Residual Local Feature Network for Efficient Super-Resolution

Fangyuan Kong* Mingxi Li* Songwei Liu* Ding Liu Jingwen He
Yang Bai Fangmin Chen Lean Fu
ByteDance Inc
{kongfangyuan, limingqi.01, liusongwei.zju, liuding, hejingwen.2021}@bytedance.com
baiyang.87@bytedance.com cfangmin@gmail.com fulean@bytedance.com

Abstract

Deep learning based approaches has achieved great performance in single image super-resolution (SISR). However, recent advances in efficient super-resolution focus on reducing the number of parameters and FLOPs, and they aggregate more powerful features by improving feature utilization through complex layer connection strategies. These structures may not be necessary to achieve higher running speed, which makes them difficult to be deployed to resource-constrained devices. In this work, we propose a novel Residual Local Feature Network (RLFN). The main idea is using three convolutional layers for residual local feature learning to simplify feature aggregation, which achieves a good trade-off between model performance and inference time. Moreover, we revisit the popular contrastive loss and observe that the selection of intermediate features of its feature extractor has great influence on the performance. Besides, we propose a novel multi-stage warm-start training strategy. In each stage, the pre-trained weights from previous stages are utilized to improve the model performance. Combined with the improved contrastive loss and training strategy, the proposed RLFN outperforms all the state-of-the-art efficient image SR models in terms of runtime while maintaining both PSNR and SSIM for SR. In addition, we won the first place in the runtime track of the NTIRE 2022 efficient super-resolution challenge. Code will be available at https://github.com/fyan111/RLFN.

1. Introduction

SISR aims to reconstructed a high-resolution image from a low-resolution image. It is a fundamental low-level vision task and has a wide range of applications [9, 13, 40]. Currently, deep learning based approaches [2, 5, 11, 12, 18, 28, 29, 31, 32, 34, 36, 38, 39, 48, 49] have achieved great success and continuously improved the quality of reconstructed images. However, most of these advanced works require considerable computation costs, which makes them difficult to be deployed on resource-constrained devices for real-world applications. Therefore, it is essential to improve the efficiency of SISR models and design lightweight models that can achieve good trade-offs between image quality and inference time.

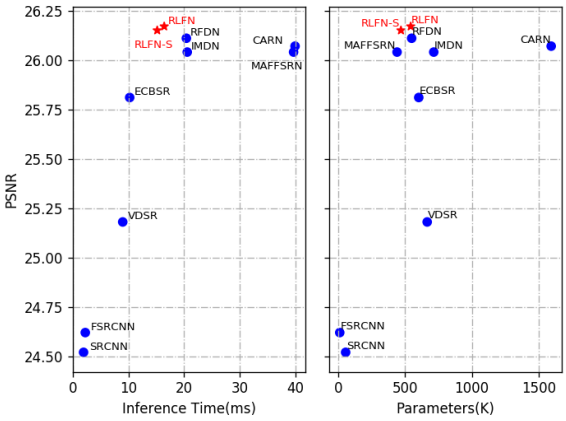


Figure 1. Illustration of PSNR, inference time and parameter numbers of different SISR models on the Urban100 dataset for 4x SR.

Many prior arts [2, 11, 12, 18, 20, 23, 29, 31, 39, 42, 45, 48, 49] have been proposed to develop efficient image super-resolution models. Most of these efficient models try to reduce model parameters or FLOPs. To reduce model parameters, recursive networks with weight sharing strategy are usually adopted [23, 42]. However, such models do not essentially decrease the number of operations and inference time because the complex graph topology is not reduced. To reduce FLOPs of SR models, it is common to employ operations like depth-wise convolutions, feature splitting and shuffling [2, 18, 19, 31], which are without the guarantee of improving computational efficiency. Some recent studies [45,48] have shown that fewer parameters and FLOPs do not always lead to better model efficiency, especially runtime, which is generally the most important factor for prac-

*Equal contribution

tical applications. Parameters and FLOPs are widely used for theoretical analysis but are only proxies for the actual inference time on physical devices. Therefore, there is a high demand to develop efficient SR models that have higher inference speed instead of fewer parameters or FLOPs, in order to better meet practical and commercial needs.

To this end, we revisit current state-of-the-art efficient SR model RFDN [31] and attempt to achieve better trade-offs between reconstructed image quality and inference time. First, we rethink the efficiency of several components of the residual feature distillation block proposed by RFDN. We observe that though feature distillation significantly reduce the number of parameters and contribute to the overall performance, it is not hardware friendly enough and limits the inference speed of RFDN. To improve its efficiency, we propose a novel Residual Local Feature Network (RLFN) that can reduce the network fragments and maintain the model capacity. To further boost its performance, we propose to employ the contrastive loss [44, 46]. We notice that the selection of intermediate features of its feature extractor has great influence on the performance. We conduct comprehensive studies on the properties of intermediate features and draw a conclusion that features from shallow layers preserve more accurate details and textures, which are critical for PSNR-oriented models. Based on this, we build an improved feature extractor that effectively extracts edges and details. To accelerates the model convergence and enhance the final SR restoration accuracy, we propose a novel multi-stage warm-start training strategy. Specifically, in each stage, the SR model can enjoy the benefit of pre-trained weight of models from all previous stages. Combined with the improved contrastive loss and the proposed warm-start training strategy, RLFN achieves state-of-the-art performance and maintain good inference speed. Figure 1 shows that RLFN has a better trade-off between image quality and inference time than other recent competitors of efficient SR models.

Our contributions can be summarized as follows:

1. We rethink the efficiency of RFDN and investigate its speed bottleneck. We propose a novel network termed *Residual Local Feature Network*, which successfully enhances the model compactness and accelerates the inference without sacrificing SR restoration accuracy.
2. We analyze intermediate features extracted by the feature extractor of the contrastive loss. We observe that features from shallow layers are critical for PSNR-oriented models, which inspires us to propose a novel feature extractor to extract more information of edges and textures.
3. We propose a multi-stage warm-start training strategy. It can utilize the trained weights from previous stages to boost the SR performance.

2. Related Work

2.1. Efficient Image Super-Resolution

Achieving real-time SISR on resource-constrained mobile devices has huge business benefits, therefore, we mainly discuss lightweight image SR methods. SCRNN [11] applied the deep learning algorithm to the SISR field for the first time. It has three layers and uses bicubic interpolation to upscale the image before the net, causing unnecessary computational cost. To address this issue, FSRCNN [12] employed the deconvolution layer as the upsampling layer and upscaled the image at the end of net. DRCN [23] introduced a deep recursive convolutional network to reduce the number of parameters. LapSRN [25] proposed the laplacian pyramid super-resolution block to reconstruct the sub-band residuals of HR images. CARN [2] proposed an efficient cascading residual network with group convolution, which obtains comparable results against computationally expensive models. IMDN [18] proposed a lightweight information multi-distillation network by constructing the cascaded information multi-distillation blocks, which extracts hierarchical features step-by-step with the information distillation mechanism (IDM). RFDN [31] refined the architecture of IMDN and proposed the residual feature distillation network, which replaced IDM with feature distillation connections. ECBSR [48] proposed an edge-oriented convolutional block based on the reparameterization technique [10], which can improve the learning ability of the model without increasing the inference time. In order to obtain better results with limited computational effort, the above studies tend to utilize various complex inter-layer connections which affect the inference speed. In this paper we propose a simple network structure with enhanced training strategies to obtain better a trade-off between SR quality and model inference speed.

2.2. Train Strategy for PSNR-oriented SISR

According to machine learning theory, good prediction results come from the combined optimization of architecture, training data, and optimization strategies. Previous works on SISR mainly focused on network architecture optimization [2, 18, 23, 29, 48], while the importance of training strategies that contribute collaboratively to the performances is rarely explored. These SR networks are usually trained by the ADAM optimizer with standard l1 loss for hundreds of epoches. To improve the robustness of training, they usually adopt a smaller learning rate and patch size. Recent works on image recognition [3] and optical flow estimation [41] have demonstrated that advanced training strategies can enable older network architectures to match or surpass the performance of novel architectures. Such evidence motivates us to interrogate the training strategies for SR models and unlocking their potential. RFDN [31]

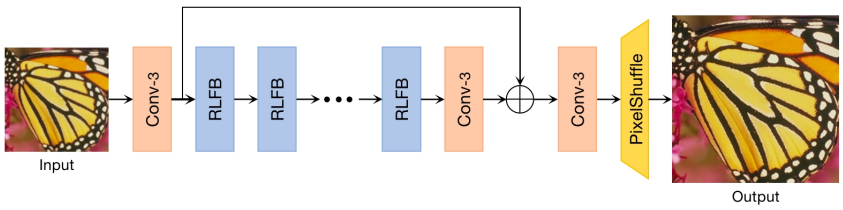
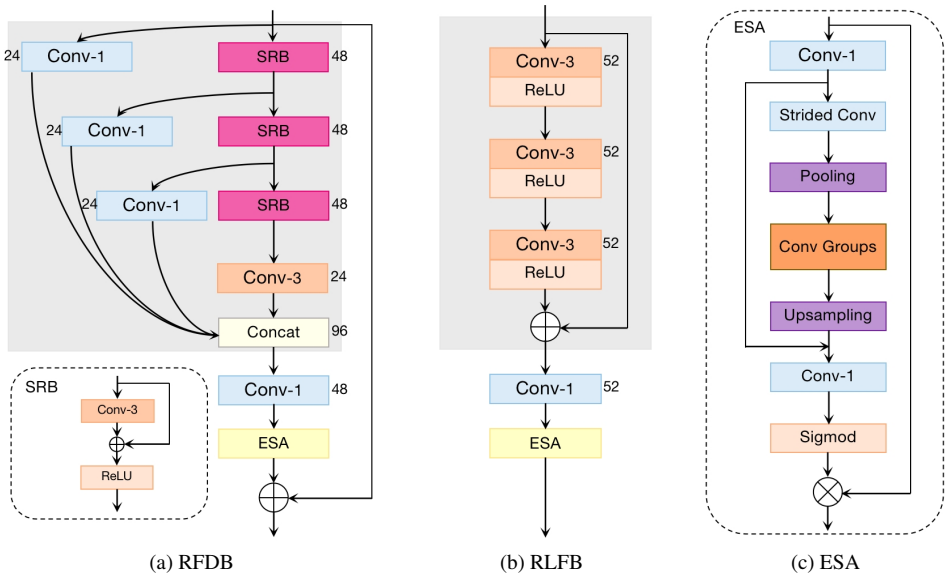


Figure 2. The architecture of residual local feature network.



(a) RFDB (b) RLFB (c) ESA

Figure 3. Conv-1 denotes 1×1 convolution and Conv-3 denotes 3×3 convolution. (a) RFDB: residual feature distillation block. (b) RLFB: residual local feature block. (c) ESA: Enhanced Spatial Attention. In (a) and (b) channel numbers of output feature maps are shown next to each layer.

demonstrated that both fine-tuning the network with l2 loss and initializing a 4x SR model with pretrained 2x model can effectively improve PSNR. RRCAN [30] revisited the popular RCAN model and demonstrated that increasing training iterations clearly improves the model performance.

3. Method

In this section, we first introduce our proposed RLFN in Section 3.1. In Section 3.2, we revisit the contrastive loss and analyze several limitations of its feature extractor. Then we propose an improved feature extractor which can provide more stronger guidance during the training process. In Section 3.3, we describe a novel multi-stage warm-start training strategy, which can effectively improve the performance of lightweight SR models.

3.1. Network Architecture

The overall network architecture of our proposed Residual Local Feature Network (RLFN) is decipted in Figure 2. Our RLFN mainly consists of three parts: the first feature extraction convolution, multiple stacked residual local feature blocks (RLFBs), and the reconstruction module. We denote I_{LR} and I_{SR} as the input and output of RLFN. In the first stage, we use a single 3×3 convolution layer to extract the coarse features:

$$F_0 = h_{ext}(I_{LR}), \tag{1}$$

where $h_{ext}(\cdot)$ denotes the convolution operation for feature extraction and F_0 is the extracted feature maps. Then we use multiple RLFBs in a cascade manner for deep feature extraction. This process can be expressed by

$$F_n = h^n_{RLFB}(h^{n-1}_{RLFB}(\dots h^0_{RLFB}(F_0)\dots)), \tag{2}$$

where $h^n_{RLFB}(\cdot)$ denotes the n-th RLFB function, and F_n is the n-th output feature maps.

In addition, we use one 3×3 convolution layer to smooth the gradually refined deep feature maps. Next, the reconstruction module is applied to generate the final output I_{SR}.

$$I_{SR} = f_{rec}((f_{smooth}(F_n) + F_0)), \tag{3}$$

where $f_{rec}(\cdot)$ represents the reconstruction module which consists of one single 3×3 convolution layer and one non-parametric sub-pixel operation. Besides, f_{smooth} denotes a 3×3 convolution operation.

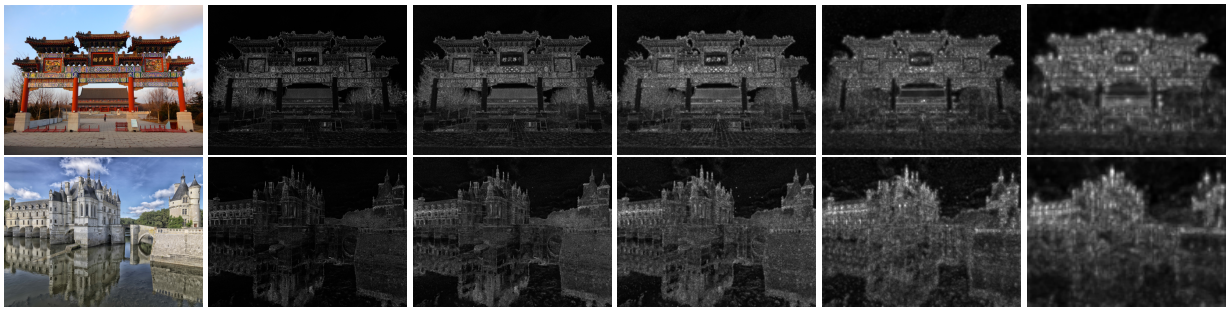


Figure 4. Visualization of extracted features from the 1st, 3rd, 5th, 9th and 13th layer of the pre-trained VGG-19.

Rethinking the RFDB In this subsection, we rethink the efficiency of the residual feature distillation block (RFDB) proposed by RFDN. As shown in Figure 3a, RFDB adopts a progressive feature refinement and distillation strategy in the beginning, and then use a 1×1 convolution for channel reduction. In the end, it applies an enhanced spatial attention (ESA) [32] layer and a residual connection.

Specifically, the feature refinement and distillation pipeline contains several steps. For each stage, RFDB adopts one refinement module that consists of one shallow residual block [32] (SRB) to refine the extracted features, and use a distillation module (a single 1×1 convolution layer) to distill features. Here we denote the refinement and distillation modules as RM and DM, respectively. Given the input features F_{in}, the whole structure can be described by as

$$\begin{aligned} F_{d_1}, F_{r_1} &= DM_1(F_{in}), RM_1(F_{in}) \\ F_{d_2}, F_{r_2} &= DM_2(F_{r_1}), RM_2(F_{r_1}) \\ F_{d_3}, F_{r_3} &= DM_3(F_{r_2}), RM_2(F_{r_2}) \\ F_{d_4} &= DM_4(F_{r_3}), \end{aligned} \tag{4}$$

where DM_j, RM_j denote the j-th distillation and refinement modules, respectively. F_{d_j} represents the j-th distilled features, and F_{r_j} is the j-th refined features that will be further processed by succeeding layers. Lastly, all the distilled features produced by previous distillation steps are concatenated together:

$$F_d = Concat(F_{d_1}, F_{d_2}, F_{d_3}, F_{d_4}), \tag{5}$$

where $Concat$ represents the concatenation operation along the channel dimension.

Overall, RFDB utilizes progressive feature refinement together with multiple feature distillation connections for discriminative feature representations. Practically, the feature distillation connections implemented by several 1×1 convolution operations as well as a concatenation operation could significantly reduce the number of parameters as well as boost the restoration performance. However, this design severely deteriorates the inference speed.

We carefully analyze the efficiency of RFDB in Table 2. In particular, we remove the hierarchical distillation connections and create two variants of RFDB, namely, RFDB_R_48 (**R**efinement with convolution channel **48**) and RFDB_R_52 as shown in Figure 7. From Table 2, it is observed that RFDB_R_48 could reduce 25% inference time compared to the original RFDB. Fortunately, the induced performance drop could be compensated by increasing the channel number of convolution layers. RFDB_R_52 surpasses RFDB_R_48 by a large margin in PSNR with just a slight increase regarding the inference time. More importantly, RFDB_R_52 yields comparable results with RFDB but shows great superiority in terms of inference speed. Therefore, in this work, we directly get rid of the feature distillation branch and make better use of the remaining progressive refinement on local features.

Residual Local Feature Block In this subsection, we introduce the residual local feature block (RLFB) that could significantly reduce the inference time while the model capacity is maintained. As shown in Figure 3b, our proposed RLFB discards the multiple feature distillation connections, and only uses a few stacked CONV+RELU layers for local feature extraction. In particular, each feature refinement module in RLFB contains one 3×3 convolution layer followed by a ReLU activation function layer. Given the input features F_{in}, the whole structure is described by as

$$\begin{aligned} F_{refined_1} &= RM_1(F_{in}), \\ F_{refined_2} &= RM_2(F_{refined_1}) \\ F_{refined_3} &= RM_3(F_{refined_2}), \end{aligned} \tag{6}$$

where RM_j denotes the j-th refinement module, and $F_{refined_j}$ is the j-th refined features. After multiple local feature refinement steps, we add the lastly refined features $F_{refined_3}$ with the skipped features F_{in}. Then we have

$$F_{refined} = F_{in} + F_{refined_3}, \tag{7}$$

where $F_{refined}$ is the final refined output features.

Next, we follow RFDB to feed $F_{refined}$ to a 1×1 convolution layer and a subsequent ESA block to obtain the final output of RLFB. To further reduce the inference time, we developed a pruning sensitivity analysis tool based on one-shot structured pruning algorithm [26] to analyze the

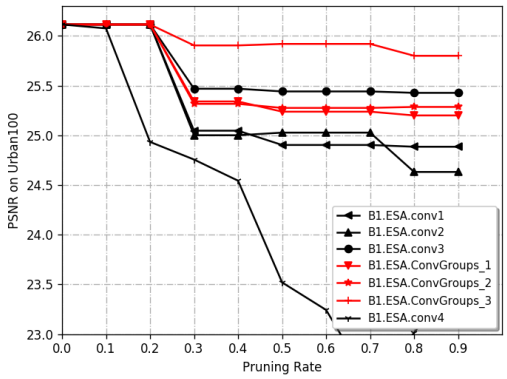


Figure 5. Parameter redundancy analysis of ESA modules on the Urban100 dataset. The red lines represent convolutional layers located in ConvGroups.

redundancy of ESA blocks in RFDB. As shown in Figure 5, the three convolution layers in ConvGroups rank top-1, top-3 and top-4 in redundancy, respectively. Thus, for each ESA block, we reduce the number of convolution layers in ConvGroups to one. From Table 3, we can find that this modification does not incur performance degradation. Instead, it brings a slight improvement regarding the inference time and model parameters.

3.2. Revisiting the Contrastive Loss

Contrastive learning has shown impressive performance in self-supervised learning [6–8, 15, 21]. The basic idea behind is to push positives closer to anchors, and push negatives away from anchors in the latent space. Recent works [44, 46] propose a novel contrastive loss, and demonstrate its effectiveness by improving the quality of reconstructed images. The contrastive loss is defined as:

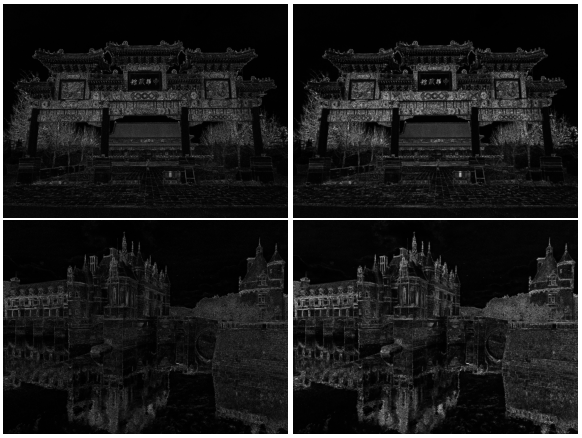
$$CL = \sum_{i=1}^{n} \lambda_i \frac{d(\phi_i(Y_{anchor}), \phi_i(Y_{pos}))}{d(\phi_i(Y_{anchor}), \phi_i(Y_{neg}))}, \quad (8)$$

where ϕ_j denotes the intermediate features from the j-th layer. $d(x, y)$ is the L1-distance between x and y, and λ_j is the balancing weight for each layer. AECR-Net [46] and CSD [44] extract the features from the 1st, 3rd, 5th, 9th and 13th layers of the pre-trained VGG-19. However, we experimentally find that the PSNR is decreased when we employ the contrastive loss.

We next try to investigate its reason to explain the discrepancy. The contrastive loss defined in Eq. (8) mainly depends on the difference of feature maps between two images Y_1 and Y_2. Therefore, we try to visualize the difference map of their feature maps extracted by a pre-trained model ϕ:

$$DMAP_{i,j} = \sqrt{\sum_{k=1}^{K} (\phi(Y_1)_{i,j,k} - \phi(Y_2)_{i,j,k})^2}, \quad (9)$$

where i, j are the spatial coordinates of Y_1 and Y_2, while k is the channel index of Y_1 and Y_2. We use the 100 validation high-resolution images in DIV2K dataset as Y_1, the corresponding images degraded blur kernels as Y_2. Figure 4 presents visualization examples. A surprising observation is that the difference map of features extracted from deeper layers are more semantic, but lacking accurate details. For example, the edges and textures are mostly preserved by the 1st layer, while features from the 13th layer only preserve the overall spatial structure and details are generally missing. In summary, features from deep layers can improve the performance in terms of real perceptual quality because it provides more semantic guidance. Features from shallow layers preserve more accurate details and textures, which are critical for PSNR-oriented models. It suggests that we should utilize features from shallow layers to improve the PSNR of the trained model.



(a) $DMAP_p$ (b) $DMAP_r$

Figure 6. The difference maps of the pre-trained VGG-19 $DMAP_p$ and our proposed feature extractor $DMAP_r$. $DMAP_r$ has stronger response and can capture more details and textures compared with $DMAP_p$.

To further improve the contrastive loss, we revisit the architectures of the feature extractor. The original contrastive loss tries to minimize the distance between two activated features after the ReLU activation function. However, the ReLU function is unbounded above and the activated feature map is sparse, which results in loss of information and provides weaker supervision. Therefore, we replace the ReLU activation function of the feature extractor with the Tanh function.

Moreover, since the VGG-19 is trained with the ReLU activation function, the performance is not guaranteed if the ReLU activation is replaced with the Tanh function without any training. Some recent works [33, 43] shows that a randomly initialized network with good architecture is sufficient to capture perceptual details. Inspired by these works, we build a randomly initialized two-layer feature extractor,

Scale	Model	Params (K)	Runtime (ms)	Set5 PSNR↑ / SSIM↑	Set14 PSNR↑ / SSIM↑	BSD100 PSNR↑ / SSIM↑	Urban100 PSNR↑ / SSIM↑
× 2	SRCNN [11]	24	6.92	36.66 / 0.9542	32.42 / 0.9063	31.36 / 0.8879	29.50 / 0.8946
	FSRCNN [12]	12	9.02	36.98 / 0.9556	32.62 / 0.9087	31.50 / 0.8904	29.85 / 0.9009
	VDSR [22]	666	35.37	37.53 / 0.9587	33.05 / 0.9127	31.90 / 0.8960	30.77 / 0.9141
	DRCN [23]	1774	716.45	37.63 / 0.9588	33.04 / 0.9118	31.85 / 0.8942	30.75 / 0.9133
	LapSRN [25]	251	53.98	37.52 / 0.9591	32.99 / 0.9124	31.80 / 0.8952	30.41 / 0.9103
	CARN [2]	1592	159.10	37.76 / 0.9590	33.52 / 0.9166	32.09 / 0.8978	31.92 / 0.9256
	IMDN [18]	694	77.34	38.00 / 0.9605	33.63 / 0.9177	32.19 / 0.8996	32.17 / 0.9283
	RFDN [31]	534	74.51	38.05 / 0.9606	33.68 / 0.9184	32.16 / 0.8994	32.12 / 0.9278
	MAFFSRN [37]	402	152.91	37.97 / 0.9603	33.49 / 0.9170	32.14 / 0.8994	31.96 / 0.9268
	ECBSR [48]	596	39.96	37.90 / 0.9615	33.34 / 0.9178	32.10 / 0.9018	31.71 / 0.9250
	FDIWN-M [14]	-	-	- / -	- / -	- / -	- / -
	RLFN-S (ours)	454	56.09	38.05 / 0.9607	33.68 / 0.9172	32.19 / 0.8997	32.17 / 0.9286
	RLFN (ours)	527	60.39	38.07 / 0.9607	33.72 / 0.9187	32.22 / 0.9000	32.33 / 0.9299
× 4	SRCNN [11]	57	1.90	30.48 / 0.8628	27.49 / 0.7503	26.90 / 0.7101	24.52 / 0.7221
	FSRCNN [12]	13	2.22	30.72 / 0.8660	27.61 / 0.7550	26.98 / 0.7150	24.62 / 0.7280
	VDSR [22]	666	8.95	31.35 / 0.8838	28.01 / 0.7674	27.29 / 0.7251	25.18 / 0.7524
	DRCN [23]	1774	176.59	31.53 / 0.8854	28.02 / 0.7670	27.23 / 0.7233	25.14 / 0.7510
	LapSRN [25]	502	66.81	31.54 / 0.8852	28.09 / 0.7700	27.32 / 0.7275	25.21 / 0.7562
	CARN [2]	1592	39.96	32.13 / 0.8937	28.60 / 0.7806	27.58 / 0.7349	26.07 / 0.7837
	IMDN [18]	715	20.56	32.21 / 0.8948	28.58 / 0.7811	27.56 / 0.7353	26.04 / 0.7838
	RFDN [31]	550	20.40	32.24 / 0.8952	28.61 / 0.7819	27.57 / 0.7360	26.11 / 0.7858
	MAFFSRN [37]	441	39.69	32.18 / 0.8948	28.58 / 0.7812	27.57 / 0.7361	26.04 / 0.7848
	ECBSR [48]	603	10.21	31.92 / 0.8946	28.34 / 0.7817	27.48 / 0.7393	25.81 /0.7773
	FDIWN-M [14]	454	-	32.17 / 0.8941	28.55 / 0.7806	27.58 / 0.7364	26.02 / 0.7844
	RLFN-S (ours)	470	15.16	32.23 / 0.8961	28.61 / 0.7818	27.58 / 0.7359	26.15 / 0.7866
	RLFN (ours)	543	16.41	32.24 / 0.8952	28.62 / 0.7813	27.60 / 0.7364	26.17 / 0.7877

Table 1. Quantitative results of the state-of-the-art models on four benchmark datasets. The best and second-best results are marked in red and blue colors, respectively.

which has an architecture of Conv_k3s1-Tanh-Conv_k3s1. The difference maps of the pre-trained VGG-19 and our proposed feature extractor are presented in Figure 6. We can observe that the difference map of our proposed feature extractor has stronger response and can capture more details and textures, compared with the difference map of the pre-trained VGG-19. This also provides evidence that a randomly initialized feature extractor can already capture some structural information and pre-training is not necessary.

3.3. Warm-Start Strategy

For large scale factors like 3 or 4 in the SR task, some previous works [31] use the 2x model as a pre-trained network instead of training them from scratch. The 2x model provides good initialized weights which accelerates the convergence and improves the final performance. However, we can only enjoy the benefit once because the scale factors of pre-trained models and target models are different.

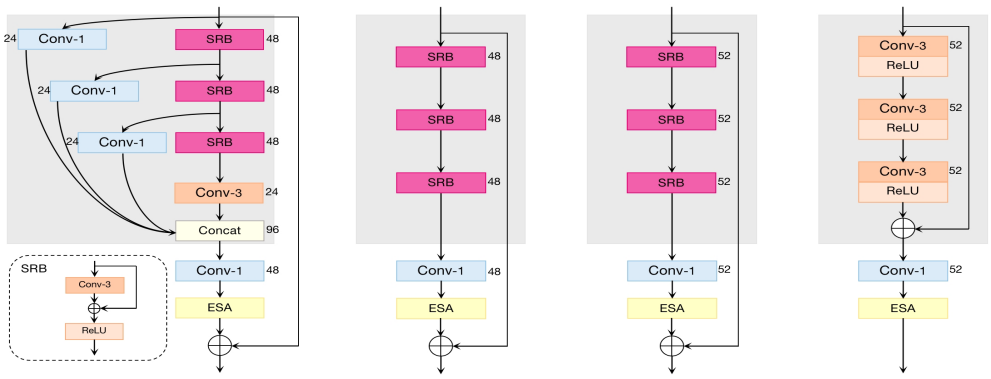
To address this issue, we propose a novel multi-stage warm-start training strategy, which can empirically improve the performance of SISR models. In the first stage, we train RLFN from scratch. Then in the next stage, instead of training from scratch, we load the weights of RLFN of the previous stage, which is referred to as the warm-start policy. The training settings, such as batch size and the learning rate, follow exactly the same training scheme in the first stage. In the following of this paper, we use **RFLN_ws**_i to denote the trained model which employs warm-start i times (after $i+1$ stages). For example, RFLN_ws_1 denotes a two-stage training process. In the first stage, we train RLFN from scratch. Then in the second stage, RLFN loads the pre-trained weights and is trained following the same training scheme as the first stage.

4. Experiments

4.1. Setup

Datasets and Metrics We use the 800 training images in DIV2K dataset [1] for training. We test the performance of our models on four benchmark dataset: Set5 [4], Set14 [47], BSD100 [35] and Urban100 [17]. We evaluate the PSNR and SSIM on the Y channel of YCbCr space.

Training Details Our models are trained on RGB channels and we augment the training data with random flipping and 90 degree rotations. LR images are generated by downsampling HR images with bicubic interpolation in MATLAB. We randomly crop HR patches of size 256×256 from ground truth, and the mini-batch size is set to 64. The training process has three stages. In the first stage, we train the model from scratch. Then we employ the warm-start strat-



(a) RFDB (b) RFDB_R_48 (c) RFDB_R_52 (d) RLFB

Figure 7. The blocks used in ablation study. RFDB_R represent the refinement part of RFDB.

egy twice. In each stage, we adopt Adam optimizer [24] by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ and minimize the L1 loss following the training process of RFDN [31]. The initial learning rate is 5e-4, and is halved every 2×10^5 iterations. Moreover, we additionally employ the widely used contrastive loss [44] in the third stage. This training strategy is adopted in the following of this paper unless otherwise stated. We implement two models, RLFN-S and RLFN. The number of RLFB is set to 6 in both models. We set the number of channels of RLFN to 52. To achieve better runtime, RLFN-S has a smaller channel number of 48.

4.2. Quantitative Results

We compare our models with several advanced efficient super-resolution models [12, 14, 18, 22, 25, 31, 37, 48] with scale factor of 2 and 4. The quantitative performance comparison on several benchmark datasets is shown in Table 1. Compared with other state-of-the-art models, the proposed RLFN-S and RLFN achieve superior performance in terms of both PSNR and SSIM. RLFN-S can achieve comparable or even better performance than RFDN [31], with 80K less parameters. With similar model size, RLFN outperforms other methods by a large margin on all benchmark datasets. We also visualize the trade-off between performance and inference time, and the trade-off between performance and parameters in Figure 1, respectively. The inference time in Figure 1 is the average of 10 runs with CUDA Toolkit 9.0.176 on the NVIDIA 1080Ti GPU. From Figure 1 we can see that our RLFN obtains a better trade-off between quality and inference time than other existing methods.

4.3. Ablation Study

Effectiveness of Architecture Optimization To evaluate the effectiveness of our model architecture optimization, we design two variants of RFDB. As shown in Figure 7, we remove the feature distillation layers in RFDB to get RFDB_R_48, then RFDB_R_52 increases the number of channel to 52 and the middle channel in ESA to 16 for reducing the performance drop, and RLFB removes the intensive addition operations inside SRB based on RFDB_R_52. RFDB, RFDB_R_48, RFDB_R_52, and RLFB are stacked as the body part of the SR network (Figure 2) and trained with the settings of the first stage described in Section 4.1. As shown in Table 2, RLFB maintains the same level of restoration performance with RFDB but has obvious speed advantages.

We also investigate the effect of reducing convolution layers in ESA ConvGroups in the same setting. RLFB_esa_g3 and RLFB maintains the similar restoration performance in Table 3, which means reducing two layers in ESA ConvGroups does not sacrifice performance, but accelerates the network inference.

Model	Params (K)	Runtime (ms)	Set5 PSNR / SSIM	Set14 PSNR / SSIM	BSD100 PSNR / SSIM	Urban100 PSNR / SSIM
RFDB	568	82.3	32.22 / 0.8949	28.64 / 0.7824	27.58 / 0.7361	26.09 / 0.7859
RFDB_R_48	470	61.7	32.12 / 0.8942	28.59 / 0.7806	27.54 / 0.7351	26.00 / 0.7832
RFDB_R_52	572	67.0	32.20 / 0.8948	28.62 / 0.7817	27.57 / 0.7358	26.11 / 0.7861
RLFB(ours)	543	63.2	32.22 / 0.8952	28.61 / 0.7817	27.58 / 0.7359	26.11 / 0.7865

Table 2. Comparison of RFDB, its two variants: RFDB_R_48, RFDB_R_52, and our RLFN for 4x SR. Runtime is the average of 10 runs on DIV2K validation set.

Model	Params (K)	Runtime (ms)	Set5 PSNR / SSIM	Set14 PSNR / SSIM	BSD100 PSNR / SSIM	Urban100 PSNR / SSIM
RLFB_esa_g3	572	63.6	32.20 / 0.8952	28.61 / 0.7818	27.57 / 0.7363	26.12 / 0.7869
RLFB (ours)	543	63.2	32.22 / 0.8952	28.61 / 0.7817	27.58 / 0.7359	26.11 / 0.7865

Table 3. Comparison of RLFB_esa_g3 and our RLFB for 4× SR. RLFB_esa_g3 uses three convolution layers in ConvGroups, while our RLFB uses just one. Runtime is the average of 10 runs on DIV2K validation set for 4x SR.

Effectiveness of Contrastive Loss To investigate the effectiveness of contrastive loss, we remove the contrastive loss in the second warm-start stage and only employs L1 loss. As shown in Table 4, the contrastive loss consistently improves the performance in terms of both PSNR and SSIM on four benchmark datasets.

Effectiveness of Warm-Start Strategy To demonstrate the effectiveness of our proposed warm-start strategy, we com-

Model	Set5 PSNR / SSIM	Set14 PSNR / SSIM	BSD100 PSNR / SSIM	Urban100 PSNR / SSIM
RLFN-S_ws_2	32.22 / 0.8960	28.60 / 0.7818	27.57 / 0.7359	26.13 / 0.7865
RLFN-S_ws_2 + CL	32.23 / 0.8961	28.61 / 0.7818	27.58 / 0.7359	26.15 / 0.7866

Table 4. Effect of contrastive loss for 4x SR. RLFN-S_ws_2 applies warm-start twice and RLFN-S_ws_2 + CL employs contrastive loss in the second warm-start stage. Contrastive loss improves the performance in terms of both PSNR and SSIM.

pare RLFN-S_ws_1 as the baseline and two variants of different learning rate strategies, RLFN-S_e2000 and RLFN-S_clr. Contrastive loss is not used in this comparison while other training settings remain the same. They set the total epochs to 2000 to be compared with RLFN-S_ws_1. RLFN-S_e2000 halves the learning rate every 4×10^5 iterations. RLFN-S_clr applies a cyclical learning rate policy, which is the same as RLFN-S_ws_1. However, it loads the state of the optimizer while RLFN-S_ws_1 applies the default initialization. As shown in Table 5, RLFN-S_e2000 and RLFN-S_clr decrease PSNR and SSIM compared with our proposed warm-start strategy. It indicates that the warm-start strategy helps to jump out of the local minimum during the optimization process and improve the overall performance.

Model	Set5 PSNR / SSIM	Set14 PSNR / SSIM	BSD100 PSNR / SSIM	Urban100 PSNR / SSIM
RLFN-S_e2000	32.17 / 0.8953	28.58 / 0.7815	27.57 / 0.7354	26.08 / 0.7849
RFLN-S_clr	32.20 / 0.8959	28.59 / 0.7818	27.56 / 0.7359	26.12 / 0.7865
RLFN-S_ws_1	32.21 / 0.8959	28.60 / 0.7818	27.57 / 0.7360	26.12 / 0.7864

Table 5. Effect of learning rate strategy for 4x SR. RLFN-S_e2000 and RLFN-S_clr set the total epochs to 2000 to be compared with our proposed strategy RLFN-S_ws_1. RLFN-S_e2000 halves the learning rate every 4×10^5 iterations. RLFN-S_clr applies a cyclical learning rate policy. The best and second-best results are marked in red and blue colors, respectively.

Generalization We also investigate the generalization of our proposed contrastive loss and warm-start strategy. We apply contrastive loss and warm-start strategy individually to EDSR [29]. The quantitative comparison is shown in Table 6, which demonstrates that our proposed methods are generic and can be applied to other existing SISR models.

Model	Set5 PSNR / SSIM	Set14 PSNR / SSIM	BSD100 PSNR / SSIM	Urban100 PSNR / SSIM
EDSR	32.06 / 0.8945	28.57 / 0.7816	27.56 / 0.7359	26.08 / 0.7859
EDSR + CL	32.07 / 0.8946	28.58 / 0.7818	27.57 / 0.7360	26.08 / 0.7861
EDSR_ws_1	32.07 / 0.8947	28.59 / 0.7821	27.59 / 0.7363	26.09 / 0.7865

Table 6. Generalization of our proposed contrastive loss and warm-start strategy. We compare EDSR [29] and its two variants which employ the contrastive loss and warm-start strategy, respectively. It can be seen that our proposed methods are generic to existing SISR models.

Team name	PSNR [val]	PSNR [test]	Ave Time [ms]	Parameters [M]	FLOPs [G]	Activations [M]	Memery [M]	Conv
ByteESR(ours)	29.00	28.72	27.11	0.317	19.7	80.05	377.91	39
NJU_Jet	29.00	28.69	28.07	0.341	22.28	72.09	204.6	34
NEESR	29.01	28.71	29.97	0.272	16.86	79.59	575.99	59
Super	29.00	28.71	32.09	0.326	20.06	93.82	663.07	59
MegSR	29.00	28.68	32.59	0.29	17.7	91.72	640.63	64
RFDN(Winner AIM20)	29.04	28.75	41.97	0.433	27.1	112.03	788.13	64
IMDN(Baseline)	29.13	28.78	50.86	0.894	58.53	154.14	471.76	43

Table 7. Runtime track results of NTIRE 2022 efficient SR challenge. Only the top five methods are included.

4.4. RLFN for NTIRE 2022 challenge

Our team won the 1st place in the main track (Runtime Track) and the 2nd place in the sub-track2 (Overall Performance Track) of NTIRE 2022 efficient super-resolution challenge [27]. The model structure and training strategy are slightly different from the above. The proposed RLFN-cut has 4 RLFBs, in which the number of feature channels is set to 48 while the channel number of ESA is set to 16. During training, DIV2K and Flickr2K datasets are used for the whole process. First, the model is trained from scratch. HR patches of size 256×256 are randomly cropped from HR images, and the mini-batch size is set to 64. The model is trained by minimizing L1 loss function with Adam optimizer. The initial learning rate is set to 5e-4 and halved at every 200 epochs.The total number of epochs is 1000. Then we employ warm-start policy and train the model with the same settings twice. After that, we change the loss to L1 loss + 255×Contrastive loss, and train with warm-start policy again. At last, we reduce the channels of conv-1 and its dependent conv-3 layers from 48 to 46 using Soft Filter Pruning [16]. Training settings remain the same except that the size of HR patches changes to 512×512. After pruning stage, L2 loss is used for fine-tuning with 640×640 HR patches and a learning rate of 1e-5 for 200 epochs. We include the top five methods in Table 7, Compared to baseline IMDN and the first place method RFDN in AIM 2020 Efficient Super-Resolution Challenge, our method achieves significant improvements in all metrics, and we achieve the shortest running time.

5. Conclusion

In this paper, we propose a Residual Local Feature Network for efficient SISR. By reducing the number of layers and simplifying the connections between layers, our network is much lighter and faster. Then we revisit the use of contrastive loss, change the structure of the feature extractor and re-select the intermediate features used by contrastive loss. We also propose a warm-start strategy, which is beneficial on the training of lightweight SR models. Extensive experiments have shown that our overall scheme, including the model structure and training method, achieves a commendable balance of quality and inference speed.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 6

[2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, volume 11214 of *Lecture Notes in Computer Science*, pages 256–272. Springer, 2018. 1, 2, 6

[3] Irwan Bello, William Fedus, Xianzhi Du, Ekin Dogus Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. Revisiting resnets: Improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 34, 2021. 2

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

[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 *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 12299–12310. Computer Vision Foundation / IEEE, 2021. 1

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 5

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 5

[8] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020. 5

[9] Yu Chen, Ying Tai, Xiaoming Liu, Chunhua Shen, and Jian Yang. Fsrnet: End-to-end learning face super-resolution with facial priors. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2492–2501. Computer Vision Foundation / IEEE Computer Society, 2018. 1

[10] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13733–13742, 2021. 2

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV*, volume 8692 of *Lecture Notes in Computer Science*, pages 184–199. Springer, 2014. 1, 2, 6

[12] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, volume 9906 of *Lecture Notes in Computer Science*, pages 391–407. Springer, 2016. 1, 2, 6, 7

[13] Ying Fu, Tao Zhang, Yinqiang Zheng, Debing Zhang, and Hua Huang. Hyperspectral image super-resolution with optimized RGB guidance. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 11661–11670. Computer Vision Foundation / IEEE, 2019. 1

[14] Guangwei Gao, Wenjie Li, Juncheng Li, Fei Wu, Huimin Lu, and Yi Yu. Feature distillation interaction weighting network for lightweight image super-resolution. *CoRR*, abs/2112.08655, 2021. 6, 7

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 5

[16] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft filter pruning for accelerating deep convolutional neural networks. *arXiv preprint arXiv:1808.06866*, 2018. 8

[17] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015. 6

[18] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In Laurent Amsaleg, Benoit Huet, Martha A. Larson, Guillaume Gravier, Hayley Hung, Chong-Wah Ngo, and Wei Tsang Ooi, editors, *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*, pages 2024–2032. ACM, 2019. 1, 2, 6, 7

[19] Zheng Hui, Xiumei Wang, and Xinbo Gao. Fast and accurate single image super-resolution via information distillation network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 723–731, 2018. 1

[20] Andrey Ignatov, Radu Timofte, Maurizio Denna, and Abdel Younes. Real-time quantized image super-resolution on mobile npus, mobile AI 2021 challenge: Report. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*, pages 2525–2534. Computer Vision Foundation / IEEE, 2021. 1

[21] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2021. 5

[22] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1646–1654. IEEE Computer Society, 2016. 6, 7

[23] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1637–1645. IEEE Computer Society, 2016. 1, 2, 6

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 7

[25] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5835–5843. IEEE Computer Society, 2017. 2, 6, 7

[26] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016. 4

[27] Yawei Li, Kai Zhang, Luc Van Gool, Radu Timofte, et al. Ntire 2022 challenge on efficient super-resolution: Methods and results. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2022. 8

[28] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *IEEE/CVF International Conference on Computer Vision Workshops, ICCVW 2021, Montreal, BC, Canada, October 11-17, 2021*, pages 1833–1844. IEEE, 2021. 1

[29] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1132–1140. IEEE Computer Society, 2017. 1, 2, 8

[30] Zudi Lin, Prateek Garg, Atmadeep Banerjee, Salma Abdel Magid, Deqing Sun, Yulun Zhang, Luc Van Gool, Donglai Wei, and Hanspeter Pfister. Revisiting rcan: Improved training for image super-resolution. *arXiv preprint arXiv:2201.11279*, 2022. 3

[31] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part III*, volume 12537 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2020. 1, 2, 6, 7

[32] Jie Liu, Wenjie Zhang, Yuting Tang, Jie Tang, and Gangshan Wu. Residual feature aggregation network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2359–2368, 2020. 1, 4

[33] Yifan Liu, Hao Chen, Yu Chen, Wei Yin, and Chunhua Shen. Generic perceptual loss for modeling structured output dependencies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5424–5432, June 2021. 5

[34] Salma Abdel Magid, Yulun Zhang, Donglai Wei, Won-Dong Jang, Zudi Lin, Yun Fu, and Hanspeter Pfister. Dynamic high-pass filtering and multi-spectral attention for image super-resolution. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 4268–4277. IEEE, 2021. 1

[35] 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 *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001. 6

[36] Yiqun Mei, Yuchen Fan, and Yuqian Zhou. Image super-resolution with non-local sparse attention. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 3517–3526. Computer Vision Foundation / IEEE, 2021. 1

[37] Abdul Muqeet, Jiwon Hwang, Subin Yang, Jung Heum Kang, Yongwoo Kim, and Sung-Ho Bae. Multi-attention based ultra lightweight image super-resolution. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part III*, volume 12537 of *Lecture Notes in Computer Science*, pages 103–118. Springer, 2020. 6, 7

[38] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XII*, volume 12357 of *Lecture Notes in Computer Science*, pages 191–207. Springer, 2020. 1

[39] Wenzhe Shi, Jose Caballero, Ferenc Huszar, 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 *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1874–1883. IEEE Computer Society, 2016. 1

[40] Xibin Song, Yuchao Dai, Dingfu Zhou, Liu Liu, Wei Li, Hongdong Li, and Ruigang Yang. Channel attention based iterative residual learning for depth map super-resolution. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 5630–5639. Computer Vision Foundation / IEEE, 2020. 1

[41] Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T Freeman, and Ce Liu. Autoflow: Learning a better

training set for optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10093–10102, 2021. 2

[42] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2790–2798. IEEE Computer Society, 2017. 1

[43] Pei Wang, Yijun Li, and Nuno Vasconcelos. Rethinking and improving the robustness of image style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 124–133, June 2021. 5

[44] Yanbo Wang, Shaohui Lin, Yanyun Qu, Haiyan Wu, Zhizhong Zhang, Yuan Xie, and Angela Yao. Towards compact single image super-resolution via contrastive self-distillation. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 1122–1128. ijcai.org, 2021. 2, 5, 7

[45] Pengxu Wei, Hannan Lu, Radu Timofte, Liang Lin, Wangmeng Zuo, Zhihong Pan, Baopu Li, Teng Xi, Yanwen Fan, Gang Zhang, Jingtuo Liu, Junyu Han, Errui Ding, Tangxin Xie, Liang Cao, Yan Zou, Yi Shen, Jialiang Zhang, Yu Jia, Kaihua Cheng, Chenhuan Wu, Yue Lin, Cen Liu, Yunbo Peng, Xueyi Zou, Zhipeng Luo, Yuehan Yao, Zhenyu Xu, Syed Waqas Zamir, Aditya Arora, Salman H. Khan, Munawar Hayat, Fahad Shahbaz Khan, Keon-Hee Ahn, Jun-Hyuk Kim, Jun-Ho Choi, Jong-Seok Lee, Tongtong Zhao, Shanshan Zhao, Yoseob Han, Byung-Hoon Kim, JaeHyun Baek, Haoning Wu, Dejia Xu, Bo Zhou, Wei Guan, Xiaobo Li, Chen Ye, Hao Li, Haoyu Zhong, Yukai Shi, Zhijing Yang, Xiaojun Yang, Xin Li, Xin Jin, Yaojun Wu, Yingxue Pang, Sen Liu, Zhi-Song Liu, Li-Wen Wang, Chu-Tak Li, Marie-Paule Cani, Wan-Chi Siu, Yuanbo Zhou, Rao Muhammad Umer, Christian Micheloni, Xiaofeng Cong, Rajat Gupta, Feras Almasri, Thomas Vandamme, and Olivier Debeir. AIM 2020 challenge on real image super-resolution: Methods and results. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part III*, volume 12537 of *Lecture Notes in Computer Science*, pages 392–422. Springer, 2020. 1

[46] Haiyan Wu, Yanyun Qu, Shaohui Lin, Jian Zhou, Ruizhi Qiao, Zhizhong Zhang, Yuan Xie, and Lizhuang Ma. Contrastive learning for compact single image dehazing. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 10551–10560. Computer Vision Foundation / IEEE, 2021. 2, 5

[47] 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. Springer, 2010. 6

[48] Xindong Zhang, Hui Zeng, and Lei Zhang. Edge-oriented convolution block for real-time super resolution on mobile devices. In Heng Tao Shen, Yueting Zhuang, John R. Smith, Yang Yang, Pablo Cesar, Florian Metze, and Balakrishnan Prabhakaran, editors, *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*, pages 4034–4043. ACM, 2021. 1, 2, 6, 7

[49] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, volume 11211 of *Lecture Notes in Computer Science*, pages 294–310. Springer, 2018. 1