This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

SLVR: Super-Light Visual Reconstruction via Blueprint Controllable Convolutions and Exploring Feature Diversity Representation

Ning Ni Libao Zhang*
School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China
ningni@mail.bnu.edu.cn libaozhang@bnu.edu.cn

Abstract

Recently, improving the residual structure and designing efficient convolutions have become important branches of lightweight visual reconstruction model design. We have observed that the feature addition mode (FAM) in existing residual structure tends to lead to slow feature learning or stagnation in feature evolution, a phenomenon we define as network inertia. In addition, although blueprint separable convolutions (BSConv) have proved the dominance of intra-kernel correlation, BSConv forces the blueprint to perform scale transformation on all channels, which may lead to incorrect intra-kernel correlation and introduce useless or disruptive features on some channels and hinder the effective propagation of features. Therefore, in this paper, we rethink the FAM and BSConv for super-light visual reconstruction framework design. First, we design a novel linking mode, called feature diversity evolution link (FDEL), which aims to alleviate the phenomenon of network inertia by reducing the retention of previous low-level features, thereby promoting the evolution of feature diversity. Second, we propose blueprint controllable convolutions (B2Conv). The B2Conv can adaptively pick accurate intra-kernel correlation in the depth-axis, effectively preventing the introduction of useless or disruptive features. Based on FDEL and B2Conv, we develop a super-light super-resolution (SR) framework SLVR for visual reconstruction. Both FDEL and B2Conv can serve as efficient plugins. Extensive experimental results demonstrate the effectiveness of our proposed B2Conv, FDEL, and SLVR. Our code will be available at https://github.com/chongningni/SLVR.

1. Introduction

As an important direction in the field of computer vision, visual reconstruction focuses on improving the visual quality of images or videos. Image super-resolution (SR) is a core task in visual reconstruction, which aims to reconstruct high-resolution details from a low-resolution image. SR is an ill-posed problem because a low-resolution (LR) image can be mapped to multiple high-resolution (HR) spaces, which makes it challenging. Recently, convolutional neural networks (CNNs) based on residual learning [1, 9, 21, 22, 30, 56] have dominated the SR field because they can constrain mapping space and reduce the difficulty of SR learning, which creates conditions for lightweight SR design. Furthermore, lightweight convolution has become an important research branch of lightweight model design. In this paper, we aim to rethink the residual structure and blueprint separable convolutions (BSConv) to build a super-light SR reconstruction framework. The issues we discovered are as follows:

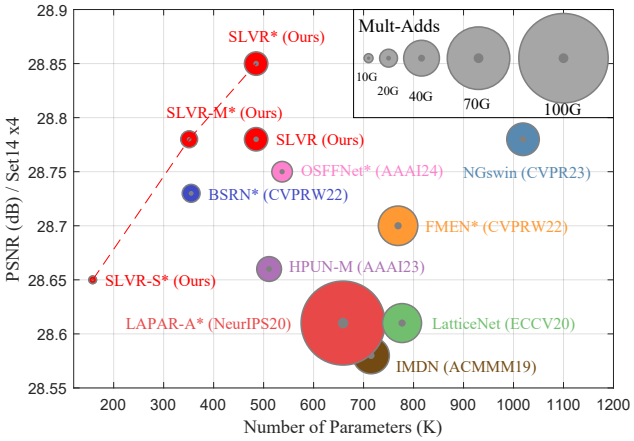


Figure 1. Performance and model complexity comparison on Set14 dataset for upscaling factor $\times 4$. * denotes the model trained on DF2K dataset and others trained on DIV2K.

1) Existing feature addition mode easily causes network inertia: SR tasks require recovering detailed textures, which require rich high-level features as support. The existing residual structure uses the feature addition mode (FAM) of input and output $(x + \mathcal{F}(x))$, retaining many previous low-level features (x), which is not conducive to the diver-

*Corresponding Author. This work is supported by the National Natural Science Foundation of China under Grant 62271060 and 61571050.

sity evolution of features and easily causes network inertia. $\mathcal{F}(\cdot)$ represents the calculation function. Furthermore, retaining redundant previous low-level features will also affect the presentation of high-level features.

2) Incorrect intra-kernel correlation of the BSConv: Although BSConv proves the dominance of intra-kernel correlation, we find that kernel correlation is not present on all channels in an ordinary convolution. However, BSConv forces the blueprint to perform linear transformations on all channels, which may lead to incorrect intra-kernel correlation and result in introducing many useless or disruptive features on some channels to hinder the effective propagation of features.

To solve the above problems, this work rethinks the residual structure and BSconv. First, to alleviate the network inertia, we propose a new linking mode, named feature diversity evolution link (FDEL). Our FDEL aims to minimize the proportion of low-level features during feature propagation and presentation, which is beneficial to distill rich high-level features with different representations and alleviate network inertia. In addition, to maximize the effect of FDEL, we introduce a diversified feature aggregation strategy (DFAS) to enhance feature representation. Second, to prevent the incorrect inter-kernel correlation of the BSConv (inter-blueprint correlation), we propose new blueprint controllable convolutions (B2Conv). Our B2Conv sets up an access control unit (ACU) in the channel dimension to prevent incorrect intra-blueprint correlation. The ACU is like a guard, which can adaptively pick relevant channels to associate and learn the correct inter-blueprint correlation. Our B2Conv can reduce the introduction of useless or disruptive features and promote the effective propagation of feature information.

The main contributions of this paper are summarized as:

1) We propose a new convolution B2Conv. Our B2Conv introduces the ACU to adaptively control intra-blueprint correlation, which can precisely prevent the introduction of useless or disruptive features and promote effective propagation of feature information. Besides, the B2Conv can also serve as a highly efficient building block for CNNs.

2) We propose a new linking mode FDEL, which effectively alleviates the network inertia problem. Our FDEL aims to reduce the retention of previous low-level feature information and can effectively promote the diverse evolution of features, which is beneficial to the extraction and presentation of high-level features.

3) Based on FDEL and B2Conv, we design a novel lightweight SR framework SLVR. Extensive experiments demonstrate that SLVR achieves record high performance (*e.g.*, ***28.85***dB *on Set14* $\times 4$ *with only* ***485K*** *parameters*) on lightweight SR benchmarks (shown in Fig. 1).

2. Related Works

Residual Network: Inspired by [47], He *et al*. [13] proposed first residual network for image recognition and achieved state-of-the-art performance. Then, many different residual networks were proposed [8, 14, 36, 42, 46] for different computer vision tasks. Residual networks are typically built by stacking a sequence of residual blocks. Recently, the residual block was further developed, such as widening the channels [60], introducing attention mechanism [17, 55], and combining with dense connections [64]. The residual network has achieved great success in heavyweight networks because it overcomes the vanishing gradient problem. Besides, residual learning can reduce the learning difficulty of computer vision tasks, which also provides opportunities for lightweight network design. To reduce the model complexity, MobileNet [16] introduced depthwise separable convolutions as key elements in the residual blocks. Then, MobileNetV2 [43] proposed an inverted residual block where the shortcut connections are between the thin bottleneck layers. ShuffleNetV2 [38] inserts a channel split component before the inverted residual block. After that, Zhou *et al*. [67] rethought the inverted residual block and proposed a sandglass block, which aims to perform spatial transformation and identity mapping at higher dimensions to alleviate information loss and gradient confusion problems. Although previous works have effectively improved the residual structure, most proposed residual blocks now still use the addition mode of input and output features.

Efficient Convolution: To reduce network parameters, AlexNet [25] first proposed the concept of group convolution and ResNext [57] demonstrated its effectiveness. After that, the group convolution was further modified by this works [18, 52, 62, 63]. VGG [45] suggested using two 3×3 convolutions instead of one 5×5 convolution. Xception [5] proposed the depthwise separable convolution (DSConv) to further reduce the parameters of convolution and then it was used in MobileNet [16] and gained state-of-the-art results. Partial convolution (PConv) [33] was first proposed for image inpainting, which greatly improved computing efficiency. To solve the unreasonable problem of mask update in the partial convolution, Gated convolution (GConv) [59] was proposed. Octave convolution [3] was proposed to extract the high-frequency and low-frequency signals of the image. Recently, DSConv was decomposed [41] into two components: variable quantization kernels and distribution offsets, thereby achieving memory reduction and operation acceleration. BSConv [12] rethought the DSConv and considered the dominance of intra-kernel correlations when performing convolution operations.

Lightweight SR: Although deep models can achieve good SR performance, they are not conducive to deployment on small devices. Therefore, many works were proposed

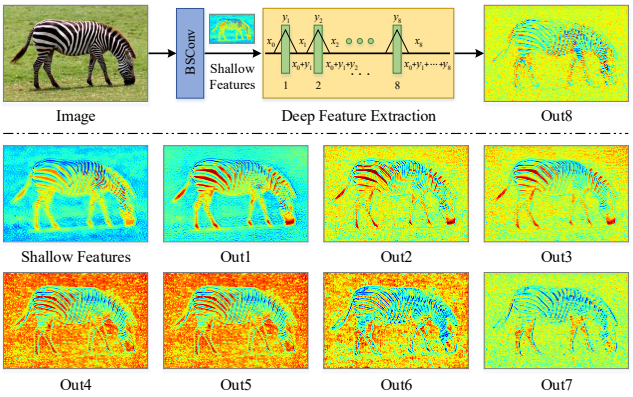


Figure 2. Visualization of intermediate features, above the dotted line is the pipeline of BSRN [30] and green cuboids represent the ESDB modules. Out$i$ represents the output of the i^{th} ESDB module. Out$i = x_i = x_{i-1} + y_i$.

[6, 20, 44] to reduce the computational complexity. LapSRN [26] removed the bicubic interpolation pre-processing step. CARN [1] introduced group convolution and cascading residual blocks. IMDN [21] used channel splitting operations to complete information multi-distillation and selective feature fusion. After that, improving information distillation structure has become an important branch of lightweight SR network design, and many methods [9, 30, 34] were proposed. Then, PAN [66] introduced pixel attention to measure pixel contribution to achieve competitive performance by using fewer parameters. Recently, a series of transformer-based SR methods [4, 7, 11, 29, 31, 35, 48, 51] emerge with superior performance. HNCT [7] cleverly combined CNN and transformer to reduce the parameters while ensuring network performance. CFIN [29] design a novel cross-receptive field-guided transformer to adaptively select the required contextual information. NGswin [4] introduced the N-Gram context to the low-level vision with transformers and achieve superior performance.

3. Methodology

3.1. Motivation

First, we define the phenomenon of network inertia as a situation where the input and output of a network module are similar, indicating that the module fails to learn meaningful features. As shown in Fig. 2, we find that the FAM $(x_{i-1} + y_i)$ easily causes network inertia because it retains many previous input low-level features (x_{i-1}) information. There are 8 ESDBs with this addition mode in BSRN, we can find that the outputs of most adjacent ESDB modules are similar, especially Out2 and Out3, Out4 and Out5. In addition, we replaced Out5 with Out4 in the subsequent feature fusion and were surprised to find that this replacement had almost no impact on the image reconstruction quality,

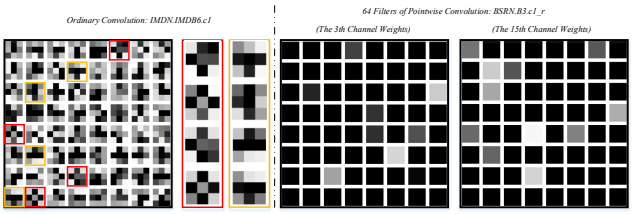


Figure 3. Exemplary filter weights of ordinary convolution and BSConv. One filter kernel of ordinary convolution size $M \times K \times K = 64 \times 3 \times 3$ is visualized and split into 64 images. Filter kernels of pointwise convolution size $N \times M \times K \times K = 64 \times 64 \times 1 \times 1$ and two channels are visualized.

and the PSNR only decreased by less than 0.005 dB. The similarity between features Out4 and Out5 can be inferred that the features y_4 extracted by the 4^{th} ESDB module are meaningless and network inertia occurs.

Second, as shown in the left of Fig. 3, although the weights of each kernel are highly correlated along the channel dimension, differences also exist. Besides, BSConv [12] also proved that not all the slices (52/128 for the ResNet-50 example) of the kernel are highly correlated along the depth axis. However, as shown in the right of Fig. 3, we find that the pointwise convolution of BSConv associates almost all the channel dimensions along the depth axis (the darker the color, the higher the correlation), which results in introducing many useless or disruptive features on some channels and is not conducive to effective extraction and propagation of useful features.

3.2. Entropy Delta Extraction Link

In this section, we will explain the definition of FDEL in detail.

In the network with stacked RBs, as shown in Fig. 5 (a), the initial shallow (low-level) features x_1 will be superimposed and retained until the last layer x_L, as follows

$$\begin{aligned} x_2 &= \mathcal{F}_1(x_1) + x_1, \cdots, \\ x_l &= \mathcal{F}_{l-1}(x_{l-1}) + x_{l-1}, \cdots, \\ x_L &= \mathcal{F}_{L-1}(x_{L-1}) + x_{L-1}, \end{aligned} \tag{1}$$

where $\mathcal{F}_l$ represents the operations of the l^{th} RB. Equation (1) can be regarded as a feature superposition operation. It may let the feature representation x_l of l^{th} RB rely too much on the input features x_{l-1}, which will limit the learning direction of features and even lead to stagnation in feature evolution. This phenomenon can be evidenced in Fig. 2. In addition, the superposition of many low-level input features is not conducive to the presentation of high-level features, especially in structure Fig. 5 (c).

In view of the aforementioned limitations of the FAM of the RBs, we rethink its design principles and propose an FDEL that can tackle the above issues by flipping the FAM,

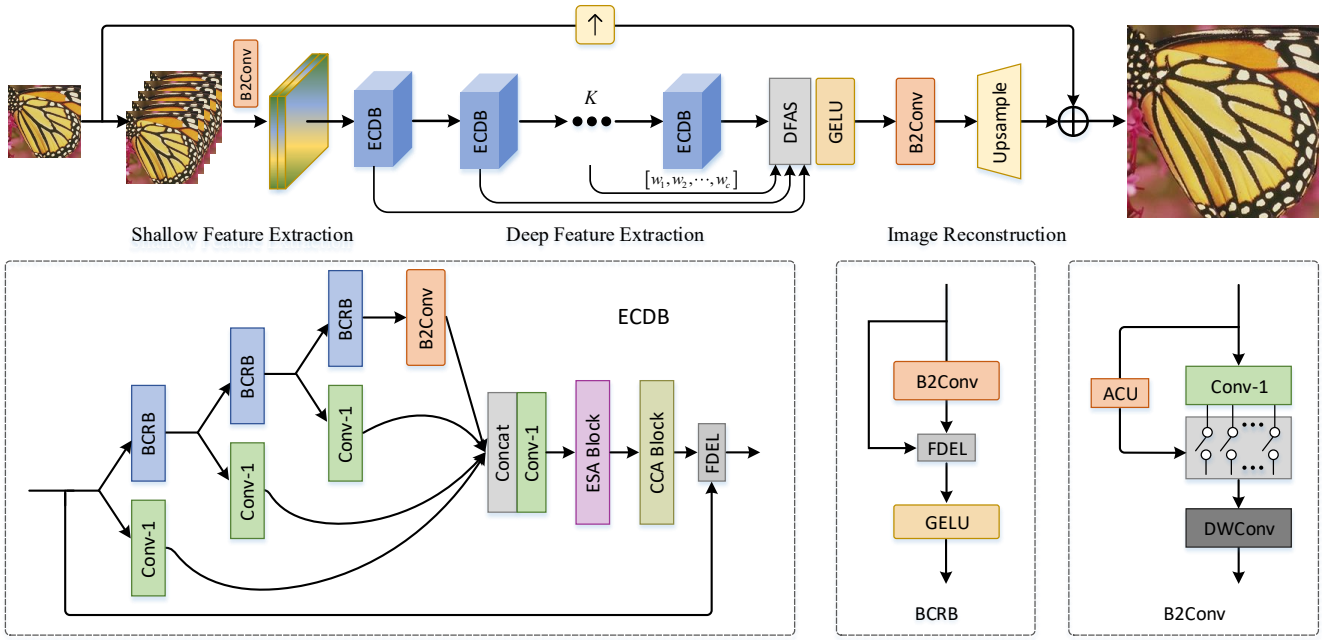


Figure 4. The overall architecture of the proposed SLVR and the structure of BCRB and blueprint controllable convolutions (B2Conv).

the structure of RB with FDEL is shown in Fig. 5 (b). Our FDEL aims to remove excessive low-level features. The specific mathematical formula of FDEL is

$$x_{l+1} = \mathcal{F}_{\text{FDEL}}(x_l, \mathcal{F}_l(x_l)), \tag{2}$$

where $\mathcal{F}_{\text{FDEL}}$ denotes the operations of the FDEL. Our FDEL uses the output of the $\mathcal{F}_l$ minus its input. Because the output feature representation is more high-level than the input, each RB can effectively extract the different high-level features, which promotes the diverse evolution of features. As we all know, we usually refer to the output of a function minus its input as the gain of the function in a physical sense. Therefore, in an RB with FDEL, the gain is the high-level features extracted by this RB.

3.3. Blueprint Controllable Convolutions

Although BSConv [12] demonstrates the importance of intra-kernel correlations in separating ordinary convolution, it does not consider the differences of the filters in the depth-axis. In other words, in some channel dimensions, this blueprint cannot obtain the original filter of the ordinary convolution through scale transformation, so filters after forced scale transformation don't effectively work. Therefore, we sparse the intra-kernel correlation and proposed a new B2Conv. Our B2Conv can adaptively pick appropriate channels along the depth axis to perform blueprint scale transformation, which can promote the extraction and efficient propagation of feature information.

In a convolution operation, assume that the input is $U \in \mathbb{R}^{M\times H\times W}$, the output is $V \in \mathbb{R}^{N\times H\times W}$, and the convolution kernels are $\left\{E^{(1)}, E^{(2)}, \cdots, E^{(N)}\right\}$ and $E^{(n)} \in \mathbb{R}^{M\times H\times W}$. Then the process can be expressed as

$$V_{n,:,:} = U * E^{(n)}, \tag{3}$$

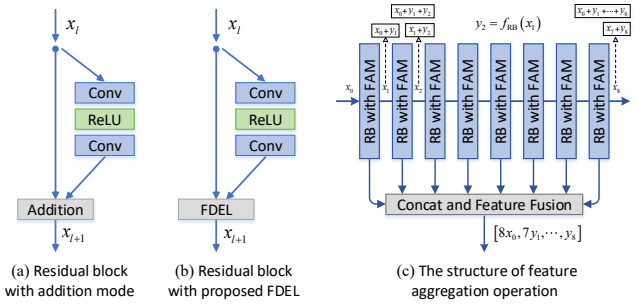
where $*$ denotes the convolution operation and $n \in \{1, 2, \cdots, N\}$, the weights of these N kernels can be learned during the training. BSConv proved that in practice these weights usually converge to a state that shows a substantial amount of correlation Therefore, BSConv proposes to decompose an ordinary convolution into the interaction of a blueprint $\mathcal{B}$ and a set of scale-transforming weights $\{w_{n,1}, w_{n,2}, \cdots, w_{n,M}\}$, as follows

$$E^{(n)}_{m,:,:} = w_{n,m} \cdot \mathcal{B}^{(n)}, \tag{4}$$

where $m \in \{1, 2, \cdots, M\}$ and $n \in \{1, 2, \cdots, N\}$ (see Fig. 6, left column). However, in ordinary convolution, not all the filters of channel dimension can be translated by Equation (4). Therefore, to solve the problem, we propose to add an access control unit (ACU) before scale-transforming. Specifically, in all channel dimensions, if there is a strong intra-kernel correlation with $\mathcal{B}$, we perform Equation (4), otherwise we do not perform any operation or multiply Equation (4) by a small correlation coefficient to ensure that the useless features extracted in this dimension will not affect the efficient propagation of features, as follows

$$E^{(n)}_{m,:,:} = \mathcal{T}_{n,m} \cdot w_{n,m} \cdot \mathcal{B}^{(n)}, \tag{5}$$

where $\mathcal{T}_{n,m}$ denotes the output of the ACU, ideally its value is 0 or 1. 1 means allowing scale-transforming and 0 means



(a) Residual block with addition mode (b) Residual block with proposed FDEL (c) The structure of feature aggregation operation

Figure 5. Comparison of residual blocks (RBs) with addition mode and FDEL. For the specific mathematical formula of FDEL, refer to Equation (2).

the opposite Here, to make the function continuous and differentiable, we soften the output of the ACU and limit its value range to $(0, 1)$. The closer the $\mathcal{T}_{n,m}$ is to 1, the stronger the intra-kernel correlation. The characteristics of features can reflect the characteristics of the convolution kernels. Therefore, $\mathcal{T}_{n,m}$ can be obtained in an embedding space by feature space mapping and weights of the $\mathcal{T}_{n,m}$ and $w_{n,m}$ can be learned directly via backpropagation. Specifically, we first rewrite the Equation (3) by split 3d convolution into a set of M 2d convolutions, yielding

$$V_{n,:,:} = U * E^{(n)} = \sum_{m=1}^{M} \left(U_{m,:,:} * E^{(n)}_{m,:,:} \right). \tag{6}$$

Secondly, we replace the ordinary convolution $E^{(n)}_{m,:,:}$ with the B2Conv repersentation as given in Equation (5), as follows

$$V_{n,:,:} = \sum_{m=1}^{M} \left(U_{m,:,:} * \left(\mathcal{T}_{n,m} \cdot w_{n,m} \cdot \mathcal{B}^{(n)} \right) \right). \tag{7}$$

Because each blueprint $\mathcal{B}^{(n)}$ is independent in the channel dimension. So, we can rewrite the Equation (7) (Note: As shown in right column of Fig. 6, the input $U_{m,:,:}$ also needs to be copied ($\bar{U}_{m,:,:}$) for feature space mapping)

$$V_{n,:,:} = \left(\underbrace{\sum_{m=1}^{M} \left(\bar{U}_{m,:,:} \cdot \mathcal{T}_{n,m} \right)}_{\bar{V}'_{n,:,:}} \cdot \underbrace{\sum_{m=1}^{M} \left(U_{m,:,:} \cdot w_{n,m} \right)}_{V'_{n,:,:}} \right) * \mathcal{B}^{(n)}. \tag{8}$$

Then, we further rearrange $\{w_{n,1}, w_{n,2}, \cdots, w_{n,M}\}$ and $\{\mathcal{T}_{n,1}, \mathcal{T}_{n,2}, \cdots, \mathcal{T}_{n,M}\}$ into a $M \times 1 \times 1$ arrays $\tilde{w}_n$ and $\tilde{\mathcal{T}}_n$ respectively, the sum operations of Equation (8) can be repalced by convolution operations, as follows

$$\bar{V}'_{n,:,:} = U * \tilde{\mathcal{T}}_n, \tag{9}$$

$$V'_{n,:,:} = U * \tilde{w}_n, \tag{10}$$

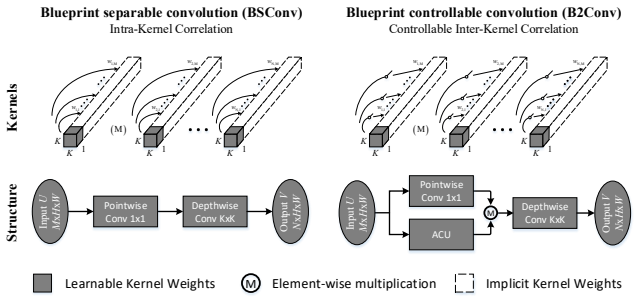


Figure 6. Interpretations for BSConv and B2Conv. B2Conv can adaptively pick correct intra-kernel correlations.

$$V_{n,:,:} = \left(\sigma \left(\bar{V}'_{n,:,:} \right) \cdot V'_{n,:,:} \right) * \mathcal{B}^{(n)}, \tag{11}$$

where σ denotes the sigmoid function, which limits $\bar{V}'_{n,:,:}$ to $(0, 1)$. For a concrete implementation, Equation (9) and Equation (10) can be translated into a 1×1 pointwise convolution with the kernels $\{\tilde{w}_1, \tilde{w}_2, \cdots, \tilde{w}_N\}$ and $\left\{\tilde{\mathcal{T}}_1, \tilde{\mathcal{T}}_2, \cdots, \tilde{\mathcal{T}}_N\right\}$ respectively. The Equation (11) can be translated into a $K \times K$ depthwise convolution with the kernels $\left\{\mathcal{B}^{(1)}, \mathcal{B}^{(2)}, \cdots, \mathcal{B}^{(N)}\right\}$.

3.4. The Overall Structure

Based on FDEL and B2Conv, we further develop a lightweight SLVR framework to achieve high-performance image SR. As shown in Fig. 4, SLVR consists of three parts, including shallow feature extraction, deep feature extraction, and image reconstruction. Specifically, given a LR input $I_{\text{LR}} \in \mathbb{R}^{H \times W \times C_{in}}$, we first use a B2Conv $\mathcal{H}_{\text{B2Conv}}$ to extract shallow feature $X_0 \in \mathbb{R}^{H \times W \times C_{mid}}$, as follows

$$X_0 = \mathcal{H}_{\text{B2Conv}} \left(Concat[I_{\text{LR}}]_n \right), \tag{12}$$

where C_{in} and C_{mid} denote the channel number of the input and shallow feature. $Concat[\cdot]_n$ means concatenating n times input. Then we use K efficient controllable distillation block (ECDB) to extract deep features. To maximize the effect of FDEL, feature aggregation is performed by DFAS and GELU [15] activation to enhance feature representation. After that, we use a B2Conv to continue refining the features, the above process can be formulated as

$$X_i = \mathcal{H}_{\text{ECDB}_i} \left(X_{i-1} \right), i = 1, 2, \cdots, K, \tag{13}$$

$$X_{\text{Agg}} = \mathcal{H}_{\text{DFAS}} \left((w_1, w_2, \cdots, w_c), (X_1, X_2, \cdots, X_K) \right), \tag{14}$$

$$X_{\text{Refined}} = \mathcal{H}_{\text{B2Conv}} \left(\text{GELU} \left(X_{\text{Agg}} \right) \right), \tag{15}$$

where $\mathcal{H}_{\text{ECDB}_i}$ represents the i^{th} ECDB, $w_1, w_2, \cdots, w_c$ are the learnable weights, $X_1, X_2, \cdots, X_K$ denote the intermediate features. $\mathcal{H}_{\text{DFAS}}(\cdot)$ denotes the operations of the DFAS, which can be achieved by a convolution. The ECDB is designed based on the ESDB of the BSRN. Specifically, first, we replace the BSRB with the blueprint controllable residual block (BCRB). Second, we replace the

Method	T-Dataset	Scale	#Params	Mult-Adds	Set5		Set14		BSD100		Urban100		Manga109	
					PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
VDSR [22]	291	×2	666K	612.6G	37.53	0.9587	33.03	0.9124	31.90	0.8960	30.76	0.9140	37.22	0.9750
CARN [1]	D2K+291	×2	1,592K	222.8G	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.92	0.9256	38.36	0.9765
IMDN [21]	D2K	×2	694K	158.8G	38.00	0.9605	33.63	0.9177	32.19	0.8996	32.17	0.9283	38.88	0.9774
LAPAR-A [27]	DF2K	×2	548K	171G	38.01	0.9605	33.62	0.9183	32.19	0.8999	32.10	0.9283	38.67	0.9772
LatticeNet [37]	D2K	×2	756K	169.5G	38.06	0.9607	33.70	0.9187	32.20	0.8999	32.25	0.9288	38.94	0.9774
RLFN [24]	D2K	×2	527K	97.9G	38.07	0.9607	33.72	0.9187	32.22	0.9000	32.33	0.9299	-	-
SRPN-Lite [65]	DF2K	×2	609K	139.9G	38.10	0.9608	33.70	0.9189	32.25	0.9005	32.26	0.9294	-	-
FMEN [6]	DF2K	×2	748K	172.0G	38.10	0.9609	33.75	0.9192	32.26	0.9007	32.41	0.9311	38.95	0.9778
MLRN [10]	D2K	×2	488k	90.4G	38.07	0.9607	33.59	0.9180	32.21	0.9000	32.28	0.9297	38.76	0.9773
NGswin [4]	D2K	×2	998K	140.4G	38.05	0.9610	33.79	0.9199	32.27	0.9008	32.53	0.9324	38.97	0.9777
SLVR (Ours)	**D2K**	×2	**464K**	**102G**	**38.14**	**0.9610**	**33.87**	**0.9202**	**32.28**	**0.9009**	**32.57**	**0.9318**	**39.08**	**0.9776**
VDSR [22]	291	×3	666K	612.6G	33.66	0.9213	29.77	0.8314	28.82	0.7976	27.14	0.8279	32.01	0.9340
CARN [1]	D2K+291	×3	1,592K	118.8G	34.29	0.9255	30.29	0.8407	29.06	0.8034	28.06	0.8493	33.50	0.9440
IMDN [21]	D2K	×3	703K	71.5G	34.36	0.9270	30.32	0.8417	29.09	0.8046	28.17	0.8519	33.61	0.9445
LAPAR-A [27]	DF2K	×3	594K	114G	34.36	0.9267	30.34	0.8421	29.11	0.8054	28.15	0.8523	33.51	0.9441
LatticeNet [37]	D2K	×3	765K	76.3G	34.40	0.9272	30.32	0.8416	29.10	0.8049	28.19	0.8513	33.63	0.9442
RFDN [34]	D2K	×3	633K	65.6G	34.47	0.9280	30.35	0.8421	29.11	0.8053	28.32	0.8547	33.78	0.9458
SRPN-Lite [65]	DF2K	×3	615K	62.7G	34.47	0.9276	30.38	0.8425	29.16	0.8061	28.22	0.8534	-	-
FMEN [6]	DF2K	×3	757K	77.2G	34.45	0.9275	30.40	0.8435	29.17	0.8063	28.33	0.8562	33.86	0.9462
MLRN [10]	D2K	×3	496K	40.9G	34.46	0.9267	30.35	0.8426	29.10	0.8054	28.20	0.8533	33.66	0.9450
NGswin [4]	D2K	×3	1,007K	66.6G	34.52	0.9282	30.53	0.8456	29.19	0.8078	28.52	0.8603	33.89	0.9470
SLVR (Ours)	**D2K**	×3	**473K**	**45.3G**	**34.58**	**0.9285**	**30.53**	**0.8460**	**29.21**	**0.8080**	**28.56**	**0.8603**	**34.07**	**0.9475**
VDSR [22]	291	×4	666K	612.6G	31.35	0.8838	28.01	0.7674	27.29	0.7251	25.18	0.7524	28.83	0.8870
CARN [1]	D2K+291	×4	1,592K	90.9G	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837	30.47	0.9084
IMDN [21]	D2K	×4	715K	40.9G	32.21	0.8948	28.58	0.7811	27.56	0.7353	26.04	0.7838	30.45	0.9075
LAPAR-A [27]	DF2K	×4	659K	94G	32.15	0.8944	28.61	0.7818	27.61	0.7366	26.14	0.7871	30.42	0.9074
LatticeNet [37]	D2K	×4	777K	43.6G	32.18	0.8943	28.61	0.7812	27.57	0.7355	26.14	0.7844	30.54	0.9075
RFDN [34]	D2K	×4	550K	23.9G	32.24	0.8952	28.61	0.7819	27.57	0.7360	26.11	0.7858	30.58	0.9089
RLFN [24]	D2K	×4	543K	25.3G	32.24	0.8952	28.62	0.7813	27.60	0.7364	26.17	0.7877	-	-
SRPN-Lite [65]	DF2K	×4	623K	35.8G	32.24	0.8958	28.69	0.7836	27.63	0.7373	26.16	0.7875	-	-
BSRN [30]	DF2K	×4	352K	19.4G	32.35	0.8966	28.73	0.7847	27.65	0.7387	26.27	0.7908	30.84	0.9123
FMEN [6]	DF2K	×4	769K	44.2G	32.24	0.8955	28.70	0.7839	27.63	0.7379	26.28	0.7908	30.70	0.9107
MLRN [10]	D2K	×4	507K	23.5G	32.30	0.8956	28.62	0.7824	27.57	0.7365	26.10	0.7867	30.56	0.9092
HPUN-M [48]	D2K	×4	511K	27.7G	32.24	0.8950	28.66	0.7828	27.60	0.7371	26.12	0.7878	30.55	0.9089
SAFMN [49]	DF2K	×4	240K	14.0G	32.18	0.8948	28.60	0.7813	27.58	0.7359	25.97	0.7809	30.43	0.9063
OSFFNet [53]	DF2K	×4	537K	22.0G	32.39	0.8976	28.75	0.7852	27.66	0.7393	26.36	0.7950	30.84	0.9125
NGswin [4]	D2K	×4	1,019K	36.4G	32.33	0.8963	28.78	0.7859	27.66	0.7396	26.45	0.7963	30.80	0.9128
SLVR-S (Ours)	**DF2K**	×4	**158K**	**8.5G**	**32.20**	**0.8952**	**28.65**	**0.7822**	**27.57**	**0.7361**	**26.08**	**0.7854**	**30.53**	**0.9084**
SLVR (Ours)	**D2K**	×4	**485K**	**26.2G**	**32.40**	**0.8975**	**28.78**	**0.7850**	**27.65**	**0.7388**	**26.41**	**0.7951**	**30.93**	**0.9133**
SLVR* (Ours)	**DF2K**	×4	**485K**	**26.2G**	**32.44**	**0.8980**	**28.85**	**0.7865**	**27.70**	**0.7400**	**26.58**	**0.8000**	**31.10**	**0.9141**

Table 1. Quantitative comparison (PSNR/SSIM) for lightweight SR with state-of-the-art (SOTA) methods on five benchmark datasets. The best and second-best results are marked in red and blue colors respectively. Mult-Adds is evaluated on a 1280×720 HR image. T-Dataset denotes training dataset. D2K stands for the DIV2K dataset. DF2K indicates a merged dataset containing $800 + 2,650$ HR-LR image pairs. 291 images dataset is from [58].

addition mode of the residual structure with the proposed FDEL. Third, all the BSConvs are replaced with the proposed B2Conv. Please see Fig. 4 for detailed changes. After performing the above operations, we obtain the final refined features X_{Refined}. Finally, to take advantage of residual learning, we use long skip connections to reduce the difficulty of learning. SR results can be obtained by

$$I_{\text{SR}} = \mathcal{H}_{\text{Upsample}}\left(X_{\text{Refined}}\right) + \text{Bicubic}^{\uparrow}\left(I_{\text{LR}}\right), \tag{16}$$

where $\mathcal{H}_{\text{Upsample}}(\cdot)$ represents the upsampling operation, which is implemented with the pixelshuffle operation [44]. Bicubic$^{\uparrow}$ denotes bicubic upsampling. L_1 loss function is used to optimize the network, as follows

$$L_1 = \left\| I_{\text{HR}} - I_{\text{SR}} \right\|_1, \tag{17}$$

where I_{HR} is the HR label.

4. Experiments

4.1. Experimental Setup

Datasets and Metrics. We use DIV2K [50] dataset as the training dataset, which contains 800 HR-LR image pairs. To show more fair results, we used DF2K (DIV2K + Flickr2K [50]) to additionally train an $\times 4$ model. Note that the model trained with DF2K is marked with a small *. For testing, we evaluated the performances on the five benchmark datasets, including Set5 [2], Set14 [61], B100 [39], Urban100 [19], Manga109 [40]. The PSNR and SSIM [54] are used as the evaluation metrics, which are evaluated on the Y channel of the YCbCr space.

Implementation Details. ECDB number is set to 8, and the channel number is set to 64. The kernel size of the depth-

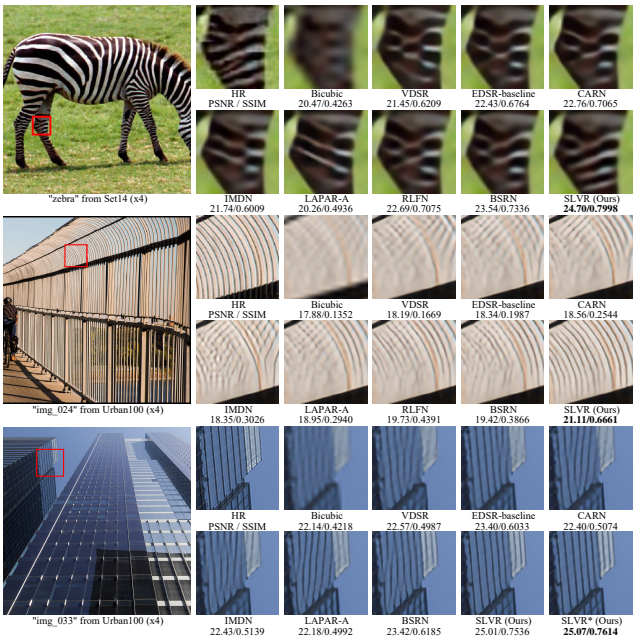


Figure 7. Visual comparison for $\times 4$ SR methods. The patches for comparison are marked with red boxes.

wise convolutions is set to 3. We additionally built a smaller model SLVR-S to compare the latest methods with less than 200K parameters and it trained on DF2K. The ECDB number of the SLVR-S is set to 6 and the channel number is set to 40. During training, we augment the data with random horizontal flips and rotation (90°, 180°, 270°). We randomly crop LR patches of size 64×64 as input. Our models are trained by Adam optimizer [23] for 1000K iterations. The initial learning rate is set to 1×10^{-3} and halved for every 200k iterations. The batch size of SLVR and SLVR-S is set to 64 and 128 respectively. LR image patches are obtained by the bicubic kernel from corresponding HR images matching each SR task. Our networks are trained on an RTX3090 GPU. Note that no other training skills (e.g., pre-training [28]) are employed.

4.2. Comparison with the SOTA SR methods

To evaluate the effectiveness of SLVR, we compare our model with other efficient SR models on scale factors $\times 2, \times 3, \times 4$, including VDSR (CVPR16) [22], CARN (ECCV18) [1], IMDN (ACMMM19) [21], RFDN (ECCV20) [34], LAPAR-A (NeurIPS20) [27], LatticeNet (ECCV20) [37], RLFN (CVPRW22) [24], SRPN-Lite (ICLR22) [65], BSRN (CVPRW22) [30], FMEN (CVPRW22) [6], MLRN (CVPRW23) [10], NGswin (CVPR23) [4], HPUN-M (AAAI23) [48], SAFMN (ICCV23) [49], OSFFNet (AAAI24) [53].

As shown in Table 1, we reported the training dataset, the parameters, Mult-Adds, and PSNR and SSIM on five benchmarks. Compared with models with similar parameters and Mult-Adds, the performance of our SLVR surpasses existing methods with a notable margin on all benchmarks. Compared with the latest transformer-based method NGswin, although the number of parameters of our SLVR is less than 50% of NGswin and the number of Mult-Adds is less than 75% of NGswin, our SLVR still has an advantage in performance. Compared with the latest extremely lightweight method SAFMN trained on the same dataset, our SLVR-S obtains better performance with a smaller model size.

We also perform a visual comparison of different SOTA lightweight SR methods on scale factor $\times 4$. As shown in Fig. 7, we can observe that the SR results generated by SLVR contain more correct and fine-grained textures, while other methods generate many blurred edges and artifacts. For example, in the second row of Fig. 7, our SLVR can restore the texture of the fence, while other methods cannot. We think that the visual results can prove the effectiveness of the proposed B2conv, which can prevent incorrect intra-kernel correlations to reduce useless or disruptive feature extraction and propagation.

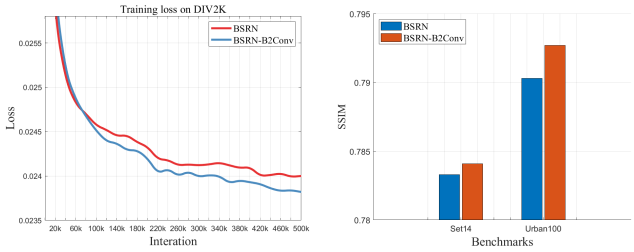


Figure 8. Ablation studies on B2Conv. BSRN-B2Conv denotes the BSConv of BSRN is replaced by B2Conv. All the results are reported on scale factor $\times 4$.

Method	FDEL	B2Conv	Set5	Set14	Urban100
Model-I	✓	×	32.37/0.8969	28.71/0.7839	26.30/0.7917
Model-II	×	✓	32.39/**0.8975**	28.76/**0.7851**	26.40/0.7943
SLVR	✓	✓	**32.40/0.8975**	**28.78**/0.7850	**26.41/0.7951**

Table 2. Ablation studies of FDEL and B2Conv on scale factor $\times 4$. The $\times$ of FDEL denotes it is replaced by addition mode, and the $\times$ of B2Conv denotes it is replaced by BSConv.

4.3. Ablation Studies

The core contributions of our framework are the proposed FDEL and B2Conv. To demonstrate their effectiveness, we first conduct ablation studies on the main framework of our SLVR. Specifically, we remove FDEL and B2Conv and replace them with the original linking mode FAM of RBs and BSConv, respectively. The experimental results are reported in Table 2. We can observe that our FDEL and B2Conv can improve network performance, especially B2Conv, improved by **0.11** dB on the dataset *Urban100*.

Method	Scale	#Params	Mult-Adds	Set5	Set14	Urban100
EDSR-baseline	×4	1,518K	114.0G	32.08/0.8938	28.58/0.7810	25.98/0.7824
EDSR+FDEL		1,518K	114.0G	**32.12/0.8944**	**28.61/0.7819**	**26.05/0.7845**

Table 3. Ablation studies of FDEL. EDSR+FDEL denotes that all addition mode of EDSR-baseline [32] are repalced by FDEL.

RFDN	Scale	#Params	Mult-Adds	Set5	Set14	Urban100
w BSConv	×3	197K	25.8G	34.37/0.9268	30.34/0.8418	28.06/0.8503
w B2Conv		261K	34.5G	34.44/0.9273	30.36/0.8426	28.18/0.8525
w Conv		633K	65.6G	34.47/0.9280	30.35/0.8421	28.21/0.8525
w BSConv	×4	210K	14.9G	32.12/0.8941	28.59/0.7806	26.01/0.7829
w B2Conv		273K	20.1G	32.22/0.8949	28.65/0.7822	26.06/0.7843
w Conv		643K	37.4G	32.28/0.8957	28.61/0.7818	26.11/0.7858

Table 4. Comprehensive comparison between ordinary convolution (Conv), BSConv, and B2Conv. The best and second-best results are marked in red and blue colors.

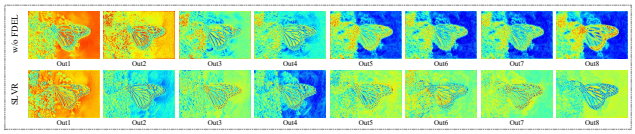


Figure 9. Visual comparison of feature maps between SLVR and SLVR without FDEL. Out$i$ represents the output of the i^{th} ECDB module.

Second, to verify that FDEL can be used as a more efficient linking mode of RBs, we use FDEL to simply replace the traditional linking mode FAM in the classic residual architecture EDSR [32]. From Table 3, we can conclude that our FDEL is more effective than the traditional FAM without additional parameters and Mult-Adds consumed.

Third, to further verify the effectiveness of our proposed B2Conv and confirm that B2Conv can be used as highly efficient building blocks. We denote BSConv of BSRN [30] replaced by B2Conv as BSRN-B2Conv. As shown in Fig. 8, we can observe that our B2Conv can significantly improve model performance without changing model architecture. Besides, we also conducted a comprehensive comparison between ordinary convolution, BSConv, and B2Conv on the network architecture RFDN [34], as shown in Table 4, compared with BSConv, our B2Conv greatly improves the performance (*e.g.*, ***0.1** dB on Set5* ×4) of the RFDN with few additional parameters (***63K***) consumed, which also illustrates the rationality of our improvement.

Finally, we perform a visual comparison of features between SLVR and SLVR without FDEL. From Fig. 9, we find that our FDEL can effectively alleviate network inertia. In addition, we visualize the scale transformation state of 8 blueprints on 64 channels. As shown in Fig. 10, we can observe that our B2Conv does not perform scale transformation on all 64 channel dimensions (only on the channel denoted by black squares). It proves that our B2Conv can adaptively select the appropriate channel to perform blueprint scale transformation, which can effectively prevent the introduction of disruptive or useless features. Furthermore, we perform a visual comparison of features between SLVR and SLVR without B2Conv. From Fig. 11, we can find that the intermediate features generated by the SLVR without B2Conv contain more interference and disruptive features, resulting in poorer feature representation. The intermediate features generated by our SLVR are more accurate and have better feature representation ability. It also indirectly proves that the proposed B2Conv can adaptively select the correct intra-kernel correlation, prevent the introduction of disruptive features, and is more conducive to feature propagation and representation.

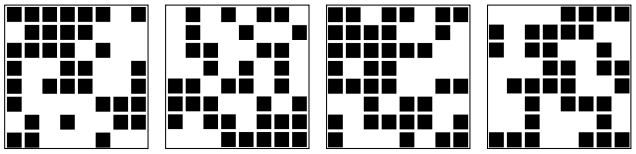


Figure 10. Visualization of scale transformation state of 4 blueprints on 64 channels. Black squares denote strong intra-kernel correlation, and white squares are the opposite.

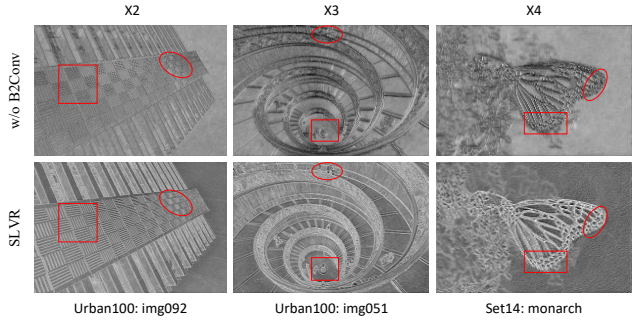


Figure 11. Visual comparison of intermediate feature maps between SLVR and SLVR without B2Conv.

5. Conclusion

In this paper, we propose blueprint controllable convolutions (B2Conv). B2Conv can precisely prevent incorrect intra-kernel correlation, which effectively reduces the extraction of useless or disruptive features and promotes the effective propagation of feature information. Besides, we propose a novel linking mode FDEL for RBs. Our FDEL reduces the retention of the input low-level features and can promote the diversity evolution of high-level features, which can effectively alleviate the occurrence of network inertia. Finally, based on B2Conv and FDEL, we propose an efficient lightweight SR framework SLVR. Furthermore, our B2Conv and FDEL can also serve as highly efficient components to improve the performance of other CNN-based models (such as EDSR, BSRN, and RFDN). Extensive experiments on public benchmark datasets show the prominent SR performance of SLVR and comprehensive analytical studies validate the reasonableness and effectiveness of the proposed framework and components.

References

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV*, pages 252–268, 2018. 1, 3, 6, 7

[2] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *BMVC*, 2012. 6

[3] Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Yan Shuicheng, and Jiashi Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *ICCV*, pages 3434–3443, 2019. 2

[4] Haram Choi, Jeongmin Lee, and Jihoon Yang. N-gram in swin transformers for efficient lightweight image super-resolution. In *CVPR*, pages 2071–2081, 2023. 3, 6, 7

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pages 1800–1807, 2017. 2

[6] Zongcai Du, Ding Liu, Jie Liu, Jie Tang, Gangshan Wu, and Lean Fu. Fast and memory-efficient network towards efficient image super-resolution. In *CVPRW*, pages 852–861, 2022. 3, 6, 7

[7] Jinsheng Fang, Hanjiang Lin, Xinyu Chen, and Kun Zeng. A hybrid network of cnn and transformer for lightweight image super-resolution. In *CVPRW*, pages 1102–1111, 2022. 3

[8] Mengyang Feng, Huchuan Lu, and Yizhou Yu. Residual learning for salient object detection. *IEEE TIP*, 29:4696–4708, 2020. 2

[9] Guangwei Gao, Wenjie Li, Juncheng Li, Fei Wu, Huimin Lu, and Yi Yu. Feature distillation interaction weighting network for lightweight image super-resolution. In *AAAI*, pages 661–669, 2022. 1, 3

[10] Garas Gendy, Nabil Sabor, Jingchao Hou, and Guanghui He. Mixer-based local residual network for lightweight image super-resolution. In *CVPRW*, pages 1593–1602, 2023. 6, 7

[11] Garas Gendy, Nabil Sabor, Jingchao Hou, and Guanghui He. A simple transformer-style network for lightweight image super-resolution. In *CVPRW*, pages 1484–1494, 2023. 3

[12] Daniel Haase and Manuel Amthor. Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved mobilenets. In *CVPR*, pages 14588–14597, 2020. 2, 3, 4

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645. Springer, 2016. 2

[15] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 5

[16] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 2

[18] Gao Huang, Shichen Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Condensenet: An efficient densenet using learned group convolutions. In *CVPR*, pages 2752–2761, 2018. 2

[19] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, pages 5197–5206, 2015. 6

[20] Zheng Hui, Xiumei Wang, and Xinbo Gao. Fast and accurate single image super-resolution via information distillation network. In *CVPR*, pages 723–731, 2018. 3

[21] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *ACM MM*, pages 2024–2032, 2019. 1, 3, 6, 7

[22] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654, 2016. 1, 6, 7

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[24] Fangyuan Kong, Mingxi Li, Songwei Liu, Ding Liu, Jingwen He, Yang Bai, Fangmin Chen, and Lean Fu. Residual local feature network for efficient super-resolution. In *CVPRW*, pages 765–775, 2022. 6, 7

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25, 2012. 2

[26] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, pages 5835–5843, 2017. 3

[27] Wenbo Li, Kun Zhou, Lu Qi, Nianjuan Jiang, Jiangbo Lu, and Jiaya Jia. Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. *NeurIPS*, 33:20343–20355, 2020. 6, 7

[28] Wenbo Li, Xin Lu, Shengju Qian, Jiangbo Lu, Xiangyu Zhang, and Jiaya Jia. On efficient transformer-based image pre-training for low-level vision. *arXiv preprint arXiv:2112.10175*, 2021. 7

[29] Wenjie Li, Juncheng Li, Guangwei Gao, Weihong Deng, Jiantao Zhou, Jian Yang, and Guo-Jun Qi. Cross-receptive focused inference network for lightweight image super-resolution. *IEEE TMM*, pages 1–13, 2023. 3

[30] Zheyuan Li, Yingqi Liu, Xiangyu Chen, Haoming Cai, Jinjin Gu, Yu Qiao, and Chao Dong. Blueprint separable residual network for efficient image super-resolution. In *CVPRW*, pages 832–842, 2022. 1, 3, 6, 7, 8

[31] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV*, pages 1833–1844, 2021. 3

[32] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, pages 1132–1140, 2017. 8

[33] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, pages 85–100, 2018. 2

[34] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. In *ECCV*, pages 41–55. Springer, 2020. 3, 6, 7, 8

[35] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. Transformer for single image super-resolution. In *CVPRW*, pages 456–465, 2022. 3

[36] Zhihe Lu, Zeyu Xiao, Jiawang Bai, Zhiwei Xiong, and Xinchao Wang. Can sam boost video super-resolution? *arXiv preprint arXiv:2305.06524*, 2023. 2

[37] Xiaotong Luo, Yuan Xie, Yulun Zhang, Yanyun Qu, Cuihua Li, and Yun Fu. Latticenet: Towards lightweight image super-resolution with lattice block. In *ECCV*, pages 272–289. Springer, 2020. 6, 7

[38] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, pages 116–131, 2018. 2

[39] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–423 vol.2, 2001. 6

[40] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76:21811–21838, 2017. 6

[41] Marcelo Gennari Do Nascimento, Victor Prisacariu, and Roger Fawcett. Dsconv: Efficient convolution operator. In *ICCV*, pages 5147–5156, 2019. 2

[42] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *CVPR*, pages 3309–3318, 2017. 2

[43] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 2

[44] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, pages 1874–1883, 2016. 3, 6

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[46] Ayan Sinha, Asim Unmesh, Qixing Huang, and Karthik Ramani. Surfnet: Generating 3d shape surfaces using deep residual networks. In *CVPR*, pages 791–800, 2017. 2

[47] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. *NeurIPS*, 28, 2015. 2

[48] Bin Sun, Yulun Zhang, Songyao Jiang, and Yun Fu. Hybrid pixel-unshuffled network for lightweight image super-resolution. In *AAAI*, pages 2375–2383, 2023. 3, 6, 7

[49] Long Sun, Jiangxin Dong, Jinhui Tang, and Jinshan Pan. Spatially-adaptive feature modulation for efficient image super-resolution. In *ICCV*, pages 13190–13199, 2023. 6, 7

[50] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *CVPRW*, pages 1110–1121, 2017. 6

[51] Hang Wang, Xuanhong Chen, Bingbing Ni, Yutian Liu, and Jinfan Liu. Omni aggregation networks for lightweight image super-resolution. In *CVPR*, pages 22378–22387, 2023. 3

[52] Xijun Wang, Meina Kan, Shiguang Shan, and Xilin Chen. Fully learnable group convolution for acceleration of deep neural networks. In *CVPR*, pages 9041–9050, 2019. 2

[53] Yang Wang and Tao Zhang. Osffnet: Omni-stage feature fusion network for lightweight image super-resolution. In *AAAI*, pages 5660–5668, 2024. 6, 7

[54] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 6

[55] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, pages 3–19, 2018. 2

[56] Zeyu Xiao, Jiawang Bai, Zhihe Lu, and Zhiwei Xiong. A dive into sam prior in image restoration. *arXiv preprint arXiv:2305.13620*, 2023. 1

[57] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017. 2

[58] Jianchao Yang, John Wright, Thomas S. Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE TIP*, 19(11):2861–2873, 2010. 6

[59] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas Huang. Free-form image inpainting with gated convolution. In *ICCV*, pages 4470–4479, 2019. 2

[60] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 2

[61] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7*, pages 711–730. Springer, 2012. 6

[62] Ting Zhang, Guo-Jun Qi, Bin Xiao, and Jingdong Wang. Interleaved group convolutions. In *ICCV*, pages 4383–4392, 2017. 2

[63] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018. 2

[64] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, pages 2472–2481, 2018. 2

[65] Yulun Zhang, Huan Wang, Can Qin, and Yun Fu. Learning efficient image super-resolution networks via structure-regularized pruning. In *ICLR*, 2021. 6, 7

[66] Hengyuan Zhao, Xiangtao Kong, Jingwen He, Yu Qiao, and Chao Dong. Efficient image super-resolution using pixel attention. In *ECCV*, pages 56–72. Springer, 2020. 3

[67] Daquan Zhou, Qibin Hou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Rethinking bottleneck structure for efficient mobile network design. In *ECCV*, pages 680–697. Springer, 2020. 2