This CVPR workshop 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.

Efficient Image Super-Resolution with Collapsible Linear Blocks

Li Wang, Dong Li, Lu Tian, Yi Shan
Advanced Micro Devices, Inc., Beijing, China.
{li.wang, d.li, lu.tian, yi.shan}@amd.com

Abstract

In this paper, we propose a simple but effective architecture for fast and accurate single image super-resolution. Unlike other compact image super-resolution methods based on hand-crafted designs, we first apply coarse-grained pruning for network acceleration, and then introduce collapsible linear blocks to recover the representative ability of the pruned network. Specifically, each collapsible linear block has a multi-branch topology during training, and can be equivalently replaced with a single convolution in the inference stage. Such decoupling of the training-time and inference-time architecture is implemented via a structural re-parameterization technique, leading to improved representation without introducing extra computation costs. Additionally, we adopt a two-stage training mechanism with progressively larger patch sizes to facilitate the optimization procedure. We evaluate the proposed method on the NTIRE 2022 Efficient Image Super-Resolution Challenge and achieve a good trade-off between latency and accuracy. Particularly, under the condition of limited inference time ($\leq 49.42ms$) and parameter amount ($\leq 0.894M$), our solution obtains the best fidelity results in terms of PSNR, i.e., 29.05dB and 28.75dB on the DIV2K validation and test sets, respectively.

1. Introduction

Single Image Super-Resolution (SISR), which refers to the task of enhancing the resolution of an image from low-resolution (LR) to high (HR), has been studied in computer vision for a long time and has a growing range of applications in fields such as surveillance, the automotive industry, or medical image analysis, etc.

Since the dawn of deep learning, CNN-based methods have made further progress in SISR. [8] innovatively employed a three-layer CNN to directly learn the mapping function and led to significant improvements compared with conventional methods. More follow-up creative ideas are introduced, such as residual learning [29], feature fusion, and attention mechanism [28] [5] [18], advancing the performance of SISR. However, most prior methods use heavy context modeling modules and limit their real-world applications. Thus, research along the line of designing efficient solutions gains increasing attention. Recently, various light-weight CNN-based SISR solutions show great progress with promising results. Many works mainly focus on designing compact networks with efficient techniques. [2] exploits collapsible linear blocks to create an efficient model architecture. [12] adopts channel split to reduce the convolution computation cost. Some others adopt distillation [27] or compression techniques (e.g., pruning [10] and kernel decomposition [20]) for acceleration. Apart from hand-designed architecture, [1] adopts neural architecture search to achieve an excellent balance among the number of parameters and performance.

In this paper, we introduce a simple but effective architecture for fast and accurate single image super-resolution. Firstly, We choose IMDN [12] as our baseline model for its simplicity and effectiveness, and adopt a coarse-grained pruning strategy to get a more shallow network. However, the model performance will often degrade after pruning. To improve the accuracy while maintaining the same inference latency, we introduce collapsible linear blocks to recover the representative ability of the pruned super-resolution network. Specifically, the training-time model has a multi-branch topology, and each branch is a single convolutional layer with different kernel sizes. During inference, these parallel branches can be equivalently converted to a single convolution. Such decoupling of the training-time and inference-time architecture is implemented by a structural re-parameterization technique, and leads to strong representation without introducing extra inference computation cost. Additionally, we apply a two-stage training strategy, which progressively larger the training patch sizes from 64×64 to 160×160, to facilitate the optimization procedure. The proposed method achieves a good trade-off between latency and accuracy. We evaluate the proposed algorithm on the NTIRE 2022 Efficient Image Super-Resolution Challenge [17], and achieve the best fidelity results under the condition of limited inference time ($\leq 49.42ms$) and parameter amount ($\leq 0.894M$). Specifically, our solu-

tion achieves PSNR scores of 29.05dB and 28.75dB on the DIV2K validation and test sets, respectively.

2. Related Work

2.1. Overview of Image Super-Resolution

Driven by the rapid development of deep learning, CNN-based SISR methods have been widely proposed and have achieved state-of-the-art performance on various benchmarks. Usually, the SISR network can be decomposed into feature extraction and upsampling modules. According to the position of the upsampling module in the network, supervised image super-resolution methods can be categorized as Pre-upsampling SR, Post-upsampling SR, Progressive-upsampling SR [13, 14, 24] and Iterative up-and-down SR [9,11,22]. Pre-upsampling SR methods [7,8] often employ an upsampling operation (e.g. bicubic) to first enlarge the size of the low-resolution image, and then use a network to extract the image features. Such framework needs high computation cost since most CNN operations are performed in the high-dimensional space. To achieve a better balance between performance and efficiency, [14,21,23] replace the predefined upsampling operations with end-to-end upsampling layers (e.g., transposed convolution or sub-pixel convolution) which are integrated at the end of the models. Such post-upsampling design can greatly reduce the computational complexity. [13, 14, 24] propose a progressive upsampling mechanism to reduce the learning difficulty by gradually reconstructing high-resolution images, and can cope with the need for multi-scale SISR. In addition, [9, 11, 22] exploit iterative upsampling and downsampling layers to generate intermediate images, and then fuse them to reconstruct final high-resolution images.

Considering high-quality results and low computational cost, post-sampling SR is currently the most widely used pipeline. Based on post-sampling SR, follow-up creative ideas are introduced to improve the fidelity. For example, [15, 29, 29] introduced residual blocks to maximize the power of residual learning. [4, 28] integrated channel or spatial attention mechanism into residual blocks and adopted residual-in-residual structure to form a very deep network. Furthermore, transformer-based solutions, such as [5, 18] have attracted much attention and show impressive performance on SISR.

2.2. Efficient Image Super-Resolution

Most prior methods use heavy context modeling modules, and the huge amount of parameters and the expensive computational cost limit their real-world applications. Designing efficient SISR models attracts much attention from the community and shows great progress with promising results. Many works mainly focus on designing compact networks with efficient techniques. [2] exploits collapsible linear blocks to create an efficient model architecture. [12] adopts channel split to reduce the convolution computation cost. Some others adopt distillation [27] or compression techniques (e.g., pruning [10], quantization and kernel decomposition [20]) for acceleration. Apart from hand-designed architecture, [1, 16, 25] adopt neural architecture search to achieve an excellent balance among the number of parameters and performance. However, their performance is far inferior to the state-of-the-art models. Simple and useful technology remains to be explored.

3. Proposed Method

The overall network architecture is depicted in Figure 1. We introduce more details of our baseline model and optimization strategy in the following sections.

3.1. Baseline Model

We begin by briefly reviewing our baseline architecture, Information Multi-distillation Network (IMDN) [12]. It is a simple but effective method for efficient SISR. As visualized in Figure 1 (a), we first conduct LR feature extraction implemented by one 3×3 convolution with 64 output channels. Then, the key component of the network utilizes multiple stacked information multi-distillation blocks (IMDB) and assembles all intermediate features to fuse by residual connection. The details of IMDB block are illustrated in Figure 1 (b), from which we can see that IMDB contains a distillation branch and a fusion branch. The distillation branch extracts hierarchical features step-by-step, and the fusion branch aggregates them by simply using a 1×1 convolution. Specifically, the distillation branch cascades a series of 3×3 convolution layer and channel split operations. The convolution layers are responsible to extract hierarchical representations. The split operations enable an excellent balance among the number of parameters, inference speed and PSNR performance by reducing the input channels. The final upsampler only consists of one learnable layer and a non-parametric operation (Pixel-shuffle) for saving parameters.

3.2. Optimization Strategy

Network Pruning. Pruning is an effective neural network compression technique, and can be categorized into fine-grained and coarse-grained pruning currently. (1) Fine-grained pruning aims to set individual parameters to zero and make the network sparse. This would lower the number of parameters in the model while keeping the architecture the same. This pruning method requires special hardware optimization (e.g., sparse convolution support) to speed up the inference process. (2) Coarse-grained pruning aims to reduce the model size by directly removing an entire node from the network. It would make the network architecture

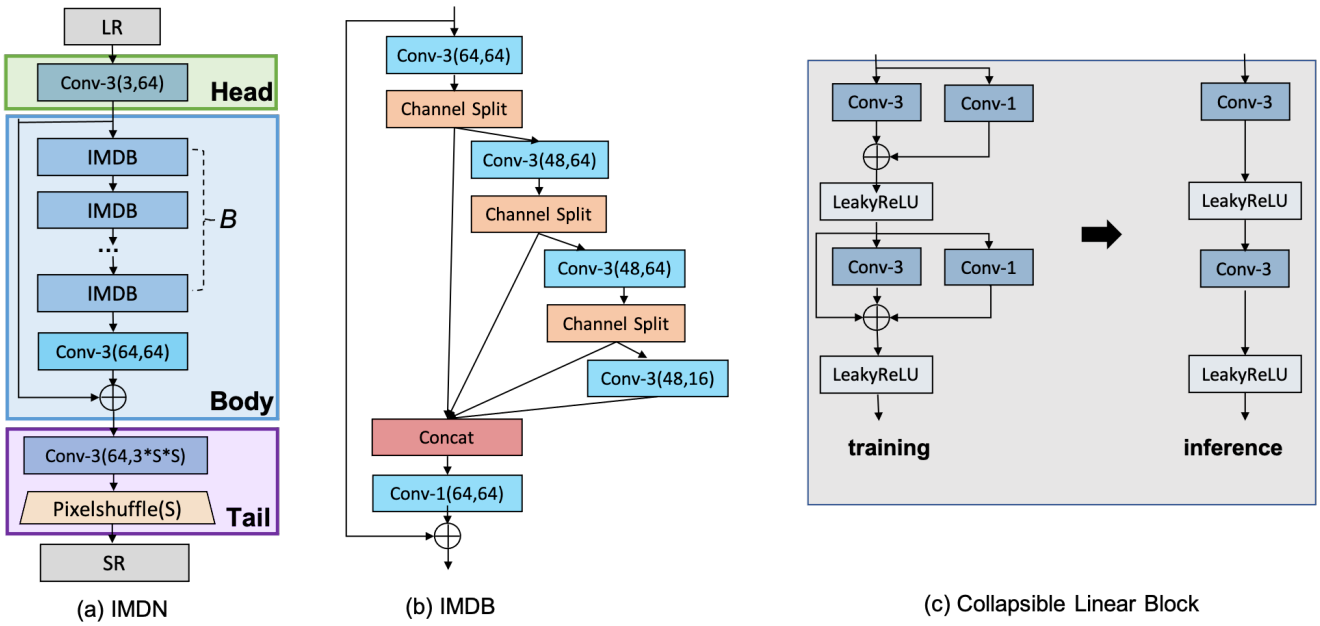

Figure 1. Overview of the proposed architecture. (a) IMDN structure. (b) IMDB block. (c) Collapsible linear block. B denotes the number of IMDB block ($B = 8$ for the baseline model). S denotes the upsampling scale ($S = 4$).

itself smaller. Considering that Titan Xp GPU does not support sparse operations, we apply coarse-grained pruning to compress the network in this work. Specifically, we manually reduce the amount of IMDB modules in IMDN and retrain the pruned network. In our experiments, we finally set $B = 7$ to achieve a good trade-off between model accuracy and efficiency.

Collapsible Linear Block. Network pruning without other optimization may result in degraded accuracy. To improve the representative ability of the pruned network, we are inspired by [2, 6] and introduce a collapsible linear block to replace the general single convolution layer in each IMDB. The structure of the collapsible linear block is shown in Figure 1 (c). Typically, each IMDB module is stacked with several 3×3 convolutions followed by a non-linearity layer. To widen the network, we add a 1×1 branch in parallel with 3×3 convolution before the non-linear layer, so that the training-time information flow of a building block is y = Conv-3×3(x) + Conv-1×1(x), they can be equivalently folded to a single narrow convolution layer at inference time. Such mechanism is hence named Collapsible Linear Block. The final collapsed convolution has 3×3 kernel size. In summary, we train a large and wide network at training time and it gets analytically collapsed into a highly efficient network at inference time. This simple yet powerful re-parameterization method shows significant benefits in image quality without introducing extra computation costs.

Two-Stage Training. We employ a two-stage training strategy that gradually increases the training patch size. Specifically, in the first stage, we take 10 random crops of size 64×64 from each image. In the second stage, we increase the patch size to 160×160 for training. Empirically, we find such two-stage training pipeline can not only speed up the training process, but also improve the accuracy. For both training stages, we use L1 loss which can be formulated as follows:

$$L = \frac{1}{N}\sum_{i=1}^{N} \| f_\theta(I^i_{\text{LR}}) - I^i_{\text{HR}} \| \tag{1}$$

where θ denotes the parameters of the proposed network f, and N is the total number of training samples. I^i_{LR} and I^i_{HR} denote the i-th LR patch and the corresponding HR ground truth.

4. Experiments

4.1. Experimental Settings

Datasets. The NTIRE 2022 Efficient Super-Resolution Challenge proposed to work with the popular DIV2K [22] dataset. It consists from 1000 divers 2K resolution RGB images: 800 are used for training, 100 for validation and 100 for testing purposes. We also use Flick2K [19] as the extra dataset. The total training set contains 3450 pairs of low-resolution and high-resolution RGB images.

Implementation details. We conduct our experiments on PyTorch. For training, we use Adam optimizer with betas = (0.9, 0.999), learning is scheduled via multistep decay

Team	Val PSNR [dB]	Test PSNR [dB]	Avg. PSNR [dB]	Val Time [ms]	Test Time [ms]
Target	≥ 29.00	-	-	≤ 49.42	≤ 52.3
Ours	**29.05**	**28.75**	**28.90**	48.86	47.55
imglhl	29.03	28.75	28.89	57.65	56.11
IMGWLH	29.01	28.72	28.87	56.14	56.53
Dragon	29.01.	28.69	28.85	42.40	41.2
VMCL-Taobao	29.01	28.68	28.85	34.70	33.79
XPixel	29.01	28.69	28.85	142.58	138.37
Alpan Team.	29.01	28.75	28.88	40.17	39.08
NEESR	29.01	28.71	28.86	30.37	29.58
rainbow	29.01	28.74	28.88	34.69	33.52
ByteESR	29.00	28.72	28.86	27.46	26.76
IPCV-IITM[†]	29.10	28.68	28.89	64.00	-
AiriA-CG[†]	29.00	28.70	28.85	37.00	-

Table 1. Ranking results by Val PSNR in the NTIRE 2022 Efficient SISR Challenge. All results are evaluated on the online test server. Our method achieves the best fidelity results under the condition of limited inference time ($\leq 49.42ms$) and parameter amount ($\leq 0.894M$). [†] denotes the results in AIM 2020 [26] challenge.

Model	Val PSNR [dB]	Parameters [M]	Val Time [ms]
Target	≥ 29.00	≤ 0.8939	≤ 49.42
IMDN Baseline	29.13	0.8939	49.42
+ Network Pruning	28.97	0.7905	48.86
++ Collapsible Linear Blocks	29.00	0.7905	48.86
+++ Two-Stage Training	29.05	0.7905	48.86

Table 2. Effect of our each algorithmic component.

Method	B	Parameters [M]	PSNR [dB]
Target		≤ 0.8939	≥ 29.00
IMDN	8	0.8939	29.13
IMDN	7	0.7905	28.97
IMDN	6	0.6871	28.93
IMDN	5	0.5836	28.91
IMDN	4	0.4802	28.85

Table 3. Comparison of different numbers of IMDB in terms of parameters and PSNR scores on the DIV2K validation set.

from base learning rate 2e-4 with decay steps = [200, 400, 450, 475] epochs. The total number of epochs is 500 with a batch size of 32. All LR RGB patches are augmented by random flipping and rotation and sent to the network for training. We follow the NTIRE challenge [17] to measure the SR results.

4.2. Results of Efficient SISR Challenge

As shown in Table 1, our proposed method achieves the best PSNR scores under the condition of limited inference time and parameter amount. Specifically, we achieve 29.05dB and 28.75dB on the DIV2K validation and test sets, respectively. Regarding runtime on the GPU server, our method is still comparable with other teams. For example, compared with the *imglhl* team, we achieve 0.02dB higher Val PSNR with 8.8ms less latency, which demonstrated the superiority of our method.

Table 1 also provides the comparison with AIM 2020 [26] participant teams. Compared with *IPCV-IITM* with a higher Val PSNR, we achieved the best average PSNR (28.90dB vs 28.89dB) which demonstrated the robustness of the proposed method. Besides, our solution also performs better than *AiriA CG*, which shared a similar idea with ours by decoupling training time and inference time with Asymmetric Convolution layers.

4.3. Ablation Study

Effect of algorithmic components To study the effect of each component of the proposed method, we show quantitative results of ablation experiments in Table 3 and Table 2. The PSNR is measured on the validation set. The runtime is measured on the online server by the organizers. Table 3 shows that PSNR scores increase and network parameters also increase with the increasing amount of IMDB modules. Considering the trade-off between accuracy and latency, we choose B=7 IMDB modules with 0.7905M pa-

Method	Val PSNR [dB]	Parameters [M]	FLOPs [G]	#Conv	Val Time [ms]
Target	$\geq$ 29.00	$\leq$ 0.8939	58.53	43	$\leq$ 49.42
OFA-based	28.84	**0.3915**	**33.71**	42	**42.72**
Pruning-based	**28.97**	0.7905	51.76	**37**	48.86

Table 4. Performance comparison between our pruning-based and OFA-based methods.

Model	Patch size	Val PSNR [dB]	Training Time / Epoch [s]	GPU Memory [M]
Stage 1	64×64	29.00	63	3358
Stage 2	160×160	**29.05**	342	1,4747
Stage 2	256×256	29.02	796	2,8991

Table 5. Performance comparisons using different patch sizes for two-stage training.

Figure 2. Examples of the image super-resolution results obtained by our method on the DIV2K validation set.

rameters. To further improve the representative ability of the network, we introduce collapsible linear blocks to keep the high efficiency of the network at inference time. In Table 2, the model IMDN ($B = 7$) with collapsible linear blocks achieves 0.03dB PSNR improvement on DIV2K validation without introducing extra computation. Moreover, the two-stage training strategy further improves PSNR to 29.05dB. These results demonstrate the effectiveness of the proposed method.

Effect of Network Pruning To verify the effectiveness of the pruning method, We also do a comparison with other advanced pruning methods. One-for-all (OFA) [3] is a generalized pruning method that can fit different hardware platforms and constraints like latency, FLOPS and parameters. Table 4 presents the SISR accuracy under constrained latency and parameters between our method and OFA-based method, and we can conclude that although OFA-based pruning method has fewer parameters and FLOPs than our method, PSNR is far inferior to ours (28.97dB vs 28.97dB) with comparable GPU latency (42.72ms vs 48.86ms). The

results highlight that our method can provide a good trade-off between accuracy and latency.

Effect of Two-Stage Training Strategy We also conduct experiments using different patch sizes for two-stage training. From Table 5 shows that increasing the patch size from 160×160 to 256×256 in the second stage drops 0.03dB of PSNR. This is because some training images have a resolution smaller than 256×256 and they need zero padding during the training process, which will affect the accuracy. In addition, training with 256×256 requires more training resources and training time. Considering the training efficiency, we adopt 160×160 as the second stage training patch size.

4.4. Visualization

We visualize the image super-resolution results obtained by the proposed method in Figure 2. The qualitative results show that our method can reconstruct more textures and edges, and obtain high-fidelity super-resolution effect.

5. Conclusion

In this paper, we propose a simple but effective method for single image super-resolution. We first adopt coarse-grained pruning for network slimming. We then introduce collapsible linear blocks to improve the performance of the pruned network while maintaining the same inference speed. We also use a two-stage training strategy to further improve the results. The proposed method achieves a good trade-off between latency and accuracy and obtains PSNR scores of 29.05dB and 28.75dB on the benchmark of NTIRE 2022 Efficient SISR Challenge.

References

[1] Joon Young Ahn and Nam Ik Cho. Neural architecture search for image super-resolution using densely constructed search space: Deconas. In *ICPR*, pages 4829–4836, 2021. 1, 2

[2] Kartikeya Bhardwaj, Milos Milosavljevic, Liam O'Neil, Dibakar Gope, Ramon Matas, Alex Chalfin, Naveen Suda, Lingchuan Meng, and Danny Loh. Collapsible linear blocks for super-efficient super resolution. *arXiv preprint arXiv:2103.09404*, 2021. 1, 2, 3

[3] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019. 5

[4] Haoyu Chen, Jinjin Gu, and Zhi Zhang. Attention in attention network for image super-resolution. *arXiv preprint arXiv:2104.09497*, 2021. 2

[5] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *cvpr*, pages 12299–12310, 2021. 1, 2

[6] 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. 3

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

[8] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, pages 391–407, 2016. 1, 2

[9] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *cvpr*, 2018. 2

[10] Zejiang Hou and Sun-Yuan Kung. Efficient image super resolution via channel discriminative deep neural network pruning. In *ICASSP*, pages 3647–3651, 2020. 1, 2

[11] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *cvpr*, 2017. 2

[12] 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

[13] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *cvpr*, 2017. 2

[14] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. 2018. 2

[15] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *cvpr*, pages 4681–4690, 2017. 2

[16] Yawei Li, Shuhang Gu, Kai Zhang, Luc Van Gool, and Radu Timofte. Dhp: Differentiable meta pruning via hypernetworks. In *ECCV*, pages 608–624, 2020. 2

[17] Yawei Li, Kai Zhang, Luc Van Gool, Radu Timofte, et al. Ntire 2022 challenge on efficient super-resolution: Methods and results. In *CVPR Workshop*, 2022. 1, 4

[18] 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. 1, 2

[19] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *cvpr*. 3

[20] Qian Ning, Weisheng Dong, Guangming Shi, Leida Li, and Xin Li. Accurate and lightweight image super-resolution with model-guided deep unfolding network. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):240–252, 2020. 1, 2

[21] 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*, 2016. 2

[22] Radu Timofte, Shuhang Gu, Jiqing Wu, and Luc Van Gool. Ntire 2018 challenge on single image super-resolution: Methods and results. In *cvpr*. 2, 3

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

[24] Yifan Wang, Federico Perazzi, Brian McWilliams, Alexander Sorkine-Hornung, Olga Sorkine-Hornung, and Christopher Schroers. A fully progressive approach to single-image super-resolution. In *cvpr*. 2

[25] Yan Wu, Zhiwu Huang, Suryansh Kumar, Rhea Sanjay Sukthanker, Radu Timofte, and Luc Van Gool. Trilevel neural architecture search for efficient single image super-resolution. *arXiv preprint arXiv:2101.06658*, 2021. 2

[26] 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 *ECCV*, pages 5–40, 2020. 4

[27] 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. 1, 2

[28] 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, 2

[29] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *cvpr*, pages 2472–2481, 2018. 1, 2