Efficient Dual-Branch Information Interaction Network for Lightweight Image Super-Resolution

Haonan Jin, Guangwei Gao, *Senior Member, IEEE*, Juncheng Li, Zhenhua Guo, and Yi Yu, *Senior Member, IEEE*

***Abstract*— Recently, deep convolutional neural networks (CNNs) have achieved remarkable success in single-image super-resolution (SISR) tasks. However, these methods often suffer from high computational and memory requirements, limiting their practicality for real-world applications. To address this challenge, we propose a lightweight and efficient dual-branch information interaction network (DIIN) for SISR. DIIN adopts a dual-branch structure that differs from the typical serial network architectures. Specifically, we design the CNN branch and Transformer branch as parallel structures. In the CNN branch, we employ a symmetric dual-branch feature interaction module (DFIM) to extract valuable local feature information. Concurrently, the Transformer branch utilizes a recursive Transformer to capture long-term global information and enhance reconstructed image details. By simultaneously considering these two branches, our model effectively combines the strengths of CNN in extracting local information and Transformer in capturing global information. Recognizing the complementarity of these two branches in SISR, we further incorporate a coefficient learning scheme to enhance their information interaction and obtain more comprehensive feature information, thereby improving overall model performance. Extensive experiments demonstrate that our DIIN outperforms competitive methods while consuming fewer computational resources and memory.**

***Index Terms*— Information interaction, lightweight network, single image super-resolution (SISR).**

I. Introduction

IMAGE super-resolution (SR) is a fundamental task to enhance low-resolution (LR) images by generating high-resolution (HR) counterparts, thereby improving their visual quality and capturing finer details. For more accurate analysis and measurement, SR is used to enhance the quality of images captured in instrumentation and measurement processes, leading to more precise and reliable results [1], [2]. However, this LR-to-HR mapping is ill-posed, as multiple HR images can be downsampled to yield the same LR image. In recent years, convolutional neural networks (CNNs) have gained significant attention in SR methods due to their exceptional feature extraction capabilities, surpassing traditional approaches [3], [4]. The groundbreaking work by Dong et al. [5] introduced the super-resolution CNN (SRCNN), which served as a foundation for subsequent CNN-based single-image SR (SISR) models. Building upon this, Kim et al. [6] proposed a highly deep SR model called very deep SR (VDSR), comprising 20 convolutional layers. VDSR exhibited superior performance compared to SRCNN, as the authors observed that deeper network architectures enable larger receptive fields. This capability allows the model to capture more contextual information and ultimately achieve better SR results.

In recent years, there has been a noticeable trend in SR algorithms to incorporate additional convolutional layers to extract more image features and enhance performance. However, this approach often results in larger model parameters, increased memory consumption, and slower training and testing speeds. For instance, RCAN [7], despite demonstrating promising results, consists of over 800 convolutional layers and has a parameter volume of around 15 M, making it unsuitable for devices with limited resources. Consequently, there is an increasing demand for the development of lightweight and efficient models that can be deployed on mobile devices while still achieving high SR performance within the constraints of available resources.

To address this issue, research on lightweight backbone networks has become a hot topic in recent years for mobile devices. Many methods adopt recursive techniques or parameter-sharing strategies to reduce the number of parameters [8]. Increasing network depth or width can compensate for the diminished performance caused by the reduced number of parameters, but it also requires more computation time. CARN [9] leverages weight sharing and group convolution to reduce network parameters. IMDN [10] employs residual feature distillation and contrast-aware channel attention (CA) to ensure model efficiency. LCRCA [11] proposed a lightweight and effective residual block that improves residual information in the same computational budget. SFFN [12] suggested using

Manuscript received 8 April 2024; revised 17 July 2024; accepted 10 August 2024. Date of publication 26 August 2024; date of current version 6 September 2024. This work was supported in part by the Key Laboratory of Artificial Intelligence of Ministry of Education under Grant AI202404, in part by the National Natural Science Foundation of China under Grant 62301306, and in part by the Open Fund Project of Provincial Key Laboratory for Computer Information Processing Technology (Soochow University) under Grant KJS2274. The Associate Editor coordinating the review process was Dr. Chuang Sun. *(Corresponding author: Guangwei Gao.)*

Haonan Jin and Guangwei Gao are with the Intelligent Visual Information Perception Laboratory, Institute of Advanced Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China, also with the Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai 200240, China, and also with the Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006, China (e-mail: 710714437@qq.com; csggao@gmail.com).

Juncheng Li is with the School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China (e-mail: cvjunchengli@gmail.com).

Zhenhua Guo is with the Tianyijiaotong Technology Ltd., Suzhou 215100, China (e-mail: cszguo@gmail.com).

Yi Yu is with the Graduate School of Advanced Science and Engineering, Hiroshima University, Hiroshima 739-8511, Japan (e-mail: yiyu@hiroshima-u.ac.jp).

Digital Object Identifier 10.1109/TIM.2024.3450063

a general-purpose, lightweight, and efficient feature fusion block that substitutes the commonly used 1×1 convolution. ETDS [13] converted time-consuming operators into more efficient alternatives and introduced a dual-stream network to improve the capability of feature extraction. With the continuous development of Transformers in natural language processing, researchers have started exploring the possibility of applying them to computer vision tasks. While the Transformer excels at enhancing long-term dependencies in image data and significantly improves image detail restoration, most previous methods simply replaced CNN structures with Transformers, which can cause the network to lose its ability to extract local features. These local features play a crucial role in image understanding and reconstruction by maintaining their stability under different viewing angles. Therefore, fully integrating the advantages of CNN and Transformer for improved image reconstruction remains a significant challenge.

Most of the current SR model backbones use a serial structure, which is a connection structure between modules. This allows feature information to flow unidirectionally and gradually extract more feature information to restore higher quality images. For models that integrate CNN and Transformer, such as LBNet [8], the model backbone is divided into CNN and Transformer parts and connected in series, combining the advantages of CNN and Transformer to achieve good performance. At the same time, inspired by LatticeNet [14], the butterfly structure in it has inspired us. Different operations are performed in the upper and lower branches of this structure, which gives them incomplete identical feature information. By interacting and fusing this information, more effective feature information can be obtained to restore better images. Therefore, we attempt to create a butterfly structure between CNN and Transformer, allowing short-term and long-term information to flow through the network for better performance.

To this end, we propose a lightweight dual-branch information interaction network (DIIN). Unlike common serial networks, which typically use a CNN to extract local feature information followed by a Transformer module (TM) to further extract global feature information, our DIIN forms a parallel structure where CNN and Transformer are combined. This enables the simultaneous flow of both local and global feature information within our network. To further enhance performance by capturing fine-grained texture details and acquiring global information, we adopt a recursive strategy on the basic Transformer in the Transformer branch. This recursive mechanism reduces the number of network parameters and computational costs. Furthermore, DIIN utilizes combination coefficients (CCs) between the CNN branch and Transformer branch to facilitate information exchange, allowing our network to obtain more feature information and produce better image restoration results.

Our contributions can be summarized as follows.

1) We propose an effective symmetric dual-branch feature interaction module (DFIM) that utilizes multiscale feature extraction units (MFEUs) to extract local features for image reconstruction.
2) We use a recursive Transformer in the Transformer branch to learn long-term dependencies in images, allowing us to obtain global information and refine texture details while reducing the number of network parameters and computation costs.
3) We introduce the new lightweight DIIN, which integrates CNN and Transformer mechanisms using an interactive scheme to provide complementary information yielding an optimal balance between model size and performance.

II. Related Work

A. CNN-Based Lightweight Super-Resolution

Recently, researchers have been striving to achieve a better balance between model size and performance [13]. Hui et al. [15] developed a lightweight information distillation network (IDN) for SR that achieved good results with a small number of parameters. Using IDN as a basis, IMDN [10] constructed a cascaded information multilayer distillation block that extracts hierarchical features from the image step by step. Liu et al. [16] comprehensively analyzed the information distillation mechanism, introduced residual mechanisms into IMDN, and proposed the residual feature distillation network (RFDN). Other approaches such as ESRN [17] utilized the neural architecture search [18] strategy to create a lightweight model structure. Lan et al. [19] leveraged MADNet's dense lightweight network to enhance the representation and learning of the metric features. Sun et al. [20] designed a hybrid pixel unshuffled network (HPUN) introducing an effective downsampling module. Additionally, Wang et al. [21] proposed a lightweight attention-directed feature aggregation network (AFAN), which consists of a simple CA module and stacked multiaware attention modules. Gao et al. [22] proposed a lightweight image SR method called feature distillation interaction weighting network (FDIWN), which utilizes residual distillation to extract deep feature information and enhance the restoration of super-resolved images. At last, Park et al. [23] devised a lightweight dynamic residual self-attention network (DRSAN), incorporating the automated design of residual connections.

B. Transformer-Based Super-Resolution

The Transformer was initially used in natural language processing, but its research has extended to the field of computer vision. With further exploration, the use of Transformers has proven to be effective in dealing with long-term dependencies in images. Consequently, numerous Transformer-based methods have been presented and applied to various computer vision tasks. For instance, Chen et al. [24] developed a pretraining image processing Transformer that achieves promising results in SR, noise removal, and rain removal. Liang et al. [25] transplanted the Swin Transformer directly into image restoration tasks to merge the benefits of convolution networks and Transformers. Compared to previous models, SwinIR has fewer parameters and has achieved better results. Later on, Lu et al. [26] presented an efficient SR transformer (ESRT), combining lightweight CNN and Transformer backbones to use global information for better performance. Gao et al. [8]

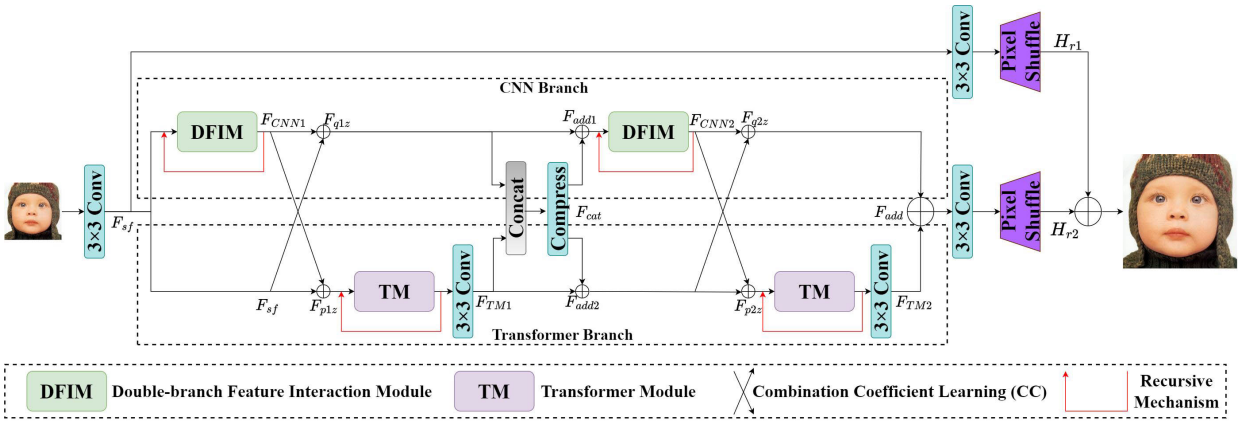


Fig. 1. Architecture of the proposed DIIN.

presented a lightweight bimodal network for SISR using cascading symmetric CNN and recursive Transformers. Li et al. [27] proposed a lightweight cross-receptive focused inference network (CFIN). Its objective is to adaptively modify network weights by incorporating modulated convolutional kernels with local representative semantic information. In Chen et al. [28] dual aggregation transformer for image super resolution, it stitches together spatial attention and CA, and parallelizes the Transformer with CNN in the module, while extracting local attention and global attention. However, these approaches failed to utilize CNNs and Transformers fully. Furthermore, balancing the size and performance of the model remains challenging.

III. Proposed Method

A. Network Framework

In this section, we provide a detailed description of the lightweight DIIN, which comprises the CNN branch, Transformer branch, and image reconstruction component (as shown in Fig. 1). The CNN branch utilizes a series of symmetrical DFIMs for local feature extraction, while the Transformer branch focuses on global feature extraction. Let I_{LR}, I_{SR}, and I_{HR} represent the input LR images, reconstructed SR images, and corresponding HR images, respectively. At the beginning of the model, we apply a 3×3 convolutional layer to extract shallow features

$$F_{\mathrm{sf}} = H_{\mathrm{sf}}(I_{\mathrm{LR}}) \tag{1}$$

where $H_{\mathrm{sf}}(\cdot)$ represents the operation for extracting shallow features and $F_{\mathrm{sf}} \in R^{C\times W\times H}$ (C denotes the number of channels and is set as 32 in our model) denotes the extracted shallow features. Then, these shallow features are fed into both the CNN branch and Transformer branch for local and global feature extraction

$$F_{\mathrm{CNN1}} = H_{\mathrm{DFIM1}}(F_{\mathrm{sf}}) \tag{2}$$

where $H_{\mathrm{DFIM1}}(\cdot)$ denotes the first CNN-based symmetrical DFIM, while $F_{\mathrm{CNN1}} \in R^{C\times W\times H}$ represents the locally extracted features of the first CNN branch. Symmetric DFIM is one of the key elements of DIIN and consists of multiple MFEUs. These modules will be discussed in detail in Section III-B. To enable information interaction between the upper and lower branches, the combining coefficient (CC) learning scheme is employed to obtain corresponding weights for the feature information

$$\begin{cases} F_{\mathrm{q1z}} = F_{\mathrm{CNN1}} + F_{\mathrm{sf}} \times H_{\mathrm{CC}}(F_{\mathrm{sf}}) \\ F_{\mathrm{p1z}} = F_{\mathrm{sf}} + F_{\mathrm{CNN1}} \times H_{\mathrm{CC}}(F_{\mathrm{CNN1}}) \end{cases} \tag{3}$$

where $H_{\mathrm{CC}}(\cdot)$ represents the CC operation, $F_{\mathrm{q1z}} \in R^{C\times W\times H}$ and $F_{\mathrm{p1z}} \in R^{C\times W\times H}$ refer to the first interactive information from the upper and lower branches, respectively. Subsequently, F_{p1z} is inputted into the first Transformer unit in the Transformer branch to obtain the global information:

$$F_{\mathrm{TM1}} = H_{\mathrm{TM1}}\left(F_{\mathrm{p1z}}\right) \tag{4}$$

where $H_{\mathrm{TM1}}(\cdot)$ refers to the Transformer operation, while $F_{\mathrm{TM1}} \in R^{C\times W\times H}$ denotes the global feature information extracted by the first Transformer. Afterward, the local information from the upper branch and global information from the lower branch are combined to further enhance the interaction between the two branches

$$F_{\mathrm{cat}} = H_{\mathrm{compress}}\left(\mathrm{Concat}\left(F_{\mathrm{q1z}}, F_{\mathrm{TM1}}\right)\right) \tag{5}$$

$$\begin{cases} F_{\mathrm{add1}} = F_{\mathrm{cat}} + F_{\mathrm{q1z}} \\ F_{\mathrm{add2}} = F_{\mathrm{cat}} + F_{\mathrm{TM1}} \end{cases} \tag{6}$$

where Concat$(\cdot)$ represents the concatenation operation along the channel dimension, $H_{\mathrm{compress}}(\cdot)$ denotes the 1×1 compression channel convolution layer, $F_{\mathrm{cat}} \in R^{C\times W\times H}$ represents the result of the channel compression, and $F_{\mathrm{add1}} \in R^{C\times W\times H}$ and $F_{\mathrm{add2}} \in R^{C\times W\times H}$ refer to the interactive information from the upper and lower branches, respectively. The subsequent operation is similar to that of the previous module and can be expressed as follows:

$$F_{\mathrm{CNN2}} = H_{\mathrm{DFIM2}}(F_{\mathrm{add1}}) \tag{7}$$

$$\begin{cases} F_{\mathrm{q2z}} = F_{\mathrm{CNN2}} + F_{\mathrm{add2}} \times H_{\mathrm{CC}}(F_{\mathrm{add2}}) \\ F_{\mathrm{p2z}} = F_{\mathrm{add2}} + F_{\mathrm{CNN2}} \times H_{\mathrm{CC}}(F_{\mathrm{CNN2}}) \end{cases} \tag{8}$$

$$F_{\mathrm{TM2}} = H_{\mathrm{TM2}}\left(F_{\mathrm{p2z}}\right) \tag{9}$$

where $H_{\mathrm{DFIM2}}(\cdot)$ denotes the second CNN-based symmetrical DFIM, $F_{\mathrm{CNN2}} \in R^{C\times W\times H}$ represents the locally extracted

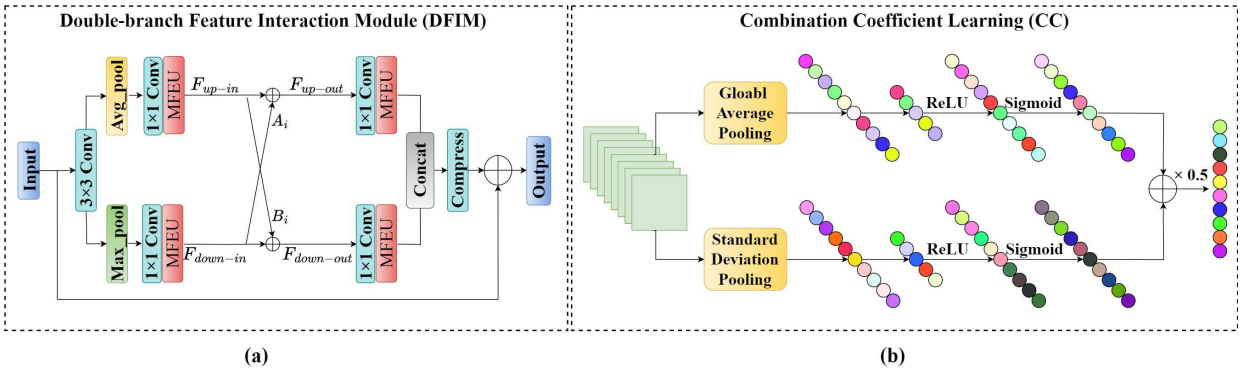


Fig. 2. Architecture of the DFIM and the CC learning.

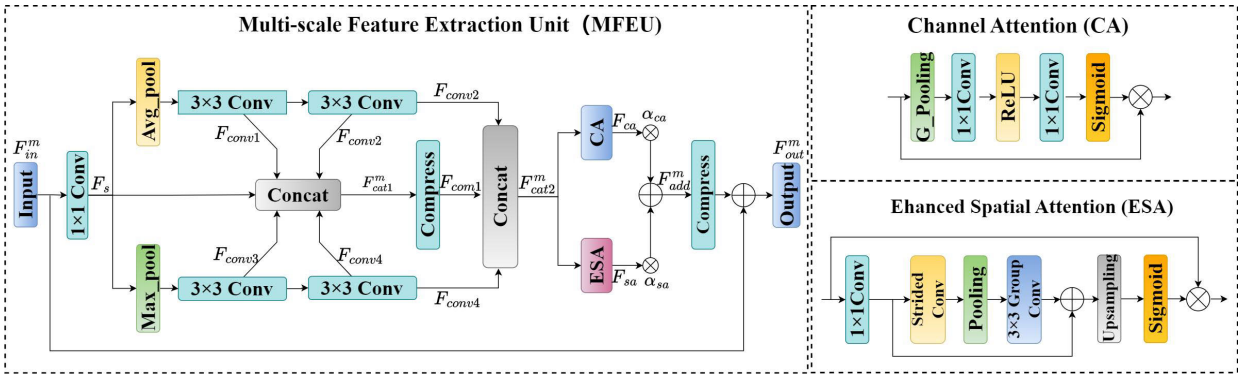


Fig. 3. Architecture of MFEU.

features from the second CNN branch, F_{add1} and F_{add2} represents the addition of the upperand down branches, $H_{\text{CC}}(\cdot)$ represents the CC operation, $F_{\text{q2z}} \in R^{C\times W\times H}$ and $F_{\text{p2z}} \in R^{C\times W\times H}$ refer to the second interactive information from the upper and lower branches, $H_{\text{TM2}}(\cdot)$ refers to the Transformer operation, and $F_{\text{TM2}} \in R^{C\times W\times H}$ denotes the global feature information extracted by the second Transformer. Once the local characteristics and global information from the upper and lower branches are added, it is fed into the reconstruction module for SR image reconstruction. Meanwhile, the shallow features are also included to create the final SR image

$$I_{\text{SR}} = H_{r1}\left(F_{\text{q2z}} + F_{\text{TM2}}\right) + H_{r2}(F_{\text{sf}}) = H_{\text{DIIN}}(L_{\text{LR}}) \tag{10}$$

where $H_r(\cdot)$ refers to the image reconstruction module, which is composed of a 3×3 convolutional layer and the pixel-shuffle layer, and H_{DIIN} denotes the proposed network. Similar to previous work, we use the L_1 loss function to optimize the model during training. Given a training dataset $\{I^i_{\text{LR}}, I^i_{\text{HR}}\}_{i=1}^N$, we solve

$$\mathcal{L}(\Theta) = \frac{1}{N}\sum_{i=1}^{N}\left\| H_{\text{DIIN}}\left(I^i_{\text{LR}}, \Theta\right) - I^i_{\text{HR}}\right\|_1 \tag{11}$$

where Θ indicates the parameters set of the proposed DIIN, N denotes the total number of the training images.

B. Dual-Branch Feature Interaction Module

As illustrated in Fig. 2(a), our DFIM consists of multiscale feature extraction units (MFEUs), CC learning, and a 3×3 convolution that reduces the number of feature channels. Furthermore, a residual link is incorporated to preserve the original information of the features.

1) Multiscale Feature Extraction Unit: As depicted in Fig. 3, the MFEU begins with a 1×1 convolutional layer to extract shallow features. The data are then enriched through the average-pooling and max-pooling layers to obtain more useful feature information. Moreover, each branch incorporates two 3×3 convolution layers to expand the receptive field and facilitate the fusion of features. These operations can be summarized as follows:

$$\begin{aligned} F_s &= H_{1\times1}\left(F^m_{\text{in}}\right) \\ F_{\text{conv1}} &= H_{3\times3}(H_{\text{avg}}(F_s)) \\ F_{\text{conv2}} &= H_{3\times3}(F_{\text{conv1}}) \\ F_{\text{conv3}} &= H_{3\times3}(H_{\max}(F_s)) \\ F_{\text{conv4}} &= H_{3\times3}(F_{\text{conv3}}) \\ F^m_{\text{cat1}} &= \text{Concat}(F_s, F_{\text{conv1}}, F_{\text{conv2}}, F_{\text{conv3}}, F_{\text{conv4}}) \end{aligned} \tag{12}$$

where $F_s, F_{\text{conv1}}, F_{\text{conv2}}, F_{\text{conv3}}, F_{\text{conv4}} \in R^{C\times W\times H}$, and $F^m_{\text{cat1}} \in R^{5C\times W\times H}$. $F^m_{\text{in}} \in R^{C\times W\times H}$ denotes the input feature of the MFEU, $H_{1\times1}(\cdot)$ denotes the 1×1 convolutional layer, $H_{\text{avg}}(\cdot)$ indicates the average-pooling operation, $H_{\max}(\cdot)$ denotes the max-pooling operation, and $H_{3\times3}(\cdot)$ indicates the 3×3 convolutional layer. Then, the integrated features are compressed by the 1×1 convolutional layer and combined with the previous feature information to get a more informative feature representation. To extract channel statistics and spatial contextual information, the CA module and the enhance spatial attention

(ESA) module are utilized, with adaptive weights computed on the output of the two attention operations. Finally, the output information is obtained through residual links that connect the original feature information with the output from the above operations. All of these steps can be summarized as follows:

$$\begin{aligned} F_{\text{com1}} &= H_{\text{compress1}}\left(F^m_{\text{cat1}}\right) \\ F^m_{\text{cat2}} &= \text{Concat}(F_{\text{com1}}, F_{\text{conv2}}, F_{\text{conv4}}) \\ F_{\text{ca}} &= H_{\text{ca}}\left(F^m_{\text{cat2}}\right) \\ F_{\text{sa}} &= H_{\text{esa}}\left(F^m_{\text{cat2}}\right) \\ F^m_{\text{add}} &= \alpha_{\text{ca}} F_{\text{ca}} + \alpha_{\text{sa}} F_{\text{sa}} \\ F^m_{\text{out}} &= H_{\text{compress2}}\left(F^m_{\text{add}}\right) + F^m_{\text{in}} \end{aligned} \tag{13}$$

where $F_{\text{com1}}, F^m_{\text{out}} \in R^{C\times W\times H}$, $F^m_{\text{cat2}}, F_{\text{ca}}, F_{\text{sa}}, and F^m_{\text{add}} \in R^{3C\times W\times H}$. $H_{\text{compress1}}(\cdot)$ and $H_{\text{compress2}}(\cdot)$ represents the 1×1 compression channel convolution layer, $H_{\text{ca}}(\cdot)$ and $H_{\text{esa}}(\cdot)$ indicate the CA and ESA, and α indicates the corresponding adapter.

2) Combination Coefficient Learning: Inspired by the butterfly structure proposed by Luo et al. [32], we utilize the CC learning scheme as a bridge for feature information circulation between the CNN and Transformer structures, to achieve the information interaction of the two branches. As depicted in Fig. 2(a), vectors A_i and B_i are employed as link weights in the module. These operations can be summarized as follows:

$$\begin{aligned} A_i &= H_{\text{CC}}(F_{\text{down-in}}) \\ B_i &= H_{\text{CC}}(F_{\text{up-in}}) \\ F_{\text{up-out}} &= F_{\text{up-in}} + A_i(F_{\text{down-in}}) \\ F_{\text{down-out}} &= F_{\text{down-in}} + B_i(F_{\text{up-in}}) \end{aligned} \tag{14}$$

where A_i and B_i are calculated as weight values by the lower and upper branches with the CC operation H_{CC}, respectively. The final outputs $F_{\text{up-out}} \in R^{C\times W\times H}$ and $F_{\text{down-out}} \in R^{C\times W\times H}$ are determined according to the format described earlier.

As illustrated in Fig. 2(b), the CC scheme is made up of two statistical measures, namely the average value and the standard deviation of the feature map. Given a set of feature maps, the upper branch utilizes an average-pooling layer to obtain the mean value of each feature map, while the lower branch calculates the standard deviation of each feature map. The statistical vector generated by each branch is then fed into two 1×1 convolutional layers, followed by a ReLU activation layer and a Sigmoid activation layer. Finally, the outputs of the two branches are combined to form the output of the CC.

C. Recursive Efficient Transformer

As previously mentioned, a symmetrical cross-feature CNN is used to extract local features in diagrams. However, local features alone are insufficient for rebuilding high-quality SR images, and it is necessary to extract global feature information. To address this issue, we incorporated Transformer to learn long-term image dependencies. For the Transformer, we employed the coding portion of the standard Transformer structure proposed by Lu et al. in ESRT [26]. As illustrated in Fig. 4, the Transformer comprises an efficient multihead attention (EMHA) and a multilayer perceptron (MLP). Additionally, layer normalization (Norm) [33] is implemented before each block, with a residual connection applied after each block. For the input feature $F^t_{\text{in}} \in R^{C\times W\times H}$ of the TM, these operations can be summarized as follows:

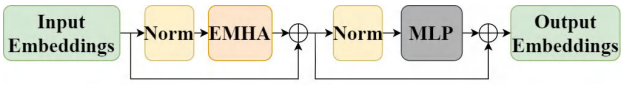


Fig. 4. Architecture of TM.

$$\begin{aligned} F^t_{\text{mid}} &= H_{\text{EMHA}}\left(H_{\text{Norm}}\left(F^t_{\text{in}}\right)\right) \\ F^t_{\text{out}} &= H_{\text{MLP}}\left(H_{\text{Norm}}\left(F^t_{\text{mid}}\right)\right) \end{aligned} \tag{15}$$

where $F^t_{\text{mid}}, F^t_{\text{out}} \in R^{C\times W\times H}$. $H_{\text{Norm}}(\cdot)$ represents layer normalization operations, $H_{\text{EMHA}}(\cdot)$ and $H_{\text{MLP}}(\cdot)$ represent the EMHA and MLP modules, respectively. Following [34], each head of the EMHA must perform scaled dot product attention, and then concatenate all the outputs before performing a linear transformation to obtain the output. The scaled dot product attention can be expressed as

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{16}$$

where Softmax represents the softmax operation, and Q, K, and V denotes the matrix of the query, key, and value. To make better use of the Transformer's long-term dependencies without increasing the number of parameters, we implement a recursive mechanism that facilitates parameter sharing. This is expressed as

$$F^{\text{rt}}_{\text{out}} = H_{3\times 3}\left(H^{\circ}_{\text{TM2}}\left(H^{\circ}_{\text{TM1}}\left(F^{\text{rt}}_{\text{in}}\right)\right)\right) \tag{17}$$

where $F^{\text{rt}}_{\text{in}}, F^{\text{rt}}_{\text{out}} \in R^{C\times W\times H}$, $\circ$ indicates the recursion operation and the output of TM is recalculated as the input.

IV. Experiments

A. Datasets and Evaluation Metrics

Similar to previous work, we employ DIV2K as the primary dataset for model training. To evaluate the effectiveness of DIIN, we use five benchmark test datasets, including Set5 [36], Set14 [37], Urban100 [38], BSDS100 [39], and Manga109 [40]. The amplification factors are set to $\times$ 2, $\times$ 3, and $\times$ 4, respectively. Furthermore, we employ peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [41] as evaluation metrics to assess the performance of SR images on the Y channel of the YCbCr color space.

B. Implementation Details

During the training process, we randomly extract a series of 48×48 patches from the training data as inputs and augment them using random rotation and horizontal flips. We employ an initial learning rate of 2×10^{-4}, which is subsequently reduced to 6.25×10^{-6} using cosine annealing. The network is trained using the Adam optimizer in the PyTorch tool with an NVIDIA RTX 2080Ti GPU. In the final model configuration, each module has input and output channels set to 32, and two DFIMs are used. The TM undergoes one recursive iteration.

TABLE I

AVERAGE PSNR/SSIM VALUES FOR SCALE FACTOR × 2, × 3 AND × 4 ON SET5, SET14, BSD100, URBAN100, AND MANGA109 DATASETS. THE BEST AND SECOND BEST INDEXES ARE **HIGHLIGHTED** AND UNDERLINED

Methods	Scale	Params	Multi-Adds	Set5	Set14	BSD100	Urban100	Manga109
				PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM
IDN [15]		553K	124.6G	37.83 / 0.9600	33.30 / 0.9148	32.08 / 0.8985	31.27 / 0.9196	38.01 / 0.9749
IMDN [10]		694K	158.8G	38.00 / 0.9605	33.63 / 0.9177	32.19 / 0.8996	32.17 / 0.9283	38.88 / 0.9774
AWSRN-M [29]		1,063K	244.1G	38.04 / 0.9605	33.66 / 0.9181	**32.21** / 0.9000	32.23 / 0.9294	38.66 / 0.9772
MADNet [19]		878K	187.1G	37.85 / 0.9600	33.38 / 0.9161	32.04 / 0.8979	31.62 / 0.9233	-
RFDN [16]		534K	95.0G	38.05 / 0.9606	33.68 / 0.9184	32.16 / 0.8994	32.12 / 0.9278	38.88 / 0.9773
SMSR [30]		985K	351.5G	38.00 / 0.9601	33.64 / 0.9179	32.17 / 0.8990	32.19 / 0.9284	38.76 / 0.9771
LAPAR-A [31]		548K	171.0G	38.01 / 0.9605	33.62 / 0.9183	32.19 / 0.8999	32.10 / 0.9283	38.67 / 0.9772
DRSAN-48s [23]		650K	150.0G	**38.08** / 0.9609	33.62 / 0.9175	32.19 / **0.9002**	32.16 / 0.9286	-
HPUN-M [20]		492K	106.2G	38.03 / 0.9604	33.60 / 0.9185	32.20 / 0.9000	32.09 / 0.9282	38.83 / 0.9775
LatticeNet [14]		756K	169.5G	38.06 / 0.9607	33.70 / 0.9187	32.20 / 0.8999	32.25 / 0.9288	-
AFAN-M [21]		682K	163.4G	37.99 / 0.9605	33.57 / 0.9175	32.14 / 0.8994	32.08 / 0.9277	38.58 / 0.9769
SFFN [12]		912K	138.7G	38.02 / 0.9606	33.59 / 0.9177	32.20 / 0.9000	32.34 / 0.9298	-
LCRCA [11]		813K	186.0G	38.05 / 0.9607	33.65 / 0.9181	32.17 / 0.8994	32.19 / 0.9285	-
DIIN (Ours)		726K	**92.1G**	38.06 / **0.9610**	**33.73** / **0.9189**	32.20 / 0.8998	**32.37** / **0.9301**	**38.95** / **0.9778**
IDN [15]		553K	56.3G	34.11 / 0.9253	29.99 / 0.8354	28.95 / 0.8013	27.42 / 0.8359	32.71 / 0.9381
IMDN [10]		703K	71.5G	34.36 / 0.9270	30.32 / 0.8417	29.09 / 0.8046	28.17 / 0.8519	33.61 / 0.9445
AWSRN-M [29]		1,143K	116.6G	34.42 / 0.9275	30.32 / 0.8419	**29.13** / 0.8059	28.26 / 0.8545	33.64 / 0.9450
MADNet [19]		930K	88.4G	34.16 / 0.9253	30.21 / 0.8398	28.98 / 0.8023	27.77 / 0.8439	-
RFDN [16]		541K	42.2G	34.41 / 0.9273	30.34 / 0.8420	29.09 / 0.8050	28.21 / 0.8525	33.67 / 0.9449
SMSR [30]		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 [31]		594K	114.0G	34.36 / 0.9267	30.34 / 0.8421	29.11 / 0.8054	28.15 / 0.8523	33.51 / 0.9441
DRSAN-48s [23]		750K	78.0G	34.47 / 0.9274	30.35 / 0.8422	29.11 / 0.8060	28.26 / 0.8542	-
HPUN-M [20]		500K	48.1G	34.39 / 0.9269	30.33 / 0.8420	29.11 / 0.8052	28.06 / 0.8508	33.54 / 0.9441
LatticeNet [14]		765K	76.3G	34.40 / 0.9272	30.32 / 0.8416	29.10 / 0.8049	28.19 / 0.8513	-
AFAN-M [21]		681K	80.8G	34.35 / 0.9263	30.31 / 0.8423	29.06 / 0.8053	28.11 / 0.8522	33.44 / 0.9440
SFFN [12]		916K	69.4G	34.42 / 0.9274	30.34 / 0.8419	29.11 / 0.8055	28.26 / 0.8543	-
LCRCA [11]		822K	83.6G	34.40 / 0.9269	30.36 / 0.8422	29.09 / 0.8049	28.21 / 0.8532	-
DIIN (Ours)		735K	**41.8G**	**34.48** / **0.9280**	**30.44** / **0.8436**	**29.13** / **0.8062**	**28.35** / **0.8551**	**33.87** / **0.9461**
IDN [15]		553K	32.3G	31.82 / 0.8903	28.25 / 0.7730	27.41 / 0.7297	25.41 / 0.7632	29.41 / 0.8942
IMDN [10]		715K	40.9G	32.21 / 0.8948	28.58 / 0.7811	27.56 / 0.7353	26.04 / 0.7838	30.45 / 0.9075
AWSRN-M [29]		1,254K	72.0G	32.21 / 0.8954	28.65 / 0.7832	27.60 / 0.7368	26.15 / 0.7884	30.56 / 0.9093
MADNet [19]		1,002K	54.1G	31.95 / 0.8917	28.44 / 0.7780	27.47 / 0.7327	25.76 / 0.7746	-
RFDN [16]		550K	23.9G	32.24 / 0.8952	28.61 / 0.7819	27.57 / 0.7360	26.11 / 0.7858	30.58 / 0.9089
SMSR [30]		1,006K	89.1G	32.12 / 0.8932	28.55 / 0.7808	27.55 / 0.7351	26.11 / 0.7868	30.54 / 0.9085
LAPAR-A [31]		659K	94.0G	32.15 / 0.8944	28.61 / 0.7818	27.61 / 0.7366	26.14 / 0.7871	30.42 / 0.9074
DRSAN-48s [23]		730K	57.6G	32.25 / 0.8945	28.55 / 0.7817	27.59 / 0.7374	26.14 / 0.7875	-
HPUN-M [20]		511K	27.7G	32.19 / 0.8946	28.61 / 0.7818	27.58 / 0.7364	26.04 / 0.7851	30.49 / 0.9078
LatticeNet [14]		777K	43.6G	32.18 / 0.8943	28.61 / 0.7812	27.57 / 0.7355	26.14 / 0.7844	-
AFAN-M [21]		692K	50.9G	32.18 / 0.8939	28.62 / 0.7826	27.58 / 0.7373	26.13 / 0.7876	30.45 / 0.9085
SFFN [12]		923K	34.6G	32.23 / 0.8950	28.58 / 0.7813	27.56 / 0.7361	26.15 / 0.7877	-
LCRCA [11]		834K	47.7G	32.20 / 0.8948	28.60 / 0.7807	27.57 / 0.7653	26.10 / 0.7851	-
DIIN (Ours)		747K	**24.2G**	**32.35** / **0.8963**	**28.73** / **0.7842**	**27.63** / **0.7378**	**26.35** / **0.7920**	**30.81** / **0.9119**

TABLE II

COMPARISONS WITH SOME TRANSFORMER-BASED METHODS. DIIN CAN ACHIEVE COMPETITIVE RESULTS WITH FEWER MULTIADDS

Methods	Params	Multi-Adds	Set5	Set14	BSD100	Urban100	Manga109	Average
SwinIR [25]	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 [26]	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 [8]	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
DIIN (ours)	747K	**24.2G**	32.35/0.8963	28.73/0.7842	27.63/0.7378	26.35/0.7920	30.81/0.9119	29.17/0.8244

C. Comparison With State-of-the-Arts

In Table I, we compare DIIN with several lightweight SISR models, including VDSR [6], IDN [15], CARN [9], IMDN [10], AWSRN-M [29], MADNet [19], RFDN [16], SMSR [30], LAPAR-A [31], DRSAN-48s [23], HPUN-M [20], LatticeNet [14], AFAN-M [21], SFFN [12], and LCRCA [11]. These models have shown promising results in lightweight SISR tasks. From the table, it can be observed that DIIN achieves competitive performance. Moreover, compared to these models, DIIN exhibits lower computational cost and a moderate number of parameters, demonstrating the effectiveness of our approach. Specifically, on the Manga109 dataset, DIIN achieves an average PSNR improvement over RFDN of 0.07, 0.20, and 0.23 dB for different scaling factors. In the ×4 SR task, DIIN outperforms several other models on all datasets while maintaining a reasonable computational cost, striking a good balance between performance and efficiency.

Recently, several SISR methods based on Transformers have emerged. To compare our DIIN with these recent approaches, we conduct a detailed evaluation and comparison with SwinIR [25], ESRT [26], and LBNet [8]. The results are presented in Table II. From the table, it can be observed that DIIN requires fewer calculations compared to the aforementioned three methods, making it more competitive

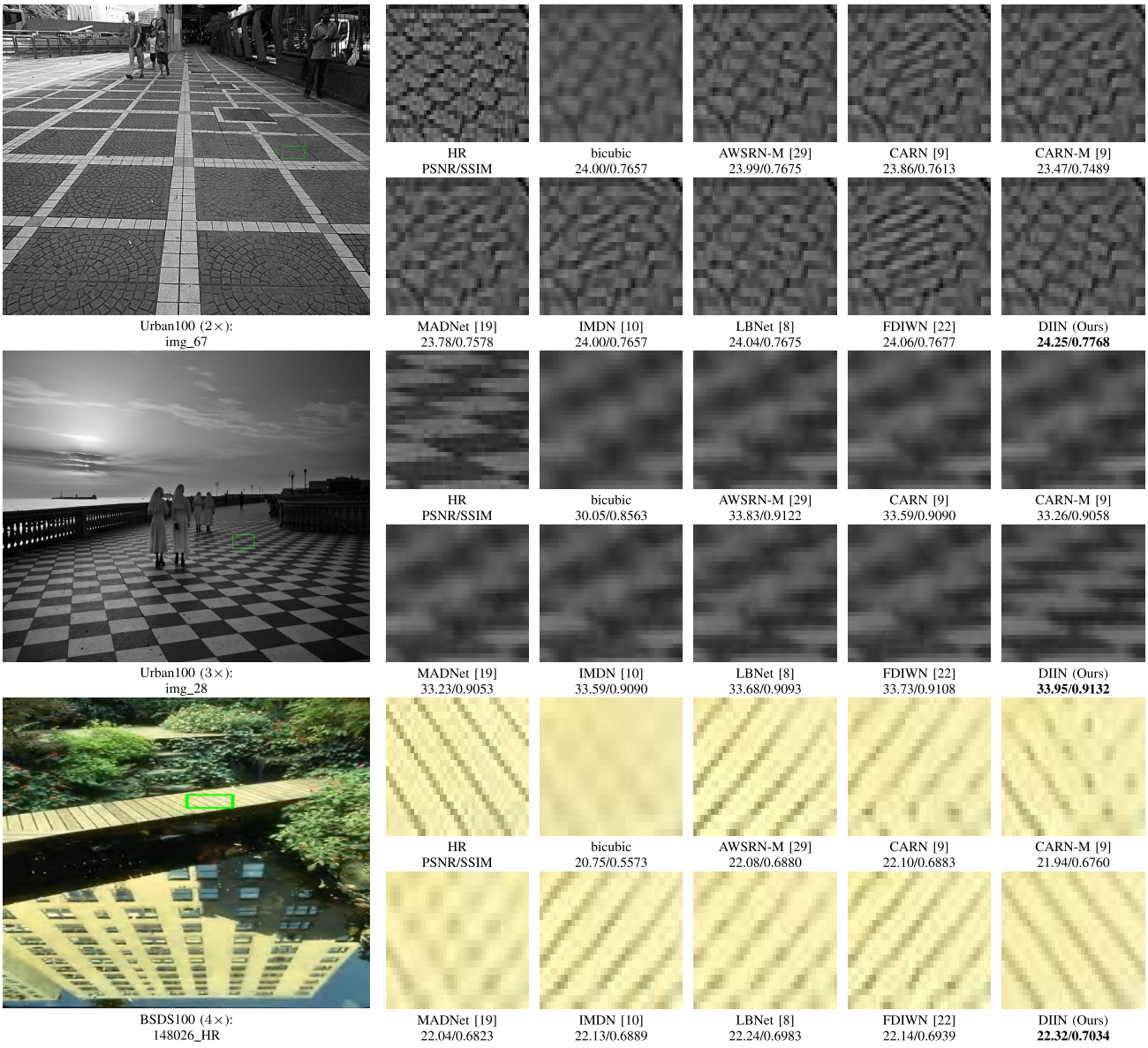


Fig. 5. Visual comparisons of DIIN with other SR methods on BSDS100 and Urban100 datasets.

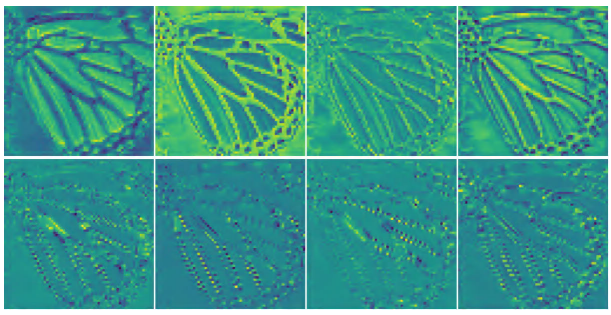


Fig. 6. Visualization of upper (CNN) branch and lower (transformer) branch feature maps.

in terms of computational efficiency. Regarding performance, DIIN achieves a lower PSNR value than SwinIR, but higher values than ESRT and LBNet. Similarly, DIIN demonstrates a higher SSIM value than LBNet and comparable performance to ESRT, albeit lower than that of SwinIR. It is worth noting that SwinIR utilizes an additional dataset (Flickr2K) for training, which may contribute to its superior model performance. Overall, these observations further validate the effectiveness of DIIN.

TABLE III
STUDY OF RECURSIVE TIMES ON SET5 DATASET (×4)

Method	Params	Multi-Adds	Running time	PSNR/SSIM
RE-0	747K	14.0G	0.0231s	32.15/0.8939
RE-1	747K	24.2G	0.0424s	**32.35/0.8963**
RE-2	747K	34.4G	0.0799s	32.33/0.8961

Additionally, we provide visual comparisons between DIIN and other lightweight SISR models in Fig. 5. Specifically, in img_067 (×2), DIIN successfully restores the ground texture, closely resembling the HR image. In_img 28 (×3), although the SR image generated by DIIN is slightly blurred, it effectively restores the line details. In img_148026 (×4), DIIN accurately preserves the direction of the lines and enhances the image clarity and details, while the SR images

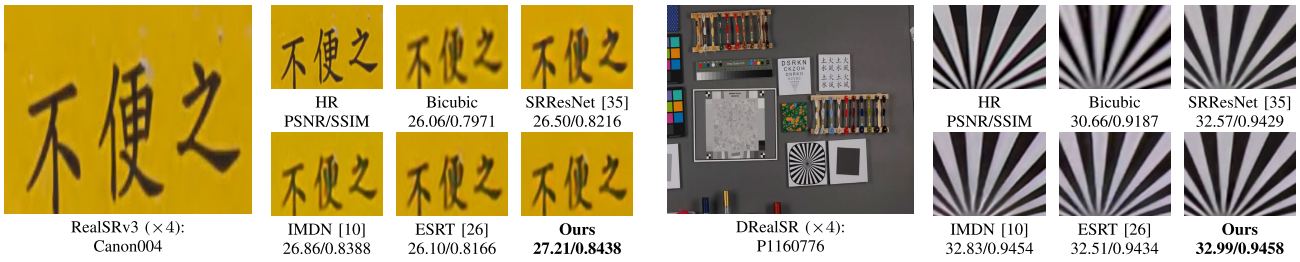


Fig. 7. Visual comparisons on real-world datasets (including RealSRv3 and DRealSR).

TABLE IV
STUDY OF DIFFERENT MODULES IN THE DUAL-BRANCH STRUCTURE ON URBAN100 DATASET (×4)

CNN	Trans	Interaction	Params	Multi-Adds	PSNR/SSIM
✗	✓	✗	542K	4.83G	25.80/0.7747
✗	✓	✓	543K	4.83G	25.99/0.7809
✓	✗	✗	489K	43.61G	26.07/0.7861
✓	✗	✓	493K	43.61G	26.17/0.7885
✓	✓	✗	744K	24.22G	26.13/0.7857
✓	✓	✓	747K	24.22G	**26.35/0.7920**

TABLE V
PERFORMANCE COMPARISONS OF DFIM WITH OTHER BASIC MODULES ON MANGA109 DATASET (×4)

Method	Params	Multi-Adds	PSNR/SSIM
DIIN+IMDB	577K	9.10G	30.31/0.9055
DIIN+RFDB	603K	12.2G	30.43/0.9074
DIIN+DFIM	747K	24.2G	**30.81/0.9119**

TABLE VI
QUANTITATIVE COMPARISONS ON REAL-WORLD DATASETS

Scale	Methods	RealSRv3	DRealSR
		PSNR↑ / SSIM↑ / LPIPS↓	PSNR↑ / SSIM↑ / LPIPS↓
×4	SRResNet [35]	27.70 / 0.7788 / 0.4010	31.18 / 0.8737 / 0.3618
	IMDN [10]	27.76 / 0.7834 / 0.3766	31.30 / 0.8764 / 0.3411
	ESRT [26]	27.61 / 0.7788 / 0.3895	31.17 / 0.8737 / 0.3556
	DIIN (Ours)	**27.77 / 0.7845 / 0.3743**	**31.63 / 0.8793 / 0.3408**

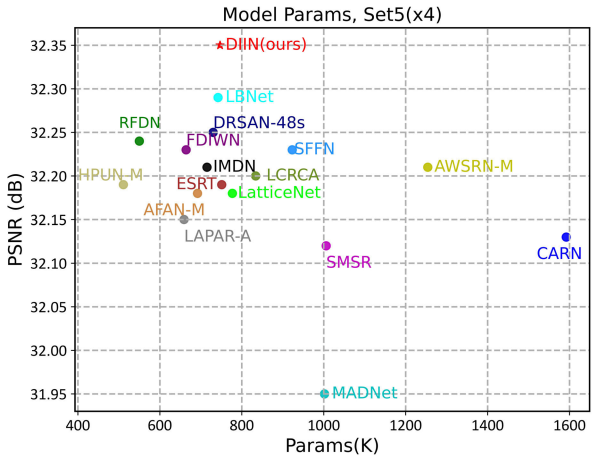


Fig. 8. Model parameters study on Set5 dataset (×4).

generated by other models appear either blurry or incorrectly restored. Overall, the SR images generated by DIIN are more similar to the ground truth compared to other methods.

D. Ablation Studies

1) Recursive Investigations: To utilize our designed modules without increasing the model parameters, we incorporated recursive mechanisms in both the CNN branch and the Transformer branch. To assess the effectiveness of these recursive mechanisms, we conducted several research experiments with varying recursion times. The results are presented in Table III, where RE-*N* denotes the module recursively applied *N* times, and RE-0 indicates that the recursive mechanism was not employed. As expected, the model's performance improves as the number of recursive times increases. However, when the number of recursions reaches 2 (RE-2), compared to the significant increase in computational cost, the performance is accidentally reduced. Consequently, we determined that setting the number of recursive times to one strikes a better balance between performance, computational cost, and execution time.

2) Dual-Branch Structure Investigations: The proposed dual-branch structure consists of two branches, with the upper branch corresponding to the CNN module and the lower branch corresponding to the TM. We conducted a series of experiments to evaluate the effectiveness of this design in consistently improving performance. Additionally, to visually observe the feature changes inside the model, we separately visualize the output of the CNN branch and the Transformer branch, as shown in Fig. 6. It is apparent from these visualizations that the feature maps in the CNN branch contain more color and texture information, while the feature maps in the Transformer branch contain more edge and contour features. By combining these two types of feature information, we can interactively amalgamate the complementary local texture and global edge information, ultimately leading to the production of higher quality SR images. Table IV displays the SISR results of these dual-branch structures under different conditions. Despite the increase in parameters caused by the introduction of the CNN and TMs in the dual branches, there is a significant improvement in performance. This improvement is substantial and fully demonstrates the effectiveness of the designed dual-branch structure.

To evaluate the effectiveness of the interaction scheme (CC) used in our design, we also conducted an additional experiment by removing this scheme from the model (as depicted in the

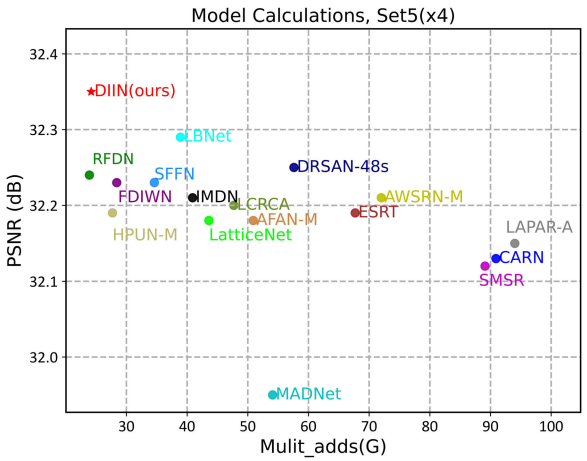


Fig. 9. Model multiadds study on Set5 dataset (×4).

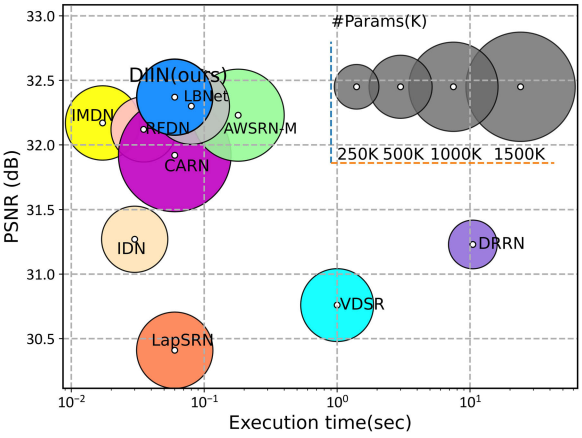


Fig. 10. Model complexity study on Set5 dataset (×4).

third column in Table IV). Despite the fact that there is not much difference in the number of parameters and calculations, there was a noticeable decrease in both PSNR and SSIM performance metrics. This finding shows that the interaction scheme (CC) in DIIN plays an important role in improving performance.

Furthermore, we assessed the effectiveness of DFIM by replacing it with commonly used feature extraction modules in lightweight SISR models, such as IMDB [10] and RFDB [16]. As presented in Table V, DFIM does increase the network's parameters and computational costs compared to IMDB and RFDB. However, substantial performance improvements are observed. We believe that the increase in computational costs is reasonable, considering the significant benefits gained from using DFIM as an effective feature extraction module.

3) Real-World Image Super-Resolution: To demonstrate the generalization and effectiveness of our model, we compare DIIN with some classic lightweight SR models, namely SRResNet [35], IMDN [10], and ESRT [26], using the RealSR [42] dataset. We retrain these models on the RealSR dataset to ensure fairness in comparison. Both DIIN and the compared models are trained uniformly on the RealSR dataset, and a standardized × 4 test is conducted. The hyperparameters remain consistent with those used during DIV2K training. The results are presented in Table VI. As indicated in the table, DIIN outperforms all other methods across all three metrics, exhibiting a significant margin over the second-best approach. Additionally, visual comparison charts in Fig. 7 demonstrate that DIIN effectively captures fine texture details. These experiments highlight the applicability of our proposed model and its ability to deliver excellent SR performance in real-world scenarios.

E. Model Complexity Studies

Table I illustrates the effective balance between model size and performance achieved by our model. Moreover, the execution time of a model serves as a crucial indicator of its complexity. To compare our DIIN with other SISR methods, parameter comparison is presented in Fig. 8, while multiadds comparison is displayed in Fig. 9. Additionally, to facilitate comparative visualization with other models, we also provide comparisons of the number of parameters, execution time, and model performance in Fig. 10. It can be observed that our DIIN delivers considerable PSNR results while maintaining a relatively balanced configuration in terms of parameters and execution time. These findings further affirm that DIIN is a lightweight and efficient SISR model.

V. Conclusion

This article presents a lightweight DIIN for efficient image SR. DIIN incorporates two branches, each serving a specific purpose. In one branch, an efficient symmetric CNN-based model is employed to extract local information. Meanwhile, the other branch utilizes a recursive Transformer to capture long-term dependencies in images. In the CNN branch, we introduce a DFIM and a MFEU to extract more representative feature information. Additionally, in the Transformer branch, we leverage a recursive mechanism to effectively refine global information without increasing the number of parameters. In summary, DIIN adopts a dual-branch approach to extract and combine both local and global information, achieving a better balance between model size, execution time, and performance in the context of efficient image SR.

References

[1] W.-Y. Hsu and P.-W. Jian, "Detail-enhanced wavelet residual network for single image super-resolution," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–13, 2022.

[2] A. S. Tomar, K. Arya, and S. S. Rajput, "Deep hyfeat based attention in attention model for face super-resolution," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, 2023.

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

[4] J. Li et al., "A systematic survey of deep learning-based single-image super-resolution," *ACM Comput. Surveys*, vol. 56, no. 10, pp. 1–40, Oct. 2024.

[5] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[6] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646–1654.

[7] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 286–301.

[8] G. Gao, Z. Wang, J. Li, W. Li, Y. Yu, and T. Zeng, "Lightweight bimodal network for single-image super-resolution via symmetric CNN and recursive transformer," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 913–919.

[9] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 252–268.

[10] Z. Hui, X. Gao, Y. Yang, and X. Wang, "Lightweight image super-resolution with information multi-distillation network," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 2024–2032.

[11] C. Peng, P. Shu, X. Huang, Z. Fu, and X. Li, "LCRCA: Image super-resolution using lightweight concatenated residual channel attention networks," *Int. J. Speech Technol.*, vol. 52, no. 9, pp. 10045–10059, Jul. 2022.

[12] Z. Wang, Y. Liu, R. Zhu, W. Yang, and Q. Liao, "Lightweight single image super-resolution with similar feature fusion block," *IEEE Access*, vol. 10, pp. 30974–30981, 2022.

[13] J. Chao et al., "Equivalent transformation and dual stream network construction for mobile image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14102–14111.

[14] X. Luo, Y. Qu, Y. Xie, Y. Zhang, C. Li, and Y. Fu, "Lattice network for lightweight image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4826–4842, Apr. 2023.

[15] Z. Hui, X. Wang, and X. Gao, "Fast and accurate single image super-resolution via information distillation network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 723–731.

[16] J. Liu, J. Tang, and G. Wu, "Residual feature distillation network for lightweight image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, Jan. 2020, pp. 41–55.

[17] D. Song, C. Xu, X. Jia, Y. Chen, C. Xu, and Y. Wang, "Efficient residual dense block search for image super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, 2020, pp. 12007–12014.

[18] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 1997–2017, 2019.

[19] R. Lan, L. Sun, Z. Liu, H. Lu, C. Pang, and X. Luo, "MADNet: A fast and lightweight network for single-image super resolution," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1443–1453, Mar. 2021.

[20] B. Sun, Y. Zhang, S. Jiang, and Y. Fu, "Hybrid pixel-unshuffled network for lightweight image super-resolution," 2022, *arXiv:2203.08921*.

[21] L. Wang, K. Li, J. Tang, and Y. Liang, "Image super-resolution via lightweight attention-directed feature aggregation network," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 19, no. 2, pp. 1–23, May 2023.

[22] G. Gao, W. Li, J. Li, F. Wu, H. Lu, and Y. Yu, "Feature distillation interaction weighting network for lightweight image super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 1, pp. 661–669.

[23] K. Park, J. W. Soh, and N. I. Cho, "A dynamic residual self-attention network for lightweight single image super-resolution," *IEEE Trans. Multimedia*, vol. 25, pp. 907–918, 2023.

[24] H. Chen et al., "Pre-trained image processing transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12299–12310.

[25] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "SwinIR: Image restoration using Swin transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1833–1844.

[26] Z. Lu, J. Li, H. Liu, C. Huang, L. Zhang, and T. Zeng, "Transformer for single image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 457–466.

[27] W. Li et al., "Cross-receptive focused inference network for lightweight image super-resolution," *IEEE Trans. Multimedia*, vol. 26, pp. 1–13, 2023.

[28] Z. Chen, Y. Zhang, J. Gu, L. Kong, X. Yang, and F. Yu, "Dual aggregation transformer for image super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Sep. 2023, pp. 1–24.

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

[30] L. Wang et al., "Exploring sparsity in image super-resolution for efficient inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4917–4926.

[31] W. Li, K. Zhou, L. Qi, N. Jiang, J. Lu, and J. Jia, "LAPAR: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 20343–20355.

[32] X. Luo, "LatticeNet: Towards lightweight image super-resolution with lattice block," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 272–289.

[33] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[34] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–24.

[35] C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.

[36] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. 23rd Brit. Mach. Vis. Conf. (BMVC)*, vol. 135, 2012, p. 135.

[37] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. Curves Surf.*, 2012, pp. 711–730.

[38] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 5197–5206.

[39] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, Jul. 2001, pp. 416–423.

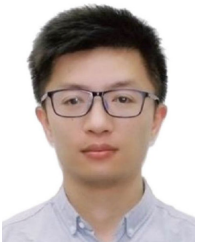
[40] Y. Matsui et al., "Sketch-based Manga retrieval using manga109 dataset," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 21811–21838, Oct. 2017.

[41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[42] J. Cai, H. Zeng, H. Yong, Z. Cao, and L. Zhang, "Toward real-world single image super-resolution: A new benchmark and a new model," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3086–3095.

Haonan Jin received the B.S. degree in automation from Yancheng Institute of Technology, Jiangsu, China, in 2021, and the M.S. degree from the College of Automation, College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing, China, in 2024.

His research interests include lightweight image super-resolution.

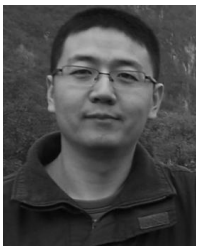


Guangwei Gao (Senior Member, IEEE) received the Ph.D. degree in pattern recognition and intelligence systems from Nanjing University of Science and Technology, Nanjing, China, in 2014.

He was a Visiting Student with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, in 2011 and 2013, respectively. From 2019 to 2021, he was a Project Researcher with the National Institute of Informatics, Tokyo, Japan. He is currently an Associate Professor with the Institute of Advanced Technology, Nanjing University of Posts and Telecommunications. He has published more than 70 scientific papers in IEEE TRANSACTIONS ON IMAGE PROCESSING/IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY/IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS/IEEE TRANSACTIONS ON MULTIMEDIA/IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, *ACM Transactions on Internet Technology/ACM Transactions on Multimedia Computing, Communications, and Applications*, PR, AAAI, and *International Journal of Computing and Artificial Intelligence*. His research interests include pattern recognition and computer vision. Personal website: *https://guangweigao.github.io*.

Juncheng Li received the Ph.D. degree in computer science and technology from East China Normal University, Shanghai, China, in 2021.

He was a Post-Doctoral Fellow with the Center for Mathematical Artificial Intelligence (CMAI), The Chinese University of Hong Kong, Hong Kong. He is currently an Assistant Professor with the School of Communication and Information Engineering, Shanghai University, Shanghai. He has published more than 40 scientific papers in the *ACM Computing Survey*, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON MEDICAL IMAGING, CVPR, ICCV, ECCV, AAAI, and IJCAI. His main research interests include image restoration, computer vision, and medical image processing.



Zhenhua Guo received the Ph.D. degree in computer science from The Hong Kong Polytechnic University, Hong Kong, in 2010.

He was a Visiting Scholar of electrical and computer engineering (ECE) with Carnegie Mellon University, Pittsburgh, PA, USA, from 2018 to 2019. Since September 2022, he has been working with Tianyijiaotong Technology Ltd., Suzhou, China. His research interests include pattern recognition, computer vision, biometrics, and object detection.

Yi Yu (Senior Member, IEEE) received the Ph.D. degree in information and computer science from Nara Women's University, Nara, Japan, in 2009.

She is an Associate Professor with the Graduate School of Advanced Science and Engineering, Hiroshima University, Higashihiroshima, Japan. She was previously an Assistant Professor with the National Institute of Informatics, Tokyo, Japan. She is a Senior Research Fellow with the School of Computing, National University of Singapore, Singapore. She has supervised more than 60 students, including postdoctoral researchers, Ph.D. students, and master students. She has published more than 100 high-quality papers. Her research interests include interdisciplinary study of multimedia content understanding and artificial intelligence, particularly in multimodal representation learning, generative modeling, and multimodal information fusion for multimedia and music.

Dr. Yu and her team have received several awards, including the Best Poster Award from IEEE ISM 2018, the Best Paper Runner-Up at APWeb-WAIM 2017, and recognition as a Finalist for the World's FIRST 10K Best Paper Award at ICME 2017. They also won the Second Prize in the Yahoo Flickr Grand Challenge 2015, were among the top winners (out of 29 teams) at the ACM SIGSPATIAL GIS Cup 2013, and received the Best Paper Award from IEEE ISM 2012. She has served as the Co-Chair for the IEEE ICDM Ph.D. Forum in both 2018 and 2019, is an Associate Editor for IEEE TRANSACTIONS ON MULTIMEDIA.