Feature Distillation Interaction Weighting Network for Lightweight Image Super-Resolution

Guangwei Gao[1]†, Wenjie Li[1]†, Juncheng Li[2]*, Fei Wu[1], Huimin Lu[3], Yi Yu[4]

[1] Nanjing University of Posts and Telecommunications [2] The Chinese University of Hong Kong
[3] Kyushu Institute of Technology [4] National Institute of Informatics
{csggao,cvjunchengli}@gmail.com, liwj0824@163.com, wufei_8888@126.com, luhuimin@ieee.org, yiyu@nii.ac.jp

Abstract

Convolutional neural networks based single-image super-resolution (SISR) has made great progress in recent years. However, it is difficult to apply these methods to real-world scenarios due to the computational and memory cost. Meanwhile, how to take full advantage of the intermediate features under the constraints of limited parameters and calculations is also a huge challenge. To alleviate these issues, we propose a lightweight yet efficient Feature Distillation Interaction Weighted Network (FDIWN). Specifically, FDIWN utilizes a series of specially designed Feature Shuffle Weighted Groups (FSWG) as the backbone, and several novel mutual Wide-residual Distillation Interaction Blocks (WDIB) form an FSWG. In addition, Wide Identical Residual Weighting (WIRW) units and Wide Convolutional Residual Weighting (WCRW) units are introduced into WDIB for better feature distillation. Moreover, a Wide-Residual Distillation Connection (WRDC) framework and a Self-Calibration Fusion (SCF) unit are proposed to interact features with different scales more flexibly and efficiently. Extensive experiments show that our FDIWN is superior to other models to strike a good balance between model performance and efficiency. The code is available at https://github.com/IVIPLab/FDIWN.

Introduction

Due to the huge computational overhead of traditional super-resolution, it is difficult to be applied to mobile devices with limited computing capabilities. The main goal of lightweight single-image super-resolution (SISR) is to reconstruct super-resolution (SR) images from the low-resolution (LR) one with fewer parameters and calculations (Yao et al. 2020; Li et al. 2020a; Hou, Zhou, and Feng 2021). In the past ten years, deep learning has made amazing achievements in various computer vision tasks, which also greatly promoted the development of SISR.

Recently, many convolutional neural networks (CNNs) based SISR methods have been proposed (Dong et al. 2015; Han, Mao, and Dally 2015; Dong, Loy, and Tang 2016; Tian et al. 2020; He et al. 2019). Compared with the traditional methods, CNN-based SISR methods are more versatile and can reconstruct higher-quality SR images with more texture details. In 2014, Dong *et al.* introduced the deep learning technology into SISR and proposed the first CNN-based SISR model, named SRCNN (Dong et al. 2015). Although SRCNN has only three convolutional layers, its performance has far surpassed traditional methods and achieved state-of-the-art results at the time. Now, we know that deeper and more complex networks can achieve better performance (Lim et al. 2017; Ahn, Kang, and Sohn 2018; Haris, Shakhnarovich, and Ukita 2018; Zhang et al. 2018a,b; Li et al. 2018; Zhang et al. 2020; Li et al. 2020b). However, their parameters and calculations are also huge and are difficult to be used on mobile devices. To address this issue, many lightweight SISR models have also been proposed. For instance, CARN (Ahn, Kang, and Sohn 2018) is a lightweight residual network composed of multiple residual connections. ECBSR (Zhang, Zeng, and Zhang 2021) is a lightweight and efficient network whose features are extracted in multiple paths. The purpose of these models is to reduce the complexity of the model and facilitate the application in the real world. The demand for lightweight practical models motivates us to propose the Feature Distillation Interaction Weighted Network (FDIWN). The computational costs of FDIWN are lower than most existing lightweight SISR models, but it is not inferior to them in terms of performance.

As we know, as the depth of the network increases, information will be lost during transmission. Therefore, under the constraints of parameters and calculations, how to prevent information loss, and how to make full use of intermediate features is important. To achieve this, we introduce Wide-residual Distillation Interaction Blocks (WDIB) in the Feature Shuffle Weighted Group (FSWG) for pairwise feature fusion, and then the features are shuffled and weighted. This operation can improve model performance while only increasing a small amount of computational cost since the wide-residual attention weighting units are extensively used in WDIB, including Wide Identical Residual Weighting (WIRW) units and Wide Convolutional Residual Weighting (WCRW) units. WIRW and WCRN allow more features to pass and be activated, thereby increasing the transmission and utilization of the features. Meanwhile, the carefully designed Self-Calibration Fusion (SCF) unit integrates different levels of features by the jump splicing strategy to achieve a good SR reconstruction. In general, our

*Corresponding author, †Equal contributions.

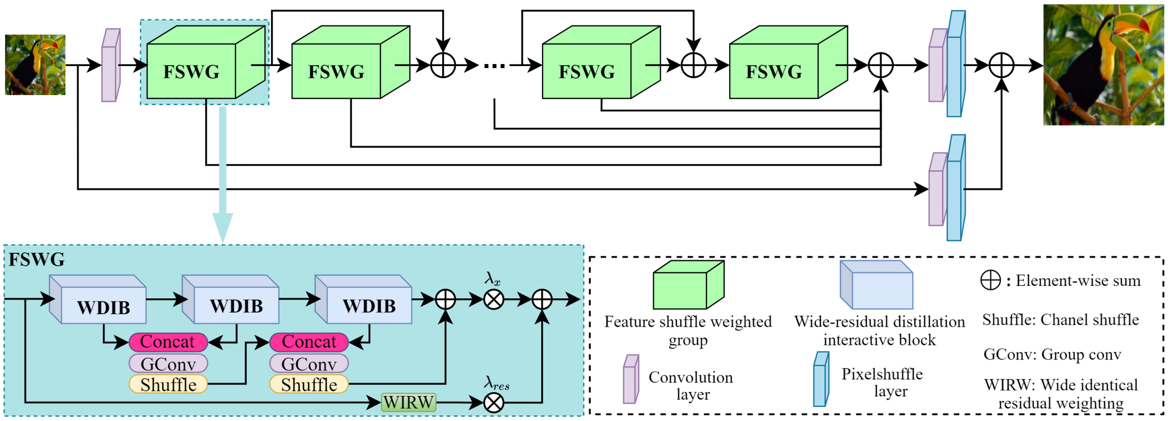

Figure 1: The architecture of the proposed Feature Distillation Interaction Weighting Network (FDIWN).

main contributions can be summarized as follows:

- We propose a wide-residual attention weighting unit for lightweight SISR, including Wide Identical Residual Weighting (WIRW) unit and Wide Convolutional Residual Weighting (WCRW) unit, which has stronger feature distillation capabilities than ordinary residual blocks.
- We propose a novel Self-Calibration Fusion (SCF) module to replace the traditional concatenate operation for efficient feature interaction and fusion, which can aggregate more representative features and self-calibrate the input and output features.
- We propose a Wide-Residual Distillation Connection (WRDC) framework, which connects the coarse and distilled fine features within the module and allows features from different scales to interact with each other.
- We design a Feature Shuffle Weighted Group (FSWG) for pairwise feature fusion, which consists of a series of interactional WDIBs. Meanwhile, it serves as a basic component of the proposed Feature Distillation Interaction Weighting Network (FDIWN).

Related Work

Lightweight SISR

Recently, more and more effective deep neural networks have been introduced into SISR. However, most of them are often accompanied by a large number of model parameters and need large calculation costs, which limits their applications on mobile devices. To address this issue, researchers began to explore lightweight and efficient SISR methods. For instance, CARN-M (Ahn, Kang, and Sohn 2018) used group convolution to reduce the model parameters, which even achieved better super-resolved effects than some large SISR models. Hui *et al.* (Hui, Wang, and Gao 2018) presented the Information Distillation Network (IDN) to extract more useful information with fewer convolutional layers. Meanwhile, IMDN (Hui et al. 2019) was modified based on the IDN with a faster and lighter structure. After that, RFDN (Liu, Tang, and Wu 2020) changed the channel splitting method based on IMDN and adopted skip connections for the convolutional layers in the residual block. Wang *et al.* (Wang, Li, and Shi 2019) proposed an adaptive weighted super-resolution network with efficient residual learning and local residual fusion. Wang (Wang et al. 2021) proposed a multi-scale feature interaction network (MSFIN) for lightweight SISR. IMRN (Jiang et al. 2021) uses the pruning method to reduce the model size without significantly reducing the performance, and achieves good performance. Nowadays, lightweight SISR is getting more and more important since its great application value. Although the aforementioned models achieved good results, they ignored the use of intermediate features, resulting in sub-optimal performance.

Wide-Residual Attention Weighting Learning

Studies have shown that the deeper the network, the better the performance of the model. However, it was later discovered that as the number of the network layers increased, the performance of the model does not rise but falls. To solve this problem, the residual block was introduced into the network, thus the network can reach very deep, and the effect of the network will also become better. This method is also used in many SISR models. For example, VDSR (Kim, Lee, and Lee 2016a) is a 20 layers network, EDSR (Lim et al. 2017) is a 65 layers network, and RCAN (Zhang et al. 2018a) has more than 800 layers. All these models introduced various skip connections and concatenation operations between shallow layers and deep layers to make full use of the shallow feature information. Different from the above methods, Yu *et al.* (Yu et al. 2018) found that the model with wider features before ReLU activation can achieve better performance. Therefore, they proposed the WDSR with the wide activation mechanism, which expanded features before ReLU and allowed more information to pass through without additional parameters. Meanwhile, the attention mechanism has been widely used in deep learning tasks. For instance, Zhang *et al.* (Zhang et al. 2018a) and Dai *et al.* (Dai et al. 2019) proposed the first-order statistics and second-order attention networks to pursue better feature extraction. Inspired by this, we try to introduce the second-order attention mechanism into our

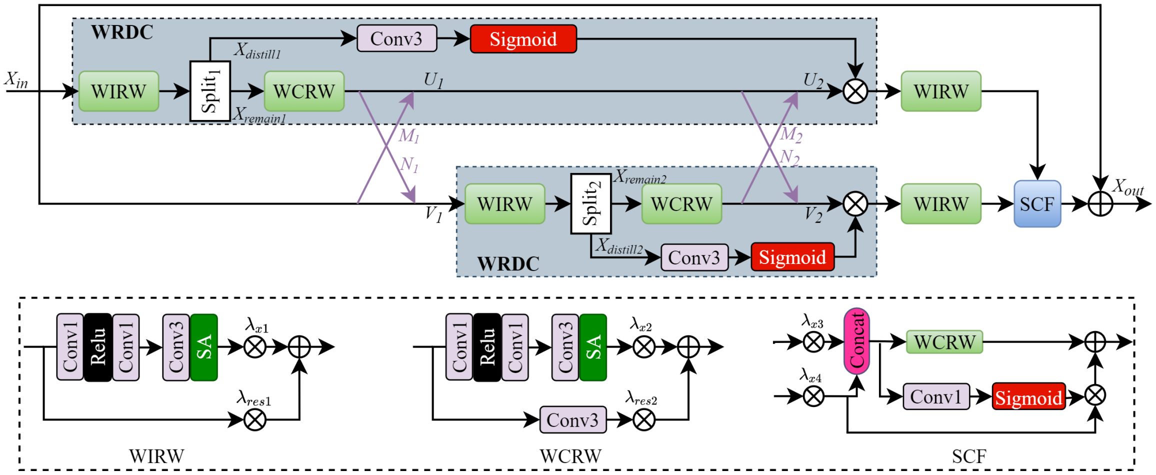

Figure 2: The structure of the proposed Wide-residual Distillation Interaction Block (WDIB). Conv1 and Conv3 represent the convolutional layer with the kernel size of 1 and 3, respectively.

modified wide activation residual block to further improve the feature extraction ability of the model.

Proposed Method

To build a lightweight and accurate SISR model, we focus on the use of intermediate features and the interaction of feature information. To achieve this, we propose a Feature Distillation Interaction Weighting Network (FDIWN). FDIWN consists of a series of novel and efficient modules, including Wide-residual Distillation Interaction Block (WDIB), Wide-Residual Distillation Connection (WRDC), Wide Identical Residual Weighting (WIRW) unit, Wide Convolutional Residual Weighting (WCRW) unit, and Self-Calibration Fusion (SCF) module.

Feature Distillation Interaction Weighting Network

As shown in Figure 1, FDIWN consists of three parts: the shallow feature extraction part, the non-linear deep feature acquisition part, and the upsampling recovery part. Following previous works, we use a 3×3 convolutional layer to extract the shallow features X_0 from the input LR image

$$X_0 = C_e(I_{LR}), \tag{1}$$

where C_e represents the feature extraction layer, I_{LR} is the LR image, and X_0 is the extracted shallow features.

After that, the non-linear feature mapping module is followed, which is formed by several Feature Shuffle Weighted Groups (FSWGs) through jump connections. The operation can be expressed as follow

$$X_1 = F^0_{FSWG}(X_0), \tag{2}$$

$$X_{n-1} = F^{n-2}_{FSWG}(\dots(F^1_{FSWG}(X_1) + X_1)\dots) + X_{n-2}, \tag{3}$$

$$X_n = F^{n-1}_{FSWG}(X_{n-1}), \tag{4}$$

where F^k_{FSWG} represents the k-th FSWG, and X_n denotes the extracted non-linear deep features.

The features used for the final SR image reconstruction in the upsampling recovery module come from two parts, one comes from the non-linear deep feature extraction module and the other comes from the input LR image. We hope that superimposing the low-frequency and high-frequency feature information in this way can reconstruct high-quality SR images with more texture details. Therefore, the final SR image can be expressed as

$$I_{SR} = F_{UP1}(\sum_{i=0}^{n-1} F^i_{FSWG}(X_i)) + F_{UP2}(I_{LR}), \tag{5}$$

where I_{SR} is the reconstructed SR image, F_{UP1} and F_{UP2} represent the upsampling modules.

Wide-Residual Distillation Connection

As shown in Figure 2, Wide-Residual Distillation Connection (WRDC) is an important component in the model, which consists of the Wide Identical Residual Weighting (WIRW) unit, the Wide Convolutional Residual Weighting (WCRW) unit, and the distillation jump connection. Both WIRW and WCRW introduced the wide activation mechanism, thus it can distill richer features with fewer parameters. Recently, with the emphasis on the importance of channel attention in RCAN (Zhang et al. 2018a), many SR methods focus on the attention mechanism. As shown in Figure 3, Zhang et al. (Zhang and Yang 2021) proposed a new attention paradigm, which is a combination of spatial attention and channel attention, called Shuffle Attention (SA). Inspired by this, we introduce the SA into our WIRW and WCRW to further enhance their feature extraction abilities. Since the SA mechanism is placed in each wide-residual unit, we set the numbers of the group g to be large enough to keep the SA lightweight. After the channel splitting operation, the number of input channels of WCRW is only half of the original input. Therefore, compared with WIRW, a 3×3 convolutional layer is added to the shortcut path of WCRW to increase the number of output channels so that it

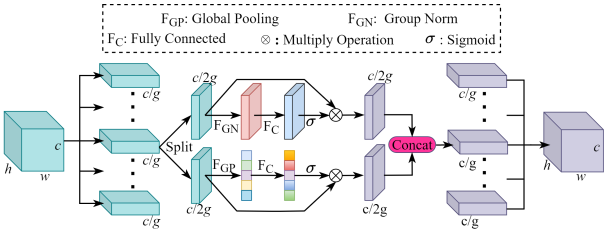

Figure 3: The principle of shuffle attention (SA) mechanism. h, w, c, and g represent the height, width, number of channels, and number of groups, respectively.

can match the original input size and achieve the interaction between different features. Meanwhile, WCRW and WIRW both introduce adaptive weights operation in the main path and shortcut path for adaptive feature learning. When input I is fed into WIRW and WCRW, the output X_{WIRW} and X_{WCRW} can be expressed as

$$X_{WIRW} = \lambda_{x1} F_{SA}[F_{CR}(I)] + \lambda_{res1} I, \tag{6}$$

$$X_{WCRW} = \lambda_{x2} F_{SA}[F_{CR}(I)] + \lambda_{res2} F_{C3}(I), \tag{7}$$

where λ_{xk} and λ_{resk} $(k = 1, 2)$ represent the adaptive multiplier of the k-th wide-residual unit branch, F_{SA} represents the SA operation, F_{CR} represents a series of (conv + relu) operations before the attention mechanism, and F_{C3} represents the 3×3 convolutional layer. The complete structure of WIRW and WCRW can be seen in Figure 2.

Apart from the above operation, the distillation connection part is applied to segment the channel features through the convolutional layer and Sigmoid function. The convolutional layer is introduced to expand the dimension of the splitting channel, while the Sigmoid function non-linearizes the obtained coarse high-frequency features to obtain fine features maps. Finally, these features are multiplied with the low-frequency attention features obtained after the wide-residual unit refinement process to realize the interaction of the features from different scales.

Wide-Residual Distillation Interaction Block

Inspired by the lattice block (LB) (Luo et al. 2020), we design a Wide-Residual Distillation Interaction Block (WDIB) based on WRDC. As shown in Figure 2, WDIB utilizes the butterfly structure described in LB to realize the interaction of intermediate features. Define W_{ir} and W_{cr} represent the WIRW and WCRW units, the first butterfly structure can be expressed as

$$X_{remain1}, X_{distill1} = Split_1(W_{ir}(X_{in})), \tag{8}$$

$$U_1 = M_1 \langle X_{in} \rangle + W_{cr}(X_{in}), \tag{9}$$

$$V_1 = N_1 \langle W_{cr}(X_{remain1}) \rangle + X_{in}, \tag{10}$$

where $Split_i(\cdot)$ represents the i-th channel splitting operation, $X_{remaini}$ represents the rough feature of the input subsequent wide-residual unit, and $X_{distilli}$ represents the i-th refinement feature that is split out and jump-connected with the next butterfly structure. The combination coefficients M_i and N_i are the two vectors connecting the upper and lower branches. It is worth noting that $M \langle X_{in} \rangle = M(X_{in}) \times X_{in}$, and the learning details of the combination coefficients M_i and N_i are provided in Figure 4. U_i and V_i are the output features after the i-th butterfly structure, and then they are fed into the second butterfly structure

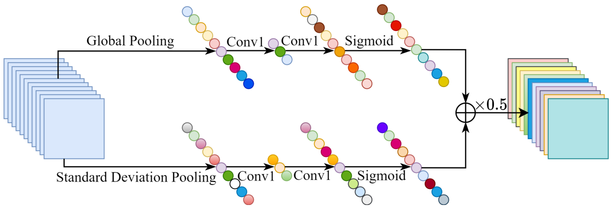

Figure 4: The diagram of combination coefficient learning.

$$X_{remain2}, X_{distill2} = Split_2(W_{ir}(V_1)), \tag{11}$$

$$U_2 = M_2 \langle W_{cr}(X_{remain2}) \rangle + U_1, \tag{12}$$

$$V_2 = N_2 \langle U_1 \rangle + W_{cr}(X_{remain2}). \tag{13}$$

After that, the output X_{out} can be expressed as

$$X_{out1} = W_{ir}(U_2 \times S_3(X_{distill1})), \tag{14}$$

$$X_{out2} = W_{ir}(V_2 \times S_3(X_{distill2})), \tag{15}$$

$$X_{out} = C_{SCF}[X_{out1}, X_{out2}] + X_{in}, \tag{16}$$

where C_{SCF} represents the proposed Self-Calibration Fusion (SCF) module. X_{out1} and X_{out2} represent the upper branch and the lower branch entering the SCF module, respectively. In addition, S_k represents a $k \times k$ convolutional layer followed by a cascade of Sigmoid function. The structure of the SCF module is provided in Figure 2. Obviously, the features of the upper and lower branches are first fused, and then the different scale features from two branches are interacted and fused. The output X_{SCF} of the SCF module can be expressed as

$$X_{concat} = C_{Concat}[\lambda_{x3} X_{out1}, \lambda_{x4} X_{out2}], \tag{17}$$

$$X_{SCF} = \lambda_{x4} X_{out2} \times S_1(X_{concat}) + W_{cr}(X_{concat}), \tag{18}$$

where C_{Concat} represents the Concat operation, λ_{x3} and λ_{x4} represent the adaptive weights, and X_{concat} represents the output after the upper and lower branches are multiplied by the adaptive weight and then the Concat is performed. Subsequently, the fused features are nonlinearized, then multiplied with the features of the lower branch, and finally added to the refined fusion features to achieve the interaction of features from different scales. Since there are a large number of adaptive multipliers in the module, the output features can be adjusted and calibrated continuously during the training, thus it can achieve better performance than the traditional Concat operation.

Feature Shuffle Weighted Group

As shown in Figure 1, Feature Shuffle Weighted Group (FSWG) consists of three interactional WDIBs and it serves as the basic component of FDIWN. Specifically, we fuse and shuffle the features extracted by WDIB one by one. The cascaded operation F_{CGS} can be expressed as

$$F_{CGS} = F_{Shuffle}(F_{GConv}(C_{Concat}[x_i, x_{i+1}])), \tag{19}$$

where $F_{Shuffle}$ represents the channel shuffle operation and F_{GConv} represents the group convolution operation. x_i and x_{i+1} represent the two features to be merged. After that, we add the shuffled and fused features with the original features that have not been operated to achieve the interaction of feature information. Meanwhile, we set a larger group in the shuffle and fusion operation to reduce the parameter burden. Moreover, to reduce the redundant information, the primary features and the features after the information interaction are self-adaptively fused to distill the desired important features. Define the input of FSWG as W_0, the output W_{out} can be formulated as

$$W_{CGS} = {F^2}_{CGS}({F^1}_{CGS}(W_1, W_2), W_3), \tag{20}$$

$$W_{out} = \lambda_x(W_{CGS} + W_3) + \lambda_{res} W_{ir}(W_0), \tag{21}$$

where W_i represents the output of the i-th WDIB, ${F^i}_{CGS}$ represents the i-th F_{CGS} operation, W_{CGS} represents the extracted features after a series of shuffle and fusion operations. In addition, λ_x and λ_{res} are used to adaptively adjust the weight of each channel. After these operations, the features from each WDIB are adequately interacted and distilled to achieve better SR image reconstruction.

Experiments

Datasets and Evaluation Metrics

Following previous works, we use the DIV2K (Agustsson and Timofte 2017) as the training dataset, which contains 800 pairs of images. For testing, we use Set5 (Bevilacqua et al. 2012), Set14 (Zeyde, Elad, and Protter 2010), BSDS100 (Martin et al. 2001), and Urban100 (Huang, Singh, and Ahuja 2015) to verify the effectiveness of the proposed FDIWN. Meanwhile, two metrics on the Y channel in the YCbCr color space, namely PSNR and SSIM are used to evaluate the model performance.

Implementation Details

Each mini-batch during the training consists of 16 RGB image blocks with the size of 48 × 48, which are randomly cropped from the LR image. Meanwhile, the training dataset is enhanced by random and horizontal rotation at different angles for data augmentation. The learning rate is initialized to 2e-4 and a total of 1000 epochs are updated. We implement our model with the PyTorch framework and update it with Adam optimizer. All our experiments are performed on NVIDIA RX 2080TI GPUs.

As for the model set, the final version of FDIWN consists of 6 FSWGs, while the tiny version of FDIWN-M only consists of 4 FSWGs. The number of input channels is initialized to 24 and the value of the adaptive weight is 1.

Ablation Studies

To verify the efficiency and effectiveness of the proposed modules, we conducted a series of ablation studies and all of these studies are tested on the Set5 dataset.

The effectiveness of WRDC and SCF. To verify the effectiveness of WRDC and SCF, we replace the WRDC module in FDIWN with a three-layer cascaded 3 × 3 convolution plus ReLU layer, and replace the SCF module with the concatenate operation. We set this structure as the Baseline1. Baseline2 has a similar structure while the three-layer cascaded convolution plus ReLU layer is placed by the cascaded WIRW plus WCRW units. Then we add WRDC and SCF step by step and compare their performance with the Baseline1. It can be observed from Table 1 that our proposed FDIWN improves the performance of Baseline1 by 0.12 dB, which proves the effectiveness of the WRDC and SCF module. Specifically, with the DC mechanism, PSNR is increased from 37.53 dB to 37.59 dB, while the number of parameters only increases 6K. Meanwhile, the SCF module can provide a 0.06 dB improvement in model performance with an acceptable increase in the number of parameters.

Table 1: Impact analysis of WRDC and SCF. WR: Wide-Residual, DC: Distillation Connection, SCF: Self-Calibration Fusion.

Method	WR	DC	SCF	Params	Multi-adds	PSNR	SSIM
Baseline1	✗	✗	✗	59K	3.3G	37.52	0.9587
Baseline2	✓	✗	✗	59K	4.9G	37.53	0.9589
FDIWN	✗	✗	✓	89K	6.5G	37.58	0.9591
FDIWN	✓	✓	✗	65K	6.5G	37.59	0.9590
FDIWN	✓	✓	✓	96K	9.7G	**37.64**	**0.9592**

Table 2: Impact analysis of WIRW and WCRW.

Case	Method	Channels	Params	Multi-adds	PSNR	SSIM
1	Baseline	24	152K	23.2G	37.70	0.9594
2	FDIWN	48	96K	9.7G	37.64	0.9592
3	FDIWN	120	131K	9.7G	**37.72**	**0.9596**

The effectiveness of WIRW and WCRW. To evaluate the role of our wide-residual units in the module, we replaced all WIRW and WCRW units in our module with the basic residual block and treat this structure as the baseline model. The kernel size of the convolutional layer in the basic residual block is 3 × 3. To explore the impact of the number of channels before the activation function in our designed wide-residual units on the SR performance, we set the number of channels as 48 and 120, respectively. Due to the lightweight character of the 1×1 convolution function, we can see from Table 2 that i) Compared with the Baseline model (Case 1 and 3), FDIWN achieves better performance with fewer parameters and computational costs; ii) Increasing the number of channels (Case 2 and 3), the model performance can be further improved. This fully demonstrates the effectiveness of the introduced wide activation mechanism and the proposed WIRW and WCRW.

To further verify the impact of WIRW and WCRW units on SISR, we visualize the features produced by the different numbers of WIRW and WCRW units. The network here is composed of one 3 × 3 convolution and several wide-residual units. Figure 5 shows that the 3 × 3 convolution can only extract shallow image information but cannot extract image details well. However, it can be seen that as the

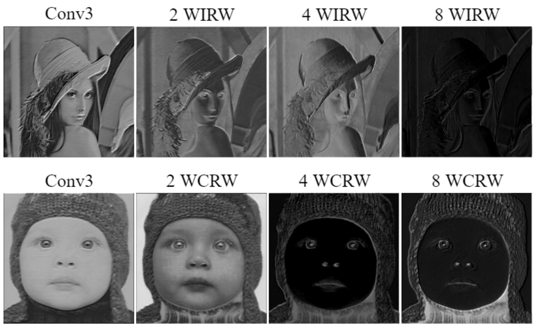

Figure 5: Feature visualization of different module.

Table 3: Evaluation of the combination structure of WDIB. The best results are **highlighted**. BI: Block Interaction.

Method	BI	WIRW	Params	Multi-adds	PSNR	SSIM
Baseline	✗	✗	215K	22.0G	37.81	0.9598
FDIWN	✓	✗	225K	24.4G	37.85	**0.9600**
FDIWN	✓	✓	230K	24.4G	**37.88**	**0.9600**

number of WIRW and WCRW units increases, more edge detail information will be extracted. This also means that as the number of WIRW and WCRW units increases, the ability of the model to capture high-frequency information will be greatly improved. This further demonstrates the effectiveness of WIRW and WCRW.

The combination structure of WDIB. The FSWG in FDIWN is composed of three WDIBs, which are interacted with each other to yield more representative features. To verify the effect of this combination of WDIB, we chose three cascaded WDIBs as the baseline. In other words, three WDIBs are simply connected without any interaction. According to Table 3, we can observe that the information blending structure we used is more effective. It improves the PSNR from 37.81 dB to 37.85 dB with limited computational costs, which further proves the effectiveness of the proposed information interaction structure. In addition, the long-skip connection provided by the WIRW unit can further improve model performance. All these experiments show that the combined structure of WDIB is effective.

Efficiency trade-off. In Figure 6, we show the performance change of the model under different numbers of FWSGs and WDIBs. Among them, the circular point Gm denotes that m FSWGs are cascaded. According to the figure, we can observe that i) As the number of FSWG and WDIB increases, the model performance can be further improved; ii) The PSNR does not increase when the number of WDIB increases from 3 to 4. Therefore, we use 6 FSWGs and 3 WDIBs in the final version model to achieve a good balance between model performance, size, and computational costs.

Model complexity analysis. In Figure 7, we show the execution time comparisons between our FDIWN with other classic lightweight SISR models. Obviously, FDIWN achieves competitive results with fewer parameters. Although our FDIWN is not the fastest model, the execution time is acceptable. Meanwhile, we show the multi-adds comparison between FDIWN and other SISR methods in Figure 8. Obviously, our FDIWN achieves the best balance. Therefore, we can draw a conclusion that FDIWN gains a better trade-off between model size, performance, inference speed, and multi-adds.

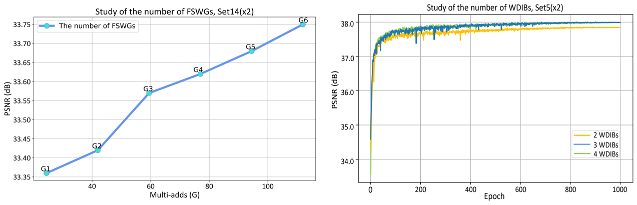

Figure 6: Study of different numbers of FSWGs and WDIBs.

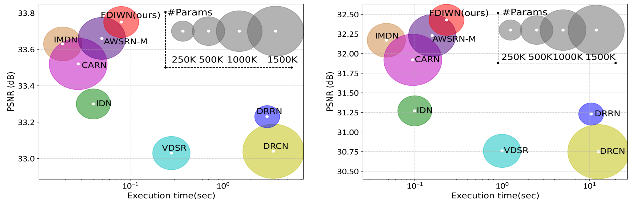

Figure 7: Inference speed study on Set14 (left) and Urban100 (right) with x2 SR.

Comparisons with State-of-the-Art Methods

We compare our proposed FDIWN-M and FDIWN with several representative state-of-the-art methods in this section. In Table 4, we provide the detail quantitative comparison. According to the table, we can observe that i) Models with a similar number of parameters to our model perform worse than ours; ii) The models with the same effects have more parameters than ours. Therefore, we can draw a conclusion that our proposed FDIWN-M and FDIWN stand out from these methods and perform very competitively in balancing model size, performance, and computational cost.

Figure 9 allows us to visually compare our method with other advanced methods on the Urban100 dataset. Through a horizontal comparison of the SR results, we can qualitatively see the advantages of our proposed method. Meanwhile, their PSNR and SSIM results are also provided. Our method not only has better visual details but also outperforms existing advanced methods in terms of quantitative data comparisons. All these results further prove the effectiveness and excellence of the proposed FDIWN.

Conclusions

In this paper, we proposed an effective and lightweight Feature Distillation Interaction Weighting Network (FDIWN) for SISR. Compared to other lightweight SISR models, FDIWN not only reduces the computational overhead but also improves the SR performance. In summary, the improvement of our FDIWN is mainly due to the following parts: (i) The specially designed wide-residual weighting units (including WIRW and WCRW) have a stronger ability to distill useful features than ordinary residual blocks; (ii)

Table 4: Quantitative comparisons of the state-of-the-art models with two scales on Set5, Set14, BSD100, and Urban100 datasets. The best and second-best results are marked in red and blue colors, respectively.

Algorithm	Scale	Params	Multi-adds	Set5		Set14		BSDS100		Urban100	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SRCNN (Dong et al. 2015)	×3	57K	52.7G	32.75	0.9090	29.30	0.8215	28.41	0.7863	26.24	0.7889
FSRCNN (Dong, Loy, and Tang 2016)		12K	5.0G	33.16	0.9140	29.43	0.8242	28.53	0.7910	26.43	0.8080
VDSR (Kim, Lee, and Lee 2016a)		665K	612.6G	33.67	0.9210	29.78	0.8320	28.83	0.7990	27.14	0.8290
DRCN (Kim, Lee, and Lee 2016b)		1774K	17974.3G	33.82	0.9226	29.76	0.8311	28.80	0.7963	27.15	0.8276
IDN (Hui, Wang, and Gao 2018)		590K	105.6G	34.11	0.9253	29.99	0.8354	28.95	0.8013	27.42	0.8359
CARN-M (Ahn, Kang, and Sohn 2018)		412K	46.1G	33.99	0.9236	30.08	0.8367	28.91	0.8000	27.55	0.8385
CARN (Ahn, Kang, and Sohn 2018)		1592K	118.8G	34.29	0.9255	30.29	0.8407	29.06	0.8034	28.06	0.8493
IMDN (Hui et al. 2019)		703K	71.5G	34.36	0.9270	30.32	0.8417	29.09	0.8046	28.17	0.8519
AWSRN-M (Wang, Li, and Shi 2019)		1143K	116.6G	34.42	0.9275	30.32	0.8419	29.13	0.8059	28.26	0.8545
MADNet (Lan et al. 2020)		930K	88.4G	34.16	0.9253	30.21	0.8398	28.98	0.8023	27.77	0.8439
RFDN (Liu, Tang, and Wu 2020)		541K	55.4G	34.41	0.9273	30.34	0.8420	29.09	0.8050	28.21	0.8525
MAFFSRN (Muqeet et al. 2020)		418K	34.2G	34.32	0.9269	30.35	0.8429	29.09	0.8052	28.13	0.8521
LAPAR-A (Li et al. 2021)		594K	114G	34.36	0.9267	30.34	0.8421	29.11	0.8054	28.15	0.8523
FDIWN-M (Ours)		446K	35.9 G	34.46	0.9274	30.35	0.8423	29.10	0.8051	28.16	0.8528
FDIWN (Ours)		645K	51.5G	34.52	0.9281	30.42	0.8438	29.14	0.8065	28.36	0.8567
SRCNN (Dong et al. 2015)	×4	57K	52.7G	30.48	0.8628	27.49	0.7503	26.90	0.7101	24.52	0.7221
FSRCNN (Dong, Loy, and Tang 2016)		12K	4.6G	30.71	0.8657	27.59	0.7535	26.98	0.7150	24.62	0.7280
VDSR (Kim, Lee, and Lee 2016a)		665K	612.6G	31.35	0.8838	28.01	0.7674	27.29	0.7251	25.18	0.7524
DRCN (Kim, Lee, and Lee 2016b)		1774K	17974.3G	31.53	0.8854	28.02	0.7670	27.23	0.7233	25.14	0.7510
LapSRN (Lai et al. 2017)		813K	149.4G	31.54	0.8850	28.19	0.7720	27.32	0.7280	25.21	0.7560
IDN (Hui, Wang, and Gao 2018)		590K	81.9G	31.82	0.8903	28.25	0.7730	27.41	0.7297	25.41	0.7632
CARN-M (Ahn, Kang, and Sohn 2018)		412K	32.5G	31.92	0.8903	28.42	0.7762	27.44	0.7304	25.62	0.7694
CARN (Ahn, Kang, and Sohn 2018)		1592K	90.9G	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837
IMDN (Hui et al. 2019)		715K	40.9G	32.21	0.8948	28.58	0.7811	27.56	0.7353	26.04	0.7838
AWSRN-M (Wang, Li, and Shi 2019)		1254K	72.0G	32.21	0.8954	28.65	0.7832	27.60	0.7368	26.15	0.7884
MADNet (Lan et al. 2020)		1002K	54.1G	31.95	0.8917	28.44	0.7780	27.47	0.7327	25.76	0.7746
RFDN (Liu, Tang, and Wu 2020)		550K	31.6G	32.24	0.8952	28.61	0.7819	27.57	0.7360	26.11	0.7858
MAFFSRN (Muqeet et al. 2020)		441K	19.3G	32.18	0.8948	28.58	0.7812	27.57	0.7361	26.04	0.7848
ECBSR (Zhang, Zeng, and Zhang 2021)		603K	34.73G	31.92	0.8946	28.34	0.7817	27.48	0.7393	25.81	0.7773
LAPAR-A (Li et al. 2021)		659K	94G	32.15	0.8944	28.61	0.7818	27.61	0.7366	26.14	0.7871
FDIWN-M (Ours)		454K	19.6G	32.17	0.8941	28.55	0.7806	27.58	0.7364	26.02	0.7844
FDIWN (Ours)		664K	28.4G	32.23	0.8955	28.66	0.7829	27.62	0.7380	26.28	0.7919

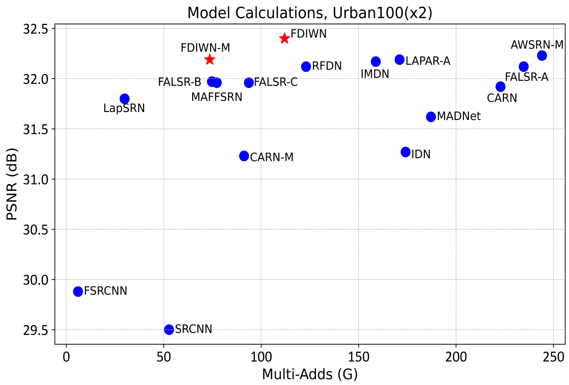

Figure 8: Investigations of the model size and performance.

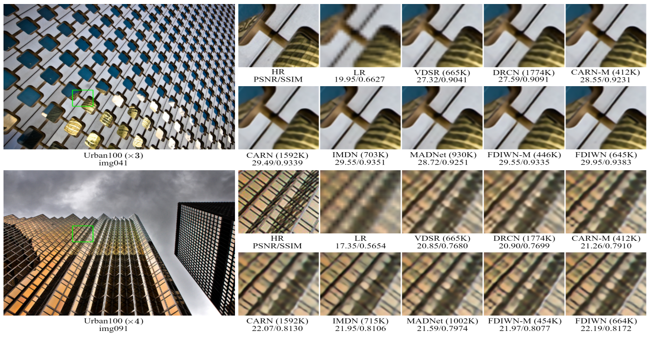

Figure 9: Visual comparisons on the Urban100 dataset. Due to the page limit, please zoom in for details.

The shuffle attention (SA) mechanism makes the feature extraction concentrating on the key information; (iii) The well-designed wide-residual units based WRDC module and SCF module can flexibly aggregate and distill more representative features, allowing features from different scales to efficiently interact with each other. Therefore, the contextual and intermediate features can be well interacted, which benefits high-quality SR image reconstruction. Evaluation results on benchmarks have shown that the proposed FDIWN achieved a good balance between model size, performance, and computational cost.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China (Nos. 2018AAA0100102 and 2018AAA0100100), the National Natural Science Foundation of China (Nos. 61972212, 61772568, 62076139 and 61833011), the Natural Science Foundation of Jiangsu Province (No. BK20190089), the Six Talent Peaks Project in Jiangsu Province (No. RJFW-011), and Research Project of Zhejiang Lab (No. 2021KF0AB05).

References

Agustsson, E.; and Timofte, R. 2017. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition Workshops*, 126–135.

Ahn, N.; Kang, B.; and Sohn, K.-A. 2018. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European Conference on Computer Vision*, 252–268.

Bevilacqua, M.; Roumy, A.; Guillemot, C.; and Alberi-Morel, M. L. 2012. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the British Machine Vision Conference*, 135.1–135.10.

Dai, T.; Cai, J.; Zhang, Y.; Xia, S.-T.; and Zhang, L. 2019. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11065–11074.

Dong, C.; Loy, C. C.; He, K.; and Tang, X. 2015. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2): 295–307.

Dong, C.; Loy, C. C.; and Tang, X. 2016. Accelerating the super-resolution convolutional neural network. In *Proceedings of the European Conference on Computer Vision*, 391–407.

Han, S.; Mao, H.; and Dally, W. J. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*.

Haris, M.; Shakhnarovich, G.; and Ukita, N. 2018. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1664–1673.

He, Z.; Cao, Y.; Du, L.; Xu, B.; Yang, J.; Cao, Y.; Tang, S.; and Zhuang, Y. 2019. Mrfn: Multi-receptive-field network for fast and accurate single image super-resolution. *IEEE Transactions on Multimedia*, 22(4): 1042–1054.

Hou, Q.; Zhou, D.; and Feng, J. 2021. Coordinate attention for efficient mobile network design. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13713–13722.

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

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

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

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

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

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

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

Lan, R.; Sun, L.; Liu, Z.; Lu, H.; Pang, C.; and Luo, X. 2020. Madnet: A fast and lightweight network for single-image super resolution. *IEEE Transactions on Cybernetics*, 51(3): 1443–1453.

Li, B.; Wang, B.; Liu, J.; Qi, Z.; and Shi, Y. 2020a. s-lwsr: Super lightweight super-resolution network. *IEEE Transactions on Image Processing*, 29: 8368–8380.

Li, J.; Fang, F.; Li, J.; Mei, K.; and Zhang, G. 2020b. MDCN: Multi-scale dense cross network for image super-resolution. *IEEE Transactions on Circuits and Systems for Video Technology*, 31: 2547–2561.

Li, J.; Fang, F.; Mei, K.; and Zhang, G. 2018. Multi-scale residual network for image super-resolution. In *Proceedings of the European Conference on Computer Vision*, 517–532.

Li, W.; Zhou, K.; Qi, L.; Jiang, N.; Lu, J.; and Jia, J. 2021. Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. *arXiv preprint arXiv:2105.10422*.

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

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

Luo, X.; Xie, Y.; Zhang, Y.; Qu, Y.; Li, C.; and Fu, Y. 2020. Latticenet: Towards lightweight image super-resolution with lattice block. In *Proceedings of the European Conference on Computer Vision*, 272–289.

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

Muqeet, A.; Hwang, J.; Yang, S.; Kang, J.; Kim, Y.; and Bae, S.-H. 2020. Multi-attention based ultra lightweight image super-resolution. In *Proceedings of the European Conference on Computer Vision*, 103–118.

Tian, C.; Xu, Y.; Zuo, W.; Zhang, B.; Fei, L.; and Lin, C.-W. 2020. Coarse-to-fine CNN for image super-resolution. *IEEE Transactions on Multimedia*, 23: 1489–1502.

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

Wang, Z.; Gao, G.; Li, J.; Yu, Y.; and Lu, H. 2021. Lightweight Image Super-Resolution with Multi-scale Feature Interaction Network. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, 1–6.

Yao, X.; Wu, Q.; Zhang, P.; and Bao, F. 2020. Weighted Adaptive Image Super-Resolution Scheme Based on Local Fractal Feature and Image Roughness. *IEEE Transactions on Multimedia*, 23: 1426–1441.

Yu, J.; Fan, Y.; Yang, J.; Xu, N.; Wang, Z.; Wang, X.; and Huang, T. 2018. Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*.

Zeyde, R.; Elad, M.; and Protter, M. 2010. On single image scale-up using sparse-representations. In *Proceedings of the International Conference on Curves and Surfaces*, 711–730.

Zhang, L.; Nie, J.; Wei, W.; Zhang, Y.; Liao, S.; and Shao, L. 2020. Unsupervised adaptation learning for hyperspectral imagery super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3073–3082.

Zhang, Q.-L.; and Yang, Y.-B. 2021. Sa-net: Shuffle attention for deep convolutional neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2235–2239.

Zhang, X.; Zeng, H.; and Zhang, L. 2021. Edge-oriented Convolution Block for Real-time Super Resolution on Mobile Devices. In *Proceedings of the ACM International Conference on Multimedia*.

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

Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; and Fu, Y. 2018b. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2472–2481.