Large Kernel Distillation Network for Efficient Single Image Super-Resolution

Chengxing Xie[1*] Xiaoming Zhang[1*] Linze Li[1] Haiteng Meng[1] Tianlin Zhang[2] Tianrui Li[1] Xiaole Zhao[1†]
[1]Southwest Jiaotong University, China
[2]National Space Science Center, Chinese Academy of Science, China
zxc0074869@gmail.com, xiaoming.zhang@my.swjtu.edu.cn, zxlation@foxmail.com
https://github.com/stella-von/LKDN

Abstract

Efficient and lightweight single-image super-resolution (SISR) has achieved remarkable performance in recent years. One effective approach is the use of large kernel designs, which have been shown to improve the performance of SISR models while reducing their computational requirements. However, current state-of-the-art (SOTA) models still face problems such as high computational costs. To address these issues, we propose the Large Kernel Distillation Network (LKDN) in this paper. Our approach simplifies the model structure and introduces more efficient attention modules to reduce computational costs while also improving performance. Specifically, we employ the re-parameterization technique to enhance model performance without adding extra cost. We also introduce a new optimizer from other tasks to SISR, which improves training speed and performance. Our experimental results demonstrate that LKDN outperforms existing lightweight SR methods and achieves SOTA performance.

1. Introduction

Single image super-resolution (SISR) is an essential problem in low-level computer vision (CV) that involves reconstructing a high-resolution (HR) image from its low-resolution (LR) counterpart. After the introduction of deep learning to super-resolution by SRCNN [9], there has been a significant surge in the development of deep-learning-based SR models. Due to their impressive ability to reconstruct high-resolution images from low-resolution observations, these algorithms have gained popularity in the CV community. Although deeper and larger models are often considered the optimal approach for designing SR models with strong representation ability [31,57], there is a growing emphasis on developing lightweight models that can approximate the performance of larger models with greatly reduced parameters and less computational complexity.

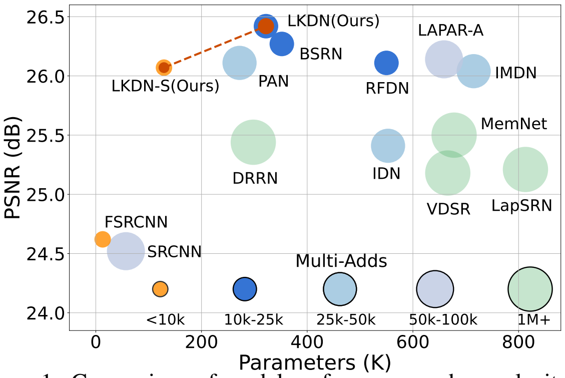

Figure 1. Comparison of model performance and complexity on Urban100 [18] with SR(×4).

Among the numerous design approaches for lightweight super-resolution models, information distillation connections [19] have been identified as a highly effective method. This approach fuses features of varying hierarchies to facilitate the transmission of more distinctive features into the network, resulting in efficient feature reuse and achieving a better balance between reconstruction accuracy and computational efficiency.

RFDN [32] reevaluated the information multi-distillation network [19] and introduced multi-feature distillation connections that are highly adaptable and lightweight, and it won the champion of AIM 2020 Efficient SR Challenge [54]. Meanwhile, BSRN [30] achieved the first place in the model complexity track in NTIRE 2022 Efficient SR Challenge [28] by incorporating residual feature distillation connections with effective attention modules and re-parameterizing the redundant convolution of RFDN by using blueprint separable convolution [14] (BSConv). Besides, RFLN [25] emerged as the champion of the runtime track by improving the efficiency of RFDN through the use of residual local feature blocks that reduce network fragments while maintaining model capacity. The newly proposed VAPSR [60] is designed with a concise structure and fewer parameters while achieving SOTA performance. By enhancing the pixel attention mechanism [59], incorporat-

[*]Equal contributions to this work. †Corresponding author.

ing large kernel convolutions, and implementing efficient depth-wise separable large kernel convolutions. Additionally, VAPSR achieves comparable performance to RFDN while utilizing only 28.18% of its parameters and outperforms BSRN.

Although BSRN has made great progress in terms of model parameter and computation, excessive residual connections and complex attention modules (i.e. contrast-aware attention [19] (CCA) and enhanced spatial attention [32] (ESA)) have led to low model computation efficiency. As analyzed in RLFN, the complex ESA module is redundant and it is difficult to quantitatively analyze which parts are truly useful. While VAPSR achieves better performance than BSRN, the model's running speed is slower due to the presence of a large number of inefficient element-wise multiplications. By removing redundant modules in the network and introducing more efficient ones, we can build a more efficient SR network.

In this paper, we propose a novel lightweight SR network, named large kernel distillation network (LKDN), which builds upon the baseline model of BSRN. Our approach simplifies the model structure and employs a more efficient attention module, called large kernel attention (LKA), to improve model performance and computational costs. We demonstrate the effectiveness of LKA in lightweight SR tasks. Additionally, we leverage the re-parameterization technique to further enhance the performance without adding any additional computational cost. To achieve faster convergence and state-of-the-art (SOTA) performance, we incorporate the recently proposed Adan optimizer [52], which has shown success in various tasks such as high-level CV, natural language processing (NLP), and reinforcement learning. Our proposed LKDN achieves SOTA performance among existing efficiency-oriented SR networks, as shown in Figure 1. The main contributions of this article are:

(**1**) After analyzing the computational efficiency of BSRN [30] and VAPSR [60], We achieved better performance while reducing the number of parameters and computational consumption through simplifying the model structure and introducing a more efficient attention module.

(**2**) We used the technique of re-parameterization to improve the model performance without introducing any additional inference burden.

(**3**) We introduced a new optimizer that can simultaneously boost the training speed and performance of SISR models.

2. Related Work

2.1. Efficient SR Models

The development of lightweight super-resolution networks has received increasing attention in recent years due to their practical applications in resource-constrained scenarios such as mobile devices and embedded systems. Many lightweight SR models have been proposed to reduce the computational cost and memory footprint while maintaining the model capacity and achieving satisfactory performance. The common strategies for lightweight SR networks include network pruning [15, 36, 58], parameter sharing [23, 45], knowledge distillation [12, 16, 56], depth-wise separable convolutions [2, 30, 41, 43], attention mechanisms [13, 19, 32, 59, 60], efficient upsampling methods [26,40,47,59] and re-parameterization technique [3,55]. In addition to the aforementioned techniques, in terms of network structure design, information distillation connections [19, 25, 30, 32] has also been demonstrated as an effective approach to building lightweight SR network architectures. We thus maintain the network topology design of information distillation while enhancing the attention mechanism and implementing re-parameterization in LKDN.

2.2. Large Kernel Design

Since the introduction of VGG [42], small convolution kernels such as 3×3 have been widely used due to their high efficiency and lightweight nature. Transformer [49], as a model that achieves a larger receptive field through global self-attention operation, has achieved excellent performance in NLP. In addition, both global [11] and local [34] vision-Transformers have demonstrated impressive performance in the field of CV. This characteristic has inspired researchers to design better convolutional neural networks (CNNs) by utilizing larger convolution kernels. For example, ConvNeXt [35] uses large convolution kernels to obtain a larger receptive field and achieve comparable performance to Swin-Transformer. RepLKNet [7] scales up kernels to 31×31 using depth-wise convolution and re-parameterization, achieving comparable or superior results to Swin-Transformer on various tasks. VAN [13] explores the effective application of attention mechanisms in CV and proposes a new large kernel attention (LKA) module. SLaK [33] proposes a recipe for applying extremely large kernels from the perspective of sparsity, allowing for the smooth scaling up of kernels to 61×61 with improved performance. Drawing inspiration from such designs, we developed a large kernel distillation block (LKDB) with large kernel attention (LKA) to further improve the representation ability of LKDN.

2.3. Re-parameterization

Re-parameterization is a piratical technique for designing lightweight models that can improve performance without increasing the inference burden. ACNet [5] employs asymmetric convolution to strengthen kernel structures, yielding better results than normal convolution. RepVGG [8] decomposes a standard 3×3 convolution into a multi-branch topology comprising identity mapping,

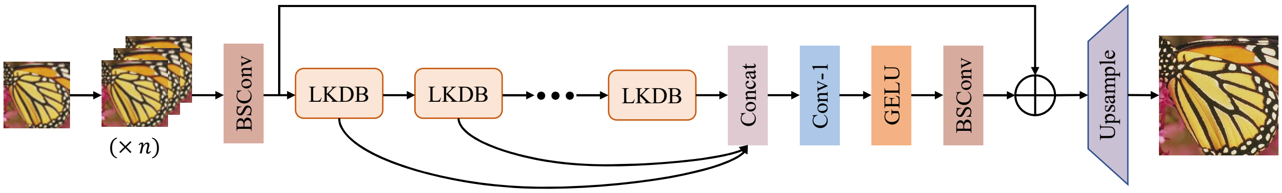

Figure 2. The architecture of large kernel distillation network (LKDN).

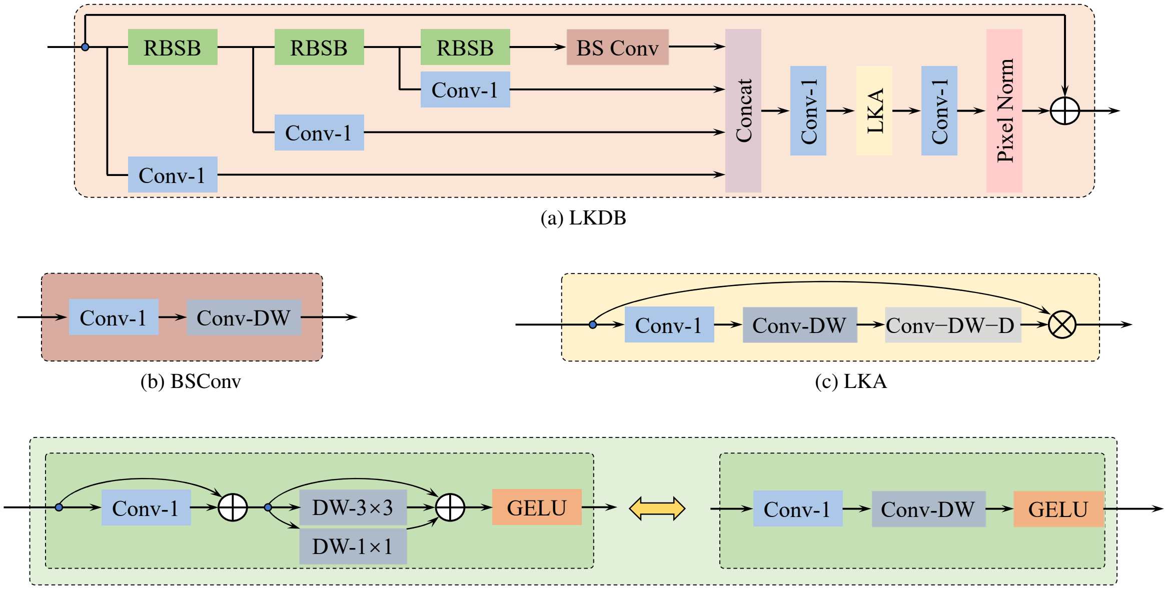

Figure 3. The details of each component. (a) LKDB: Large Kernel Distillation Block; (b) BSConv: Blueprint Separable Convolution; (c) LKA: Large Kernel Attention; (d) RBSB: Re-parameterized Blueprint Shallow Block.

1×1 convolution, and 3×3 convolution, enabling traditional VGG-style CNNs to achieve similar performance and faster inference speed than SOTA on several high-level vision tasks. Diverse Branch Block [6] combines diverse branches of varying scales and complexities to enrich the feature space, constructing a convolutional network unit resembling Inception [44]. MobileOne [48] leverages the re-parameterization technique to enhance the model's accuracy and speed, eventually achieving SOTA performance within effective architectures while being many times faster on mobile devices. As a result. we employed re-parameterization techniques to optimize the feature extraction block in LKDN.

2.4. Adan Optimizer

The Adam optimizer [24] is widely utilized in various deep learning domains. By utilizing first and second order gradient moment estimation, it dynamically adjusts the learning rate of each parameter to achieve faster convergence than stochastic gradient descent (SGD). However, Adam can also suffer from non-convergence [39] and local optima [21, 51]. Recently, Adan optimizer [52] combines modified Nesterov impulse, adaptive optimization, and decoupling weight attenuation. Adan uses extrapolation points to perceive gradient information beforehand, allowing for efficient escape from sharp local minima and increasing model generalization. Based on extensive experiments, it has been shown that the Adan optimizer outperforms existing SOTA optimizers for both CNNs and Transformers. Therefore, we aim to apply the Adan algorithm to lightweight super-resolution tasks.

3. Method

3.1. Network Architecture

Our approach follows the same framework design as BSRN [30], as depicted in Figure 2. It comprises four components: shallow feature extraction, multiple stacked feature distillation blocks, multi-layer feature fusion, and image reconstruction block.

Compared to traditional super-resolution models, our approach involves replicating the input image n times during the pre-processing stage, followed by a concatenation of the replicated images. Given the input I_{LR}, this procedure can be expressed as:

$$I^n_{LR} = \text{Concat}(I^n_{LR}), \tag{1}$$

where $\text{Concat}(\cdot)$ denotes the concatenation operation along the channel dimension, and n is the number of replicated input image I_{LR}. Then the initial feature extraction is implemented by a 3×3 BSConv to generate shallow features from the input LR image as: $F_0 = h_{ext}(I^n_{LR})$, where $h_{ext}(\cdot)$ denotes the module of shallow feature extraction, and F_0 denotes shallow feature. The structure of BSconv is shown in Figure 3(b), which consists of a 1×1 convolution and a depth-wise convolution.

The next part of LKDN is to extract deep features through a stack of LKDBs, which can be formulated as:

$$F_k = H^m_{LKDB}(\ldots H^1_{LKDB}(F_0)\ldots), \quad 1 \leq k \leq m, \tag{2}$$

where $H^k_{LKDB}(\cdot)$ denotes the kth LKDB, m is the number of used LKDBs, and F_k represents the output feature of the kth LKDB.

After gradually refining by the LKDBs, all the intermediate features are fused and activated by a 1×1 convolution layer and a GELU [17] activation. A 3×3 BSConv layer is used to smooth the fused features. The process of multi-layer feature fusion can be formulated as:

$$F_{fusion} = H_{fusion}(\text{Concat}(F_1, \ldots, F_k)), \tag{3}$$

where $H_{fusion}(\cdot)$ denotes the feature fusion module, and F_{fusion} is the fused features.

Finally, a skip connection is employed in the model to enhance the residual learning and the SR images are obtained through image reconstruction as:

$$I_{SR} = H_{rec}(F_{fusion} + F_0), \tag{4}$$

where $H_{rec}(\cdot)$ denotes the image reconstruction module, and I_{SR} is the output of the model. The reconstruction process only includes a 3×3 convolution and pixel-shuffle operation [40].

3.2. Rethinking the BSRN

The performance of BSRN models has been improved by ESA, CCA, and multiple residual connections. However, the complex structure results in lower computational efficiency. RLFN used a pruning sensitivity analysis tool based on a one-shot structured pruning algorithm to analyze the redundancy of the ESA block and discovered a significant amount of redundancy. Therefore, we removed the ESA and CCA modules of BSRN and introduced more efficient attention modules.

Recent studies [13, 60] have shown that the performance of a model can be improved while maintaining acceptable complexity by using large kernel convolution reasonably. VAN [13] utilizes convolution decomposition to split a large kernel convolution into three parts: a spatial local convolution, a spatial long-range convolution, and a channel convolution. Specifically, a $K \times K$ convolution is decomposed into a $(2d-1) \times (2d-1)$ depth-wise convolution, a $\lceil \frac{K}{d} \rceil \times \lceil \frac{K}{d} \rceil$ depth-wise dilation convolution with dilation d, and a 1×1 convolution. By decomposing large kernel convolution, the model can capture long-range relationships with minimal computational cost and parameters. Similar to VAPSR, we perform convolution decomposition in a different order, and as a result, the LKA module is shown in Figure 3(c) and can be expressed as follows:

$$X_{atten} = \text{Conv}_{DW-D}(\text{Conv}_{DW}(\text{Conv}_{1\times1}(F)), \tag{5}$$

$$Output = X_{atten} \otimes F, \tag{6}$$

where $\text{Conv}_{DW-D}(\cdot)$ and $\text{Conv}_{DW}(\cdot)$ denotes dilated depth-wise convolution and depth-wise convolution respectively, X_{atten} denotes attention map, $\otimes$ denotes element-wise product operation, and F denotes the input feature. By decomposing a large 17×17 convolution into a 1×1 point-wise convolution, a depth-wise 5×5 convolution, and a depth-wise dilation convolution with a kernel size of 5 and dilation of 3, our model can reduce complexity and improve performance. Replacing ESA and CCA modules with LKA modules can also further improve inference speed.

3.2.1 Re-parameterization

According to [55], excessive skip connection operations can increase memory access cost and inference time. However, the efficient separable distillation block (ESDB) used in BSRN includes four skip connections. To address this issue, we replace the residual connection of the ESDB with a re-parameterizable skip connection and add the branch of BSConv. To replace the Blueprint Shallow Residual Block (BSRB) in BSRN, we introduce the Re-parameterized Blueprint Shallow Block (RBSB) structure, as shown in Figure 3(d). Specifically, we introduce a re-parameterizable skip connection on the 1×1 point-wise convolution and the 3×3 depth-wise convolution. Additionally, we parallel a 1×1 depth-wise convolution on the 3×3 depth-wise convolution.

The re-parameterization process consists of two steps. First, the input feature F_0 is passed through a re-parameterized 1×1 point-wise convolution:

$$F_1 = \text{Conv}_{1\times1}(F_0) + F_0. \tag{7}$$

Then, a re-parameterized 3×3 depth-wise convolution is applied:

$$F_2 = \text{Conv}_{DW_{3\times3}}(F_1) + \text{Conv}_{D_{1\times1}}(F_1) + F_1. \tag{8}$$

The expressions for F_1 and F_2 during inference using the re-parameterization technique are thus as follows:

$$\begin{aligned} F_1 &= \text{Conv}_{1\times1}(F_0), \\ F_2 &= \text{Conv}_{DW_{3\times3}}(F_0). \end{aligned} \tag{9}$$

3.2.2 Large Kernel Distillation Block

After analyzing the characteristics of BSRN and VAPSR, we developed an even more efficient large kernel distillation block (LKDB). The complete structure of LKDB can be seen in Figure 3(a). It comprises four components: feature distillation, feature fusion, feature enhancement, and feature transformation. In the first stages, given the input F_{in}, the feature distillation operation can be described as:

$$\begin{aligned} F_{d_1}, F_{r_1} &= D_1(F_{in}), R_1(F_{in}), \\ F_{d_2}, F_{r_2} &= D_2(F_{r_1}), R_2(F_{r_1}), \\ F_{d_3}, F_{r_3} &= D_3(F_{r_2}), R_3(F_{r_2}), \\ F_{d_4} &= D_4(F_{r_3}), \end{aligned} \tag{10}$$

where D_i, R_i denote the ith distillation and ith refinement layer, respectively. F_{d_i}, F_{r_i} represents the ith distilled features and ith refined features, respectively. In the feature fusion stage, all the distilled features produced by previous distillation steps are concatenated together and then fused by a 1×1 convolution as:

$$F_{fusion} = H_{fusion}(\text{Concat}(F_{d_1}, F_{d_2}, F_{d_3}, F_{d_4})), \tag{11}$$

where H_{fusion} denotes the 1×1 convolution layer, and F_{fusion} is the fused feature. For the feature enhancement stage, we introduce an efficiency large kernel attention (LKA) block as:

$$F_{enhance} = H_{LKA}(F_{fusion}), \tag{12}$$

where H_{LKA} denotes the LKA module, and $F_{enhance}$ is the enhanced feature. To enhance the performance of the model, we employ a 1×1 convolution in the feature transformation stage, while a pixel normalization [60] module is incorporated to ensure stable model training as:

$$F_{trans} = \text{Norm}_{\text{pixel}}(H_{trans}(F_{enhanced})), \tag{13}$$

where H_{trans} denotes the 1×1 convolution transformation, F_{trans} is the transformed feature, and $\text{Norm}_{\text{pixel}}$ refers to the pixel normalization operation [60]. Finally, a long skip connection is used to strengthen the residual learning ability of the model as:

$$F_{out} = F_{trans} + F_{in}. \tag{14}$$

4. Experiments

4.1. Datasets and Evaluation Metrics

We utilized a training set of 800 images from DIV2K [1] and 2650 images from Flickr2K [31]. Our evaluation of the models is performed on commonly used benchmark datasets, including Set5 [4], Set14 [53], B100 [37], Urban100 [18], and Manga109 [38]. The training data was augmented with random horizontal flips and 90-degree rotations. The evaluation metrics used are the average peak-signal-to-noise ratio (PSNR) and the structural similarity [50] (SSIM) on the luminance (Y) channel.

4.2. Implementation Details

The proposed LKDN model is composed of 8 LKDBs with a distillation structure channel number and attention module channel number set to 56, and is trained with BSB to reduce training time. The mini-batch size and input patch size for each LR input are set to 64 and 48×48, respectively. We train the model using the common L_1 loss function and the Adan optimizer [52], with $\beta_1 = 0.98$, $\beta_2 = 0.92$ and $\beta_3 = 0.99$. We set the exponential moving average (EMA) to stabilize training to 0.999. The learning rate is set to a constant 5×10^{-3} for the entire 1×10^6 training iterations.

We propose a smaller version of LKDN, called LKDN-S, for the NTIRE 2023 Efficient SR Challenge [29]. LKDN-S comprises 5 LKDBs and 42 channels, and is trained with RBSB. We also employ re-parameterization techniques in the up-sample layer of LKDN-S. The training process of LKDN-S involves two stages: an initial training stage and a fine-tuning stage. In the initial training stage, we randomly crop 128 mini-batch HR patches with a size of 256×256. We train LKDN-S using the common L1 loss function with a learning rate of 5×10^{-3} and 9.5×10^5 iterations. In the fine-tuning stage, we set the patch size of HR images and batch size to 480×480 and 64, respectively. LKDN-S is fine-tuned using the L_2 loss function with a learning rate of 2×10^{-5}, and a total of 5×10^4 iterations. The EMA is set to 0.999 and Adan optimizer [52], with $\beta_1 = 0.98$, $\beta_2 = 0.92$ and $\beta_3 = 0.99$ is applied in both stages.

We implement all our models using PyTorch 1.11 and Nvidia GeForce RTX 3080 GPUs.

4.3. Ablation Study

4.3.1 Large Kernel Attention

We conducted ablation studies to verify the efficacy of the LKA module, as presented in Table 1. In this table, C and A denote the input channels of the distillation structure and attention module, respectively. Removing the ESA and CCA modules from BSRN resulted in a significant drop in model performance. However, utilizing the LKA module increased the receptive field of the model, leading to

Table 1. Ablation study on large kernel attention.

Method	Params[K]	Multi-Adds[G]	Set5 [4]	Set14 [53]	B100 [37]	Urban100 [18]	Manga109 [18]
BSRN	352	19.4	32.29 / 0.8950	28.63 / 0.7824	27.59 / 0.7365	26.12 / 0.7870	30.58 / 0.9091
BSRN-woESA&CCA	315	18.2	32.13 / 0.8945	28.56 / 0.7814	27.54 / 0.7355	25.92 / 0.7811	30.31 / 0.9071
LKDN_*C64_A32*	322	18.4	32.28 / **0.8963**	28.69 / 0.7835	27.63 / 0.7381	26.19 / 0.7894	30.72 / 0.9113
LKDN_*C56_A56*	322	18.3	**32.30** / 0.8962	**28.70 / 0.7838**	**27.65 / 0.7385**	**26.22 / 0.7901**	**30.76 / 0.9114**

Table 2. PSNR / SSIM comparison of different basic blocks in the feature distillation connections of LKDN-S.

Method	Set5 [4]	Set14 [53]	B100 [37]	Urban100 [18]	Manga109 [18]
BSRB	32.05 / 0.8932	28.52 / 0.7801	27.53 / 0.7349	25.85 / 0.7787	30.23 / 0.9050
BSB	32.06 / 0.8935	28.54 / **0.7805**	27.54 / 0.7352	25.89 / 0.7797	**30.29** / 0.9057
RBSB	**32.11 / 0.8936**	**28.55 / 0.7805**	**27.55 / 0.7354**	**25.90 / 0.7803**	**30.29 / 0.9060**

Table 3. PSNR / SSIM comparison of applying Adam [24] and Adan [52] optimizers.

Method	Training-time[h]	Set5 [4]	Set14 [53]	B100 [37]	Urban100 [18]	Manga109 [18]
LKDN_Adam	50	**32.41** / 0.8975	28.77 / 0.7854	**27.69** / 0.7399	26.36 / 0.7949	30.93 / 0.9132
LKDN_Adan	45 ↓	32.39 / **0.8979**	**28.79 / 0.7859**	**27.69 / 0.7402**	**26.42 / 0.7965**	**30.97 / 0.9140**

Table 4. Following [28], we compare the computational costs.

Method	DIV2K val [dB]	Params[K]	Multi-Adds[G]	Runtime[ms]
BSRN [30]	29.07	352	19.4	83.75
VAPSR [60]	29.15	342	19.5	105.82
LKDN	29.18	322	18.3	85.65

improved performance while maintaining lower parameter and computation costs than BSRN. Further performance improvements were achieved by adjusting the channel numbers of the distillation structure and attention module. Compared to the original BSRN, LKDN can achieve performance gains of more than **0.1 dB** on the Urban100 [18] and Manga109 [18] while maintaining lower parameter and computation costs.

4.3.2 Re-parameterization

To demonstrate the effectiveness of the proposed RBSB in Figure 3(d), we replaced the basic blocks in the feature distillation structure for comparison. Figure 4(a) displays the BSRB used in BSRN, while Figure 4(b) illustrates the blueprint shallow block (BSB) obtained by directly removing the residual connections. The evaluation results we conducted on LKDN-S are presented in Table 2, demonstrating that eliminating unnecessary residual connections can improve performance while reducing model complexity. Re-parameterization can thus further improve performance while maintaining model complexity.

4.3.3 Optimizer

The previous optimization of SR models primarily relied on the Adam optimizer [24]. However, the Adan optimizer [52], which has recently achieved state-of-the-art results on various vision tasks, has piqued the interest of researchers in the field. Therefore, we investigated the effects of the Adan optimizer on SR tasks. Table 3 shows that using the Adan optimizer yields a training speedup of roughly **10%** compared to using the Adam optimizer, with a significant performance improvement on various benchmark datasets. Moreover, as demonstrated in Figure 5, the Adan optimizer leads to faster convergence.

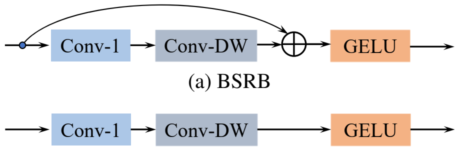

Figure 4. Non-reparameteried components. (a) BSRB: Blueprint Shallow Residual Block; (b) BSB: Blueprint Shallow Block.

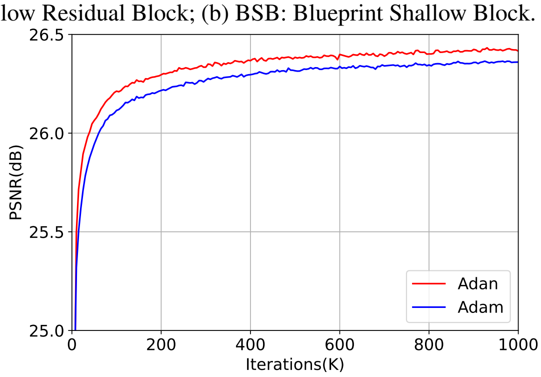

Figure 5. Convergence comparison between Adan [52] and Adam [24] optimizers, using Urban100 [18] SR(×4).

4.4. Comparison with the State-of-the-art Methods

We have evaluated our proposed LKDN with various state-of-the-art lightweight SR methods. These methods have been compared for upscale factors of ×2, ×3, and ×4. Table 5 presents the quantitative comparison results for these methods. With the incorporation of efficient attention modules, LKDN has outperformed other methods in terms of achieving the best performance while maintaining a lightweight model.

Figure 6 presents a qualitative comparison of our proposed method. The results indicate that our model is more capable of reconstructing high-similarity structures than existing methods. As an example, in images ”img_11” and ”img_92”, existing methods typically generate obvious distortion and blurriness, while our approach accurately cap-

Table 5. Quantitative comparison with state-of-the-art methods on benchmark datasets. ’Multi-Adds’ is calculated with a 1280 × 720 GT image. The best and second best are in red and blue respectively.

Method	Scale	Params.[K]	Multi-Adds[G]	Set5 [4]	Set14 [53]	B100 [37]	Urban100 [18]	Manga109 [38]
Bicubic	×2	-	-	33.66 / 0.9299	30.24 / 0.8688	29.56 / 0.8431	26.88 / 0.8403	30.80 / 0.9339
SRCNN [9]	×2	8	52.7	36.66 / 0.9542	32.45 / 0.9067	31.36 / 0.8879	29.50 / 0.8946	35.60 / 0.9663
FSRCNN [10]	×2	13	6.0	37.00 / 0.9558	32.63 / 0.9088	31.53 / 0.8920	29.88 / 0.9020	36.67 / 0.9710
VDSR [22]	×2	666	612.6	37.53 / 0.9587	33.03 / 0.9124	31.90 / 0.8960	30.76 / 0.9140	37.22 / 0.9750
LapSRN [26]	×2	251	29.9	37.52 / 0.9591	32.99 / 0.9124	31.80 / 0.8952	30.41 / 0.9103	37.27 / 0.9740
DRRN [45]	×2	298	6796.9	37.74 / 0.9591	33.23 / 0.9136	32.05 / 0.8973	31.23 / 0.9188	37.88 / 0.9749
MemNet [46]	×2	678	2662.4	37.78 / 0.9597	33.28 / 0.9142	32.08 / 0.8978	31.31 / 0.9195	37.72 / 0.9740
IDN [20]	×2	553	124.6	37.83 / 0.9600	33.30 / 0.9148	32.08 / 0.8985	31.27 / 0.9196	38.01 / 0.9749
CARN [2]	×2	1592	222.8	37.76 / 0.9590	33.52 / 0.9166	32.09 / 0.8978	31.92 / 0.9256	38.36 / 0.9765
IMDN [19]	×2	694	158.8	38.00 / 0.9605	33.63 / 0.9177	32.19 / 0.8996	32.17 / 0.9283	38.88 / 0.9774
PAN [59]	×2	261	70.5	38.00 / 0.9605	33.59 / 0.9181	32.18 / 0.8997	32.01 / 0.9273	38.70 / 0.9773
LAPAR-A [27]	×2	548	171.0	38.01 / 0.9605	33.62 / 0.9183	32.19 / 0.8999	32.10 / 0.9283	38.67 / 0.9772
RFDN [32]	×2	535	95.0	38.05 / 0.9606	33.68 / 0.9184	32.16 / 0.8994	32.12 / 0.9278	38.88 / 0.9773
RFLN [25]	×2	527	115.4	38.07 / 0.9607	33.72 / 0.9187	32.22 / 0.9000	32.33 / 0.9299	-
BSRN [30]	×2	332	73.0	38.10 / 0.9610	33.74 / 0.9193	32.24 / 0.9006	32.34 / 0.9303	39.14 / 0.9782
VAPSR [60]	×2	329	74.0	38.08 / 0.9612	33.77 / 0.9195	32.27 / 0.9011	32.45 / 0.9316	-
LKDN	×2	304	69.1	38.12 / 0.9611	33.90 / 0.9202	32.27 / 0.9010	32.53 / 0.9322	39.19 / 0.9784
Bicubic	×3	-	-	30.39 / 0.8682	27.55 / 0.7742	27.21 / 0.7385	24.46 / 0.7349	26.95 / 0.8556
SRCNN [9]	×3	8	52.7	32.75 / 0.9090	29.30 / 0.8215	28.41 / 0.7863	26.24 / 0.7989	30.48 / 0.9117
FSRCNN [10]	×3	13	5.0	33.18 / 0.9140	29.37 / 0.8240	28.53 / 0.7910	26.43 / 0.8080	31.10 / 0.9210
VDSR [22]	×3	666	612.6	33.66 / 0.9213	29.77 / 0.8314	28.82 / 0.7976	27.14 / 0.8279	32.01 / 0.9340
DRRN [45]	×3	298	6796.9	34.03 / 0.9244	29.96 / 0.8349	28.95 / 0.8004	27.53 / 0.8378	32.71 / 0.9379
MemNet [46]	×3	678	2662.4	34.09 / 0.9248	30.00 / 0.8350	28.96 / 0.8001	27.56 / 0.8376	32.51 / 0.9369
IDN [20]	×3	553	56.3	34.11 / 0.9253	29.99 / 0.8354	28.95 / 0.8013	27.42 / 0.8359	32.71 / 0.9381
CARN [2]	×3	1592	118.8	34.29 / 0.9255	30.29 / 0.8407	29.06 / 0.8034	28.06 / 0.8493	33.50 / 0.9440
IMDN [19]	×3	703	71.5	34.36 / 0.9270	30.32 / 0.8417	29.09 / 0.8046	28.17 / 0.8519	33.61 / 0.9445
PAN [59]	×3	261	39.0	34.40 / 0.9271	30.36 / 0.8423	29.11 / 0.8050	28.11 / 0.8511	33.61 / 0.9448
LAPAR-A [27]	×3	544	114.0	34.36 / 0.9267	30.34 / 0.8421	29.11 / 0.8054	28.15 / 0.8523	33.51 / 0.9441
RFDN [32]	×3	541	42.2	34.41 / 0.9273	30.34 / 0.8420	29.09 / 0.8050	28.21 / 0.8525	33.67 / 0.9449
BSRN [30]	×3	340	33.3	34.46 / 0.9277	30.47 / 0.8449	29.18 / 0.8068	28.39 / 0.8567	34.05 / 0.9471
VAPSR [60]	×3	337	33.6	34.52 / 0.9284	30.53 / 0.8452	29.19 / 0.8077	28.43 / 0.8583	-
LKDN	×3	311	31.4	34.54 / 0.9285	30.52 / 0.8455	29.21 / 0.8078	28.50 / 0.8601	34.08 / 0.9475
Bicubic	×4	-	-	28.42 / 0.8104	26.00 / 0.7027	25.96 / 0.6675	23.14 / 0.6577	24.89 / 0.7866
SRCNN [9]	×4	8	52.7	30.48 / 0.8626	27.50 / 0.7513	26.90 / 0.7101	24.52 / 0.7221	27.58 / 0.8555
FSRCNN [10]	×4	13	4.6	30.72 / 0.8660	27.61 / 0.7550	26.98 / 0.7150	24.62 / 0.7280	27.90 / 0.8610
VDSR [22]	×4	666	612.6	31.35 / 0.8838	28.01 / 0.7674	27.29 / 0.7251	25.18 / 0.7524	28.83 / 0.8870
LapSRN [26]	×4	813	149.4	31.54 / 0.8852	28.09 / 0.7700	27.32 / 0.7275	25.21 / 0.7562	29.09 / 0.8900
DRRN [45]	×4	298	6796.9	31.68 / 0.8888	28.21 / 0.7720	27.38 / 0.7284	25.44 / 0.7638	29.45 / 0.8946
MemNet [46]	×4	678	2662.4	31.74 / 0.8893	28.26 / 0.7723	27.40 / 0.7281	25.50 / 0.7630	29.42 / 0.8942
IDN [20]	×4	553	32.3	31.82 / 0.8903	28.25 / 0.7730	27.41 / 0.7297	25.41 / 0.7632	29.41 / 0.8942
CARN [2]	×4	1592	90.9	32.13 / 0.8937	28.60 / 0.7806	27.58 / 0.7349	26.07 / 0.7837	30.47 / 0.9084
IMDN [19]	×4	715	40.9	32.21 / 0.8948	28.58 / 0.7811	27.56 / 0.7353	26.04 / 0.7838	30.45 / 0.9075
PAN [59]	×4	272	28.2	32.13 / 0.8948	28.61 / 0.7822	27.59 / 0.7363	26.11 / 0.7854	30.51 / 0.9095
LAPAR-A [27]	×4	659	94.0	32.15 / 0.8944	28.61 / 0.7818	27.61 / 0.7366	26.14 / 0.7871	30.42 / 0.9074
RFDN [32]	×4	550	23.9	32.24 / 0.8952	28.61 / 0.7819	27.57 / 0.7360	26.11 / 0.7858	30.58 / 0.9089
RLFN [25]	×4	543	29.8	32.24 / 0.8952	28.62 / 0.7813	27.60 / 0.7364	26.17 / 0.7877	-
BSRN [30]	×4	352	19.4	32.35 / 0.8966	28.73 / 0.7847	27.65 / 0.7387	26.27 / 0.7908	30.84 / 0.9123
VAPSR [60]	×4	342	19.5	32.38 / 0.8978	28.77 / 0.7852	27.68 / 0.7398	26.35 / 0.7941	30.89 / 0.9132
LKDN-S	×4	129	7.3	32.10 / 0.8938	28.62 / 0.7821	27.59 / 0.7371	26.07 / 0.7845	30.50 / 0.9078
LKDN	×4	322	18.3	32.39 / 0.8979	28.79 / 0.7859	27.69 / 0.7402	26.42 / 0.7965	30.97 / 0.9140

tures the lines. Furthermore, in ”img_73”, VAPSR calculates the incorrect number of windows, resulting in a significant decrease in PSNR and SSIM, while our method can accurately restore the number of windows.

Table 4 provides a more in-depth analysis of BSRN, VAPSR, and LKDN. The results show that LKDN outperforms BSRN while maintaining a comparable inference speed. Moreover, LKDN achieves faster inference speeds than VAPSR while maintaining superior performance.

4.5. NTIRE 2023 Efficient SR Challenge

The goal of the NTIRE 2023 Efficient SR Challenge [29] is to develop a SISR model that improves one or more aspects such as runtime, parameters, FLOPs, activations, and

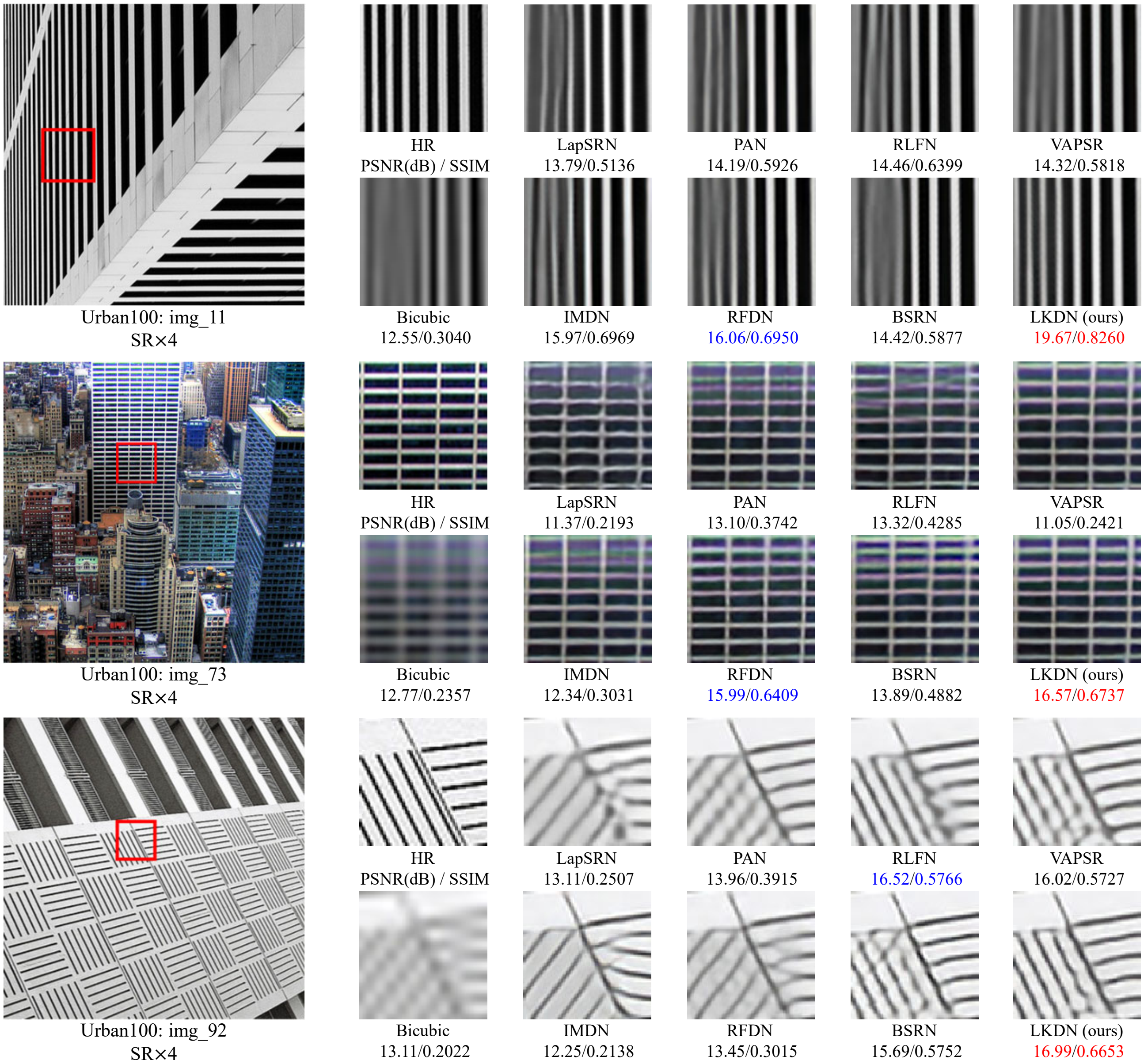

Figure 6. Qualitative and quantitative comparison on SR ($\times 4$), the best and second best are in red and blue respectively.

depth of the RFDN [32], while maintaining a PSNR of at least 29.00dB on validation datasets.

Our solution, LKDN-S, for the NTIRE 2023 Efficient SR Challenge has proven to be both efficient and effective for super-resolution tasks, achieving competitive performance with just 129K parameters and 7.3G Multi-Adds for SR $\times 4$.

5. Conclusion

We propose the Large Kernel Distillation Network (LKDN) as an efficient single-image super-resolution (SISR) solution. Through careful analysis of some state-of-the-art lightweight models, we identified their weaknesses and improved upon them to enhance the performance of LKDN. By incorporating the large kernel design, simplifying the model structure, introducing more efficient attention modules, and employing re-parameterization, we achieve a balance between performance and computational efficiency. A new optimizer is also introduced to accelerate the convergence of LKDN. Extensive experiments demonstrate that LKDN outperforms state-of-the-art efficient SR methods in terms of performance, parameters, and Multi-Adds.

Acknowledgment This work is supported in part by National Natural Science Foundation of China under Grant No. 62102330 and in part by Natural Science Foundation of Sichuan Province under Grant Nos. 2022NSFSC0947 and 2022NSFSC0945.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, pages 126–135, 2017. 5

[2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV*, pages 252–268, 2018. 2, 7

[3] Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *ICML*, pages 244–253, 2018. 2

[4] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *BMVC*, pages 1–10, 2012. 5, 6, 7

[5] Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *ICCV*, pages 1911–1920, 2019. 2

[6] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Diverse branch block: Building a convolution as an inception-like unit. In *CVPR*, pages 10886–10895, 2021. 3

[7] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *CVPR*, pages 11963–11975, 2022. 2

[8] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *CVPR*, pages 13733–13742, 2021. 2

[9] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 38(2):295–307, 2015. 1, 7

[10] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, pages 391–407, 2016. 7

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[12] Qinquan Gao, Yan Zhao, Gen Li, and Tong Tong. Image super-resolution using knowledge distillation. In *ACCV*, pages 527–541, 2019. 2

[13] Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network. *arXiv preprint arXiv:2202.09741*, 2022. 2, 4

[14] Daniel Haase and Manuel Amthor. Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved mobilenets. In *CVPR*, pages 14600–14609, 2020. 1

[15] Tianxing He, Yuchen Fan, Yanmin Qian, Tian Tan, and Kai Yu. Reshaping deep neural network for fast decoding by node-pruning. In *ICASSP*, pages 245–249, 2014. 2

[16] Zibin He, Tao Dai, Jian Lu, Yong Jiang, and Shu-Tao Xia. Fakd: Feature-affinity based knowledge distillation for efficient image super-resolution. In *ICIP*, pages 518–522, 2020. 2

[17] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[18] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, pages 5197–5206, 2015. 1, 5, 6, 7

[19] 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, 2, 7

[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. 7

[21] Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from adam to sgd. *arXiv preprint arXiv:1712.07628*, 2017. 3

[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. 7

[23] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, pages 1637–1645, 2016. 2

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3, 6

[25] 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 766–776, 2022. 1, 2, 7

[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 624–632, 2017. 2, 7

[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. *Advances in Neural Information Processing Systems*, 33:20343–20355, 2020. 7

[28] Yawei Li, Kai Zhang, Radu Timofte, Luc Van Gool, Fangyuan Kong, Mingxi Li, Songwei Liu, Zongcai Du, Ding Liu, Chenhui Zhou, et al. Ntire 2022 challenge on efficient super-resolution: Methods and results. In *CVPRW*, pages 1062–1102, 2022. 1, 6

[29] Yawei Li, Yulun Zhang, Luc Van Gool, Radu Timofte, et al. Ntire 2023 challenge on efficient super-resolution: Methods and results. In *CVPRW*, 2023. 5, 7

[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 833–843, 2022. 1, 2, 3, 6, 7

[31] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, pages 136–144, 2017. 1, 5

[32] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. In *ECCVW*, pages 41–55, 2020. 1, 2, 7, 8

[33] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022. 2

[34] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 2

[35] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pages 11976–11986, 2022. 2

[36] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *ICLR*, 2018. 2

[37] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–423, 2001. 5, 6, 7

[38] 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(20):21811–21838, 2017. 5, 7

[39] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019. 3

[40] 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. 2, 4

[41] Laurent Sifre and Stéphane Mallat. Rigid-motion scattering for texture classification. *arXiv preprint arXiv:1403.1687*, 2014. 2

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

[43] Bin Sun, Yulun Zhang, Songyao Jiang, and Yun Fu. Hybrid pixel-unshuffled network for lightweight image super-resolution. *AAAI*, 2023. 2

[44] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 3

[45] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *CVPR*, pages 3147–3155, 2017. 2, 7

[46] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. MemNet: A persistent memory network for image restoration. In *ICCV*, pages 4549–4557, 2017. 7

[47] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *ICCV*, pages 4799–4807, 2017. 2

[48] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. An improved one millisecond mobile backbone. *arXiv preprint arXiv:2206.04040*, 2022. 3

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[50] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 13(4):600–612, 2004. 5

[51] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. *Advances in neural information processing systems*, 30, 2017. 3

[52] Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models. *arXiv preprint arXiv:2208.06677*, 2022. 2, 3, 5, 6

[53] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730, 2010. 5, 6, 7

[54] Kai Zhang, Martin Danelljan, Yawei Li, Radu Timofte, Jie Liu, Jie Tang, Gangshan Wu, Yu Zhu, Xiangyu He, Wenjie Xu, et al. Aim 2020 challenge on efficient super-resolution: Methods and results. In *ECCVW*, pages 5–40, 2020. 1

[55] Xindong Zhang, Hui Zeng, and Lei Zhang. Edge-oriented convolution block for real-time super resolution on mobile devices. In *ACM MM*, pages 4034–4043, 2021. 2, 4

[56] Yiman Zhang, Hanting Chen, Xinghao Chen, Yiping Deng, Chunjing Xu, and Yunhe Wang. Data-free knowledge distillation for image super-resolution. In *CVPR*, pages 7852–7861, 2021. 2

[57] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, pages 286–301, 2018. 1

[58] Yulun Zhang, Huan Wang, Can Qin, and Yun Fu. Learning efficient image super-resolution networks via structure-regularized pruning. In *ICLR*, 2021. 2

[59] Hengyuan Zhao, Xiangtao Kong, Jingwen He, Yu Qiao, and Chao Dong. Efficient image super-resolution using pixel attention. In *ECCVW*, pages 56–72, 2020. 1, 2, 7

[60] Lin Zhou, Haoming Cai, Jinjin Gu, Zheyuan Li, Yingqi Liu, Xiangyu Chen, Yu Qiao, and Chao Dong. Efficient image super-resolution using vast-receptive-field attention. In *ECCVW*, pages 256–272, 2023. 1, 2, 4, 5, 6, 7