Residual Feature Distillation Network for Lightweight Image Super-Resolution

Jie Liu, Jie Tang*, and Gangshan Wu

State Key Laboratory for Novel Software Technology, Nanjing University, China
jieliu@smail.nju.edu.cn, {tangjie,gswu}@nju.edu.cn

Abstract. Recent advances in single image super-resolution (SISR) explored the power of convolutional neural network (CNN) to achieve a better performance. Despite the great success of CNN-based methods, it is not easy to apply these methods to edge devices due to the requirement of heavy computation. To solve this problem, various fast and lightweight CNN models have been proposed. The information distillation network is one of the state-of-the-art methods, which adopts the channel splitting operation to extract distilled features. However, it is not clear enough how this operation helps in the design of efficient SISR models. In this paper, we propose the feature distillation connection (FDC) that is functionally equivalent to the channel splitting operation while being more lightweight and flexible. Thanks to FDC, we can rethink the information multi-distillation network (IMDN) and propose a lightweight and accurate SISR model called residual feature distillation network (RFDN). RFDN uses multiple feature distillation connections to learn more discriminative feature representations. We also propose a shallow residual block (SRB) as the main building block of RFDN so that the network can benefit most from residual learning while still being lightweight enough. Extensive experimental results show that the proposed RFDN achieves a better trade-off against the state-of-the-art methods in terms of performance and model complexity. Moreover, we propose an enhanced RFDN (E-RFDN) and won the first place in the AIM 2020 efficient super-resolution challenge. Code will be available at https://github.com/njulj/RFDN.

Keywords: image super-resolution, computational photography, image processing

1 Introduction

Image super-resolution (SR) is a classic computer vision task to reconstruct a high-resolution (HR) image from its low-resolution (LR) counterpart. It is an ill-posed procedure since many HR images can be degraded to the same LR image. Image SR is a very active research area where many approaches have been proposed to generate the upscaled images. In this paper, we focus on the...
problem of lightweight image SR which is needed in time-sensitive applications such as video streaming.

Recently, various convolutional neural network (CNN) based methods have been proposed and achieved prominent performance in image SR. As a pioneering work, Dong et al. [4] proposed the super-resolution convolutional neural network (SRCNN), which is a three-layer network to directly model the mapping from LR to HR. Then, Kim et al. [12] pushed the depth of SR network to 20 and achieved much better performance than SRCNN, which indicates that the quality of upscaled images can be improved with deeper networks. The EDSR [17] network further proved this by using more than 160 layers. Although deeper networks increase the quality of SR images, they are not suitable for real-world scenarios. It is important to design fast and lightweight CNN models that have a better trade-off between SR quality and model complexity.

To reduce the number of parameters, DRCN [13] and DRRN [22] adopted a recursive network that decreases the number of parameters effectively by parameter sharing. However, it has to increase the depth or the width of the network to compensate for the loss caused by the recursive module. These models reduce the model size at the expense of increased number of operations and inference time. In real-world applications, the number of operations is also an important factor to consider so that the SR model can be performed in real-time. So, it is better to design dedicated networks that are lightweight and efficient enough for real-world scenarios.

To this end, Ahn et al. [1] proposed the CARN-M for mobile devices by using a cascading network architecture, but it is at the cost of a large PSNR drop. Hui et al. [11] proposed an information distillation network (IDN) that explicitly split the intermediate features into two parts along the channel dimension, one was retained and the other was further processed by succeeding convolution layers. By using this channel splitting strategy, IDN can aggregate current information with partially retained local short-path information and achieve good performance at a modest size. Later, IMDN [10] further improved IDN by designing an information multi-distillation block (IMDB) that extracted features at a granular level. Specifically, the channel splitting strategy was applied multiple times within a IMDB. Each time, one part of the features was retained and another was sent to the next step. IMDN has a good performance in terms of both PSNR and inference time and won the first place in the AIM 2019 constrained image super-resolution challenge [30]. However, the number of parameters of IMDN is more than most of the lightweight SR models (e.g.
VDSR [12], IDN [11], MemNet [23]). There is still room for improvement to be more lightweight.

The key component of both IDN and IMDN is the information distillation mechanism (IDM) that explicitly divides the preceding extracted features into two parts, one is retained and the other is further refined. We argue that the IDM is not efficient enough and it brings some inflexibility in the network design. It is hard to incorporate identity connections with the IDM. In this paper, we will give a more comprehensive analysis of the information distillation mechanism and propose the feature distillation connection (FDC) that is more lightweight and flexible than the IDM. We use IMDN as the baseline model since it makes a good trade-off between the reconstruction quality and the inference speed, which is very suitable for mobile devices. But the IMDN is not lightweight enough and the SR performance can still be further improved. To build a more powerful fast and lightweight SR model, we rethink the architecture of IMDN and propose the residual feature distillation network (RFDN). In comparison with IMDN, our RFDN is much more lightweight by using the feature distillation connections (FDCs). Further more, we propose a shallow residual block (SRB) that uses as the building blocks of RFDN to further improve the SR performance. The SRB consists of one convolutional layer, an identical connection and an activation unit at the end. It can benefit from the residual learning [8] without introducing extra parameters compared with plain convolutions. It is very easy to incorporate SRB with the feature distillation connection to build a more powerful SR network.

The main contributions of this paper can be summarized as follows:

1. We propose a lightweight residual feature distillation network (RFDN) for fast and accurate image super-resolution, which achieves state-of-the-art SR performance while using much fewer parameters than the competitors.
2. We give a more comprehensive analysis of the information distillation mechanism (IDM) and rethink the IMDN network. Based on these new understandings, we propose the feature distillation connections (FDC) that are more lightweight and flexible than the IDM.
3. We propose the shallow residual block (SRB) that incorporates the identity connection with one convolutional block to further improve the SR performance without introducing any extra parameters.

2 RELATED WORK

Recently, deep learning based models have achieved dramatic improvements in image SR. The pioneering work was done by Dong et al. [4], they first exploited a three-layer convolutional neural network SRCNN to jointly optimize the feature extraction, non-linear mapping and image reconstruction in an end-to-end manner. Then Kim et al. [12] proposed the very deep super-resolution (VDSR) network, which stacked 20 convolutional layers to improve the SR performance. To reduce the model complexity, Kim et al. [13] introduced DRCN that recursively applied the feature extraction layer for 16 times. DRRN [22] improved DRCN by combining the recursive and residual network schemes to
achieve better performance with fewer parameters. Lai et al. [14] proposed the laplacian pyramid super-resolution network (LapSRN) to address the speed and accuracy problem by taking the original LR images as input and progressively reconstructing the sub-band residuals of HR images. Tai et al. [23] presented the persistent memory network (MemNet) for image restoration task, which tackled the long-term dependency problem in the previous CNN architectures. To reduce the computational cost and increase the testing speed, Shi et al. [21] designed an efficient sub-pixel convolution to upscale the resolutions of feature maps at the end of SR models so that most of computation was performed in the low-dimensional feature space. For the same purpose, Dong et al. [5] proposed fast SRCNN (FSRCNN), which employed transposed convolution as upsampling layers to accomplish post-upsampling SR. Then Lim et al. [17] proposed EDSR and MDSR, which achieved significant improvements by removing unnecessary modules in conventional residual networks. Based on EDSR, Zhang et al. proposed the residual dense network (RDN) [34] by introducing dense connections into the residual block. They also proposed the very deep residual attention network (RCAN) [32] and the residual non-local attention network (RNAN) [33]. Dai et al. [3] exploited the second-order attention mechanism to adaptively rescale features by considering feature statistics higher than first-order. Guo et al. [6] developed a dual regression scheme by introducing an additional constraint such that the mappings can form a closed-loop and LR images can be reconstructed to enhance the performance of SR models.

Despite the great success of CNN-based methods, most of them are not suitable for mobile devices. To solve this problem, Ahn et al. [1] proposed the CARN-M model for mobile scenario through a cascading network architecture. Hui et al. [11] proposed the information distillation network (IDN) that explicitly divided the preceding extracted features into two parts. Based on IDN, the also proposed the fast and lightweight information multi-distillation network (IMDN) [10] that is the winner solution of the AIM 2019 constrained image super-resolution challenge [30].

3 METHOD

3.1 Information multi-distillation block

As shown in Figure 2a, the main part of information distillation block (IMDB) [10] is a progressive refinement module (PRM), which is marked with a gray background. The PRM first uses a $3 \times 3$ convolution layer to extract input features for multiple subsequent distillation steps. For each step, the channel splitting operation is employed on the preceding features and it divides the input features into two parts. One part is retained and the other part is fed into the next distillation step. Given the input features $F_{in}$, this procedure can be described
as

\[
\begin{align*}
F_{\text{distilled}_1}, F_{\text{coarse}_1} &= Split_1(L_1(F_{\text{in}})), \\
F_{\text{distilled}_2}, F_{\text{coarse}_2} &= Split_2(L_2(F_{\text{coarse}_1})), \\
F_{\text{distilled}_3}, F_{\text{coarse}_3} &= Split_3(L_3(F_{\text{coarse}_2})), \\
F_{\text{distilled}_4} &= L_4(F_{\text{coarse}_3})
\end{align*}
\]

(1)

where \( L_j \) denotes the \( j \)-th convolution layer (including the activation unit), \( Split_j \) denotes the \( j \)-th channel splitting operation, \( F_{\text{distilled}_j} \) represents the \( j \)-th distilled features, and \( F_{\text{coarse}_j} \) is the \( j \)-th coarse features that will be further processed by succeeding layers. Finally, all the distilled features are concatenated together as the output of the PRM

\[
F_{\text{distilled}} = \text{Concat}(F_{\text{distilled}_1}, F_{\text{distilled}_2}, F_{\text{distilled}_3}, F_{\text{distilled}_4})
\]

(2)

where \( \text{Concat} \) represents the concatenation operation along the channel dimension.

### 3.2 Rethinking the IMDB

Although PRM achieves prominent improvements, it is not efficient enough and introduces some inflexibility because of the channel splitting operation. The distilled features are generated by \( 3 \times 3 \) convolution filters that has many redundant parameters. Moreover, the feature refinement pipeline (along the right branch of the PRM) is coupled together with channel splitting operation so that it is hard to use identity connections only for this pipeline. Next, we will rethink the channel splitting operation and give a new equivalent architecture of the PRM to tackle the aforementioned problems.
As depicted in Figure 2b, the $3 \times 3$ convolution followed by a channel splitting layer can be decoupled into two $3 \times 3$ convolution layers $DL$ and $RL$. The layer $DL$ is responsible for producing the distilled features and $RL$ is the refinement layer that further processes the proceeding coarse features. The whole structure can be described as

$$
F_{\text{distilled},1}, F_{\text{coarse},1} = DL_1(F_{\text{in}}), RL_1(F_{\text{in}})
$$

$$
F_{\text{distilled},2}, F_{\text{coarse},2} = DL_2(F_{\text{coarse},1}), RL_2(F_{\text{coarse},1}),
$$

$$
F_{\text{distilled},3}, F_{\text{coarse},3} = DL_3(F_{\text{coarse},2}), RL_3(F_{\text{coarse},2}),
$$

$$
F_{\text{distilled},4} = DL_4(F_{\text{coarse},3})
$$

Comparing equation [1] with equation [3], we have the following relationships

$$
DL_1(F_{\text{in}}), RL_1(F_{\text{in}}) = \text{Split}_1(L_1(F_{\text{in}})),
$$

$$
DL_2(F_{\text{coarse},1}), RL_2(F_{\text{coarse},1}) = \text{Split}_2(L_2(F_{\text{coarse},1})),
$$

$$
DL_3(F_{\text{coarse},2}), RL_3(F_{\text{coarse},2}) = \text{Split}_3(L_3(F_{\text{coarse},2})),
$$

$$
DL_4(F_{\text{coarse},3}) = L_4(F_{\text{coarse},3})
$$

The above equations describe that each group of split operation can be viewed as two convolution layers that work concurrently. We call this new architecture IMDB-R, which is more flexible than the original IMDB. It has a clearer view on how the PRM works so that we can get more clues on how to design more efficient SR models.

### 3.3 Residual feature distillation block

Inspired by the rethinking of IMDB, in this section, we introduce the residual feature distillation block (RFDB) that is more lightweight and powerful than the IMDB. In Figure 2, we can see that the information distillation operation is actually implemented by a $3 \times 3$ convolution that compresses feature channels at a fixed ratio. However, we find that it is more efficient to use the $1 \times 1$ convolution for channel reduction as have done in many other CNN models. As depicted in Figure 2c, the three convolutions on the left are replaced with $1 \times 1$ convolutions, which significantly reduces the amount of parameters. The rightmost convolution still uses $3 \times 3$ kernels. This is because it locates on the main body of the RFDB and it must take the spatial context into account to better refine the features. For clarity, we call these outer connections feature distillation connections (FDC).

Despite aforementioned improvements, we also introduce more fine-grained residual learning into the network. For this purpose, we design a shallow residual block (SRB), as shown in Figure 2d, which consists of a $3 \times 3$ convolution, an identity connection and the activation unit. The SRB can benefit from residual learning without introducing any extra parameters. The original IMDB only contains mid-level residual connections that are too coarse for the network to benefit most from the residual connections. In contrast, our SRB enables deeper
residual connections and can better utilize the power of residual learning even with a lightweight shallow SR model. We use the proposed RFDB to build our residual feature distillation network (RFDN) as will be described in the next section.

### 3.4 Framework

We use the same framework as IMDN [10], as shown in Figure 1, the residual feature distillation network (RFDN) consists of four parts: the first feature extraction convolution, multiple stacked residual feature distillation blocks (RFDBs), the feature fusion part and the last reconstruction block. Specifically, the initial feature extraction is implemented by a $3 \times 3$ convolution to generate coarse features from the input LR image. Given the input $x$, this procedure can be expressed as

$$F_0 = h(x)$$

(5)

where $h$ denotes the coarse feature extraction function and $F_0$ is the extracted features. The next part of RFDN is multiple RFDBs that are stacked in a chain manner to gradually refine the extracted features. This process can be formulated as

$$F_k = H_k(F_{k-1}), \quad k = 1, \ldots, n$$

(6)

where $H_k$ denotes the $k$-th RFDB function, $F_{k-1}$ and $F_k$ represent the input feature and output feature of the $k$-th RFDB, respectively. After gradually refined by the RFDBs, all the intermediate features are assembled by a $1 \times 1$ convolution layer. Then, a $3 \times 3$ convolution layer is used to smooth the aggregated features as follows

$$F_{\text{assemble}} = H_{\text{assemble}}(\text{Concat}(F_1, \ldots, F_n))$$

(7)

where $\text{Concat}$ is the concatenation operation along the channel dimension, $H_{\text{assemble}}$ denotes the $1 \times 1$ convolution followed by a $3 \times 3$ convolution, and $F_{\text{assemble}}$ is the aggregated features. Finally, the SR images are generated through the reconstruction as follows

$$y = R(F_{\text{assemble}} + F_0)$$

(8)

where $R$ denotes the reconstruction function and $y$ is the output of the network. The reconstruction process only consists of a $3 \times 3$ convolution and a non-parametric sub-pixel operation.

The loss function of our RFDN can be expressed by

$$\mathbb{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \|H_{\text{RFDN}}(I_{iLR}^L) - I_{iHR}^H\|_1$$

(9)

where $H_{\text{RFDN}}$ represents the function of our proposed network, $\theta$ indicates the learnable parameters of RFDN and $\|\cdot\|_1$ is the $l_1$ norm. $I_{iLR}^L$ and $I_{iHR}^H$ are the input LR images and the corresponding ground-truth HR images, respectively.
Fig. 3: The Base block, SRB block, FDC block and RFDB used in ablation study.

4 EXPERIMENTS

4.1 Datasets and metrics

Following previous works [17,32,11,10], we use the recently popular dataset DIV2K [24] to train our models. The DIV2K dataset contains 800 high-quality RGB training images. For testing, we use five widely used benchmark datasets: Set5 [2], Set14 [27], BSD100 [19], Urban100 [9] and Manga109 [20]. We employ peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [26] to measure the quality of the super-resolved images. All the values are calculated on the Y channel of the YCbCr channels converted from the RGB channels as with existing works [12,22,17,34,32,1,11,10].

4.2 Implementation details

We generate the training LR images by down-sampling HR images with scaling factors (×2, ×3 and ×4) using bicubic interpolation in MATLAB. As of preparing for this paper, the IMDN has not released the training code yet. To reproduce the results that reported in the IMDN paper, we use different training settings from the original paper. More details will be discussed in section 4.5. In this paper, we randomly crop 64 patches of size 64 × 64 from the LR images as input for each training minibatch. We augment the training data with random horizontal flips and 90 rotations. We train our model with ADAM optimizer by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The learning rate is initialized as $5 \times 10^{-4}$ and halved at every $2 \times 10^5$ minibatch updates. When training the final models, the ×2 model is trained from scratch. After the model converges, we use it as a pretrained network for other scales. All the models in the ablation study are trained from scratch for saving the training time. We implement two models in this paper, which are named RFDN and RFDN-L. RFDN uses a channel number of 48 while RFDN-L uses a channel number of 52 to achieve a better reconstruction quality. We set the number of RFDB to 6 in both RFDN and RFDN-L. The networks are implemented by using PyTorch framework with a NVIDIA 1080Ti GPU.
Table 1: Investigations of FDC and SRB on the benchmark datasets with scale factor of $\times 4$. The best results are highlighted.

| Method  | Params | Set5          | Set14          | B100           | Urban100       | Manga109       |
|---------|--------|---------------|---------------|----------------|----------------|----------------|
| Base    | 652K   | 32.08/0.8932  | 28.55/0.7802  | 27.53/0.7345  | 26.05/0.7842  | 30.28/0.9050  |
| SRB     | 652K   | 32.19/0.8949  | 28.58/0.7809  | 27.53/0.7347  | 26.07/0.7849  | 30.40/0.9074  |
| FDC     | 637K   | 32.18/0.8945  | 28.58/0.7811  | 27.55/0.7352  | 26.09/0.7849  | 30.47/0.9077  |
| RFDB    | 637K   | 32.18/0.8950  | 28.61/0.7820  | 27.56/0.7356  | 26.10/0.7859  | 30.55/0.9082  |

Table 2: Investigations of the distillation rate on the benchmark datasets with scale factor of $\times 4$. The best results are highlighted. ↑ represents rising, ↓ represents falling and ∧ represents rising first and then falling.

| Ratio | Params | Set5          | Set14          | B100           | Urban100       | Manga109       |
|-------|--------|---------------|---------------|----------------|----------------|----------------|
| 0.25  | 523K   | 32.18/0.8946  | 28.57/0.7811  | 27.53/0.7348  | 26.09/0.7851  | 30.44/0.9071  |
| 0.5   | 544K   | 32.16/0.8945  | 28.60/0.7819  | 27.55/0.7351  | 26.10/0.7858  | 30.45/0.9074  |
| 0.75  | 565K   | 32.15/0.8944  | 28.61/0.7816  | 27.54/0.7350  | 26.12/0.7853  | 30.46/0.9081  |
| -     | -      | ↑/↓           | ↑/∧           | ∧/∧           | ↑/∧           | ↑/↑           |

4.3 Model analysis

Ablation study To evaluate the importance of the proposed feature distillation connection (FDC) and shallow residual block (SRB), we design four blocks that will be stacked as the body part of the SR network (Figure 1), respectively. The four blocks are depicted in Figure 3 and the evaluation results are shown in Table 1. Comparing the first two rows of Table 1, we can find that SRB improves the performance (e.g. PSNR: $+0.12\text{dB}$, SSIM: $+0.0024$ for Manga109) without introducing any extra parameters. We can also observe similar improvements when comparing the last two rows, which indicates the effectiveness of the shallow residual block. By adding FDC, the performance of the base method is improved by a large margin, for example the PSNR of Manga109 improves from 30.28 to 30.47 ($+0.19\text{dB}$). Thanks to FDC and SRB, our RFDB significantly outperforms the base block.

Investigation of distillation rate We investigate the distillation rate of the feature distillation connections in Table 2. Different distillation rates indicate different number of output channels in the feature distillation connections. As shown in the last row of Table 2 when the distillation rate increases, the growth trends of PSNR and SSIM are different on each dataset. Overall, the distillation rate of 0.5 has a good trade-off between SR performance and the number of parameters, which is adopted as the final distillation rate in our RFDN and RFDN-L.

Model complexity analysis Figure 4 depicts the comparison of PSNR vs. parameters on Set5 $\times 4$ dataset. The models depicted in Figure 4 including DRRN [22], LapSRN [13], VDSR [12], MemNet [23], IDN [11], CARN-M [1]
When evaluating a lightweight model, the number of model parameters is a key factor to take into account. From Table 3, we can observe that our RFDN achieves comparable or better performance when comparing with the state-of-the-art lightweight models with fewer parameters. As shown in Figure 4, though IMDN achieves prominent improvements compared with the previous methods, such as MemNet and IDN, it has more parameters than most of the lightweight models. In contrast, our RFDN achieves better performance than VDSR, MemNet, IDN, and IMDN with fewer parameters. When using more feature channels, our RFDN-L achieves even better results than RFDN while maintaining a modest model size. To get a more comprehensive understanding of the model complexity, we also show the comparison of PSNR vs. Mult-Adds on Set5 × 4 dataset in Figure 5. As we can see, our RFDN and RFDN-L achieve higher PSNR than IMDN while using fewer calculations. IMDN won the first place in the parameters and inference tracks of AIM 2019 constrained super-resolution challenge [30], so we compare our RFDN with IMDN in terms of FPS. Our RFDN (44 FPS) has a comparable inference speed with IMDN (49 FPS) while being more accurate and lightweight. Moreover, our method has fewer calculations than IMDN and can save more energy.

4.4 Comparison with state-of-the-arts

We compare the proposed RFDN with various lightweight SR methods on ×2, ×3 and ×4 scales, including SRCNN [4], FSRCNN [5], VDSR [12], DRCN [13], LapSRN [14], DRRN [22], MemNet [23], IDN [11], SRMDNF [31], CARN [1] and IMDN [10]. Table 3 shows the quantitative comparisons on the five benchmark datasets. We can find that the proposed RFDN can make a better trade-off than IMDN. Our RFDN can achieve comparable or better results with state-of-the-art methods while using 534/541/550K parameters for ×2/×3/×4 SR. By using slightly more parameters, our RFDN-L achieves the best in most quantitative results, especially on large scaling factors.

4.5 About the experimental settings

As described in section 4.2, we use a slightly different experimental setup when training our models. In order to get a clearer insight on the improvements of our
RFDBs (E-RFDBs) in E-RFDN. The number of feature channels in E-RFDN is the CCA layer in RFDB with the ESA block \[18\] and we use 4 such enhanced improvements on network design of our RFDN indeed boosts the performance of both PSNR and SSIM with much fewer parameters, which proves that the benchmark datasets. Our RFDN outperforms IMDN on all the datasets in terms experimental settings. Table 4 shows the performance comparison on the five RFDB-L (Ours) RFDN-L (Ours) CARN \[1\] IDN \[11\] MemNet \[23\] DRRN \[22\] LapSRN \[14\] FSRCNN \[5\] SRCNN \[4\] Bicubic RFDN-L (Ours) RFDN (Ours) IMDN \[10\] CARN \[1\] SRMDNF \[31\] IDN \[11\] MemNet \[23\] DRRN \[22\] LapSRN \[14\] VDSR \[12\] FSRCNN \[5\] SRCNN \[4\] Bicubic Method PSNR/SSIM PSNR/SSIM Table 3: Average PSNR/SSIM for scale factor 2, 3 and 4 on datasets Set5, Set14, BSD100, Urban100, and Manga109. The best and second best results are highlighted in red and blue respectively.

Table 4: Performance comparison of RFDN and IMDN under the same experimental settings. Both models are trained from scratch with scaling facotr ×4.

Table 4: Performance comparison of RFDN and IMDN under the same experimental settings. Both models are trained from scratch with scaling facotr ×4.

RFDN, we train both RFDN and the IMDN \[10\] from scratch under the same experimental settings. Table 4 shows the performance comparison on the five benchmark datasets. Our RFDN outperforms IMDN on all the datasets in terms of both PSNR and SSIM with much fewer parameters, which proves that the improvements on network design of our RFDN indeed boosts the performance of image SR.

4.6 Enhanced RFDN for AIM20 challenge

As shown in Table 5, our enhanced RFDN (E-RFDN) won the first place in the AIM 2020 efficient super-resolution challenge \[29\]. Specifically, we replace the CCA layer in RFDB with the ESA block \[19\] and we use 4 such enhanced RFDBs (E-RFDBs) in E-RFDN. The number of feature channels in E-RFDN is

| Method | Params | Set5 | Set14 | BSD100 | Urban100 | Manga109 |
|--------|--------|------|-------|--------|----------|----------|
| IMDN \[10\] | 715K 32.16/0.8908 28.39/0.7812 27.54/0.7350 26.05/0.7841 30.42/0.9074 |
| RFDN | 550K 32.24/0.8953 28.59/0.7814 27.54/0.7355 26.15/0.7868 30.48/0.9080 |
set to 50 and the feature distillation rate is 0.5. During the training of E-RFDN, HR patches of size $256 \times 256$ are randomly cropped from HR images, and the mini-batch size is set to 64. The E-RFDN model is trained by minimizing L1 loss function with Adam optimizer. The initial learning rate is set to $5 \times 10^{-4}$ and halved at every 200 epochs. After 1000 epochs, L2 loss is used for fine-tuning with learning rate of $1 \times 10^{-5}$. DIV2K and Flickr2K datasets are used for training the E-RFDN model. We include the top five methods in Table 5, the “#Activations” measures the number of elements of all outputs of convolutional layers. Compared to the first place method IMDN in AIM 2019 constrained SR challenge [28], our method provides a significant gain with respect to the runtime, parameters, FLOPs, and activations. More details and results can be found in [29].

Table 5: AIM 2020 efficient SR challenge results (we only include the first five methods).

| Team            | Author  | PSNR | Runtime | #Params | FLOPs | #Activations | Extra |
|-----------------|---------|------|---------|---------|-------|--------------|-------|
| NJU_MCG (ours)  | TinyJie | 28.75| 0.037   | 0.433   | 27.10 | 112.03       | Yes   |
| AirA_GG         | Now     | 28.70| 0.037   | 0.667   | 44.98 | 118.49       | Yes   |
| UESTC-MediaLab  | Mulns   | 28.70| 0.060   | 0.461   | 30.06 | 219.61       | Yes   |
| XPixel          | zzyhy   | 28.70| 0.066   | 0.272   | 32.19 | 270.53       | Yes   |
| HaiYun          | Sudo    | 28.78| 0.054   | 0.777   | 49.67 | 112.31       | Yes   |
| IMDN            | zheng222| 28.78| 0.050   | 0.891   | 58.53 | 154.14       | Yes   |
| Baseline        | MSRResNet| 28.70| 0.114   | 1.517   | 166.36| 292.55       | Yes   |

5 CONCLUSION

In this paper, we give a comprehensive analysis of the information distillation mechanism for lightweight image super-resolution. Then we rethink the information multi-distillation network (IMDN) and propose the feature distillation connections (FDC) that are much more lightweight and flexible. To further boost the super-resolution performance, we also propose the shallow residual block (SRB) that incorporates the identity connection with one convolutional block. By using the shallow residual blocks and the feature distillation connections, we build the residual feature distillation network (RFDN) for fast and lightweight image super-resolution. Extensive experiments have shown that the proposed method achieves state-of-the-art results both quantitatively and qualitatively. Furthermore, our model has a modest number of parameters and multi-adds such that it can be easily ported to mobile devices.

References

1. Ahn, N., Kang, B., Sohn, K.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: ECCV (10). Lecture Notes in Computer Science, vol. 11214, pp. 256–272. Springer (2018)
2. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: BMVC. pp. 1–10. BMVA Press (2012)

3. Dai, T., Cai, J., Zhang, Y., Xia, S., Zhang, L.: Second-order attention network for single image super-resolution. In: CVPR. pp. 11065–11074. Computer Vision Foundation / IEEE (2019)

4. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: ECCV (4). Lecture Notes in Computer Science, vol. 8692, pp. 184–199. Springer (2014)

5. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: ECCV (2). Lecture Notes in Computer Science, vol. 9006, pp. 391–407. Springer (2016)

6. Guo, Y., Chen, J., Wang, J., Chen, Q., Cao, J., Deng, Z., Xu, Y., Tan, M.: Closed-loop matters: Dual regression networks for single image super-resolution. CoRR abs/2003.07018 (2020)

7. Haris, M., Shakhnarovich, G., Ukita, N.: Deep back-projection networks for super-resolution. In: CVPR. pp. 1664–1673. IEEE Computer Society (2018)

8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778. IEEE Computer Society (2016)

9. Huang, J., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR. pp. 5197–5206. IEEE Computer Society (2015)

10. Hui, Z., Gao, X., Yang, Y., Wang, X.: Lightweight image super-resolution with information multi-distillation network. In: ACM Multimedia. pp. 2024–2032. ACM (2019)

11. Hui, Z., Wang, X., Gao, X.: Fast and accurate single image super-resolution via information distillation network. In: CVPR. pp. 723–731. IEEE Computer Society (2018)

12. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR. pp. 1646–1654. IEEE Computer Society (2016)

13. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: CVPR. pp. 1637–1645. IEEE Computer Society (2016)

14. Lai, W., Huang, J., Ahuja, N., Yang, M.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR. pp. 5835–5843. IEEE Computer Society (2017)

15. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR. pp. 105–114. IEEE Computer Society (2017)

16. Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., Wu, W.: Feedback network for image super-resolution. In: CVPR. pp. 3867–3876. Computer Vision Foundation / IEEE (2019)

17. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: CVPR Workshops. pp. 1132–1140. IEEE Computer Society (2017)

18. Liu, J., Zhang, W., Tang, Y., Tang, J., Wu, G.: Residual feature aggregation network for image super-resolution. In: CVPR. pp. 2356–2365. IEEE (2020)

19. Martin, D.R., Fowlkes, C.C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV. pp. 416–425 (2001)
20. Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., Aizawa, K.: Sketch-based manga retrieval using manga109 dataset. Multimedia Tools Appl. 76(20), 21811–21838 (2017)

21. Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: CVPR. pp. 1874–1883. IEEE Computer Society (2016)

22. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR. pp. 2790–2798. IEEE Computer Society (2017)

23. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. In: ICCV. pp. 4549–4557. IEEE Computer Society (2017)

24. Timofte, R., Agustsson, E., Gool, L.V., Yang, M., Zhang, L., Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M., et al.: NTIRE 2017 challenge on single image super-resolution: Methods and results. In: CVPR Workshops. pp. 1110–1121. IEEE Computer Society (2017)

25. Timofte, R., Rothe, R., Gool, L.V.: Seven ways to improve example-based single image super resolution. In: CVPR. pp. 1865–1873. IEEE Computer Society (2016)

26. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Processing 13(4), 600–612 (2004)

27. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Curves and Surfaces. Lecture Notes in Computer Science, vol. 6920, pp. 711–730. Springer (2010)

28. Zhang, K., Gu, S., Timofte, R., Hui, Z., Wang, X., Gao, X., Xiong, D., Liu, S., Gang, R., Nan, N., Li, C., Zou, X., Kang, N., Wang, Z., Xu, H., Wang, C., Li, Z., Wang, L., Shi, J., Sun, W., Lang, Z., Nie, J., Wei, W., Zhang, L., Niu, Y., Zhuo, P., Kong, X., Sun, L., Wang, W.: Aim 2019 challenge on constrained super-resolution: Methods and results. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). pp. 3565–3574 (2019)

29. Zhang, K., Danelljan, M., Li, Y., Timofte, R., et al.: AIM 2020 challenge on efficient super-resolution: Methods and results. In: European Conference on Computer Vision Workshops (2020)

30. Zhang, K., Nan, N., Li, C., Zou, X., Kang, N., Wang, Z., Xu, H., Wang, C., Li, Z., Wang, L., Shi, J., Gu, S., Sun, W., Lang, Z., Nie, J., Wei, W., Zhang, L., Niu, Y., Zhuo, P., Kong, X., Sun, L., Wang, W., Timofte, R., Hui, Z., Wang, X., Gao, X., Xiong, D., Liu, S., Gang, R.: AIM 2019 challenge on constrained super-resolution: Methods and results. In: ICCV Workshops. pp. 3565–3574. IEEE (2019)

31. Zhang, K., Zuo, W., Zhang, L.: Learning a single convolutional super-resolution network for multiple degradations. In: CVPR. pp. 3262–3271. IEEE Computer Society (2018)

32. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: ECCV (7). Lecture Notes in Computer Science, vol. 11211, pp. 294–310. Springer (2018)

33. Zhang, Y., Li, K., Li, K., Zhong, B., Fu, Y.: Residual non-local attention networks for image restoration. In: ICLR (Poster). OpenReview.net (2019)

34. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: CVPR. pp. 2472–2481. IEEE Computer Society (2018)