Efficient Image Super-Resolution With Feature Interaction Weighted Hybrid Network

Wenjie Li, Juncheng Li, Guangwei Gao, *Senior Member, IEEE*, Weihong Deng, *Senior Member, IEEE*, Jian Yang, *Member, IEEE*, Guo-Jun Qi, *Fellow, IEEE*, and Chia-Wen Lin, *Fellow, IEEE*

***Abstract*—Lightweight image super-resolution aims to reconstruct high-resolution images from low-resolution images using low computational costs. However, existing methods result in the loss of middle-layer features due to activation functions. To minimize the impact of intermediate feature loss on reconstruction quality, we propose a Feature Interaction Weighted Hybrid Network (FIWHN), which comprises a series of Wide-residual Distillation Interaction Block (WDIB) as the backbone. Every third WDIB forms a Feature Shuffle Weighted Group (FSWG) by applying mutual information shuffle and fusion. Moreover, to mitigate the negative effects of intermediate feature loss, we introduce Wide Residual Weighting units within WDIB. These units effectively fuse features of varying levels of detail through a Wide-residual Distillation Connection (WRDC) and a Self-Calibrating Fusion (SCF). To compensate for global feature deficiencies, we incorporate a Transformer and explore a novel architecture to combine CNN and Transformer. We show that our FIWHN achieves a favorable balance between performance and efficiency through extensive experiments on low-level and high-level tasks.**

Received 29 December 2023; revised 23 March 2024 and 31 July 2024; accepted 10 September 2024. Date of publication 24 December 2024; date of current version 12 May 2025. This work was supported in part by the Foundation of Key Laboratory of Artificial Intelligence of Ministry of Education under Grant AI202404 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 of this article and approving it for publication was Dr. Symeon Papadopoulos. *(Corresponding author: Guangwei Gao.)*

Wenjie Li and Weihong Deng are with the Pattern Recognition and Intelligent System Laboratory, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100080, China (e-mail: lewj2408@gmail.com; whdeng@bupt.edu.cn).

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

Guangwei Gao is with the IVIPLab, Institute of Advanced Technology, Nanjing University of Posts and Telecommunications, Nanjing 210046, 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: csggao@gmail.com).

Jian Yang is with the School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: csjyang@njust.edu.cn).

Guo-Jun Qi is with the Research Center for Industries of the Future and the School of Engineering, Westlake University, Hangzhou 310024, China, and also with the OPPO Research, Seattle, WA 98101 USA (e-mail: guojunq@gmail.com).

Chia-Wen Lin is with the Department of Electrical Engineering and the Institute of Communications Engneering, National Tsing Hua University, Hsinchu 300044, Taiwan (e-mail: cwlin@ee.nthu.edu.tw).

Codes will be available at https://github.com/IVIPLab/FIWHN.

Digital Object Identifier 10.1109/TMM.2024.3521753

***Index Terms*—Single-image super-resolution, wide-residual distillation interaction, hybrid network, transformer.**

I. Introduction

SINGLE image super-resolution (SISR) has gained increasing attention due to the demand for high-resolution in various computer vision applications, including medical image analysis, security surveillance [1], and autonomous driving. However, SISR remains a challenging problem as it involves solving an inverse problem. The advent of deep neural networks has significantly advanced the field of SISR. One of the pioneering works, SRCNN [2], employed a three-layer convolutional network and outperformed traditional methods significantly. Then VDSR [3] increased the model depth to 20 layers and achieved better performance. Subsequently, several approaches [4] utilized deeper networks to improve the performance of SISR. However, these methods have a high computational overhead, making them unsuitable for practical devices with limited computational resources.

To reduce the model size, many existing approaches have focused on designing efficient model structures, which include weight sharing [5], multi-scale structures [6], strategies for neural structure search [7], grouped convolution [8]. However, existing approaches often overlook the loss of intermediate information caused by activation functions like ReLU. This issue has been demonstrated by MobileNetV2 [9], where the reduction in intermediate information during increases in network depth can negatively affect the quality of image reconstruction. We propose the Feature Interaction Weighted Hybrid Network (FIWHN) to address this concern while maintaining lightweight models. Specifically, our Convolutional Neural Network (CNN) part incorporates wide-residual attention-weighted units, which consist of Wide Identical Residual Weight (WIRW) and Wide Convolutional Residual Weighting (WCRW). These units help compensate for the lost intermediate features by obtaining a broader feature map before applying the activation function. Additionally, we adopt Wide-residual Distillation Interaction Blocks (WDIB) with a lattice structure [10]. The WDIB includes two paired skip connections and adaptive combinations of wide residual blocks that utilize attention-based connection weights. As a result, we achieve a compact network with strong expressive power. The Wide-Residual Distillation Connection (WRDC) framework and the Self-Calibrating Fusion (SCF) unit facilitate the distillation and fusion of split features from different classes within the WDIB, thereby enhancing its

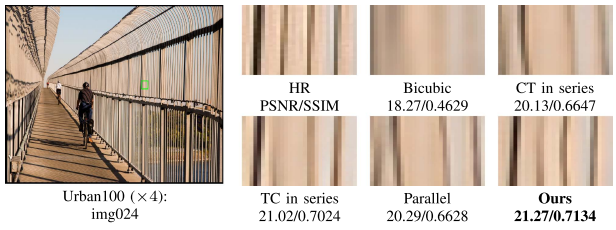


Fig. 1. Comparisons of different interaction schemes between the CNN and transformer. "CT in series" is shown in Fig. 5(a), denoting the series connection of CNN with transformer; "TC in series" is shown in Fig. 5(b), denoting the series connection of transformer with CNN; "Parallel" is shown in Fig. 5(c), denoting the parallel connection between CNN and transformer; "Ours" is shown in Fig. 5(d), denoting the potential interaction between CNN and transformer.

generalization capability. Multiple WDIBs are combined to form a Feature Shuffle Weighted Group (FSWG), which leverages information from the middle layers at the group level through blending, fusion, and weighting of the output features from each WDIB.

Recently, Transformer-based methods have demonstrated great potential for SISR. For instance, SwinIR-light [11] leveraged the Transformer's ability to model long-range dependencies by employing a sliding window mechanism to address the issue of uncorrelated edges between image patches. Hybrid networks [12], [13], [14], [15] that combine CNN and Transformer have also exhibited advantages over pure CNN or pure Transformer. Therefore, in our method, we also introduce the Transformer to enable effective long-range modeling. By combining CNN and Transformer, the weights of these models can be adjusted based on the information extracted from each other during the training process. However, existing methods, represented in Fig. 5(a) and (b), often struggle to effectively integrate local and global information flow, resulting in the generation of ambiguous artifacts, as shown in Fig. 1. To address this limitation, we propose an improved approach that combines CNN and Transformer. Our goal is to facilitate a stronger interaction between the features extracted by both networks, thus enhancing the overall performance.

In summary, the main contributions are listed as follows:

- We propose wide-residual weighting units for SISR, which consist of WIRW and WCRW. They effectively mitigate the negative impact of intermediate feature loss by incorporating a wide residual mechanism.
- We introduce WRDC, which enhances information flow by leapfrogging features at different levels within the WDIB. Additionally, we propose a SCF, allowing for a more precise and adaptive feature combination.
- We present a novel interaction framework that enhances communication between different levels of features. This includes the FSWG in the CNN part and an interaction framework between CNN and Transformers.

In this work, we have expanded on the following aspects compared to our conference version paper [5]:

- To address the issue of missing global features, we incorporate the Transformer component by designing a novel combinatorial framework that promotes better information flow, which outperforms existing frameworks.
- We conduct comprehensive experiments covering both real-world as well as segmentation scenarios, and the results consistently support the competitiveness of our proposed method in the SISR domain.

II. Related Work

A. Lightweight SISR Models

To make SISR applicable to real-world applications with limited computational resources, there has been significant research focused on developing lightweight SISR models. Existing research in this area can be broadly categorized into three main approaches: efficient model structure design-based methods [5], [10], [16], pruning or quantification techniques-based methods [17], and knowledge distillation-based methods [18]. Efficient model structure design primarily involves designing model architectures specifically tailored for lightweight SISR. For instance, some models adopt recursive cascading to learn feature representations across different layers [8], while others reuse intermediate layer features through recursive learning [5]. CFSR [19] utilized large kernel convolutions to capture long-range features. Strategies like channel splitting and hierarchical distillation have also been explored in models like IMDN [16] to extract features at different levels. Additionally, applying neural architecture search (NAS) in SISR, as demonstrated by FALSR [7], has introduced new possibilities for structural design-based methods. Knowledge distillation methods leverage the knowledge transfer from pre-trained large teacher models to smaller student models to improve their performance [18]. Pruning or quantization techniques aim to reduce the model size without sacrificing accuracy [17]. Our FIWHN belongs to the category of efficient structure design-based method, which investigates the network structure in terms of inter-block design, aiming to efficiently combine our units to achieve a lightweight model.

B. Wide-Residual Weighting Learning

Numerous studies [3], [4] have investigated the relationship between network depth and performance in deep learning models. Initially, it was believed that deeper networks would lead to better performance. For instance, VDSR [3] utilized a 20-layer network, and RCAN [4] went even further with a network depth exceeding 800 layers. However, subsequent research revealed that increasing network depth does not necessarily result in better performance and may lead to a decline. Studies, such as MobileNetV2 [9], have shown that the extensive use of activation functions in models can contribute to feature degradation. The widely-used ReLU function, in particular, can cause some neurons to become "dead", resulting in the loss of intermediate features. This degradation becomes more pronounced as the network grows deeper. Building upon the findings of WDSR [20], which demonstrated that models with wider features before the Relu activation layer can achieve better performance, we introduce adaptive multipliers that allow for the adaptive adjustment of wide-residual weighting units during training. Consequently, our proposed residual unit enables efficient feature extraction while maintaining a lightweight structure.

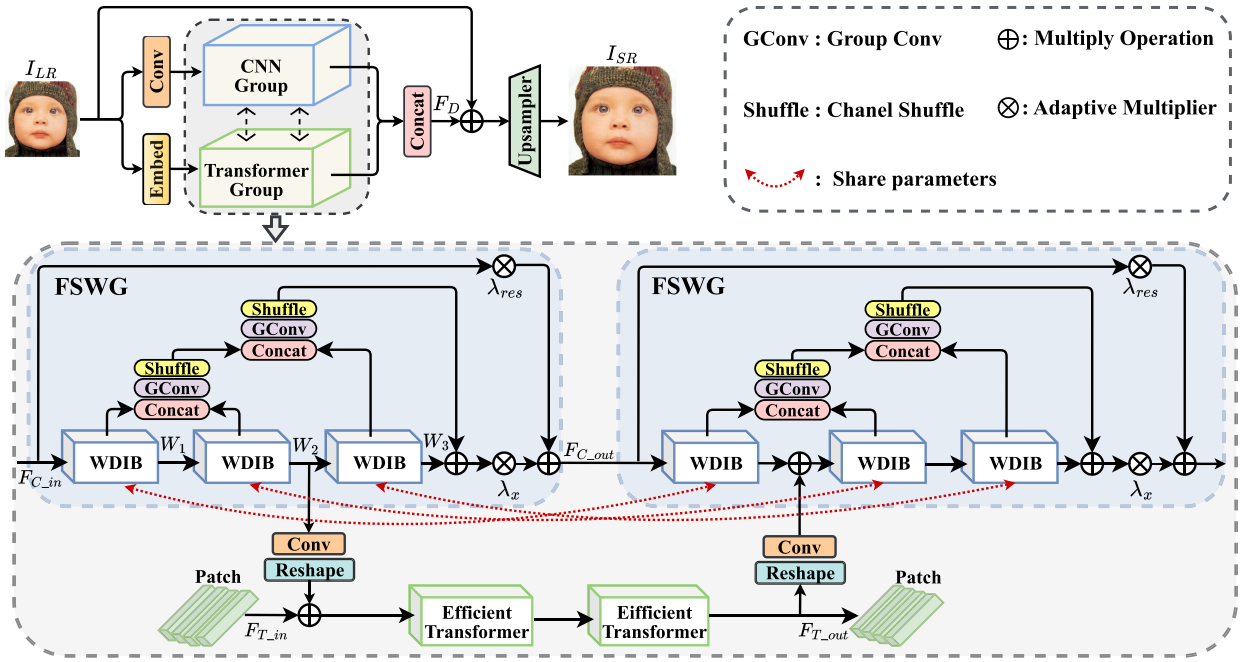


Fig. 2. Architecture of our proposed feature interaction weighted hybrid network (FIWHN).

C. Transformer-Based SISR Models

Transformers have recently demonstrated their remarkable capabilities in computer vision. As a result, Transformer-based approaches for SISR have gained significant attention. SwinIR-light [11] initially demonstrated state-of-the-art performance by introducing Transformer-based strategies to the SISR task. Building on this advancement, ESRT [12] and LBNet [13] combined lightweight CNN with lightweight Transformers in the SISR task, achieving a good balance across multiple evaluation metrics. ELAN [14] accelerates models by grouping multi-scale self-attentive schemes and attention-sharing mechanisms. SCTANet [21] and ELSFace [22] both proposed to improve performance by utilizing a parallel structure involving both CNNs and transformers. CFIN [15] and NGSwin [23] further enhanced performance by introducing the context to expand the perceptual field. However, unlike these previous works, we explore a more efficient structure of combining CNNs with Transformers to further improve the interaction ability of local and global features.

III. Proposed Method

In this section, we present an overview of FIWHN. First, we outline the general structure, which includes the backbone of the CNN and Transformer and their interaction. Next, we introduce our proposed WDIB. Finally, we provide details about the supervision functions used in training the model.

A. Feature Interaction Weighting Hybrid Network

Overview: As presented in Fig. 2, FIWHN consists of three main components: the dimensional transformation part, the interaction part between CNN and Transformer, and the upsampling part. In this setup, I_{LR} and I_{SR} represent the input LR image and the SR image, respectively.

First, the dimensional transformation part consists of a convolutional layer G_{Conv} and an embedding module G_{Embed}. The process can be expressed as

$$F_C = G_{Conv}(I_{LR}), F_T = G_{Embed}(I_{LR}). \tag{1}$$

Then, the respective outputs F_C and F_T are sent to the CNN and Transformer interaction stage:

$$F_D = G_{Concat}(CNN(F_C) \leftrightarrow Trans(F_T)), \tag{2}$$

where CNN refers to the CNN group, $Trans$ denotes the Transformer group, $\leftrightarrow$ represents the information exchange process between CNN and Transformer, and G_{Concat} denotes the concat operation. The depth feature F_D is then passed to the upsampling part $G_{Upsample}$ along with the input to complete the image reconstruction:

$$I_{SR} = G_{Upsample}(I_{LR} + F_D). \tag{3}$$

Feature Shuffle Weighted Group (FSWG): In FSWG, we have incorporated a feature shuffling and fusion mechanism to effectively combine, group, and shuffle features from different receiver domains. As depicted in Fig. 2, FSWG serves as the backbone component of the CNN part and consists of 3 interacting WDIBs. Specifically, we progressively blend and shuffle the output features of adjacent WDIB. This cascade operation, denoted as G_{CGS}, can be represented as:

$$G_{CGS} = G_{Shuffle}(G_{GConv}(G_{Concat}[x_i, x_{i+1}])), \tag{4}$$

where G_{GConv} represents the operation of group convolution, $G_{Shuffle}$ represents the operation of channel shuffle like ShuffleNet [24], x_i and x_{i+1} represent the output features of the two

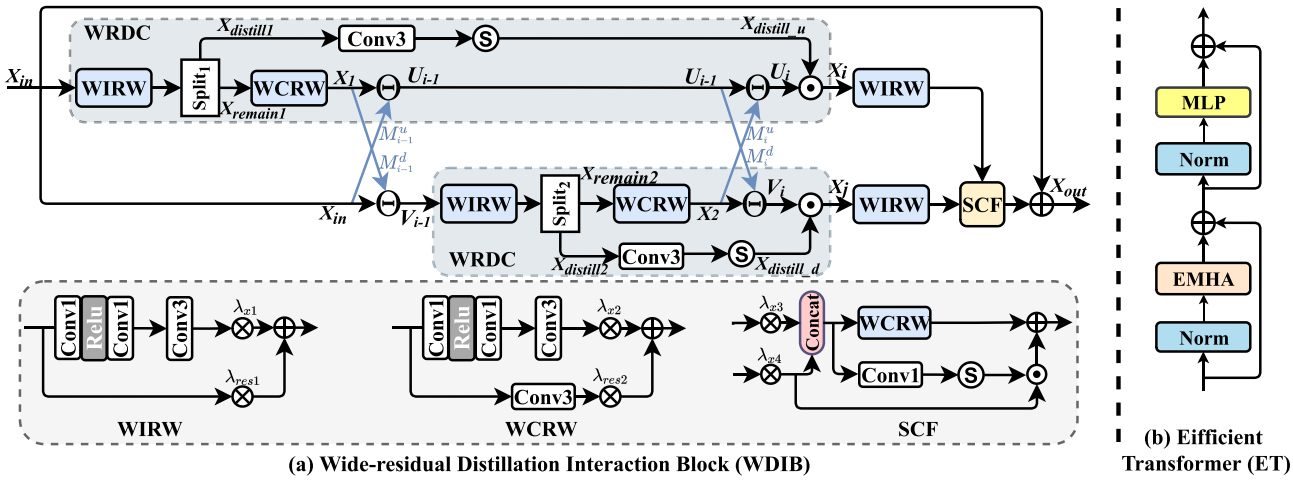


Fig. 3. (a) Structure of the wide-residual distillation interaction block (WDIB). The M_i and M_{i-1} represent the combination coefficient learning, $\odot$ represents the operation of multiplication, $\textcircled{s}$ represents the sigmoid function, and $\Theta(x_i, y_i) = x_i + y_i M_i(y_i)$; (b) the structure of the efficient transformer (ET).

blocks to be fused, respectively. Additionally, adaptive multipliers are applied to both the fused features between blocks and the original input features. This allows the module to adjust its weights dynamically in addition to the weight updates during training. Let's define the input as F_{C_in} and the output as F_{C_out}. The process is:

$$F_{CGS} = G^2{}_{CGS}(G^1{}_{CGS}(W_1, W_2), W_3), \tag{5}$$

$$F_{C\_out} = \lambda_x(F_{CGS} + W_3) + \lambda_{res} F_{C\_in}, \tag{6}$$

where W_i represents the output of the i-th WDIBs, $G^i{}_{CGS}$ denotes the function of the i-th G_{CGS}, F_{CGS} represents the output features obtained from different blocks after a series of fusion, grouping, and shuffling operations, λ_x and λ_{res} are adaptive multipliers, which can be automatically learned and modified during the training process to optimize the model.

The Interaction of CNN and Transformer: The integration architectures can be categorized into three main types, as depicted in Fig. 5 and labeled as (a), (b), and (c). Methods such as ESRT [12], LBNet [13], and CFIN [15] adopt the structures depicted in (a) and (b) in Fig. 5. These methods concatenate CNN and Transformer modules to focus on local and global features in a batch-wise manner. Another set of methods, such as Faceformer [25], utilize the structure of (c) in Fig. 5. They connect the CNN and Transformer modules in parallel and then fuse the extracted local features with the global features, enabling feature extraction for image reconstruction. However, these methods overlook the significance of interactions between local features, extracted from the middle layer of the model, and global features. Our proposed interaction approach addresses this concern by facilitating the free flow of local and global patterns within the network, enabling them to mutually guide each other. This interaction approach enables multiple interactions between local and global features, as illustrated in Fig. 5(d). Specifically, as demonstrated in Fig. 2, the local features W_2 from the first FSWG are combined with the global input features F_{T_in} from the Transformer branch after undergoing dimension and shape transformation through G_{CR}. This fusion of features is then globally modeled using Efficient Transformer G_{ET}. The resulting features, denoted as F_{T_out}, are utilized in the subsequent stages of global modeling and also fed into the second FSWG for local feature extraction. The process can be expressed as follows:

$$F_{T\_out} = G_{ET}(G_{ET}(G_{CR}(W_2) + F_{T\_in})). \tag{7}$$

Efficient Transformer (ET): In the context of lightweight models with limited network depth, using a pure CNN model alone is insufficient for reconstructing high-quality images. To address this issue, we propose a solution that involves compressing the size of the CNN and incorporating an efficient Transformer module to capture long-distance dependencies in images. Specifically, as shown in Fig. 3(b), we adopt the design philosophy of ESRT [12] for multi-headed attention (MHA), which consumes less GPU training memory. This is achieved by splitting the generated tokens Q, K, and V from the linear layer along the width and height dimensions. The formulation of this process is as follows:

$$(Q_1 \ldots Q_n), (K_1 \ldots K_n), (V_1 \ldots V_n) = Split(Q, K, V). \tag{8}$$

Then, the sub-token obtained after subsequent splitting is matrix multiplied on a field of only $\frac{1}{n}$ (where n represents the number of feature splits, and we chose four splits) of the original perception. This effectively reduces memory consumption. Finally, the sub-attention obtained from the matrix dot product is merged to obtain the final self-attention. The entire process can be described as follows:

$$O_i = Attention_i(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) \text{V}_i, \tag{9}$$

$$Attention(Q, K, V) = Concat(O_1, \ldots, O_n). \tag{10}$$

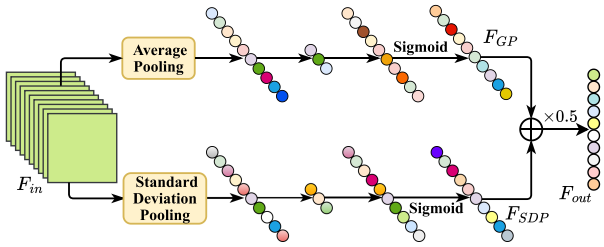


Fig. 4. Details of the combination coefficient learning, which corresponds to M_i and M_{i-1} in Fig. 3(a).

B. Wide-Residual Distillation Interaction Block

Lattice structure: Inspired by the advantages of lattice blocks [10], as depicted in Fig. 3(a), we employ this structure to combine wide-residual weighted units. The structure consists of two paired skip connections designed to connect upper and lower features by combining learning coefficients. Each paired skip connection introduces a distinct combination pattern for the residual units. Additionally, we incorporate feature splitting and information refinement when combining residual units. We also leverage the concept of feature distillation [16] to efficiently perform feature selection and fusion. Since too many distillations may affect the efficiency of the model, we only distill the information once in each of the upper and lower branches to ensure enhanced generalization of the model. Specifically, for the input feature X_{in} that feeds into the upper and lower branches, we define F_{ir} as the WIRW unit and F_{cr} as the WCRW unit. The operation of the upper branch can be described as:

$$X_{remain1}, X_{distill1} = Split(F_{ir}(X_{in})), \tag{11}$$

$$X_1 = F_{cr}(X_{remain1}). \tag{12}$$

The upper and lower branches are then connected via the first paired skip connections. This process can be described as:

$$V_{i-1} = \Theta(X_{in}, X_1) = X_{in} + X_1 M^u_{i-1}(X_1), \tag{13}$$

$$U_{i-1} = \Theta(X_1, X_{in}) = X_1 + X_{in} M^d_{i-1}(X_{in}), \tag{14}$$

where M^u_{i-1} and M^d_{i-1} represent the two combined coefficient learning mechanisms that connect the upper and lower branches in the first paired skip connections, respectively. As shown in Fig. 4, when the input is F_{in}, the output F_{out} of the mechanism can be formulated as:

$$F_{GP} = Sigmoid\left(Conv_{\uparrow}\left(Conv_{\downarrow}\left(Avg\left(F_{in}\right)\right)\right)\right), \tag{15}$$

$$F_{SDP} = Sigmoid\left(Conv_{\uparrow}\left(Conv_{\downarrow}\left(Std\left(F_{in}\right)\right)\right)\right), \tag{16}$$

$$F_{out} = (F_{GP} + F_{SDP})/2, \tag{17}$$

where Avg and Std are average pooling and standard deviation pooling, respectively, $Conv_{\downarrow}$ and $Conv_{\uparrow}$ are convolution operations for reducing channel dimensions and ascending channel dimensions, respectively, and F_{GP} and F_{SDP} are the outputs of the upper and lower branches, respectively. Compared to the average pooling used in traditional channel attention, we have incorporated a standard difference pooling branch here for improved visualization, as verified in [16]. U_{i-1} and V_{i-1} represent the output features of the upper and lower branches, respectively, after passing through the first paired skip connections. Subsequently, U_{i-1} and V_{i-1} are then fed into the second paired skip connections, which follow a similar process as the first paired skip connections. It can be listed as:

$$X_{remain2}, X_{distill2} = Split(F_{ir}(V_{i-1})), \tag{18}$$

$$X_2 = F_{cr}(X_{remain2}), \tag{19}$$

$$V_i = \Theta(X_2, U_{i-1}) = X_2 + U_{i-1} M^u_i(U_{i-1}), \tag{20}$$

$$U_i = \Theta(U_{i-1}, X_2) = U_{i-1} + X_2 M^d_i(X_2). \tag{21}$$

Similar to before, U_i and V_i represent the output features of the upper and lower branches after the second skip connection, respectively. M^u_i and M^d_i denote the combined coefficient learning mechanisms above and below the connection of the second paired skip connection. Simultaneously, the features extracted from the upper and lower branches undergo a nonlinear transformation through the operation of convolution followed by sigmoid activation. This process completes the non-linearization of the coarse features. As a result, we obtain the coarse features $X_{distill\_u}$ from the upper branch and the coarse features $X_{distill\_d}$ from the lower branch. The process can be expressed as:

$$X_{distill\_u} = F_{sigmoid}(F_{conv3}(X_{distill1})), \tag{22}$$

$$X_{distill\_d} = F_{sigmoid}(F_{conv3}(X_{distill2})). \tag{23}$$

Next, obtained coarse features $X_{distill\_u}$ and $X_{distill\_d}$ interact with the fine features U_i and V_i. They are modulated by the attention-based combined coefficient learning mechanism to achieve feature blending with varying degrees of refinement. Finally, the blended features X_i and X_j are fused using our well-designed Self-Calibration Fusion (SCF) module to facilitate the adaptive fusion of the blended features obtained from the two branches. Original input features are retained through residual concatenation. This can be formulated as:

$$X_{out} = F_{SCF}(F_{ir}(X_i), F_{ir}(X_j)) + X_{in}, \tag{24}$$

where F_{SCF} represents the SCF module. Within the SCF module, the outputs of the upper and lower branches perform weighted connections. Subsequently, different levels of refinement are applied to the fused features, resulting in a diverse range of information being incorporated into the final fused features. The module adjusts its weights during training, leading to improved performance compared to standard fusion techniques.

Wide-Residual Distillation Connection (WRDC): As depicted in Fig. 3, the Wide-Residual Distillation Connection (WRDC) is a key component of the model, comprising Wide Convolutional Residual Weighting (WCRW) and Wide Identical Residual Weighting (WIRW) units, as well as residual connections for feature refinement. Both WIRW and WCRW introduce a wide range of activation mechanisms to mitigate the loss of intermediate layer features and extract more expressive features. For WIRW, the wide residual mechanism splits the original residual's initial 3×3 convolution into two 1×1 convolution, which

the first 1×1 convolution increases the channel dimension substantially to handle the subsequent activation functions and reduce feature loss, while the second 1×1 convolution is then used for channel dimensionality reduction, preventing an excessive number of parameters that would arise from using 3×3 convolutional layers for feature extraction. For an input feature x, the broad features obtained through this process can be expressed as:

$$x_{wide} = F_{conv3}(F_{conv1\downarrow}(F_{relu}(F_{conv1\uparrow}(x)))), \tag{25}$$

where $F_{conv1\uparrow}$ represents the channel up-dimensioning operation of the first 1×1 convolution, $F_{conv1\downarrow}$ represents the channel down-dimensioning operation of the second 1×1 convolution, F_{relu} denotes the Relu function used for non-linearization, and F_{conv3} refers to the 3×3 convolution. Since all the high-dimensional channel operations are performed on the 1×1 convolution, they do not impose a significant computational burden. Subsequently, adaptive multipliers are incorporated into both the main branch and the residual branch of the residual block, enabling autonomous adjustment of the weights during training. It is worth noting that WCRW has additional 3×3 convolution layers added to its shortcut path compared to WIRW. This ensures a match with the original input channel size after channel splitting. Consequently, the outputs y_{wirw} and y_{wcrw} for WIRW and WCRW can be respectively expressed as:

$$y_{wirw} = \lambda_{x1} x_{wide} + \lambda_{res1} x, \tag{26}$$

$$y_{wcrw} = \lambda_{x2} x_{wide} + \lambda_{res2} F_{conv3}(x), \tag{27}$$

where λ_{xk} and λ_{resk} $(k = 1, 2)$ represent the adaptive weighted multipliers of the k-th wide residual weighted unit. Additionally, a convolutional layer is introduced in the distillation connection section to expand the dimensionality of the split channels. The obtained coarse features are then subjected to Sigmoid function nonlinearities, resulting in low-frequency feature maps. Finally, these features are multiplied with the high-frequency feature maps obtained through the combined action of wide residual units and combined coefficient learning to achieve the interaction of various types of pattern features.

C. Loss Function

For the pairs $\{I^i_{LR}, I^i_{HR}\}_{i=1}^N$ in the training set, the reconstruction loss of our method FIWHN during training can be expressed as:

$$Loss(\theta) = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left\| FIWHN(I^i_{LR}) - I^i_{HR} \right\|_1, \tag{28}$$

where N represents the number of LR-HR pairs in the training set, and θ represents the parameter size of FIWHN.

IV. Experiments

A. Datasets

We use 800 pairs of HR-LR images from the DIV2K [27] dataset for training, which includes images of various natural scenes. LR samples are generated using a bicubic downsampling method as used in [4]. In addition, to evaluate the effectiveness of our method, we conducted tests on commonly used benchmark datasets, including Set5 [28], Set14 [29], BSDS100 [30], Urban100 [31], and Manga109 [32].

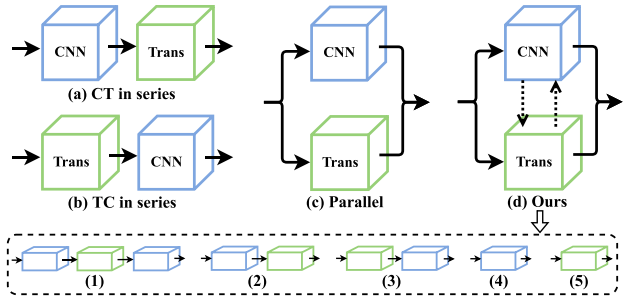


Fig. 5. Exploring how to combine CNN and transformer efficiently and the potential of our method for multiple combinations of both.

TABLE I
Analysis of the Effect of the Wide Residual Mechanism on WIRW and WCRW

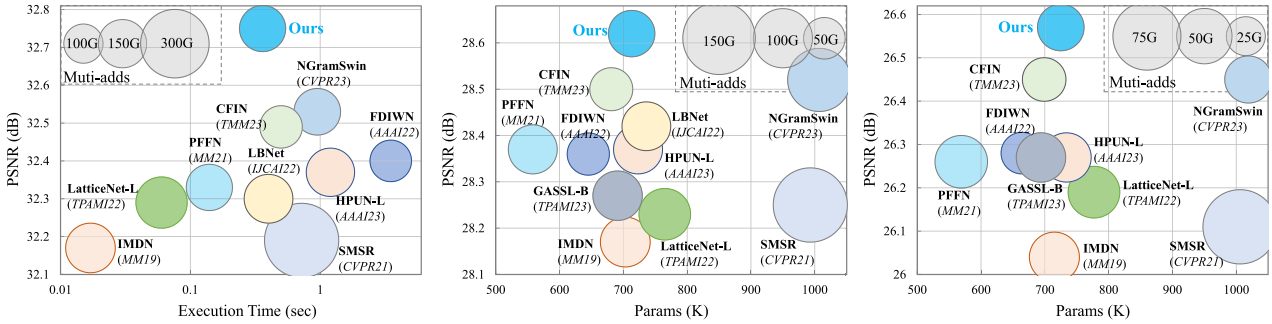
Methods	Channels	Params	Multi-adds	Set5(×4) PSNR / Time
Baseline	32	223K	12.69G	31.75 / 7.22ms
FIWHN	64	147K	4.46G	31.76 / **5.72ms**
FIWHN	120	175K	9.89G	**31.83** / 6.49ms

B. Implementation Details

During training, we initialize the learning rate to 5e-4 and use the cosine annealing strategy to gradually decay it to 6.25e-6 over 1000 epochs. The optimizer used Adam, with the β_1 parameter set to 0.9 and the β_2 parameter set to 0.999. We randomly crop patches of size 48×48 from the training set as the input for training. Additionally, we apply data augmentation techniques such as random rotation and random flipping to these patches to enhance the dataset's variability. All our training is done using the Pytorch framework on an NVIDIA RTX 2080Ti. In the final model, the initial channel is set to 32 for the CNN part and 144 for the Transformer part. Furthermore, we apply weight normalization after the convolutional layers in the wide residual block to accelerate the convergence of training.

C. Ablation Study

The effectiveness of WIRW and WCRW: To assess the superiority of the residual blocks with wide activation mechanisms over normal residual blocks, we conducted experiments by replacing the WIRW and WCRW blocks with basic residual blocks as the baseline within the WDIB. We also investigated the impact of the number of channels before the activation function on the quantitative performance of SISR by setting the number of channels to 64 and 120, respectively. The results in Table I demonstrate the following observations: i) Our proposed FIWHN model achieves better performance and faster inference speed while using fewer parameters and Multi-adds compared



(a) PSNR, Muti-adds, and Speed Tradeoffs (×2). (b) PSNR, Parmas and Muti-adds Tradeoffs (×3). (c) PSNR, Parmas and Muti-adds Tradeoffs (×4).

Fig. 6. Model complexity analysis at each scale on the Urban100 test set.

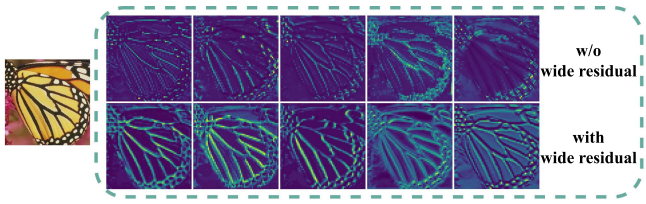


Fig. 7. Visualization of the effect of wide residual mechanism on extracted features.

TABLE III
EVALUATE THE EFFECTIVENESS OF OUR WDIB

Methods	Depth	Params	Multi-adds	Set5(×4) PSNR / Time
IMDB [16]	32	60.0K	3.42G	31.40 / 4.37ms
RFDB [26]	24	63.2K	3.51G	31.36 / 3.51ms
LB [10]	39	65.2K	3.66G	31.34 / 4.41ms
HPB [12]	48	64.5K	3.78G	31.36 / 6.81ms
LFFM [13]	25	61.2K	3.47G	31.37 / 3.05ms
WDIB	26	61.0K	2.49G	**31.44 / 3.02ms**

TABLE II
IMPACT ANALYSIS OF DIFFERENT MODULE COMBINATIONS IN THE WDIB FRAMEWORK, WHERE ⊗ IS THE ADAPTIVE MULTIPLIER

Methods	WRDC	SCF	BI	⊗	Params	Multi-adds	Set5(×4) PSNR / SSIM
Baseline	×	×	×	×	59.3K	3.36G	31.17 / 0.8791
Case 1	✓				49.2K	2.11G	31.17 / 0.8799
Case 2		✓			69.5K	3.62G	31.35 / 0.8824
Case 3			✓		59.3K	3.36G	31.22 / 0.8791
Case 4				✓	59.3K	3.36G	31.21 / 0.8794
FIWHN	✓	✓	✓	✓	69.9K	2.57G	**31.42 / 0.8835**

to the baseline model. ii) Increasing the number of channels before the activation function leads to further improvements in the model performance with a slight increase in computational load. Furthermore, we provide visual results to illustrate the beneficial effects of the wide residual mechanism on feature extraction. In Fig. 7, we display the feature maps obtained by both normal residual units and our wide residual units. It is evident that the features extracted by the normal residual units lack many details in contour texture regions, which are crucial for accurate image recovery. In contrast, our proposed WIRW and WCRW units effectively mitigate the loss of these essential intermediate features mentioned above.

The effectiveness of WDIB: Firstly, we analyze the internal components of WDIB in Table II, including our proposed WRDC (Case 1), SCF (Case 2), and ⊗ (adaptive multiplier, Case 4). By comparing case 1, case 2, and the baseline, we observe that our proposed WRDC module achieves slightly better performance than the baseline while saving approximately 13% of the number of parameters and 37% of the Multi-adds. The SCF module further improves the PSNR value by 0.18 dB with a parametric gain of less than 10 K. The adaptive multiplier improves the PSNR by 0.04 dB compared to the baseline and does not introduce additional computational load or slow down the inference process. Overall, the integration of these submodules significantly enhances the performance.

Next, we provide a comparative analysis of our proposed WDIB with the sub-block of some methods in terms of PSNR values, model complexity, and inference speed in Table III. The compared methods include state-of-the-art models such as IMDN [16], RFDN [26], LatticeNet [10], ESRT [12], and LBNet [13]. To ensure a fair comparison, we stack these blocks to achieve a similar number of parameters and then evaluate them comprehensively. As shown in the table, our proposed WDIB achieves the best performance with lower computational complexity. And our module has a shallower depth and faster inference speed compared to the other modules. Considering the trade-off between model size, inference speed, and reconstruction accuracy, WDIB proves to be a superior choice for efficient SISR.

The combination structure of FIWHN: The combination structure of our proposed model consists of two main parts. The first is the feature grouping shuffle fusion part of the combined WDIB, which specifically addresses the issue of inadequate communication between blocks. In Table II, we compare the performance of the model with and without Block Interaction (BI) between blocks. Case 3, which includes the BI part, only has one additional BI compared to the baseline and almost no increase in computational load. The PSNR value of the model improves by 0.05 dB, indicating that communication between blocks benefits image reconstruction. Moreover, we also investigate the optimal number of inter-block feature mixing and fusion. As shown in Fig. 9, the model achieves its best efficiency when the number of WDIBs forming FSWG is 3. Therefore, we use three WDIBs to form the FSWG.

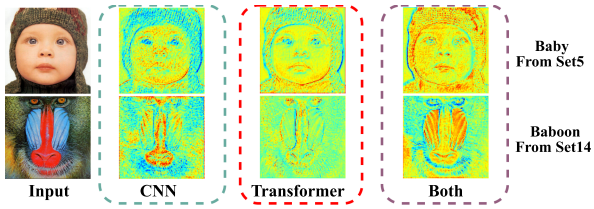
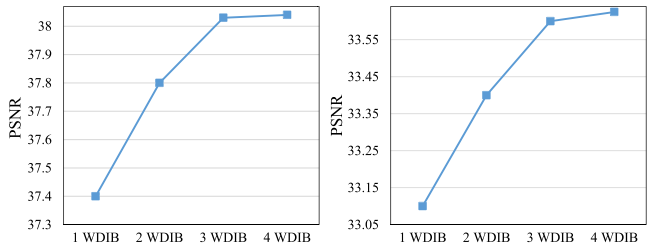


Fig. 8. Heat maps about the internal composition of FIWHN.



(a) Ablation model on Set5 (×2) (b) Ablation model on Set14 (×2)

Fig. 9. Study on different numbers of WDIBs.

TABLE IV
EXPERIMENTS ABOUT THE PERFORMANCE OF VARIOUS COMBINED CNN AND TRANSFORMER ARCHITECTURES SETTING

Scale	Architecture	Set14	B100	Urban100	Manga109
×4	CT in series	28.72	27.63	26.40	30.89
	TC in series	28.71	27.64	26.33	30.75
	Parallel	28.61	27.58	26.18	30.56
	Ours	**28.76**	**27.68**	**26.57**	**30.93**

The second part of the combined structure involves the fusion of CNN and Transformer. In Table IV, we evaluate different combinations as depicted in Fig. 5. Our proposed scheme outperforms the implementations of simple CNN followed by Transformer, simple Transformer followed by CNN, and a simple parallel combination of CNN and Transformer. Importantly, this combination approach does not increase the overall computational load. These experiments on combination architectures demonstrate that an effective combination architecture can significantly enhance the model's representation ability. Furthermore, these results emphasize the importance of well-connecting and combining features from both local-based and global-based features in the middle layer to maximize the model's generalization ability. To visualize the effects of CNN and Transformer on the attention area, we provide the feature heat maps at different branches of the model in Fig. 8. When the model contains only the CNN part, the attention is focused solely on the local area. Conversely, when the model contains only the Transformer part, it effectively captures global image information but may ignore certain local details. However, by integrating CNN and Transformer, the model can simultaneously consider both local and global areas, leading to the activation of more details.

Model complexity analysis: In Fig. 6, we present a comparison considering parameters, computational load, and inference time, in comparison to several state-of-the-art methods. The results demonstrate that our approach achieves superior performance across all scales while maintaining a reasonable number of parameters and computational effort. Notably, as shown in Fig. 6(a), our method not only excels in performance but also outperforms most methods in terms of inference speed. Compared to the conference version FDIWN, FIWHN can achieve better performance using a shallower model depth due to the complementary global-local feature interaction, and therefore also significantly improves the inference speed. However, the inclusion of the highly computationally intensive Transformer also leads to an increase in model parameters and Muti-adds. It surpasses Transformer-based methods like LBNet [13], CFIN [15], and NGramswin [23]. This demonstrates that our method effectively strikes a superior balance among model complexity, performance, and inference speed.

D. Comparisons With State-of-The-Art Methods

In this section, we provide a comprehensive comparison with state-of-the-art methods on benchmark datasets. The quantitative comparison results for ×2, ×3, and ×4 SISR are presented in Table V. It can be clearly seen that our FIWHN achieves the best performance across almost all datasets. Additionally, the number of parameters and the Multi-adds of our method is much lower than most of the methods. Furthermore, compared to our conference version FDIWN [5], we have further improved performance with only a marginal increase in computational cost. Notably, on Urban100 and Manga109 test sets, we observe an average performance gain of over 0.3 dB for all three scaling factors. These improvements clearly demonstrate the effectiveness of incorporating Transformers to complement the global features of the CNN model.

In addition, we also compare our method with several advanced Transformer-based methods in Table VI. Although our approach did not achieve optimal performance on some of the test sets, it can be clearly seen that our method outperforms LBNet [13], CFIN [15], and NGSwin [23] on various datasets while being roughly comparable to SwinIR-light [11] in terms of average performance. However, it is worth noting that SwinIR-light utilizes additional pre-training strategies to enhance model performance and employs a larger patch size for training, which contributes to better performance. Moreover, SwinIR-light has significantly higher numbers of parameters and computations compared to our method. And training of our model can be efficiently conducted on an NVIDIA RTX 2080Ti, and our method demonstrates the fastest inference speed among these compared methods. Overall, the minute performance gap is challenging to discern visually, but our method's advantage of only about three-quarters of the computational consumption of SwinIR-light, and the inference time is faster, which is important for models running on devices with limited computational resources. These experiments collectively demonstrate that our FIWHN is a competitive method that excels in terms of performance, efficiency, and inference.

A qualitative comparison between our method and other methods is shown in Fig. 10. To provide more convincing comparisons, we include the most recent CNN-based and Transformer-based methods. In addition to visual comparisons, we also provide the corresponding PSNR/SSIM values for a

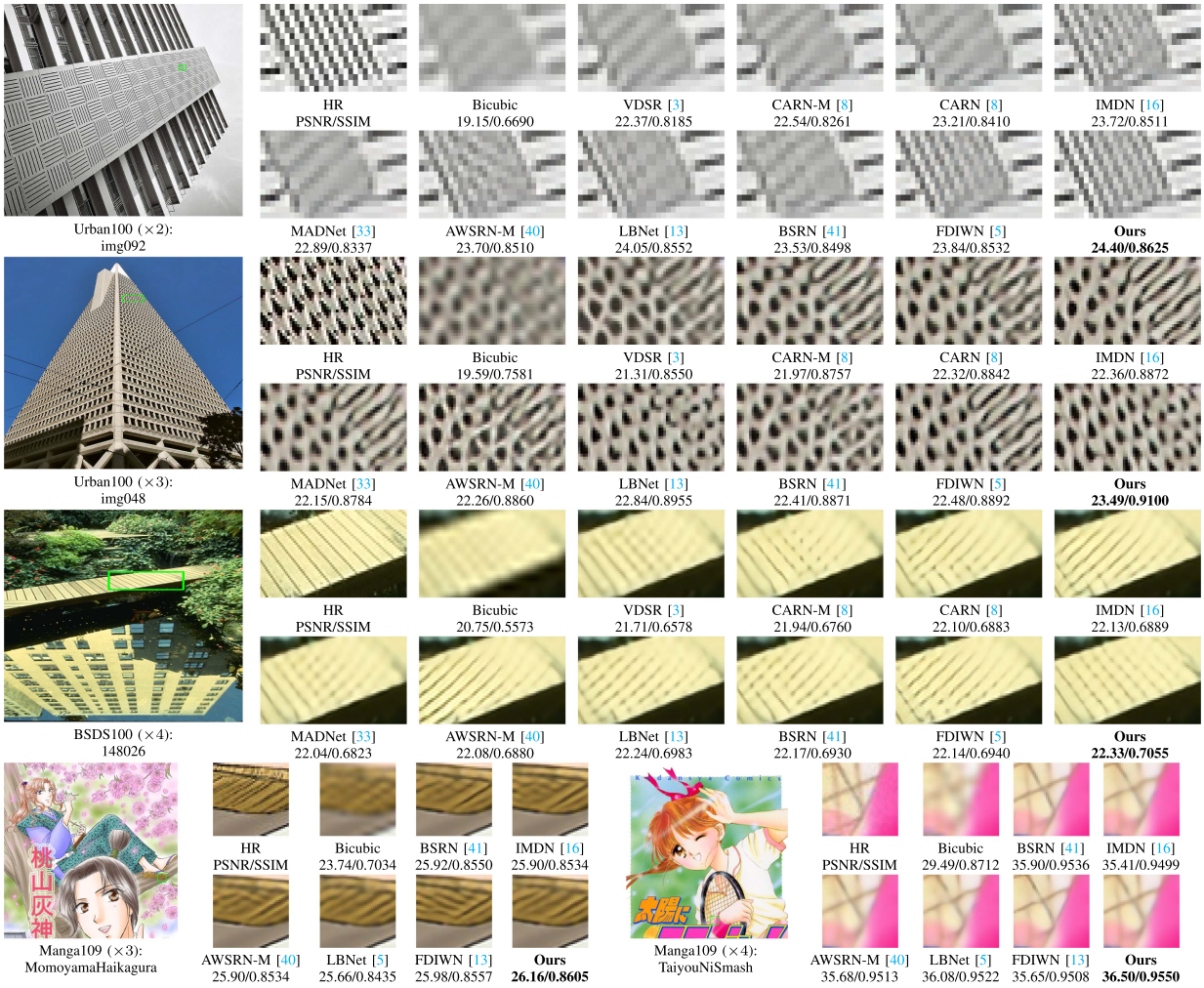


Fig. 10. Visual comparison of FIWHN with existing SISR methods.

Fig. 11. Visual comparison on RealSR [42] dataset (including nikon and canon).

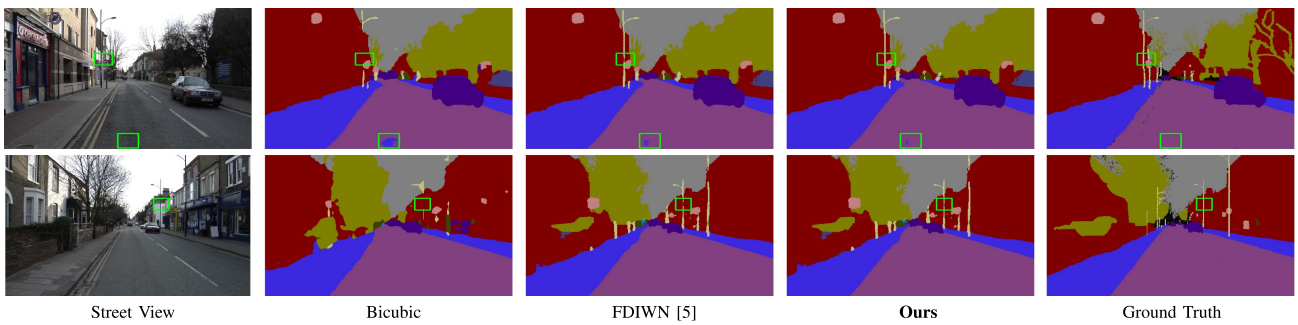


Fig. 12. Comparison of qualitative image segmentation results (×2 SR) on CamVid [44] test set.

TABLE V
THE PSNR/SSIM COMPARISON WITH THE STATE-OF-THE-ART CNN-BASED SISR MODELS

Methods	Scale	Params	Multi-adds	Set5 [28]	Set14 [29]	BSDS100 [30]	Urban100 [31]	Manga109 [32]
				PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
CARN [8]	×2	1592K	222.8G	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256	38.36/0.9765
IMDN [16]		694K	158.8G	38.00/0.9605	33.63/0.9177	32.19/0.8996	32.17/0.9283	38.88/0.9774
MADNet [33]		878K	187.1G	37.85/0.9600	33.38/0.9161	32.04/0.8979	31.62/0.9233	-
LAPAR-A [34]		548K	171.0G	38.01/0.9605	33.62/0.9183	32.19/0.8999	32.10/0.9283	38.67/0.9772
SMSR [35]		985K	351.5G	38.00/0.9601	33.64/0.9179	32.17/0.8990	32.19/0.9284	38.76/0.9771
PFFN [6]		569K	138.3G	38.07/0.9607	33.69/0.9192	32.21/0.8997	32.33/0.9298	38.89/0.9775
FDIWN [5]		629K	112.0G	38.07/0.9608	33.75/**0.9201**	32.23/0.9003	32.40/0.9305	38.85/0.9774
LatticeNet-CL [10]		756K	169.5G	38.09/0.9608	33.70/0.9188	32.21/0.9000	32.29/0.9291	-
FMEN [36]		748K	172.0G	38.10/0.9609	33.75/0.9192	32.26/0.9003	32.41/0.9311	38.95/0.9778
HPUN-L [37]		714K	151.1G	38.09/0.9608	**33.79**/0.9198	32.25/0.9006	32.37/0.9307	**39.07**/0.9779
GASSL-B [38]		689K	158.2G	38.08/0.9607	33.75/0.9194	32.24/0.9005	32.29/0.9298	38.92/0.9777
FIWHN (Ours)		705K	137.7G	**38.16/0.9613**	33.73/0.9194	**32.27/0.9007**	**32.75/0.9337**	**39.07/0.9782**
CARN [8]	×3	1592K	118.8G	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493	33.43/0.9427
IMDN [16]		703K	71.5G	34.36/0.9270	30.32/0.8417	29.09/0.8046	28.17/0.8519	33.61/0.9445
MADNet [33]		930K	88.4G	34.16/0.9253	30.21/0.8398	28.98/0.8023	27.77/0.8439	-
LAPAR-A [34]		594K	114.0G	34.36/0.9267	30.34/0.8421	29.11/0.8054	28.15/0.8523	33.51/0.9441
SMSR [35]		993K	156.8G	34.40/0.9270	30.33/0.8412	29.10/0.8050	28.25/0.8536	33.68/0.9445
PFFN [6]		558K	69.1G	34.54/0.9282	30.42/0.8435	29.17/0.8062	28.37/0.8566	33.63/0.9455
FDIWN [5]		645K	51.5G	34.52/0.9281	30.42/0.8438	29.14/0.8065	28.36/0.8567	33.77/0.9456
LatticeNet-CL [10]		765K	76.3G	34.46/0.9275	30.37/0.8422	29.12/0.8054	28.23/0.8525	-
FMEN [36]		757K	77.2G	34.45/0.9275	30.40/0.8435	29.17/0.8063	28.33/0.8562	33.86/0.9462
HPUN-L [37]		723K	69.3G	**34.56**/0.9281	30.45/0.8445	29.18/0.8072	28.37/0.8572	33.90/0.9463
DDistill-SR [39]		665K	60.1G	34.43/0.9276	30.39/0.8432	29.16/0.8070	28.31/0.8546	**33.97**/0.9465
GASSL-B [38]		691K	70.4G	34.47/0.9278	30.39/0.8430	29.15/0.8063	28.27/0.8546	33.77/0.9455
FIWHN (Ours)		713K	62.0G	34.50/ **0.9283**	**30.50/ 0.8451**	**29.19/ 0.8077**	**28.62/ 0.8607**	**33.97/0.9472**
CARN [8]	×4	1592K	90.9G	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	30.42/0.9070
IMDN [16]		715K	40.9G	32.21/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838	30.45/0.9075
MADNet [33]		1002K	54.1G	31.95/0.8917	28.44/0.7780	27.47/0.7327	25.76/0.7746	-
LAPAR-A [34]		659K	94.0G	32.15/0.8944	28.61/0.7818	27.61/0.7366	26.14/0.7871	30.42/0.9074
SMSR [35]		1006K	89.1G	32.12/0.8932	28.55/0.7808	27.55/0.7351	26.11/0.7868	30.54/0.9085
PFFN [6]		569K	45.1G	**32.36/0.8967**	28.68/0.7827	27.63/0.7370	26.26/0.7904	30.50/0.9100
FDIWN [5]		664K	28.4G	32.23/0.8955	28.66/0.7829	27.62/0.7380	26.28/0.7919	30.63/0.9098
LatticeNet-CL [10]		777K	43.6G	32.30/0.8958	28.65/0.7822	27.59/0.7365	26.19/0.7855	-
FMEN [36]		769K	44.2G	32.24/0.8955	28.70/0.7839	27.63/0.7379	26.28/0.7908	30.70/0.9107
HPUN-L [37]		734K	39.7G	32.31/0.8962	28.73/0.7842	27.66/0.7386	26.27/0.7918	30.77/0.9109
DDistill-SR [39]		675K	32.6G	32.29/0.8961	28.69/0.7833	27.65/0.7385	26.25/0.7893	30.79/0.9098
GASSL-B [38]		694K	39.9G	32.27/0.8962	28.74/**0.7850**	27.66/0.7388	26.27/0.7914	30.92/0.9122
FIWHN (Ours)		725K	35.6G	32.30/**0.8967**	**28.76**/0.7849	**27.68/0.7400**	**26.57/0.7989**	**30.93/0.9131**

The best and the second-best results are highlighted and underlined, respectively.

TABLE VI
COMPARISON WITH EXISTING TRANSFORMER-BASE METHODS FOR ×4 SR

Methods	Params	Multi-adds	GPU	Time	Set14 [29]	BSDS100 [30]	Urban100 [31]	Manga109 [32]	Average
					PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM
SwinIR-light [11]	897K	49.6G	10.5G	55ms	28.77 / 0.7858	**27.69 / 0.7406**	26.47 / 0.7980	30.92 / **0.9151**	28.46 / **0.8192**
LBNet [13]	742K	38.9G	6.4G	49ms	28.68 / 0.7832	27.62 / 0.7382	26.27 / 0.7906	30.76 / 0.9111	28.30 / 0.8147
CFIN [15]	699K	31.2G	11.5G	45ms	28.74 / 0.7849	27.68 / 0.7396	26.39 / 0.7946	30.73 / 0.9124	28.35 / 0.8169
NGSwin [23]	1019K	36.4G	>12G	67ms	**28.78 / 0.7859**	27.66 / 0.7396	26.45 / 0.7963	30.80 / 0.9128	28.39 / 0.8184
FIWHN (Ours)	725K	35.6G	7.5G	**38ms**	28.76 / 0.7849	27.68 / 0.7400	**26.57 / 0.7989**	**30.93** / 0.9131	**28.49** / 0.8186

comprehensive analysis. As observed in this figure, our method consistently achieves higher PSNR and outperforms other methods in terms of visual quality, especially in capturing fine details across multiple validation sets.

E. Real-World Image Super-Resolution

Dataset and implement details: For training and testing, we utilize the RealSR dataset [42]. This dataset captures both HR and LR images in the same scene using the same camera but with different focal lengths. The degradation model in this dataset is more complex compared to the DIV2K dataset, and the degradation kernel varies spatially. It is worth noting that the dimensions of LR and HR images in this dataset are already aligned. Therefore, in our experiments, all methods remove the upsampling operation at the back end of the model. However, pixel alignment during image restoration becomes more challenging due to issues such as pixel drift and changes in scale factors resulting from adjusting the focal length. To alleviate the difficulty caused by pixel alignment, most methods adopt the strategy of dividing image patches into large patches when feeding them into the network during training. This operation helps mitigate the challenge of aligning pixels between numerous patches that do not effectively communicate with each other. Typically, These methods use image patches of size 128 × 128 during training. However, for FDIWN and FIWHN, we set the patch to 64 × 64

TABLE VII
COMPARISON WITH EXISTING SISR MODELS ON REALSR [42] DATASET

Scale	Bicubic	SRCNN [2]	VDSR [2]	SRResNet [43]	IMDN [16]	ESRT [12]	FDIWN [5]	**FIWHN(ours)**
	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM
×2	32.61 / 0.907	33.40 / 0.916	33.64 / 0.917	33.69 / 0.919	33.85 / 0.923	33.92 / 0.924	33.68/0.9242	**33.96 / 0.927**
×3	29.34 / 0.841	29.96 / 0.845	30.14 / 0.856	30.18 / 0.859	30.29 / 0.857	30.38 / 0.857	30.38 / 0.857	**30.57 / 0.862**
×4	27.99 / 0.806	28.44 / 0.801	28.63 / 0.821	28.67 / 0.824	28.68 / 0.815	28.78 / 0.815	28.70 / 0.815	**28.82 / 0.828**

during training due to the higher training memory requirement associated with larger patch sizes. Therefore, our FIWHN can be trained on a single NVIDIA RTX 2080Ti GPU, while other methods cannot even be trained on two NVIDIA RTX 2080Ti GPUs due to their higher memory demands.

Comparison results: The final quantitative comparison results are presented in Table VII. Despite facing a disadvantage in the training scenario, our method achieves the best results across all scales, particularly at a scale factor of ×3, where our method outperforms the second-best method with a PSNR value of 0.19 dB. We also provide visual comparisons in Fig. 11, which highlight that our FIWHN can recover more textural details compared to the conference version FDIWN, resulting in restoration results that closely resemble the HR image. To further validate the effectiveness of FIWHN, we evaluate its benefits for street image semantic segmentation tasks. For this evaluation, we first downsample the images from the CamVid [44] dataset and then apply SISR methods to recover the high-quality images. Finally, we segment these images using the recently published real-time segmentation method FBSNet [45]. As shown in Fig. 12, the segmentation results obtained from the images recovered by our method are closer to the ground truth. Notably, for segmented details such as the utility poles, our FIWHN significantly outperforms simple Bicubic and our conference version FDIWN.

V. CONCLUSION

In this paper, we present a FIWHN that supports efficient SISR, which consists of groups of WDIB that are blended, fused, and weighted. The WDIB utilizes combinatorial coefficient learning to connect wide residual weighted units, which mitigates the loss of intermediate layers and induces different combinatorial structures. This enables features to exploit varying levels of information through residual connections and fusion. To further enhance the model, we explore a novel combined CNN and Transformer architecture to encourage capturing local and global feature interactions. Through extensive experiments, we demonstrate the efficacy of FIWHN in achieving efficient SISR and its applicability across diverse SISR scenarios and related downstream tasks.

REFERENCES

[1] G. Gao, Y. Yu, H. Lu, J. Yang, and D. Yue, "Context-patch representation learning with adaptive neighbor embedding for robust face image super-resolution," *IEEE Trans. Multimedia*, vol. 25, pp. 1879–1889, 2023.

[2] 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.

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

[4] Y. Zhang et al., "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 286–301.

[5] G. Gao et al., "Feature distillation interaction weighting network for lightweight image super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 1, pp. 661–669.

[6] D. Zhang, C. Li, N. Xie, G. Wang, and J. Shao, "PFFN: Progressive feature fusion network for lightweight image super-resolution," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 3682–3690.

[7] X. Chu, B. Zhang, H. Ma, R. Xu, and Q. Li, "Fast, accurate and lightweight super-resolution with neural architecture search," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2021, pp. 59–64.

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

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[10] X. Luo et al., "Lattice network for lightweight image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4826–4842, Apr. 2023.

[11] J. Liang et al., "Swinir: Image restoration using swin transformer," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2021, pp. 1833–1844.

[12] Z. Lu et al., "Transformer for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2022, pp. 457–466.

[13] G. Gao et al., "Lightweight bimodal network for single-image super-resolution via symmetric cnn and recursive transformer," in *Proc. Int. Joint Conf. Artif. Intell.*, 2022, pp. 661–669.

[14] X. Zhang, H. Zeng, S. Guo, and L. Zhang, "Efficient long-range attention network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 649–667.

[15] W. Li et al., "Cross-receptive focused inference network for lightweight image super-resolution," *IEEE Trans. Multimedia*, vol. 26, pp. 864–877, 2024.

[16] 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*, 2019, pp. 2024–2032.

[17] H. Li et al., "Pams: Quantized super-resolution via parameterized max scale," in *Proc. Eur. Conf. Comput. Vis.*. Springer, 2020, pp. 564–580.

[18] W. Lee, J. Lee, D. Kim, and B. Ham, "Learning with privileged information for efficient image super-resolution," in *Proc. Comput. Vis.–ECCV 2020: 16th Eur Conf.*, Springer, 2020, pp. 465–482.

[19] G. Wu, J. Jiang, J. Jiang, and X. Liu, "Transforming image super-resolution: A convformer-based efficient approach," *IEEE Trans. Image Process.*, vol. 33, pp. 6071–6082, 2024.

[20] J. Yu et al., "Wide activation for efficient and accurate image super-resolution," 2018, *arXiv:1808.08718*.

[21] Q. Bao, Y. Liu, B. Gang, W. Yang, and Q. Liao, "Sctanet: A spatial attention-guided CNN-transformer aggregation network for deep face image super-resolution," *IEEE Trans. Multimedia*, vol. 25, pp. 8554–8565, 2024.

[22] H. Qi, Y. Qiu, X. Luo, and Z. Jin, "An efficient latent style guided transformer-CNN framework for face super-resolution," *IEEE Trans. Multimedia*, vol. 26, pp. 1589–1599, 2024.

[23] H. Choi, J. Lee, and J. Yang, "N-gram in swin transformers for efficient lightweight image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 2071–2081.

[24] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6848–6856.

[25] Y. Wang et al., "Faceformer: Aggregating global and local representation for face hallucination," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 6, pp. 2533–2545, Jun. 2022.

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

[27] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 114–125.
[28] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. Brit. Mach. Vis. Conf., Nottingham*, 2012, pp. 135.1–135.10.
[29] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Curves Surfaces: 7th Int. Conf.*, 2010, pp. 711–730.
[30] 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. IEEE Int. Conf. Comput. Vis.*, 2001, pp. 416–423.
[31] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5197–5206.
[32] Y. Matsui et al., "Sketch-based manga retrieval using manga109 dataset," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 21811–21838, 2017.
[33] R. Lan et al., "MADNet: A fast and lightweight network for single-image super resolution," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1443–1453, Mar. 2021.
[34] W. Li et al., "Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond," in *Proc. NeurIPS*, 2020, vol. 33, pp. 20343–20355.
[35] L. Wang et al., "Exploring sparsity in image super-resolution for efficient inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4917–4926.
[36] Z. Du et al., "Fast and memory-efficient network towards efficient image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2022, pp. 853–862.
[37] B. Sun, Y. Zhang, S. Jiang, and Y. Fu, "Hybrid pixel-unshuffled network for lightweight image super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 2, pp. 2375–2383.
[38] H. Wang, Y. Zhang, C. Qin, L. Van Gool, and Y. Fu, "Global aligned structured sparsity learning for efficient image super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10974–10989, Sep. 2023.
[39] Y. Wang et al., "DDistill-SR: Reparameterized dynamic distillation network for lightweight image super-resolution," *IEEE Trans. Multimedia*, vol. 25, pp. 7222–7234, 2023.
[40] C. Wang, Z. Li, and J. Shi, "Lightweight image super-resolution with adaptive weighted learning network," 2019, *arXiv:1904.02358*.
[41] Z. Li et al., "Blueprint separable residual network for efficient image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 833–843.
[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 Int. Conf. Comput. Vis.*, 2019, pp. 3086–3095.
[43] C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4681–4690.
[44] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 44–57.
[45] G. Gao et al., "FBSNet: A fast bilateral symmetrical network for real-time semantic segmentation," *IEEE Trans. Multimedia*, vol. 25, pp. 3273–3283, 2023.

Wenjie Li received the M.S. degree from the College of Automation and College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing, China. He is currently working toward the Ph.D. degree with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include image restoration.

Juncheng Li received the Ph.D. degree in computer science and technology from East China Normal University, Shanghai, China, in 2021. He is currently an Assistant Professor with the School of Communication and Information Engineering, Shanghai University, Shanghai, China. His research interests include image restoration, computer vision, and medical image processing.

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 is currently a Professor with the Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include pattern recognition and computer vision. More information can be found at https://guangweigao.github.io.

Weihong Deng (Senior Member, IEEE) received the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2009. He was a Professor with the School of Artificial Intelligence, BUPT. His research interests include trustworthy biometrics and affective computing, with a particular emphasis on face recognition and expression analysis.

Jian Yang (Member, IEEE) received the Ph.D. degree from the Nanjing University of Science and Technology (NJUST), Nanjing, China. He is currently a professor with the School of Computer Science and Technology, NJUST. He is the author of more than 400 scientific papers in pattern recognition and computer vision. His research interests include pattern recognition, computer vision, and machine learning. He is/was an Associate Editor for Pattern Recognition and IEEE TRANSACTIONS NEURAL NETWORKS AND LEARNING SYSTEMS. He is a fellow of IAPR.

Guo-Jun Qi (Fellow, IEEE) since 2014, he has been a Faculty Member with the Department of Computer Science, University of Central Florida, Orlando, FL, USA. Since 2018, he has been the Chief Scientist leading and overseeing the International Research and Development Team for multiple artificial intelligence services on the Huawei Cloud. He is currently a Professor and the Chief Scientist who oversees the Artificial Intelligence Research Center, Westlake University, Hangzhou, China, and the OPPO U.S. Research Center. His research interests include machine learning and knowledge discovery from multi-modal data to build smart and reliable information and decision-making systems.

Chia-Wen Lin (Fellow, IEEE) received the Ph.D. degree in electrical engineering from National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 2000. From 2000 to 2007, he was with the Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan. He is currently a Distinguished Professor with the Department of Electrical Engineering and the Institute of Communications Engineering, NTHU. He is also the Deputy Director of the AI Research Center, NTHU. His research interests include image and video processing, computer vision, and video networking.