Residual Feature Aggregation Network for Image Super-Resolution

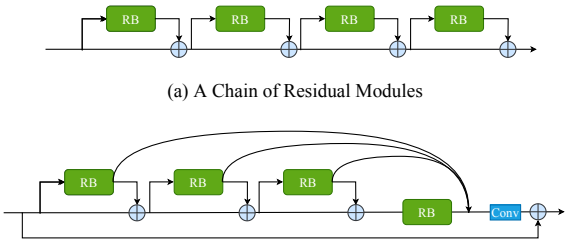
Jie Liu Wenjie Zhang Yuting Tang Jie Tang* Gangshan Wu
State Key Laboratory for Novel Software Technology, Nanjing University, China
{jieliu,zwj,MF1833070}@smail.nju.edu.cn, {tangjie,gswu}@nju.edu.cn

Abstract

Recently, very deep convolutional neural networks (CNNs) have shown great power in single image super-resolution (SISR) and achieved significant improvements against traditional methods. Among these CNN-based methods, the residual connections play a critical role in boosting the network performance. As the network depth grows, the residual features gradually focused on different aspects of the input image, which is very useful for reconstructing the spatial details. However, existing methods neglect to fully utilize the hierarchical features on the residual branches. To address this issue, we propose a novel residual feature aggregation (RFA) framework for more efficient feature extraction. The RFA framework groups several residual modules together and directly forwards the features on each local residual branch by adding skip connections. Therefore, the RFA framework is capable of aggregating these informative residual features to produce more representative features. To maximize the power of the RFA framework, we further propose an enhanced spatial attention (ESA) block to make the residual features to be more focused on critical spatial contents. The ESA block is designed to be lightweight and efficient. Our final RFANet is constructed by applying the proposed RFA framework with the ESA blocks. Comprehensive experiments demonstrate the necessity of our RFA framework and the superiority of our RFANet over state-of-the-art SISR methods.

1. Introduction

The task of single image super-resolution (SISR) is to map a degraded low-resolution (LR) image to a visually high-resolution (HR) image, which is a highly ill-posed procedure since multiple HR solutions can map to one LR input. Many image SR methods have been proposed to tackle this inverse problem, including early interpolation-based [37], reconstruction-based [34], and recent learning based methods [27, 28, 22, 4, 12, 13, 36, 3].



(a) A Chain of Residual Modules

(b) Residual Feature Aggregation (RFA)

Figure 1. (a) A chain of residual modules. A residual module consists of a residual block (RB) and an identity connection. (b) The residual feature aggregation (RFA) framework.

Recent deep convolutional neural network based methods have made great progress in reconstructing the HR images. The first successful attempt was done by Dong *et al*. [4], who proposed the three-layer SRCNN for SISR and achieved superior performance against conventional methods. Kim *et al*. further increased the depth to 20 in VDSR [13] and DRCN [14] by introducing residual learning to ease the training difficulty. Following these pioneering works, many CNN-based methods have been proposed and achieved state-of-the-art results in SISR [15, 18, 40, 38, 3, 17, 7, 9, 39].

Although considerable improvements have been achieved in SISR, existing CNN-based models are still faced with some limitations. As the network depth grows, the features in each convolutional layer would be hierarchical with different receptive fields. Most existing CNN-based models do not make fully use of the information from the intermediate layers. Especially, residual learning is widely used in CNN-based models to extract the residual information of input features, while almost all the existing SR models only use the residual learning as a strategy to ease the training difficulty. For clarity, we call the entire residual construct as a residual module and the residual branch as a residual block. Usually, a SR model is made by stacking a bunch of residual modules, where the residual features are fused with the identity features before propagating to the next module (Fig. 1(a)). As a

*Corresponding author.

result, later residual blocks can only see the complex fused features. These methods neglect to make fully use of the cleaner residual features, thereby leading to performance degradation. The residual features, however, are extremely helpful for reconstructing the HR images.

To address these problems, we propose a residual feature aggregation (RFA) framework, which aggregates the local residual features for more powerful feature representation. Fig. 1(a) shows a common network design where multiple residual modules are stacked together to build a deep network. Under this design, the residual features of preceding blocks must go through a long path to propagate to subsequent blocks. After a series of addition and convolutional operations, these features are quickly merged with the identity features to form more complex features. Therefore, these highly representative residual features are used very locally, which limits the representational power of the network. As depicted in Fig. 1(b), the proposed RFA framework reorganizes the stacked residual modules, where the last residual module is extended to cover the fist three residual modules to ease the training difficulty. Then the residual features of the first three blocks are sent directly to the output of the last residual block. Finally, these hierarchical features are concatenated together and sent to a 1×1 convolutional layer to generate a more representative feature. The only overhead is a 1×1 convolution every four residual blocks, which is negligible compared with the whole very deep networks.

As shown in Fig. 8, the residual features of different residual blocks can reflect different aspects of the spatial contents. But these residual features are not highlighted enough. It is necessary to enhance the spatial distribution of residual features with spatial attention mechanism so that the performance of our RFA framework could be further improved. However, existing spatial attention mechanisms in image SR are either less powerful or computationally intensive. For example, the plain spatial attention in [10] lacks of a large receptive field which is essential for image SR and the Non-Local mechanisms in [19, 3] consume a lot of computational resource. To solve this issue, we propose a lightweight and efficient enhanced spatial attention (ESA) block. The ESA block enables a large receptive field by the joint use of a strided convolution and a max-pooling with large window size. To keep the body of the ESA block lightweight enough, we apply a 1×1 convolution at the beginning of the ESA block for channel dimension reduction.

To verify the effectiveness of the proposed methods, we build a very deep network RFANet by combining the RFA framework with the ESA block. The RFANet achieves comparable or superior results compared with RCAN [38] (16M) and SAN [3] (15.7M) by using much fewer parameters (11M). In summary, the main contributions of this paper are as follows:

- We propose a general residual feature aggregation (RFA) framework for more accurate image SR. Comprehensive ablation study shows that the performance of residual networks as well as dense networks can get a substantial improvement.

- We propose an enhanced spatial attention (ESA) block to adaptively rescale features according to the spatial context. The ESA block allows the network to learn more discriminative features. Besides, it is lightweight and has better performance than the plain spatial attention block.

- We propose a residual feature aggregation network (RFANet) which is constructed by incorporating the proposed RFA framework with the powerful ESA block. Thanks to the enhanced spatial attention mechanism, the RFA framework can aggregate more representative features, thus generating more accurate SR results.

2. Related Work

Super-resolution can be broadly divided into two main categories: traditional and deep learning based methods. Due to the powerful learning ability, the classical methods have been outperformed by their deep learning based counterparts. In this section, we briefly review the works related to deep neural networks for single image super-resolution.

2.1. CNN-based Networks

Dong *et al*. [4] first proposed a shallow three-layer convolutional neural network (SRCNN) for image SR and achieved superior performance against previous works. Inspired by this pioneering work, Kim *et al*. designed deeper VDSR [13] and DRCN [14] with 20 layers based on residual learning. Later, Tai *et al*. introduced recursive blocks in DRRN [24] and memory blocks in MemNet [25]. These methods extract features from the interpolated LR images, which consumes a lot of memory and computation time. To address this problem, Shi *et al*. proposed an efficient sub-pixel convolutional layer in ESPCN [23], where LR feature maps are upscaled into HR output at the end of the network. Thanks to the efficient sub-pixel layer, many very deep networks have been proposed for a better performance. Lim *et al*. proposed a very deep and wide network EDSR [18] by stacking modified residual blocks in which the batch normalization (BN) layers are removed. Ledig *et al*. introduced the SRResNet in [16] and are further improved in [31] by introducing the dense connections. Zhang *et al*. also used dense connections in RDN [40] to utilize all the hierarchical features from all the convolutional layers.

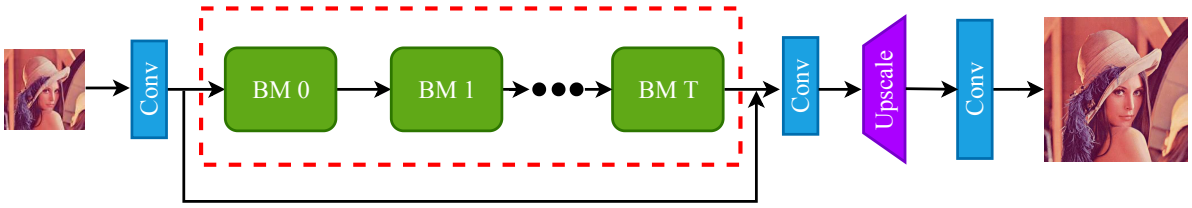


Figure 2. Basic architecture of a SR network. The red dotted rectangle represents the trunk part of the network, which consists of T base modules (BM).

2.2. Attention-based Networks

Attention mechanism are widely used in recent computer vision tasks, such as image captioning [32, 2], image and video classification [8, 30]. It can be interpreted as a way to bias the allocation of available resources towards the most informative parts of an input signal [8]. Wang *et al*. [29] proposed a powerful trunk-and-mask attention mechanism inserted between the intermediate stages of deep residual networks. Hu *et al*. [8] proposed the squeeze-and-excitation network (SENet) to exploit channel-wise relationships and achieved a significant improvement for image classification.

Recently, some attention-based models are also proposed to further improve the SR performance. Zhang *et al*. [38] proposed the residual channel attention network (RCAN) by introducing the channel attention mechanism into a modified residual block for image SR. The channel attention mechanism uses global average pooling to extract channel statistics which are called first-order statistics. On the contrary, Dai *et al*. [3] proposed the second-order attention network (SAN) to explore more powerful feature expression by using second-order feature statistics. RCAN and SAN are the two best performing methods among all currently published methods in terms of PSNR.

3. Methodology

3.1. Basic Network Architecture for Image SR

Many recent SR networks have similar network architectures. Here we introduce one of the basic architecture used by some state-of-the-art methods [18, 40, 38, 3]. As shown in Fig. 2, a basic image SR network usually consists of three parts: the head part, the trunk part and the reconstruction part. The head part is responsible for initial feature extraction with only one convolutional layer. Given the LR input I_{LR}, we can get the shallow feature F_0 through this layer

$$F_0 = \mathcal{H}(I_{LR}) \tag{1}$$

where $\mathcal{H}$ stands for the shallow feature extraction function of the head part. Then the extracted feature F_0 is sent to the trunk part for deep feature learning. The trunk part is made up of T base modules (BM), which can be formulated as

$$F_t = \mathcal{B}_t(F_{t-1}) = \mathcal{B}_t(\mathcal{B}_{t-1}(\dots(\mathcal{B}_0(F_0))\dots)) \tag{2}$$

where $\mathcal{B}_t$ denotes the t-th base module function. F_{t-1} is the input of the t-th module and F_t is the corresponding output. Finally, the extracted deep feature F_t is upscaled through the reconstruction part

$$I_{SR} = \mathcal{R}(F_t + F_0) = \mathcal{G}(I_{LR}) \tag{3}$$

where I_{SR} is the super-resolved image, $\mathcal{R}$ denotes the reconstruction function and $\mathcal{G}$ denotes the function of the SR network. Here, global residual learning is used to ease the training difficulty, so the input to $\mathcal{R}$ is the element-wise addition of F_t and F_0. The key module of the reconstruction part is the upscale module, where appropriate number of sub-pixel [23] convolutions are applied.

The SR network will be optimized with L_1 loss function. Given a training set of N LR image patches I_{LR} and their HR counterparts I_{HR}, the loss function of the basic network with the parameter set Θ is

$$\mathcal{L}(\Theta) = \frac{1}{N}\sum_{i=1}^{N} ||\mathcal{G}(I^i_{LR}) - I^i_{HR}||_1 \tag{4}$$

3.2. Residual Feature Aggregation Framework

Residual learning has demonstrated its significance for the image classification problem. Recently, residual learning is also introduced in image SR to further boost the performance. Fig. 3(Left) depicts a basic residual module used in EDSR [18] and ESRGAN [31]. The residual modules are often stacked together to form the trunk part of the SR network (Fig. 2). Each residual module consists of two branches:the residual branch (*i.e*. residual block) and the identity branch. In the task of image SR, the residual block can produce some useful hierarchical features focusing on different aspects of the original LR image. Consider the scenario of several consecutive residual modules (*e.g*. Fig. 1(a)), the feature of the first residual block must go though a long path to reach the last module via repetitive

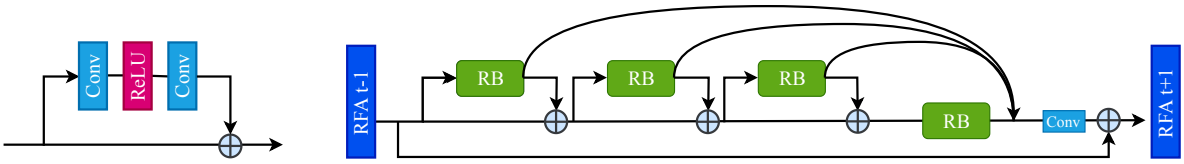


Figure 3. **Left**: A basic residual module. **Right**: Details of the RFA module, which contains 4 residual blocks (RB) and a 1×1 convolutional layer.

addition and convolution operations. As a result, the residual feature is hard to be fully utilized and plays a very local role in the learning process of the entire network.

To solve this issue, we propose a residual feature aggregation (RFA) framework to make a better use of the local residual features. Fig. 3(Right) shows the details of an RFA module which contains four residual blocks. As we can see, the residual features of the first three blocks are sent directly to the end of the RFA module and then concatenated together with the output of the last residual block. Finally, a 1×1 convolution is applied to fuse these features before the element-wise addition with the identity feature. Compared with the way of simply stacking multiple residual modules, our RFA framework enables non-local use of the residual features. The useful hierarchical information that preceding residual blocks contain can be propagated to the end of the RFA module without any loss or interference, thus leading to a more discriminative feature representation.

The proposed residual feature aggregation methodology is a general framework that can be easily applied with existing SR blocks (*e.g.* dense block [40]). We will investigate the effects in detail when our RFA framework is used in conjunction with the state-of-the-art blocks.

3.3. Enhanced Spatial Attention Block

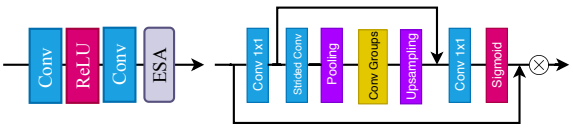


Figure 4. **Left**: The enhanced spatial attention (ESA) block. **Right**: Details of the ESA mechanism.

In order to maximize the effectiveness of our RFA framework, it is best to be used in conjunction with the spatial attention mechanism, since we need the residual features to be focused on spatial contents of key importance. To this end, we design an enhanced spatial attention (ESA) block that is more powerful than the plain one in [10]. The ESA mechanism works at the end of the residual block (Fig.4(Left)) to force the features to be more focused on the regions of interest. We can get a more representative feature when aggregating these highlighted features together. In the design of an attention block, several elements have to be carefully considered. First, the attention block must be lightweight enough since it will be inserted into every residual module of the network. Second, a large receptive field is required for the attention block to work well for the task of image SR.

As shown in Fig. 4(Right), the proposed ESA mechanism starts with a 1×1 convolutional layer to reduce channel dimensions, so that the whole block can be extremely lightweight. Then to enlarge the receptive field we use one strided convolution (with stride 2) followed by a max-pooling layer. The combination of strided convolution and max-pooling is widely used in image classification to quickly reduce the spatial dimensions at the beginning of the network. However, the receptive field enlargement brought by a regular 2×2 max-pooling layer is still very limited. So we choose to apply the max-pooling operation with a larger window (*e.g.* 7×7) and stride (*e.g.* stride 3). Corresponding to the front, an up-sampling layer is added to recover the spatial dimensions and a 1×1 convolutional layer is used to recover the channel dimensions. Finally, the attention mask is generated via a sigmoid layer. We also use a skip connection to forward the high-resolution features before spatial dimension reduction directly to the end of the block.

Put aside the amount of calculation, a potentially better way to implement the spatial attention block is to use the Non-Local block. Actually, there are works [19, 3] that have attempted to use the Non-Local block to model pixel-wise similarities in image SR. Though it brings performance boost, the huge computation overhead is unacceptable which violates the first element of our design principle.

3.4. Implementation Details

We apply the RFA framework with the ESA block to build our final SR network (RFANet). RFANet uses 30 RFA modules and each RFA module contains 4 ESA blocks. In the ESA block, the reduction ratio of the 1×1 convolutional layer is set to 4 and we use three 3×3 convolutions in the convolutional groups. For other convolution filters outside the ESA block, the number of filters are set to 64.

3.5. Discussions

Difference to MemNet. MemNet stands for the very deep persistent memory network proposed by Tai *et al*. [25]. The most crucial part of MemNet is the stacked memory blocks. A memory block consists of a recursive unit and a gate unit to explicitly mine persistent memory though an adaptive learning process. The recursive unit is implemented by a residual building block and this residual building block is executed in each recursion to generate multi-level representations. The gate unit is responsible for adaptively learning these representations. The unfolded memory block has a similar connection pattern with our RFA framework. The key difference is that memory block aggregates the output features of a whole residual module while our RFA framework concentrates on the feature of the residual branch. Moreover, the memory block operates in a recursive manner very locally. In RFA framework, the basic building blocks are organized in a chain way so that each residual branch can focus on different aspects of the LR image, so the aggregated residual features would be more diverse and discriminative.

Difference to RDN. The main building block of RDN [40] is called residual dense block (RDB). RDB combines residual skip connections with dense connections. The motivation of RDB is that the hierarchical feature representations should be fully used to learn local patterns. In a dense block, each layer can have direct access to its subsequent layers. Before merging with the identity branch, a 1×1 convolutional layer is also used to fuse features coming from all the intermediate layers. Though shares a similar motivation behind the block design, our RFA module operates in a quite different way. A RFA module contains several residual modules and mainly aggregates features from the residual branches. In contrast, the RDB collects intermediate features between plain convolutional layers. The dense block is very computationally intensive because of the dense feature fusion strategy. Our RFA module is much more lightweight since the feature aggregation only happens at the end of the module. In general, the proposed RFA module works at a higher level than the dense block and the performance can be further boosted when applying our RFA framework to the dense block (Table 1).

4. Experiments

4.1. Settings

Following previous works [40, 38, 3], we use 800 high-resolution training images from DIV2K [26] dataset as training set. During training, data augmentation is performed by randomly rotating $90°$, $180°$, $270°$ and horizontally flipping. In each training mini-batch, 16 LR color patches with size 48×48 are used. For testing, we use five standard benchmark datasets: Set5 [1], Set14 [33], B100 [20], Urban100 [12], and Manga109 [21]. Bicubic (BI) and blur-downscale (BD) degradation models [36] are used when conducting experiments. The SR results are evaluated by PSNR and SSIM metrics on Y channel of transformed YCbCr space. Our model is trained by ADAM optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 10^{-8}$. The learning rate is initialized as 5×10^{-4} and then decreases to half every 2×10^5 iterations. We use PyTorch framework to implement our models with a Titan Xp GPU.

4.2. Combination with Residual Block

In this section, we investigate the combination of our RFA framework with the basic residual block used in EDSR [18]. Different from the original residual block used in image classification, EDSR removes the Batch Normalization layers and achieved substantial improvements. The baseline model contains 120 residual modules and we refer to this model as "EDSR-Baseline". Our RFA model adopts 30 RFA modules to keep the number of residual blocks the same as EDSR-Baseline for a fair comparison. We refer to this model as "RFA-EDSR" for short. As shown in the second column of Table 1, the PSNR of EDSR-Baseline is 32.40 dB which demonstrates a strong baseline for image SR. When deploying our RFA framework with the residual block (RFA-EDSR), the PSNR reaches 32.50 dB. Compared with the EDSR-Baseline, the RFA-EDSR has only one more 1×1 convolution every four residual blocks while boosting the PSNR by 0.1 dB. We attribute this considerable improvement to the effective design of our RFA framework where the residual feature in each residual block can be better utilized by the network. These comparisons demonstrate that the proposed RFA framework is essential to very deep networks for Image SR.

4.3. Combination with Dense Block

The motivation behind dense block [40] is also to combine hierarchical cues available along the network depth to get richer feature representations. But the combination happens inside a single residual module. In contrast, our RFA framework aims to combining the residual features at a higher level. It is reasonable to apply the RFA framework in conjunction with the dense block to further improve the performance. In this ablation study, we use 42 dense blocks to maintain similar number of parameters with EDSR-Baseline and RFA-EDSR. We refer to the dense block baseline model as "Dense-Baseline". When applying RFA framework with dense blocks (RFA-Dense), we use 14 RFA modules to make these two models comparable. As shown in the third column of Table 1, RFA-Dense improves the performance of Dense-Baseline from 32.42 dB to 32.51 dB. This indicates that the proposed RFA framework can further combine the hierarchical information against the dense block. Note that this semi-trained RRA-Dense model

Table 1. Ablation results of different blocks combined with the RFA framework. We report the best PSNR (dB) values on Set5 ($\times 4$) in 4×10^5 iterations.

Name	EDSR-Baseline	RFA-EDSR	Dense-Baseline	RFA-Dense	CA	SA	ESA	RFA-CA	RFA-SA	RFA-ESA (RFANet)
Residual Block	√	√								
Dense Block			√	√						
Channel Attention Block					√			√		
Spatial Attention Block						√			√	
Enhanced Spatial Attention Block							√			√
Residual Feature Aggregation		√		√				√	√	√
PSNR	32.40	32.50	32.42	32.51	32.56	32.48	32.56	32.56	32.54	32.65

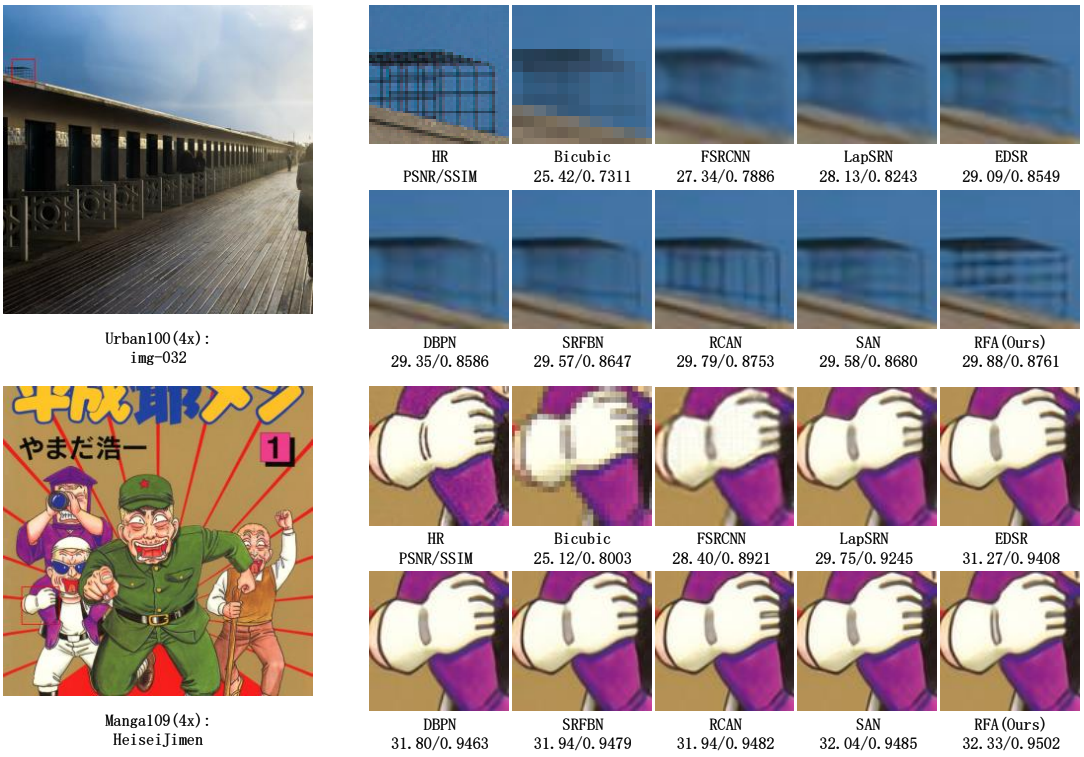


Figure 5. Visual comparisons for $\times 4$ SR with BI degradation model.

already has higher PSNR than the original RDN [40].

4.4. Combination with Attention Block

By using attention mechanism, the performance of image SR has achieved significant improvements. Here, we will comprehensively investigate the effects of applying our RFA framework to the attention blocks. Table 1 shows the ablation results including channel attention (CA) [38], spatial attention (SA) [10], enhanced spatial attention (ESA) and their combinations (*i.e.* RFA-CA, RFA-SA and RFA-ESA) with the RFA framework. As we can see, by using channel attention block alone, the PSNR already achieves 32.56 dB, which demonstrates the excellent performance of channel attention mechanism. The plain SA has a much lower PSNR than CA, but when equipped with our RFA framework, the RFA-SA achieves a comparable PSNR with CA. On the contrary, RFA-CA does not show any considerable improvement compared with CA. This indicates that the RFA framework is best to be used with spatial attention mechanism. To this end, we design an enhanced spatial attention block and it achieves the same PSNR as CA, which indicates its effectiveness for image SR. Furthermore, The RFA-ESA solution improved the ESA from 32.56 dB to 32.65 dB. This shows that the proposed RFA framework can further boost the performance of spatial attention mechanism by a large margin. Among all the investigated methods, the proposed RFA-ESA method achieves the best performance and we will use it to compare with the state-of-the-art methods. From now on, we use the name "RFANet" to represent the RFA-ESA network.

4.5. Results with Bicubic Degradation (BI)

It is widely used to simulate LR images with BI degradation model in image SR settings. To verify the effective-

Table 2. Quantitative results with BI degradation model. Best and second best results are **highlighted** and underlined.

Method	Scale	Set5	Set14	BSD100	Urban100	Manga109
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Bicubic	×2	33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403	30.80/0.9399
SRCNN [4]	×2	36.66/0.9542	32.45/0.9067	31.36/0.8879	29.50/0.8946	35.60/0.9663
FSRCNN [5]	×2	37.05/0.9560	32.66/0.9090	31.53/0.8920	29.88/0.9020	36.67/0.9710
VDSR [13]	×2	37.53/0.9590	33.05/0.9130	31.90/0.8960	30.77/0.9140	37.22/0.9750
LapSRN [15]	×2	37.52/0.9591	33.08/0.9130	31.08/0.8950	30.41/0.9101	37.27/0.9740
MemNet [25]	×2	37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195	37.72/0.9740
EDSR [18]	×2	38.11/0.9602	33.92/0.9195	32.32/0.9013	32.93/0.9351	39.10/0.9773
SRMD [36]	×2	37.79/0.9601	33.32/0.9159	32.05/0.8985	31.33/0.9204	38.07/0.9761
NLRN [19]	×2	38.00/0.9603	33.46/0.9159	32.19/0.8992	31.81/0.9246	–/–
DBPN [6]	×2	38.09/0.9600	33.85/0.9190	32.27/0.9000	32.55/0.9324	38.89/0.9775
RDN [40]	×2	38.24/0.9614	34.01/0.9212	32.34/0.9017	32.89/0.9353	39.18/0.9780
RCAN [38]	×2	38.27/0.9614	34.12/0.9216	32.41/0.9027	**33.34**/0.9384	**39.44**/0.9786
SAN [3]	×2	**38.31/0.9620**	34.07/0.9213	**32.42/0.9028**	33.10/0.9370	39.32/**0.9792**
RFANet (Ours)	×2	38.26/0.9615	**34.16/0.9220**	32.41/0.9026	33.33/**0.9389**	**39.44**/0.9783
Bicubic	×3	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349	26.95/0.8556
SRCNN [4]	×3	32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989	30.48/0.9117
FSRCNN [5]	×3	33.18/0.9140	29.37/0.8240	28.53/0.7910	26.43/0.8080	31.10/0.9210
VDSR [13]	×3	33.67/0.9210	29.78/0.8320	28.83/0.7990	27.14/0.8290	32.01/0.9340
LapSRN [15]	×3	33.82/0.9227	29.87/0.8320	28.82/0.7980	27.07/0.8280	32.21/0.9350
MemNet [25]	×3	34.09/0.9248	30.01/0.8350	28.96/0.8001	27.56/0.8376	32.51/0.9369
EDSR [18]	×3	34.65/0.9280	30.52/0.8462	29.25/0.8093	28.80/0.8653	34.17/0.9476
SRMD [36]	×3	34.12/0.9254	30.04/0.8382	28.97/0.8025	27.57/0.8398	33.00/0.9403
NLRN [19]	×3	34.27/0.9266	30.16/0.8374	29.06/0.8026	27.93/0.8453	–/–
RDN [40]	×3	34.71/0.9296	30.57/0.8468	29.26/0.8093	28.80/0.8653	34.13/0.9484
RCAN [38]	×3	34.74/0.9299	30.65/0.8482	29.32/0.8111	29.09/0.8702	34.44/0.9499
SAN [3]	×3	34.75/**0.9300**	30.59/0.8476	29.33/0.8112	28.93/0.8671	34.30/0.9494
RFANet (Ours)	×3	**34.79/0.9300**	**30.67/0.8487**	**29.34/0.8115**	**29.15/0.8720**	**34.59/0.9506**
Bicubic	×4	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866
SRCNN [4]	×4	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221	27.58/0.8555
FSRCNN [5]	×4	30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280	27.90/0.8610
VDSR [13]	×4	31.35/0.8830	28.02/0.7680	27.29/0.7260	25.18/0.7540	28.83/0.8870
LapSRN [15]	×4	31.54/0.8850	28.19/0.7720	27.32/0.7270	25.21/0.7560	29.09/0.8900
MemNet [25]	×4	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630	29.42/0.8942
EDSR [18]	×4	32.46/0.8968	28.80/0.7876	27.71/0.7420	26.64/0.8033	31.02/0.9148
SRMD [36]	×4	31.96/0.8925	28.35/0.7787	27.49/0.7337	25.68/0.7731	30.09/0.9024
NLRN [19]	×4	31.92/0.8916	28.36/0.7745	27.48/0.7346	25.79/0.7729	–/–
DBPN [6]	×4	32.47/0.8980	28.82/0.7860	27.72/0.7400	26.38/0.7946	30.91/0.9137
RDN [40]	×4	32.47/0.8990	28.81/0.7871	27.72/0.7419	26.61/0.8028	31.00/0.9151
RCAN [38]	×4	32.63/0.9002	28.87/0.7889	27.77/0.7436	26.82/0.8087	31.22/0.9173
SAN [3]	×4	32.64/0.9003	**28.92**/0.7888	27.78/0.7436	26.79/0.8068	31.18/0.9169
RFANet (Ours)	×4	**32.66/0.9004**	28.88/**0.7894**	**27.79/0.7442**	**26.92/0.8112**	**31.41/0.9187**

Table 3. Quantitative results with BD degradation model. Best and second best results are **highlighted** and underlined.

Method	Scale	Set5	Set14	BSD100	Urban100	Manga109
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Bicubic	×3	28.78/0.8308	26.38/0.7271	26.33/0.6918	23.52/0.6862	25.46/0.8149
SPMSR [22]	×3	32.21/0.9001	28.89/0.8105	28.13/0.7740	25.84/0.7856	29.64/0.9003
SRCNN [4]	×3	32.05/0.8944	28.80/0.8074	28.13/0.7736	25.70/0.7770	29.47/0.8924
FSRCNN [5]	×3	26.23/0.8124	24.44/0.7106	24.86/0.6832	22.04/0.6745	23.04/0.7927
VDSR [13]	×3	33.25/0.9150	29.46/0.8244	28.57/0.7893	26.61/0.8136	31.06/0.9234
IRCNN [35]	×3	33.38/0.9182	29.63/0.8281	28.65/0.7922	26.77/0.8154	31.15/0.9245
SRMD [36]	×3	34.01/0.9242	30.11/0.8364	28.98/0.8009	27.50/0.8370	32.97/0.9391
RDN [40]	×3	34.58/0.9280	30.53/0.8447	29.23/0.8079	28.46/0.8582	33.97/0.9465
SRFBN [17]	×3	34.66/0.9283	30.48/0.8439	29.21/0.8069	28.48/0.8581	34.07/0.9466
RCAN [38]	×3	34.70/0.9288	30.63/0.8462	29.32/0.8093	28.81/0.8647	34.38/0.9483
SAN [3]	×3	34.75/0.9290	**30.68**/0.8466	29.33/0.8101	28.83/0.8646	34.46/0.9487
RFANet (Ours)	×3	**34.77/0.9292**	**30.68/0.8473**	**29.34/0.8104**	**28.89/0.8661**	**34.49/0.9492**

ness of our RFANet, we compare RFANet with 12 state-of-the-art image SR methods: SRCNN [4], FSRCNN [5], VDSR [13], LapSRN [15], MemNet [25], EDSR [18], SRMD [36], NLRN [19], DBPN [6], RDN [40], RCAN [38] and SAN [3]. Table 2 shows all the quantitative results with BI model. In general, our RFANet can achieve comparable or superior results compared with all the other methods including the extremely competitive RCAN and SAN. Most quantitative results of RFANet are either the best or the second best. For scale ×2, RFANet achieves the best reuslts on Set14, the best SSIM on Urban100 and the highest PSNR on Manga109. For scale ×3, RFANet outperforms the other methods on all the datasets. Our RFANet also has excellent performance with scale ×4, the best results are achieved on Set5, B100, Urban100 and Manga109, respectively. Compared with other methods, we found that

Figure 6. Visual comparisons for ×4 SR with BD degradation model.

our RFANet behaves particularly well on Urban100 and Manga109 datasets. This is mainly because both datasets contain rich structured contents and our RFANet can gradually aggregate these hierarchical information to form more representative features. This property can be further verified from the SSIM scores of our RFANet. The SSIM score is focused on the visible structures in the image. For example, on Urban100 (×2) dataset, our PSNR is the second best but we achieve the best SSIM, which indicates our RFANet can recover better visible structures. Similar phenomena can also be found on Set14 (×4) dataset. The visual comparisons of Fig. 5 can also prove that our RFANet reconstructs better structural details.

4.6. Results with Blur-downscale Degradation (BD)

Following [36, 38, 3], we also provide the results with blur-downscale degradation (BD) model. We compare our RFANet with 10 state-of-the-art methods: SPMSR [22], SRCNN [4], FSRCNN [5], VDSR [13], IRCNN [35], SRMD [36], RDN [40], SRFBN [17], RCAN [38], and SAN [3]. As shown in Table 3, our RFANet outperforms other methods on all the datasets. Specifically, we achieve 0.06dB PSNR gain over SAN on Urban100 dataset. Compared with SAN, the PSNR gain on Set14 dataset is marginal but we can still achieve considerable improvement in terms of SSIM. The consistently better results of RFANet indicate that our method can adapt well to scenarios with multiple degradation models. Fig. 4.4 shows the visual superiority of our method.

4.7. Effects of Residual Feature Aggregation (RFA)

We now illustrate how our residual feature aggregation design affects the output features in different stages of the network. Inspired by [11], we adopt the weight norm as an approximate for the dependency of a convolutional layer

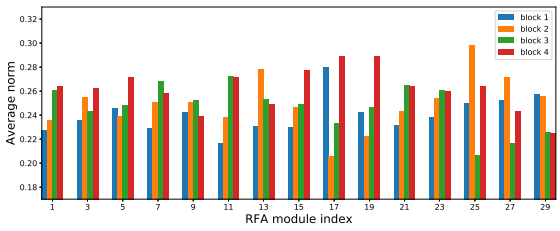


Figure 7. Average norms of filter weights. Each set of histograms corresponds to one RFA module. There are four blocks inside a RFA module. The histogram represents the norm of filter weights in the aggregation convolutional layer w.r.t. the feature map of each block.

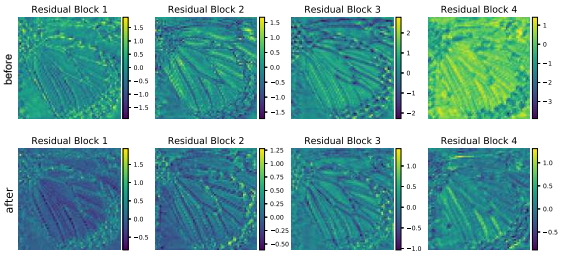


Figure 8. Average feature maps of residual blocks.

on its preceding layers. The weight norm is calculated by the corresponding weights from all filters w.r.t. each residual feature map in the aggregation 1×1 convolutional layer (see Fig. 3). In general, the larger the norm is, the stronger dependency it has on this particular feature map. For clarity, we choose to visualize every two modules in a total of 30 RFA modules. Fig. 7 presents the norm of the filter weights *vs*. feature map index. The legend of Fig.7 shows the index of residual blocks in each RFA module. Several observations can be made from the plot: (1) The aggregation layers spread their weights over all the residual blocks which indicates that all the residual features are directly used to produce the output features of the RFA module. (2) The variance of weight norms in latter modules are larger than that of the previous modules. This indicates that the network gradually learns to distinguish the residual features and assign more weights to the features of critical importance. (3) At the beginning, the last block contributes most than the other three blocks. With the depth increases, the other three blocks also play an important role in feature learning, indicating the necessity of residual feature aggregation.

4.8. Effects of Enhanced Spatial Attention

Fig. 8 visualizes the average feature maps of residual blocks within a RFA module. The top row is the feature maps before attention mechanism and the bottom row is the feature maps after attention mechanism. We can get some

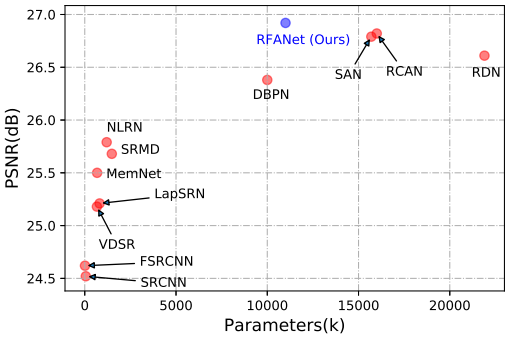


Figure 9. PSNR *vs*. Parameters on Urban100 ($\times 4$).

intuitive clues from this visualization: (1) The attention mechanism has the effect of modulating the activation values. We can see that the activation ranges of the bottom row are smaller than the top row, which can ease the training difficulty to some extent (*e.g*. residual scaling in EDSR [18]). (2) Feature maps after the attention mechanism tend to contain more negative values, showing a stronger effect of suppressing the smooth area of the input image, which further leads to a more accurate residual image.

4.9. Model Complexity Analysis

Fig. 9 shows the comparisons about model size and performance with 11 stae-of-the-art SR methods:SRCNN [4], FSRCNN [5], VDSR [13], LapSRN [15], MemNet [25], NLRN [19], SRMD [36], DBPN [6], RDN [40], RCAN [38] and SAN [3]. Our RFANet has much fewer parameters than RDN, RCAN and SAN, but obtains better performance, which verifies the effectiveness of our method. Compared with DBPN, our RFANet achieves much higher PSNR with a slightly larger model, indicating that we have a good trade-off between performance and model complexity.

5. Conclusions

In this paper, we propose a general residual feature aggregation (RFA) framework for image SR. The RFA framework effectively groups the residual blocks together, where the features of local residual blocks are sent directly to the end of the RFA framework for fully utilizing these useful hierarchical features. To maximize the power of the proposed RFA framework, we further design an enhanced spatial attention (ESA) block to make the residual features to be more focused on spatial contents of key importance. To compare with state-of-the-art methods, we propose the RFANet by applying the RFA framework in conjunction with the ESA block. Comprehensive benchmark evaluations with BI and BD degradation models well demonstrate the effectiveness of our RFANet in terms of both quantitative and visual results.

References

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

[2] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. SCA-CNN: spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*, pages 6298–6306. IEEE Computer Society, 2017.

[3] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVPR*, pages 11065–11074. Computer Vision Foundation / IEEE, 2019.

[4] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV (4)*, volume 8692 of *Lecture Notes in Computer Science*, pages 184–199. Springer, 2014.

[5] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV (2)*, volume 9906 of *Lecture Notes in Computer Science*, pages 391–407. Springer, 2016.

[6] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *CVPR*, pages 1664–1673. IEEE Computer Society, 2018.

[7] Jingwen He, Chao Dong, and Yu Qiao. Modulating image restoration with continual levels via adaptive feature modification layers. In *CVPR*, pages 11056–11064. Computer Vision Foundation / IEEE, 2019.

[8] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141. IEEE Computer Society, 2018.

[9] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *CVPR*, pages 1575–1584. Computer Vision Foundation / IEEE, 2019.

[10] Yanting Hu, Jie Li, Yuanfei Huang, and Xinbo Gao. Channel-wise and spatial feature modulation network for single image super-resolution. *CoRR*, abs/1809.11130, 2018.

[11] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 2261–2269. IEEE Computer Society, 2017.

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

[13] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654. IEEE Computer Society, 2016.

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

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

[16] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 105–114. IEEE Computer Society, 2017.

[17] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *CVPR*, pages 3867–3876. Computer Vision Foundation / IEEE, 2019.

[18] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPR Workshops*, pages 1132–1140. IEEE Computer Society, 2017.

[19] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S. Huang. Non-local recurrent network for image restoration. In *NeurIPS*, pages 1680–1689, 2018.

[20] David R. Martin, Charless C. Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–425, 2001.

[21] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools Appl.*, 76(20):21811–21838, 2017.

[22] Tomer Peleg and Michael Elad. A statistical prediction model based on sparse representations for single image super-resolution. *IEEE Trans. Image Processing*, 23(6):2569–2582, 2014.

[23] 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 *CVPR*, pages 1874–1883. IEEE Computer Society, 2016.

[24] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *CVPR*, pages 2790–2798. IEEE Computer Society, 2017.

[25] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *ICCV*, pages 4549–4557. IEEE Computer Society, 2017.

[26] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, et al. NTIRE 2017 challenge on single image super-resolution: Methods and results. In *CVPR Workshops*, pages 1110–1121. IEEE Computer Society, 2017.

[27] Radu Timofte, Vincent De Smet, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *ICCV*, pages 1920–1927. IEEE Computer Society, 2013.

[28] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: adjusted anchored neighborhood regression for fast super-resolution. In *ACCV (4)*, volume 9006 of *Lecture Notes in Computer Science*, pages 111–126. Springer, 2014.

[29] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang.

Residual attention network for image classification. In *CVPR*, pages 6450–6458. IEEE Computer Society, 2017.

[30] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803. IEEE Computer Society, 2018.

[31] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. ESRGAN: enhanced super-resolution generative adversarial networks. In *ECCV Workshops (5)*, volume 11133 of *Lecture Notes in Computer Science*, pages 63–79. Springer, 2018.

[32] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057. JMLR.org, 2015.

[33] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, volume 6920 of *Lecture Notes in Computer Science*, pages 711–730. Springer, 2010.

[34] Kaibing Zhang, Xinbo Gao, Dacheng Tao, and Xuelong Li. Single image super-resolution with non-local means and steering kernel regression. *IEEE Trans. Image Processing*, 21(11):4544–4556, 2012.

[35] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep CNN denoiser prior for image restoration. In *CVPR*, pages 2808–2817. IEEE Computer Society, 2017.

[36] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *CVPR*, pages 3262–3271. IEEE Computer Society, 2018.

[37] Lei Zhang and Xiaolin Wu. An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE Trans. Image Processing*, 15(8):2226–2238, 2006.

[38] 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 (7)*, volume 11211 of *Lecture Notes in Computer Science*, pages 294–310. Springer, 2018.

[39] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. In *ICLR (Poster)*. OpenReview.net, 2019.

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