (i = 1, 2, 3) denotes the distilled features, F_{FFLB_i} (i = 1, 2, 3) represent the function of the FFLB unit, F_{split} expresses the function of the channel split, and F_{ACRB_i} (i = 1, 2, 3) represent the function of the ACRB unit.

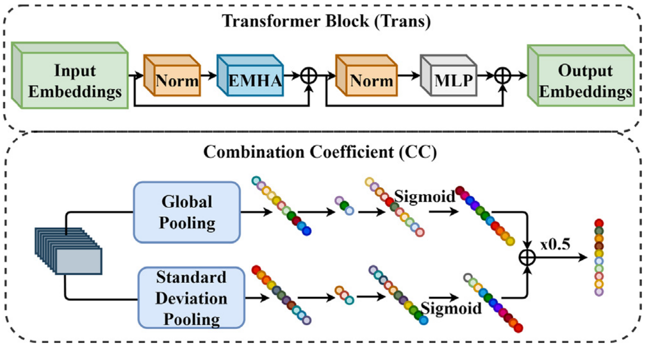


Fig. 4. The process of Transformer Block (Trans) and Combination Coefficient (CC) learning.

Following the feature distillation process, the final FFLB-operated feature in the refinement branch is extracted through a 3 × 3 convolution and combined with the previously distilled feature. After fusion, a reduction in channel count is achieved through a 3 × 3 convolution. We propose a new structure, Multiple Self-Calibrated Convolutions (MSCC), as depicted in Fig. 3. In the MSCC module, we initially conducted the Self-calibrated Convolution with Spatial Attention (SCSA) operation to extract valuable spatial information, followed by channel-level information extraction through the Self-calibrated Convolution with Channel Attention (SCCA) module. Ultimately, the output is weighted via SCCA and added as the final output. These operations are detailed in the following formula:

$$X_{concat} = Concat\left(X_{distill\_1}, X_{distill\_2}, X_{distill\_3}, X_{remain\_3}\right), \tag{26}$$

$$X_{out} = F_{MSCC}\left(X_{concat}\right) + X_{in}, \tag{27}$$

where "Concat" signifies the fusion of features from spatial and channel dimensions. F_{MSCC} represents the operation of the MSCC module, X_{concat} denotes the output of the Concat operation, and X_{out} signifies the output of the ACDM module.

3.3. Efficient transformer

A pure CNN network still faces challenges in high-quality image reproduction. CNNs typically have a fixed-size receptive field, limiting their ability to capture global image information. For high-quality image reproduction, it is essential to capture long-range dependencies and global context, but traditional CNNs often struggle with this. Additionally, while pooling layers and downsampling operations in CNNs reduce computational load and help extract high-level features, they can also cause the loss of important details, negatively impacting image quality. As a result, CNNs may fall short in capturing intricate details and structures, which can affect their ability to effectively reconstruct fine textures in an image.

While CNNs are skilled at extracting local features and details, Transformers excel in capturing global context and long-term dependencies. Combining these two approaches allows for the integration of local details with global structure, improving the overall comprehensiveness of feature extraction. Transformers, with their self-attention mechanisms, enhance the understanding of relationships between distant pixels, leading to better image quality and detailed reconstruction. Meanwhile, CNNs offer computational efficiency through convolutions, while Transformers provide superior handling of long-term dependencies. Together, they achieve a balance between computational efficiency and modeling capability. This paper introduces an efficient Transformer model based on CNN. While ensuring model lightweightness, we integrate global image information extraction to combine local and global details, enhancing model performance. Illustrated in Fig. 4, we draw inspiration from ESRT (Lu et al., 2022), utilizing Efficient Multi-head Attention (EMHA) and multi-layer Perceptrons (MLPs) to optimize GPU memory usage during training. With the input denoted as T_{in} and the output as T_{out}, the Transformer process can be summarized as follows:

$$T_{Atten} = T_{in} + F_{EMHA}\left(F_{Norm}\left(T_{in}\right)\right), \tag{28}$$

$$T_{out} = T_{Atten} + F_{MLP}\left(F_{Norm}\left(T_{Atten}\right)\right), \tag{29}$$

where F_{Norm} represents the layer normalization operation. T_{Atten} stands for the output of the attention module. Additionally, F_{EMHA} and F_{MLP} represent the functions of the EMHA and MLP modules, respectively.

Building on Vaswani et al. (2017), each head of the EMHA is required to conduct scaled dot product attention, followed by the aggregation of all outputs for a linear transformation to generate the final outputs. The scaled dot product attention operation can be represented as

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{30}$$

where Q, K, and V represent the query matrix, key matrix, and value matrix, while $Softmax$ denotes the softmax operation function.

3.4. Loss function

To ensure a fair comparison of experimental results, we also utilize the L_1 loss function to optimize our experimental model. For the training set $\{I^i_{LR}, I^i_{HR}\}_{i=1}^{N}$ with N images, the objective of the EFRDN model is to minimize the values of the following loss function formula:

$$\mathcal{L}(\Theta) = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathcal{H}_{EFRDN}(I^i_{LR}, \Theta) - I^i_{HR}\right\|_1, \tag{31}$$

where $\mathcal{H}_{EFRDN}$ denotes the parameter set of EFRDN, and $\|.\|_1$ is the L_1 norm. The stochastic gradient descent algorithm is applied to optimize this loss function.

4. Experiments

4.1. Datasets and evaluation metrics

In this experiment, we utilize the DIV2K dataset, a collection of high-definition images depicting various natural scenes, for training the model. The DIV2K dataset comprises 900 high-resolution images, with the initial 800 images used for training and the remaining 100 for validation. Low-resolution images are generated using a double-triple downscaling method. To assess the efficacy of EFRDN, we employ the Set5 (Bevilacqua et al., 2012), Set14 (Zeyde et al., 2012), Urban100 (Huang et al., 2015), BSDS100 (Martin et al., 2001), and Manga109 (Matsui et al., 2017) benchmark datasets for testing. Evaluation metrics such as PSNR and SSIM (Wang et al., 2004) are employed.

4.2. Implementation details

This study enhanced the training dataset by applying random rotations and horizontal flips at various angles to increase data diversity. During model training, we set the initial learning rate to 2×10^{-4} and decay it to 6.25×10^{-6} following the cosine annealing strategy. The model was optimized using the Adam optimizer and trained on the NVIDIA RTX 2080Ti GPU within the PyTorch framework. Throughout the training, 48 × 48 patches were randomly extracted from the training set for training input. Data augmentation techniques such as random rotation and horizontal flipping were applied to enhance the data. In the final model, both the CNN and Transformer modules had an input channel size of 32 channels.

4.3. Comparison with state-of-the-arts

The quantitative comparison results for ×2, ×3, and ×4 image super-resolution are presented in Table 1. The best and the second-best results are highlighted and underlined, respectively. EFRDN is compared against IDN (Hui et al., 2018), CARN (Ahn et al., 2018), IMDN (Hui et al., 2019), AWSRN-M Wang et al. (2019), MADNet (Lan et al., 2020), DCDN (Li et al., 2021), SMSR (Wang et al., 2021), ECBSR (Zhang et al., 2021b), LAPAR-A (Li et al., 2020), HPUN-M (Sun et al., 2022), GLADSR (Zhang et al., 2021a), LCRCA (Peng et al., 2022), DRSAN-48s (Park et al., 2023), LatticeNet (Luo et al., 2023), AFAN-M (Wang et al., 2023b), and FDSCSR-S (Wang et al., 2023a), which are leading lightweight image super-resolution models in mainstream benchmark datasets. EFRDN demonstrates superior or second-best performance across most datasets, with fewer parameters and computational requirements compared to many methods, showcasing improved performance with reduced complexity. Particularly noteworthy is the significant performance enhancement on the Urban100 and Manga109 datasets in the comparison table across the three scaling factors. Our model's training on the RTX 2080Ti GPU and its low computational demand are among our key advantages.

Additionally, we present a visual comparison of EFRDN with other models. As depicted in Fig. 5, our EFRDN excels in restoring texture details in super-resolved images. The comprehensive comparison and analysis of all data and images validate the effectiveness and efficiency of our model. In img_062 (×2), although there is a slight fuzziness, EFRDN outperforms other methods in restoring image textures. Similarly, in img_148026 (×3), EFRDN accurately restores line orientation, enhancing image clarity. Notably, in img_038 (×4), EFRDN reconstructs texture images that closely resemble HR images.

4.4. Ablation studies

4.4.1. Asymmetric convolution distillation module (ACDM)

To assess the effectiveness of our Asymmetric Convolution Distillation Module (ACDM), we replaced it with IMDB (Hui et al., 2019) and RFDB (Liu et al., 2020), respectively. To ensure a fair comparison, all models were tuned to approximately 650K parameters, trained for a ×4 upscaling factor, and evaluated on the Set5 and Manage100 test datasets. The results are presented in Table 2. The outcomes indicate that our ACDM outperforms the other two modules within the same framework and with a similar parameter count. Although ACDM requires a slightly higher computational load, this increase is minor compared to the performance enhancement achieved. The effectiveness of our ACDM module is clearly demonstrated. Subsequently, we validate the effectiveness of each module within ACDM:

The effectiveness of ACRB: To assess the effectiveness of the Asymmetric Convolution Residual Block (ACRB), we conducted two ablation experiments. In the first experiment, we directly removed the ACRB module and BN layer. The data in Table 3 indicates a significant decrease in model performance. In the second experiment, to showcase the effectiveness of the ACRB module, we substituted the ACRB and BN layers in the ACDB module with a standard 3 × 3 convolution. This operation mirrors the one in the RFDB (Liu et al., 2020) module, where the distilled feature undergoes a 3 × 3 convolution for channel transformation and feature extraction after the feature split. Table 3 illustrates that the model size and computational load remained relatively unchanged after the replacement. However, there was a decrease in model performance, highlighting the superior effectiveness of our proposed strategy over that of RFDB (Liu et al., 2020).

The effectiveness of MSCC and FFLB: To further validate the effectiveness of the Multiple Self-Calibrated Convolutions (MSCC) and Feature Fusion Lattice Block (FFLB) modules, we individually removed these modules from ACDM, trained the models under a ×4 upscaling factor, and evaluated them on the Urban100 dataset. The test results are presented in Table 3 below. The data in the table highlights that both the MSCC and FFLB modules we designed effectively enhance the performance of the network model.

Table 1
Performance comparisons with other advanced CNN-based SISR models.

Methods	Scale	Params	Multi-Adds	Set5	Set14	BSD100	Urban100	Manga109
				PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
IDN (Hui et al., 2018)	×2	553K	124.6G	37.83/0.9600	33.30/0.9148	32.08/0.8985	31.27/0.9196	38.01/0.9749
CARN (Ahn et al., 2018)		1,592K	222.8G	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256	38.32/0.9765
IMDN (Hui et al., 2019)		694K	158.8G	38.00/0.9605	33.63/0.9177	32.19/0.8996	32.17/0.9283	**38.88**/0.9774
MADNet (Lan et al., 2020)		878K	187.1G	37.85/0.9600	33.38/0.9161	32.04/0.8979	31.62/0.9233	–
DCDN (Li et al., 2021)		756K	–	38.01/0.9606	33.52/0.9166	32.17/0.8996	32.16/0.9283	38.70/0.9773
SMSR (Wang et al., 2021)		985K	351.5G	38.00/0.9601	33.64/0.9179	32.17/0.8990	32.19/0.9284	38.76/0.9771
ECBSR (Zhang et al., 2021b)		596K	137.3G	37.90/**0.9615**	33.34/0.9178	32.10/**0.9018**	31.71/0.9250	–
LAPAR-A (Li et al., 2020)		548K	171.0G	38.01/0.9605	33.62/0.9183	32.19/0.8999	32.10/0.9283	38.67/0.9772
GLADSR (Zhang et al., 2021a)		812K	187.2G	37.99/0.9608	33.63/0.9179	32.16/0.8996	32.16/0.9283	–
LCRCA (Peng et al., 2022)		813K	186.0G	38.05/0.9607	33.65/0.9181	32.17/0.8994	32.19/0.9285	–
DRSAN-48s (Park et al., 2023)		650K	150.0G	**38.08**/0.9609	33.62/0.9175	32.19/0.9002	32.16/0.9286	–
LatticeNet (Luo et al., 2023)		756K	169.5G	38.06/0.9607	**33.70/0.9187**	**32.20**/0.8999	32.25/0.9288	–
AFAN-M (Wang et al., 2023b)		682K	163.4G	37.99/0.9605	33.57/0.9175	32.14/0.8994	32.08/0.9277	38.58/0.9769
FDSCSR-S (Wang et al., 2023a)		466K	121.8G	38.02/0.9606	33.51/0.9174	32.18/0.8996	32.24/0.9288	38.67/0.9771
EFRDN (Ours)		768K	111.6G	38.03/0.9609	33.65/0.9185	32.16/0.8997	**32.34/0.9298**	38.86/**0.9776**
IDN (Hui et al., 2018)	×3	553K	56.3G	34.11/0.9253	29.99/0.8354	28.95/0.8013	27.42/0.8359	32.71/0.9381
CARN (Ahn et al., 2018)		1,592K	118.8G	34.29/0.9255	30.29/0.8407	29.06/0.8493	28.06/0.8493	33.43/0.9427
IMDN (Hui et al., 2019)		703K	71.5G	34.36/0.9270	30.32/0.8417	29.09/0.8046	28.17/0.8519	33.61/0.9445
MADNet (Lan et al., 2020)		930K	88.4G	34.16/0.9253	30.21/0.8398	28.98/0.8023	27.77/0.8439	–
DCDN (Li et al., 2021)		765K	–	34.41/0.9273	30.31/0.8417	29.08/0.8045	28.17/0.8520	33.54/0.9441
SMSR (Wang et al., 2021)		993K	156.8G	34.40/0.9270	30.33/0.8412	29.10/0.8050	28.25/0.8536	33.68/0.9445
LAPAR-A (Li et al., 2020)		594K	114.0G	34.36/0.9267	30.34/0.8421	**29.11**/0.8054	28.15/0.8523	33.51/0.9441
GLADSR (Zhang et al., 2021a)		821K	88.2G	34.41/0.9272	30.37/0.8418	29.08/0.8050	28.24/0.8537	–
LCRCA (Peng et al., 2022)		822K	83.6G	34.40/0.9269	30.36/0.8422	29.09/0.8049	28.21/0.8532	–
DRSAN-48s (Park et al., 2023)		750K	78.0G	**34.47**/0.9274	30.35/0.8422	**29.11/0.8060**	28.26/**0.8542**	–
LatticeNet (Luo et al., 2023)		765K	76.3G	34.40/0.9272	30.32/0.8416	29.10/0.8049	28.19/0.8513	–
AFAN-M (Wang et al., 2023b)		681K	80.8G	34.35/0.9263	30.31/0.8423	29.06/0.8053	28.11/0.8522	33.44/0.9440
FDSCSR-S (Wang et al., 2023a)		471K	54.6G	34.42/0.9274	33.37/**0.8429**	29.10/0.8052	28.20/0.8532	33.55/0.9443
EFRDN (Ours)		768K	49.5G	34.44/**0.9275**	**30.38**/0.8414	29.10/0.8059	**28.29/0.8542**	**33.73/0.9453**
IDN (Hui et al., 2018)	×4	553K	32.3G	31.82/0.8903	28.25/0.7730	27.41/0.7297	25.41/0.7632	29.41/0.8942
CARN (Ahn et al., 2018)		1,592K	90.9G	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	30.42/0.9070
IMDN (Hui et al., 2019)		715K	40.9G	32.21/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838	30.45/0.9075
MADNet (Lan et al., 2020)		1,002K	54.1G	31.95/0.8917	28.44/0.7780	27.47/0.7327	25.76/0.7746	–
DCDN (Li et al., 2021)		777K	–	32.21/0.8949	28.57/0.7807	27.55/0.7356	26.09/0.7855	30.41/0.9072
SMSR (Wang et al., 2021)		1,006K	89.1G	32.12/0.8932	28.55/0.7808	27.55/0.7351	26.11/0.7868	30.54/0.9085
ECBSR (Zhang et al., 2021b)		603K	34.7G	31.92/0.8946	28.34/0.7817	27.48/0.7393	25.81/0.7773	–
LAPAR-A (Li et al., 2020)		659K	94.0G	32.15/0.8944	28.61/0.7818	27.61/0.7366	26.14/0.7871	30.42/0.9074
GLADSR (Zhang et al., 2021a)		826K	52.6G	32.14/0.8940	28.62/0.7813	27.59/0.7361	26.12/0.7851	–
LCRCA (Peng et al., 2022)		834K	47.7G	32.20/0.8948	28.60/0.7807	27.57/0.7653	26.10/0.7851	–
DRSAN-48s (Park et al., 2023)		730K	57.6G	32.25/0.8945	28.55/0.7817	27.59/0.7374	26.14/0.7875	–
LatticeNet (Luo et al., 2023)		777K	43.6G	32.18/0.8943	28.61/0.7812	27.57/0.7355	26.14/0.7844	–
AFAN-M (Wang et al., 2023b)		692K	50.9G	32.18/0.8939	28.62/0.7826	27.58/0.7373	26.13/0.7876	30.45/0.9085
FDSCSR-S (Wang et al., 2023a)		478K	31.1G	32.25/0.8959	28.61/0.7821	27.58/0.7367	26.12/0.7866	30.51/0.9087
EFRDN (Ours)		767K	27.9G	**32.33/0.8964**	**28.67/0.7833**	**27.63/0.7384**	**26.37/0.7939**	**30.76/0.9113**

Table 2
Performance comparisons of ACDM with other basic modules on Manga109 dataset.

Scale	Methods	Params	Multi-Adds	PSNR/SSIM
×4	EFRDN+IMDB (Hui et al., 2019)	637K	9.8G	30.46/0.9077
	EFRDN+RFDB (Liu et al., 2020)	688K	13G	30.52/0.9083
	EFRDN+ACDM (Ours)	650K	27.9G	**30.76/0.9113**

Table 3
Study of different units in ACDM on Urban100 dataset.

Scale	ACRB+BN	FFLB	MSCC	Params	Multi-Adds	PSNR/SSIM
×4	✗	✓	✓	761K	27.0G	26.24/0.7900
	✗[a]	✓	✓	761K	27.0G	26.29/0.7909
	✓	✗	✓	640K	17.0G	25.90/0.7777
	✓	✓	✗	750K	25.8G	26.25/0.7901
	✓	✓	✓	768K	27.9G	**26.37/0.7939**

[a] Represents replacing the ACRB unit with a 3 × 3 convolution.

4.4.2. Feature fusion lattice block (FFLB)

We evaluated the effectiveness of each module within the FFLB module by individually removing the GCRB, EFFB, and CC modules, conducting training under a ×4 upscaling factor, and evaluating them on the Urban100 dataset. The results are presented in Table 4. Reducing the performance of any module has a significant decline, indicating that each structure of the FFLB module designed by us is reasonable and effective, and one is indispensable. This underscores the validity of our designed modules.

4.4.3. Dense connection (DC)

Table 5 illustrates the effectiveness of the dense connection structure utilized. Three experiments were conducted: removing the dense connection structure from the CNN part, removing it from the Transformer part, and removing it from both parts simultaneously. The data in the table indicates that the dense connection structure does not notably increase the model's parameter count or computational

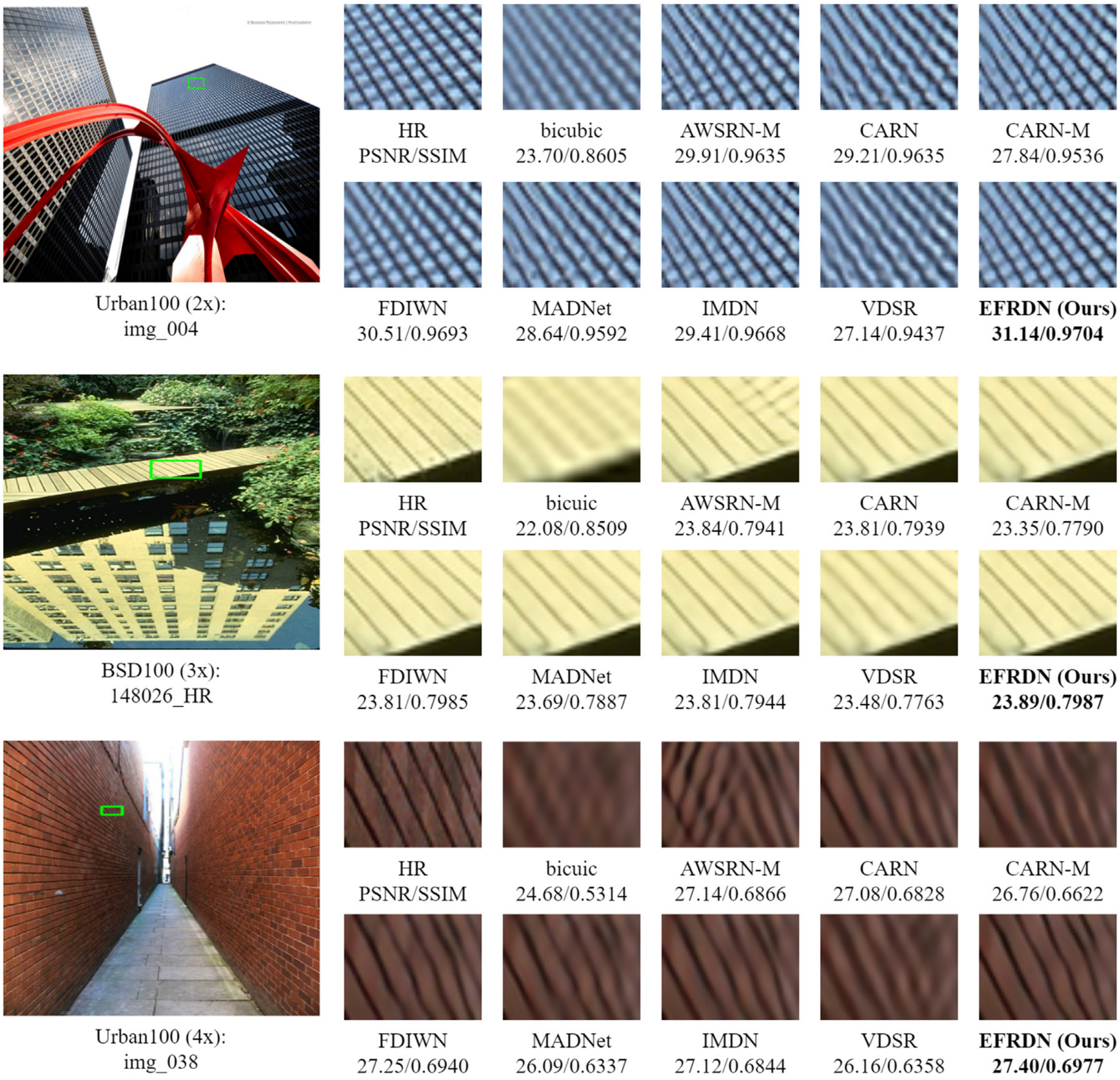


Fig. 5. Visual comparisons of EFRDN with other SR methods on BSDS100 and Urban100 datasets.

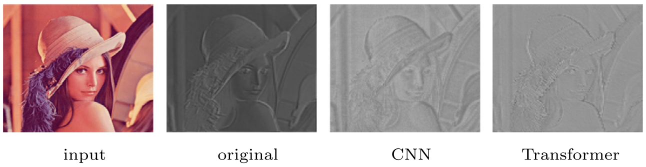


Fig. 6. CNN and Transformer module intermediate feature visualization.

Table 4
Study of different units in FFLB on Urban100 dataset.

Scale	GCRB	EFFB	CC	Params	Multi-Adds	PSNR/SSIM
×4	✗	✓	✓	762.84K	27.45G	26.34/0.7923
	✓	✗	✓	684.5K	18.58G	26.11/0.7849
	✓	✓	✗	764.0K	27.9G	26.34/0.7933
	✓	✓	✓	767.9K	27.9G	**26.37/0.7939**

load. However, the iterative utilization of extracted features significantly enhances the model's performance while maintaining model lightweightness.

Table 5
Study of DC on Urban100 dataset.

Scale	CNN-DC	Transformer-DC	Params	Multi-Adds	PSNR/SSIM
×4	✗	✓	767.9k	27.9G	26.17/0.7882
	✓	✗	767.9k	27.9G	26.22/0.7898
	✗	✗	767.9k	27.9G	26.28/0.7997
	✓	✓	767.9k	27.9G	**26.37/0.8964**

4.4.4. *Comparison with some transformer-based methods*

The Transformer-based SISR approach has gained significant attention recently. In this study, we compare EFRDN with some recent Transformer-based methods, namely SwinIR (Liang et al., 2021), ESRT (Lu et al., 2022), and LBNet (Gao et al., 2022b), as shown in Table 6. The comparison reveals that EFRDN has significantly reduced the computational load compared to the other models. EFRDN excels in computational efficiency, offering superior performance with minimal computation. In detail, our model outperforms the other two models, except for SwinIR, while also being lighter in weight. It is important to note that SwinIR was trained on a much larger dataset, Flickr2k, which includes 2650 HD images, whereas we used the DIV2k dataset

Table 6
Comparisons with some Transformer-based methods (×4).

Methods	Params	Multi-Adds	Set5	Set14	BSD100	Urban100	Manga109	Average
SwinIR (Liang et al., 2021)	897K	49.6G	32.44/0.8976	28.77/0.7858	27.69/0.7406	26.47/0.7980	30.92/0.9151	**29.26/0.8274**
ESRT (Lu et al., 2022)	751K	67.7G	32.19/0.8947	28.69/0.7833	27.69/0.7379	26.39/0.7962	30.75/0.9100	29.14/0.8244
LBNet (Gao et al., 2022b)	742K	38.9G	32.29/0.8960	28.68/0.7832	27.62/0.7382	26.27/0.7906	30.76/0.9111	29.12/0.8238
EFRDN (Ours)	767K	**27.9G**	32.33/0.8964	28.67/0.7833	27.63/0.7384	26.37/0.7939	30.76/0.9113	<u>29.15/0.8247</u>

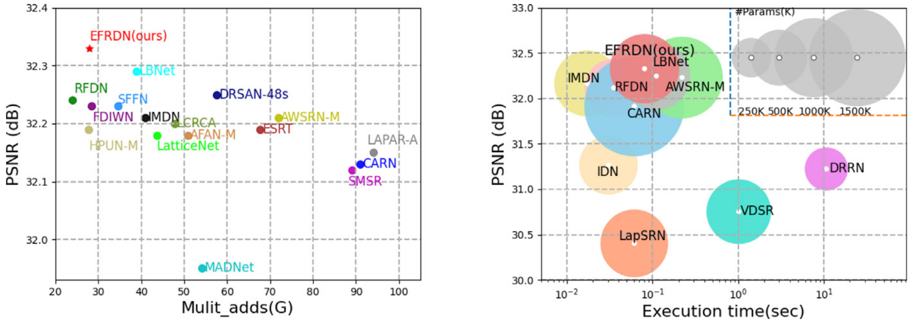


Fig. 7. Model calculations (left) and execution time (right) study on Set5 dataset (×4).

with only 1000 HD images. Despite this, our model achieves comparable performance with lower computational demands. Overall, EFRDN strikes a favorable balance between model efficiency and performance, further affirming its effectiveness.

In Fig. 6, we provide a visualization of the input image after feature extraction by the CNN and Transformer modules. Compared to the initial input image, the CNN module first captures rich local features. Following the Transformer module, the edge information and texture details become more pronounced, enhancing overall image quality and detail reconstruction. This confirms the effectiveness of our CNN and Transformer modules in extracting feature information. The combination of both modules successfully captures more detailed textures and global structures, leading to high-quality image reconstruction.

4.5. Model complexity studies

Fig. 7 distinctly illustrates the comparisons between EFRDN and existing methods regarding model size, computational efficiency, execution time, and model performance. The figure underscores the evident advantages of our model in terms of lightweight design and performance. Meanwhile, EFRDN successfully achieves a harmonious balance between parameter size and execution time, as demonstrated in Fig. 7, further emphasizing the efficiency and effectiveness of our model.

5. Conclusion

In this study, we introduce the Efficient Feature Reuse Distillation Network (EFRDN) model for efficient image super-resolution tasks. The CNN section of EFRDN comprises four Asymmetric Convolution Distillation Modules (ACDM), with shared interval parameters to reduce model parameters. ACDM utilizes a feature distillation structure, supporting model lightweightness while ensuring efficient feature extraction. Introducing the Transformer structure enables the effective capture of global feature information. The combination of CNN and Transformer in EFRDN facilitates the integration of local and global features. The framework incorporates a dense connection structure to fuse feature information at various levels, enhancing model performance. EFRDN effectively balances model size and performance, efficiently achieving super-resolution tasks. Meanwhile, our EFRDN utilizes a straightforward series combination of CNN and Transformer, which presents some limitations. Future work should explore more effective methods for integrating local and global features.

CRediT authorship contribution statement

Chunying Liu: Writing – original draft, Methodology. **Fei Wu:** Writing – review & editing, Resources, Investigation. **Zhenhua Guo:** Validation, Resources. **Yi Yu:** Writing – review & editing, Resources, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The author would like to thank the editor and the anonymous reviewers for their detailed comments and constructive suggestions which greatly contribute to this paper. This work was supported by the foundation of the Key Laboratory of Artificial Intelligence of Ministry of Education under Grant AI202404, the National Natural Science Foundation of China under Grant 62076139, and the Open Fund Project of Provincial Key Laboratory for Computer Information Processing Technology (Soochow University) under Grant KJS2274.

References

Ahn, N., Kang, B., Sohn, K.-A., 2018. Fast, accurate, and lightweight super-resolution with cascading residual network. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 252–268.
Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L., 2012. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: Proceedings of the British Machine Vision Conference. 135.1–135.10.
Chu, X., Zhang, B., Ma, H., Xu, R., Li, Q., 2021. Fast, accurate and lightweight super-resolution with neural architecture search. In: Proceedings of the International Conference on Pattern Recognition. ICPR, pp. 59–64.
Ding, M., Xiao, B., Codella, N., Luo, P., Wang, J., Yuan, L., 2022. Davit: Dual attention vision transformers. In: Proceedings of the European Conference on Computer Vision. pp. 74–92.
Dong, C., Loy, C.C., He, K., Tang, X., 2015. Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. 38 (2), 295–307.
Gao, G., Li, W., Li, J., Wu, F., Lu, H., Yu, Y., 2022a. Feature distillation interaction weighting network for lightweight image super-resolution. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, (no. 1), pp. 661–669.

Gao, G., Wang, Z., Li, J., Li, W., Yu, Y., Zeng, T., 2022b. Lightweight bimodal network for single-image super-resolution via symmetric CNN and recursive transformer. In: Proceedings of International Joint Conference on Artificial Intelligence. pp. 661–669.

Gao, G., Xu, Z., Li, J., Yang, J., Zeng, T., Qi, G.-J., 2023. CTCNet: A CNN-transformer cooperation network for face image super-resolution. IEEE Trans. Image Process. 32, 1978–1991.

Georgescu, M.-I., Ionescu, R.T., Miron, A.-I., Savencu, O., Ristea, N.-C., Verga, N., Khan, F.S., 2023. Multimodal multi-head convolutional attention with various kernel sizes for medical image super-resolution. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2195–2205.

Huang, J.-B., Singh, A., Ahuja, N., 2015. Single image super-resolution from transformed self-exemplars. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5197–5206.

Hui, Z., Gao, X., Yang, Y., Wang, X., 2019. Lightweight image super-resolution with information multi-distillation network. In: Proceedings of the ACM International Conference on Multimedia. pp. 2024–2032.

Hui, Z., Wang, X., Gao, X., 2018. Fast and accurate single image super-resolution via information distillation network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 723–731.

Jiang, X., Wang, N., Xin, J., Xia, X., Yang, X., Gao, X., 2021. Learning lightweight super-resolution networks with weight pruning. Neural Netw. 144, 21–32.

Jiang, K., Wang, Z., Yi, P., Lu, T., Jiang, J., Xiong, Z., 2022. Dual-path deep fusion network for face image hallucination. IEEE Trans. Neural Netw. Learn. Syst. 33 (1), 378–391.

Kim, J., Choi, M., Lee, S., 2024a. Efficient transformer for lightweight image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, pp. 1234–1243.

Kim, Y., Lee, J., Kim, H., 2024b. Transformer-based super-resolution with adaptive attention mechanism. Comput. Vis. Image Underst. 230, 103456. http://dx.doi.org/10.1016/j.cviu.2024.103456.

Kim, J., Lee, J.K., Lee, K.M., 2016a. Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1646–1654.

Kim, J., Lee, J.K., Lee, K.M., 2016b. Deeply-recursive convolutional network for image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1637–1645.

Lai, W.-S., Huang, J.-B., Ahuja, N., Yang, M.-H., 2017. Deep laplacian pyramid networks for fast and accurate super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 624–632.

Lan, R., Sun, L., Liu, Z., Lu, H., Pang, C., Luo, X., 2020. MADNet: a fast and lightweight network for single-image super resolution. IEEE Trans. Cybern. 51 (3), 1443–1453.

Lee, W., Lee, J., Kim, D., Ham, B., 2020. Learning with privileged information for efficient image super-resolution. In: Proceedings of the European Conference on Computer Vision. pp. 465–482.

Li, Y., Cao, J., Li, Z., Oh, S., Komuro, N., 2021. Lightweight single image super-resolution with dense connection distillation network. ACM Trans. Multimed. Comput. Commun. Appl. 17 (1s), 1–17.

Li, W., Li, J., Gao, G., Deng, W., Yang, J., Qi, G.-J., Lin, C.-W., 2022. Efficient image super-resolution with feature interaction weighted hybrid network. arXiv preprint arXiv:2212.14181.

Li, W., Li, J., Gao, G., Deng, W., Zhou, J., Yang, J., Qi, G.-J., 2024a. Cross-receptive focused inference network for lightweight image super-resolution. IEEE Trans. Multimed. 26, 864–877.

Li, J., Pei, Z., Li, W., Gao, G., Wang, L., Wang, Y., Zeng, T., 2024b. A systematic survey of deep learning-based single-image super-resolution. ACM Comput. Surv. 56 (10), 1–40.

Li, W., Zhou, K., Qi, L., Jiang, N., Lu, J., Jia, J., 2020. Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. Adv. Neural Inf. Process. Syst. 33, 20343–20355.

Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R., 2021. Swinir: Image restoration using swin transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1833–1844.

Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K., 2017. Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 136–144.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022.

Liu, Z., Liu, H., Zhang, X., Wang, X., 2024a. Top-K token selective transformer for high-resolution image reconstruction. IEEE Trans. Image Process. 33, 738–752.

Liu, J., Tang, J., Wu, G., 2020. Residual feature distillation network for lightweight image super-resolution. In: Proceedings of the European Conference on Computer Vision Workshops. pp. 41–55.

Liu, H., Wang, J., Zhang, T., 2024b. Efficient attention-based transformer for lightweight super-resolution. Comput. Vis. Image Underst. 205, 103441.

Lu, Z., Li, J., Liu, H., Huang, C., Zhang, L., Zeng, T., 2022. Transformer for single image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 457–466.

Luo, X., Qu, Y., Xie, Y., Zhang, Y., Li, C., Fu, Y., 2023. Lattice network for lightweight image restoration. IEEE Trans. Pattern Anal. Mach. Intell. 45 (4), 4826–4842.

Martin, D., Fowlkes, C., Tal, D., Malik, J., 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of the IEEE International Conference on Computer Vision, vol. 2, pp. 416–423.

Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., Aizawa, K., 2017. Sketch-based manga retrieval using manga109 dataset. Multimedia Tools Appl. 76, 21811–21838.

Park, K., Soh, J.W., Cho, N.I., 2023. A dynamic residual self-attention network for lightweight single image super-resolution. IEEE Trans. Multimed. 25, 907–918.

Peng, C., Shu, P., Huang, X., Fu, Z., Li, X., 2022. LCRCA: image super-resolution using lightweight concatenated residual channel attention networks. Appl. Intell. 1–15.

Sun, B., Zhang, Y., Jiang, S., Fu, Y., 2022. Hybrid pixel-unshuffled network for lightweight image super-resolution. arXiv preprint arXiv:2203.08921.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30, 1–11.

Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 13 (4), 600–612.

Wang, L., Dong, X., Wang, Y., Ying, X., Lin, Z., An, W., Guo, Y., 2021. Exploring sparsity in image super-resolution for efficient inference. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4917–4926.

Wang, Z., Gao, G., Li, J., Yan, H., Zheng, H., Lu, H., 2023a. Lightweight feature de-redundancy and self-calibration network for efficient image super-resolution. ACM Trans. Multimedia Comput. Commun. Appl. 19 (3), 1–15.

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

Wang, L., Li, K., Tang, J., Liang, Y., 2023b. Image super-resolution via lightweight attention-directed feature aggregation network. ACM Trans. Multimedia Comput. Commun. Appl. 19 (2), 1–23.

Xiao, Y., Yuan, Q., Zhang, Q., Zhang, L., 2023. Deep blind super-resolution for satellite video. IEEE Trans. Geosci. Remote Sens.

Zeyde, R., Elad, M., Protter, M., 2012. On single image scale-up using sparse-representations. In: Proceedings of the International Conference on Curves and Surfaces. pp. 711–730.

Zhang, W., Chen, L., Xu, L., 2024a. A lightweight transformer framework for real-time image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, pp. 2040–2050.

Zhang, Y., Dong, C., Wu, Y., 2023. Efficient super-resolution with dynamic convolution and attention mechanisms. IEEE Trans. Neural Netw. Learn. Syst. 34 (5), 6789–6801. http://dx.doi.org/10.1109/TNNLS.2023.1234567.

Zhang, X., Gao, P., Liu, S., Zhao, K., Li, G., Yin, L., Chen, C.W., 2021a. Accurate and efficient image super-resolution via global-local adjusting dense network. IEEE Trans. Multimed. 23, 1924–1937.

Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y., 2018. Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 286–301.

Zhang, L., Wang, X., Zhang, X., Liu, X., 2024b. Enhancing the super-resolution with contrastive learning. IEEE Trans. Image Process. 33 (2), 1234–1245. http://dx.doi.org/10.1109/TIP.2024.1234567.

Zhang, X., Zeng, H., Zhang, L., 2021b. Edge-oriented convolution block for real-time super resolution on mobile devices. In: Proceedings of the ACM International Conference on Multimedia. pp. 4034–4043.