

# RDRN: Recursively Defined Residual Network for Image Super-Resolution\*

Alexander Panaetov<sup>1[0000-0003-2309-9798]</sup>, Karim Elhadji Daou<sup>1[0000-0003-4677-7571]</sup>, Igor Samenko<sup>1[0000-0001-9400-312X]</sup>, Evgeny Tetin<sup>1[0000-0001-6878-8330]</sup>, and Ilya Ivanov<sup>1[0000-0001-7919-5143]</sup>

Huawei, Moscow Research Center, Russia {panaetov.alexander1, karim.daou, samenko.igor, evgeny.tetin, ivanov.ilya1}@huawei.com

**Abstract.** Deep convolutional neural networks (CNNs) have obtained remarkable performance in single image super-resolution (SISR). However, very deep networks can suffer from training difficulty and hardly achieve further performance gain. There are two main trends to solve that problem: improving the network architecture for better propagation of features through large number of layers and designing an attention mechanism for selecting most informative features. Recent SISR solutions propose advanced attention and self-attention mechanisms. However, constructing a network to use an attention block in the most efficient way is a challenging problem. To address this issue, we propose a general recursively defined residual block (RDRB) for better feature extraction and propagation through network layers. Based on RDRB we designed recursively defined residual network (RDRN), a novel network architecture which utilizes attention blocks efficiently. Extensive experiments show that the proposed model achieves state-of-the-art results on several popular super-resolution benchmarks and outperforms previous methods by **up to 0.43 dB**.

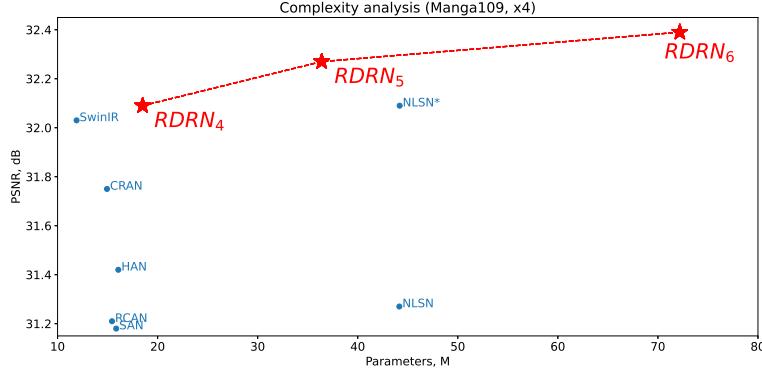
**Keywords:** Super-Resolution, Recursively Defined Residual Network, Recursively Defined Residual Block

## 1 Introduction

The main purpose of super-resolution (SR) is to reconstruct high-resolution image (HR) from given low-resolution counterpart (LR). SR is an ill-posed problem since the mapping between LR and HR images is ambiguous (one-to-many). Recovering missing details is a challenging task, especially for a high upscale factor. Despite being a difficult problem, SR plays an important role in various image processing tasks with applications in face recognition, medical imaging, surveillance, digital zoom, etc. While many existing SR methods reconstruct HR image from several LR images, in this paper we focus on single image super-resolution (SISR).

---

\* Supported by Huawei.

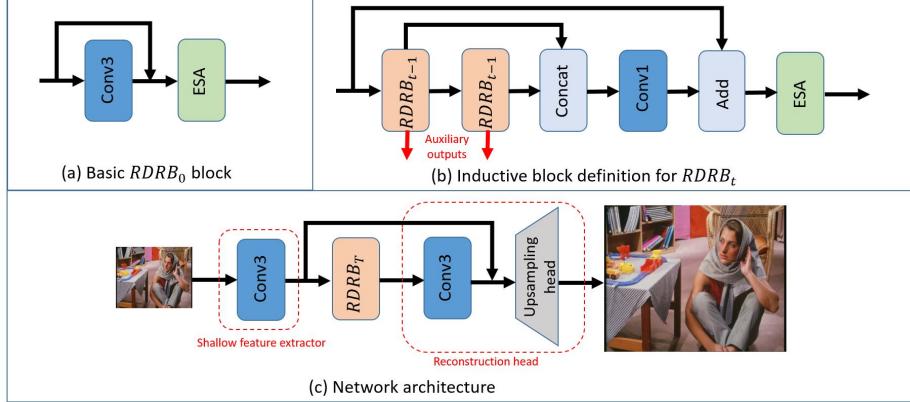


**Fig. 1.** Number of parameters and performance on Manga109 with upscale factor  $\times 4$  (BI model)

In recent years, convolutional neural networks (CNNs) have achieved remarkable results in many computer vision tasks, including SISR. Deep CNNs have shown improvement over the traditional algorithms. Network depth in existing solutions has been significantly increased from three layers in SRCNN [5] to more than 400 in recent works [35, 25]. However, very deep networks can suffer from training difficulties and hardly achieve any extra performance gain. A further increase in CNN depth does not lead to an improvement in quality and makes them unsuitable for various applications. The difficulty of training can be explained by the fact that network is not able to efficiently use information from intermediate layers. This issue can be partially solved using residual learning [8]. Combining features from different layers through skip connections is a fruitful idea in SISR. Additional connections along the network's depth could help to learn more powerful feature representations, making training more stable and accelerating convergence.

Another approach to address the training difficulty is related to the mechanism of attention. Recently, this direction has become very popular and profitable for SISR. The intuition behind attention is a simulation of the human vision system, which can focus on the most informative parts of an image and ignore the irrelevant information. Recent works show that attention can effectively reduce the width and depth of a network while maintaining comparable or better performance due to enhanced discriminative learning ability [35].

In this paper, we combine both approaches. The design of recursively defined residual block (RDRB) is shown in Figure 2. It consists of two parts: basic block (Fig. 2-a) and recursive block (Fig. 2-b). We have found that enhanced spatial attention (ESA) introduced in [20] is very effective for the super-resolution task, and we take advantage of its benefits even more than in the original paper. We include ESA in the basic block, which is repeated in the final architecture mul-



**Fig. 2.** Architecture of the proposed recursively defined residual block (RDRB) and network (RDRN). RDRB is defined in a recurrent manner. (a) We define basic block  $RDRB_0$  as a convolution layer followed by ESA block from [20]. (b) Each subsequent block  $RDRB_t$  can be defined using the architecture of previous block  $RDRB_{t-1}$  according to the scheme. (c) Full network architecture

tiple times. Compared to previous work, our RDRB contains more connections between intermediate layers. It combines hierarchical cues along the network depth to obtain richer feature representations. Experiments show that the effect of the proposed RDRB is more visible for lower upscale factors ( $\times 2, \times 3$ ). For the upscale factor  $\times 2$ , the RDRB-based model outperforms all recent solutions without any bells and whistles. It can be explained by the recurrent nature of RDRB. Shallow features are propagated to all levels of RDRB via long skip connections from input. For lower upscale factors, they contain more relevant information, as fewer details will be missed compared to higher upscale factors. Finally, to further improve RDRB, especially for large upscale factors, we insert non-local sparse attention [24] into the block.

Based on RDRB we design recursively defined residual network (RDRN) as shown in Figure 2-c. Following [19], we add batch normalization (BN) and apply adaptive deviation modulator (AdaDM) to the final model. For training, we use intermediate supervision (IS) loss, which improves convergence and allows to simultaneously train several models of different computational complexity without a large overhead.

The proposed RDRN shows superior performance on the most popular SR benchmarks. Our model produces better visual quality, recovers more details and outperforms current state-of-the-art solutions with a significant margin of up to 0.43 dB.

In summary, the main contributions of our paper are:

- We propose a novel recursive scheme for block architecture definition. Using that scheme we build a general recursively defined residual block (RDRB) for more accurate image SR.
- Based on RDRB we design a novel network architecture (RDRN). Extensive experiments on public datasets demonstrate that the proposed model outperforms current state-of-the-art SR methods.
- We introduce intermediate supervision loss. Training with IS helps to obtain additional performance gain and allows to simultaneously train several models of different computational complexity.

## 2 Related Work

Super-resolution algorithms can be categorized into two types: traditional and deep learning based methods. In this section, we will focus on the second category as the most successful in computer vision.

Dong et al.[5] proposed the first three-layer convolutional neural network (SRCNN) to learn the mapping from LR to HR directly. This pioneering work achieved superior performance against the previous traditional methods.

Following this work, many networks achieved better performance using deeper and wider architectures [12, 13]. SRCNN used interpolated image as input, however, it is more efficient to upscale the feature maps at the end of the network. To address this issue, Shi et al. proposed ESPCN [27] with a sub-pixel layer, which is widely used in modern SISR networks. As in the case of the classification task, a further increase in depth and width of plain architecture leads to quality degradation. However, residual blocks [8] and dense connections [9] allow to train more powerful networks, and Lin et al. proposed a very deep and wide EDSR [17] based on a modified residual block. Dense connections were used in RDN [36] to utilize hierarchical features from all convolutional layers.

Following [17], most of the recent SISR works do not use batch normalization (BN) [11], as it harms network’s performance. However, Liu et al. [19] showed that normalization layers reduce the standard deviation of feature pixels (the main reason for quality degradation) and proposed adaptive deviation modulator (AdaDM) to solve that issue. AdaDM can successfully enable BN layers, significantly improving performance and allowing to train larger models.

Attention mechanism is used as a simulation of the human vision system that focuses on the most informative parts of an image, ignores irrelevant information and enhances discriminative learning ability. Recently, attention has been successfully applied to the SISR problem, and attention-based methods have shown superiority over pure CNN solutions. Zhang et al. [35] proposed channel attention (CA) to adaptively rescale each feature channel-wise by modeling the interdependencies across feature channels. Such CA mechanism improves the representational ability of residual channel attention network (RCAN).

Liu et al. [20] designed lightweight and powerful enhanced spatial attention (ESA). Dai et al. [3] introduced second-order attention network (SAN) to adaptively refine features using second-order feature statistics. Niu et al.[25] presented

a new holistic attention network (HAN), which consists of a layer attention module (LAM) and a channel-spatial attention module (CSAM) and models the holistic interdependencies among layers, channels, and positions. Zhang et. al. [37] proposed a context reasoning attention network (CRAN) that can adaptively modulate the convolution kernel according to the global context enhanced by semantic reasoning.

Attention could be considered as an additional tool to improve the expressiveness and convergence of CNNs. An alternative is the self-attention mechanism [30], which was introduced for natural language processing models. Recent work has shown that even pure transformers can achieve SOTA results on computer vision tasks [7]. However, hybrid models that contain both convolution and self-attention show outstanding results as they integrate the advantages of both approaches. Following this new trend, Liang et al [16] introduced the SwinIR architecture based on Swin Transformer [21]. SwinIR has improved upon all previous methods by a significant margin.

### 3 Proposed Method

In this section, we first present the overview of RDRN. Then we give a detailed description of the proposed RDRB. We proceed with outlining the technical part of training, including the loss function and implementation details. Finally, we provide arguments about the advantages of the proposed method and differences from other approaches.

#### 3.1 Network Architecture

As shown in Figure 2, RDRN can be divided into three parts: shallow feature extractor, RDRB deep feature extractor and reconstruction head. Let us denote  $I_{LR}$  and  $I_{SR}$  the low resolution input and the output of RDRN. Following [35, 3, 25], we use only one convolutional layer to extract the shallow features  $F_{SF}$  from LR input:

$$F_{SF} = H_{SF}(I_{LR}), \quad (1)$$

where  $H_{SF}$  denotes the convolution operation. The extracted shallow feature map  $F_{SF}$  is used as input to the deep feature extractor:

$$F_{DF}^T = H_{RDRB}^T(F_{SF}), \quad (2)$$

where  $T$  is the recursion depth and  $H_{RDRB}^T$  stands for RDRB feature extraction model, which will be introduced in the next subsection. RDRB is the core of the proposed network. Finally, the extracted deep features  $F_{DF}$  are combined with the shallow features to stabilize training, after which they are processed by the reconstruction module:

$$I_{SR}^T = H_{rec}(F_{SF}, F_{DF}^T), \quad (3)$$

where  $H_{rec}$  denotes the reconstruction head that consists of a convolutional layer and a sub-pixel layer [27]. The long skip connection propagates low-frequency information directly to the reconstruction module, which can help the deep feature extractor to focus on the extraction of high-frequency information [16].

There are several choices for loss function to optimize the model, such as  $\ell_1$ ,  $\ell_2$ , perceptual, adversarial loss. We found that for the proposed method,  $\ell_1$  loss is the most suitable one, and we minimize the following loss function:

$$L^T(\Theta) = \frac{1}{m} \sum_{i=1}^m \|H_{RDRN}^T(I_{LR}^i) - I_{HR}^i\|_1 = \frac{1}{m} \sum_{i=1}^m \|I_{SR}^{T,i} - I_{HR}^i\|_1, \quad (4)$$

where  $H_{RDRN}^T$ ,  $\Theta$ , and  $m$  denote the function of the proposed RDRN, the set of learned parameters, and the number of training pairs, respectively. Following [18] we fine-tune the model using  $\ell_2$  loss. More details are provided in the experiments section.

### 3.2 Recursively Defined Residual Block (RDRB)

As discussed above, recursively defined residual block is the core of RDRN, and here we give the overall description of the proposed block architecture. The definition will be done in a recursive manner. First, we define basic block  $RDRB_0$  as a convolutional layer followed by ESA block introduced in [20]:

$$F_{DF}^0 = H_{RDRB}^0(F_{SF}) := ESA(conv_{3 \times 3}(F_{SF}) + F_{SF}). \quad (5)$$

We found that ESA mechanism is highly effective for the super-resolution task, and we take advantage of its benefits even more than in the original paper. We include ESA in the basic block, which is repeated in our architecture multiple times.

Finally, we use induction to define  $RDRB_t$  for any natural  $t$ :

$$F_{DF}^t = H_{RDRB}^t(F_{SF}), \quad (6)$$

$$F_{DF}^{t,1} = H_{RDRB}^t(F_{DF}^t), \quad (7)$$

$$H_{RDRB}^t(F_{SF}) := ESA\left(F_{SF} + conv_{1 \times 1}\left(concat(F_{DF}^{t-1}, F_{DF}^{t-1,1})\right)\right). \quad (8)$$

The scheme of building  $RDRB_0$  and  $RDRB_t$  is depicted in Figure 2. Same as in the basic block, ESA is included in  $RDRB_t$ . It's worth noticing that we don't share weights in any part of the model. To avoid any misunderstanding about the definition of the proposed RDRB, we provide PyTorch implementation of the basic block and the recursive step in the supplementary.

### 3.3 Intermediate Supervision (IS)

As shown in Figure 2, for  $t > 0$  each  $\text{RDRB}_t$  contains two additional auxiliary outputs. Each of these intermediate outputs is paired with a reconstruction head and an additional loss. The final loss function is a weighted sum of the original loss and all intermediate losses (due to recursive definition of the block we have  $2^{T+1} - 2$  additional loss terms):

$$L(\Theta) = w_0 L^T(\Theta) + \sum_{i=1}^{2^{T+1}-2} w_i L_i(\Theta), \quad (9)$$

where  $w_i$  denotes the weight and  $L_i(\Theta)$  is the loss based on the intermediate output. The computational overhead for IS training is minimal, because the size and complexity of the added heads is much smaller than the size and complexity of RDRB. Using IS loss function gives two advantages. First, it allows to simultaneously train several models of different computational complexity using the output of intermediate reconstruction head. Second, as proved in the ablation study, training with IS enables us to achieve a performance gain.

### 3.4 Implementation Details

Here we specify the implementation details of RDRN. We use  $\text{RDRB}_5$  in our final architecture. Following [19], we add batch normalization (BN) [11] and adaptive deviation modulator (AdaDM) to every  $3 \times 3$  convolution. Finally, starting from recursion level 3 we add non-linear spatial attention block [24] after ESA. According to our experiments, both tricks improve the final score and allow us to outperform SwinIR [16].

Experiments show that for IS training it is better to zero out the weights for losses based on auxiliary outputs from  $\text{RDRB}_1$  and  $\text{RDRB}_2$  and set the remaining weights to one.

### 3.5 Discussion

In our research we aim to follow the best practices from recent work. The motivation is two-fold: to outperform current SoTA models and to design general method for architecture definitions which can be used by other researchers. We conducted a large number of experiments and devised the following recipe:

- Following [18], we use advanced two-stage training procedure.
- We apply batch normalization [11] and add AdaDM to the model as in [19].
- We have an attention mechanism both inside the basic block and between blocks.

We explored deformable convolution [38] for image SR. In all experiments it improves the final score but doubles the training time, and we decided to postpone this direction for future research.

The main novelty of our work is the proposed architecture of RDRB and RDRN. We present the concept of building a network that utilizes multiple connections between different blocks. It is not limited to one specific architecture: the basic block can be replaced with any other block and the merging of two blocks can vary as well. The proposed architecture significantly differs from the ones presented in prior works.

**Difference from Residual Feature Aggregation Network (RFANet).**

RFANet [20] utilizes skip-connection only on block level while blocks are connected consequently. The intuition behind RDRN is that combining of hierarchical cues along the network depth allows to get richer feature representations. Compared to RFANet, our architecture is able to combine features at deeper layers. In ablation study we have comparison of residual-block (RB) from RFANet and RDRB<sub>0</sub> under the same conditions. We also show in Table 1 the benefits of our recursive way of building the network compared to plain connection of blocks in RFANet.

**Difference from Non-Local Sparse Network (NLSN).** NLSN [24] is a recent state-of-the-art SR method. The model achieved remarkable results after it was updated with BN and AdaDM [19]. The original architecture of NLSN is plain and made of residual blocks and non-local sparse attention (NLSA) blocks. The main difference is again coming from more effective way of merging information from different layers. Our smaller RDRN<sub>4</sub> model has similar performance with NLSN and 72% less FLOPs (Table 6).

**Difference from Deep Recursive Residual Network (DRRN).** The architecture of the recursive block (RB) from [28] is defined by a recursive scheme, similarly to the proposed RDRB. The key differences between the two approaches are as follows. First, according to the definition of RB, the weights of the same sub-blocks are shared, while RDRB does not reuse weights. However, weights sharing can be an effective way to reduce the number of parameters, and we save this direction for future research. Second, DRRN contains sequences of RB blocks. In contrast, our model is based on one large RDRB. Our recursive definition helps to stack blocks together with additional skip connections, granting extra performance gain.

## 4 Experiments

In this section, we compare RDRN to the state-of-the-art algorithms on five benchmark datasets. We first give a detailed description of experiment settings. Then we analyze the contribution of the proposed neural network. Finally, we provide a quantitative and visual comparison with the recent state-of-the-art methods for the most popular degradation models.

### 4.1 Implementation and trainings details

**Datasets and metrics.** We train our models on DF2K dataset, which combines DIV2K [29] and Flickr2K together as in [37, 16, 19]. For testing, we choose five

**Table 1.** Ablation study on the advantage of RDRB

Model / PSNR	Set5	Set14	B100	Urban100	Manga100
RFANet(RB)	32.65	29.00	27.86	27.11	31.73
RFANet(RDRB <sub>0</sub> )	32.67 +0.02	29.02 +0.02	27.86 0.00	27.16 +0.05	31.81 +0.08
RDRN(RB)	32.73 +0.08	29.02 +0.02	27.86 0.00	27.14 +0.03	31.80 +0.07

standard datasets: Set5 [2], Set14 [32], B100 [22], Urban100 [10], and Manga109 [23]. We train and test RDRN for three degradation methods. Degraded data is obtained by bicubic interpolation (BI), blur-downscale (BD) and downscale-noise (DN) models from Matlab. We employ peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [31] to measure the quality of super-resolved images. All SR results are evaluated on Y channel after color space transformation from RGB to YCbCr. Metrics are calculated using Matlab.

**Training settings.** The proposed network architecture is implemented in PyTorch framework, and models are trained from scratch. In our network, patch size is set to  $64 \times 64$ . We use Adam [14] optimizer with a batch size of 48 (12 per each GPU). The initial learning rate is set to  $10^{-4}$ . After  $7.5 \times 10^5$  iterations it is reduced by 2 times. Default values of  $\beta_1$  and  $\beta_2$  are used, which are 0.9 and 0.999, respectively, and we set  $\epsilon = 10^{-8}$ . During training we augment the images by randomly rotating  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$  and horizontal flipping.

For all the results reported in the paper, we train the network  $9 \times 10^5$  iterations with  $\ell_1$  loss function. After that we fine-tune the model  $1.5 \times 10^4$  iterations with a smaller learning rate of  $10^{-5}$  using MSE loss. Complete training procedure takes about two weeks on a server with 4 NVIDIA Tesla V100 GPUs.

## 4.2 Ablation study

**Impact of the architecture.** To highlight the advantages of the proposed architecture, we compare RDRN and RFANet [20]. For comparison we use vanilla RFANet with 32 RFA blocks and 64 channels. Each RFA block consists of 4 residual blocks (RB). First, we demonstrate that adding ESA to our basic block RDRB<sub>0</sub> is beneficial. For that purpose we train RFANet(RDRB<sub>0</sub>), changing RB to RDRB<sub>0</sub>. As shown in Table 1, our basic block gives a performance gain to RFANet. Second, we show that the proposed recurrent scheme of building a network is better than stacking RFA blocks. We train RDRN(RB) using RB instead of RDRB<sub>0</sub> to demonstrate that. Even when using the same basic block, RDRN(RB) still outperforms RFANet(RFA, RB). For fair comparison, we change the depth of both networks to keep the computation complexity for all models similar.

**Impact of Intermediate Supervision (IS).** To show the impact of using IS during training, we train the same network with and without IS. Table 2 demonstrates the effectiveness of the proposed loss function. Experiments show

**Table 2.** Ablation study on the advantage of IS

Model / PSNR	Set5	Set14	B100	Urban100	Manga100
<i>RDRN</i> <sub>4</sub>	32.68	29.00	27.84	27.07	31.77
<i>RDRN</i> <sub>4</sub> -IS	32.71	29.01	27.85	27.07	31.83
	+0.03	+0.01	+0.01	0.00	+0.06
<i>RDRN</i> <sub>5</sub>	32.67	28.99	27.86	27.09	31.86
<i>RDRN</i> <sub>5</sub> -IS	32.73	29.05	27.88	27.19	31.97
	+0.06	+0.06	+0.02	+0.10	+0.11

that IS provides a bigger gain for larger models. It corresponds with the intuition that it is harder to train a larger model using SGD, and IS helps to propagate gradients better. Using IS allows to simultaneously train several SISR models of different computational complexity.

### 4.3 Results with Bicubic (BI) Degradation Model

We compare the proposed algorithm with the following 11 state-of-the-art methods: SRCNN [4], RDN [36], RCAN [35], SAN [3], NLSN [24], DRLN [1], HAN [25], RFA-Net [20], CRAN [37], SwinIR [16] and NLSN\* [19]. We take all results from the original papers. Following [17, 3, 35, 25, 16], we provide a self-ensemble model and denote it RDRN+.

**Quantitative results.** Table 3 reports the quantitative comparison of  $\times 2$ ,  $\times 3$ ,  $\times 4$  SR. Compared to the existing methods, RDRN scores best in all scales of the reconstructed test sets. Even without self-ensemble, our model outperforms other solutions at upscale factors  $2\times$  and  $3\times$ . For  $4\times$  SR, RDRN achieves the best PSNR values, however, SSIM score of competitors on several datasets is higher. The effect of the proposed method is more significant for lower upscale factors. This can be explained by the RDRB design. Shallow features are propagated for all levels of RDRB using long skip connections from input. For lower upscale factors, shallow features contain more important information as less information will be missed compared to higher upscale factors.

**Visual results.** We give visual comparison of various competing methods on Urban100 dataset for  $\times 2$  SR in Figure 3. Our model obtains better visual quality and recovers a more detailed image. Most compared methods recover grid textures of buildings with blurring artifacts, while RDRN produces sharper images. The proposed method can maintain a regular structure in difficult cases where previous approaches fail. To further illustrate the analysis above, we show such cases for  $\times 2$  SR in Figure 4. The recovered details are more faithful to the ground truth.

### 4.4 Results with Bicubic Blur-Downscale (BD) Degradation Model

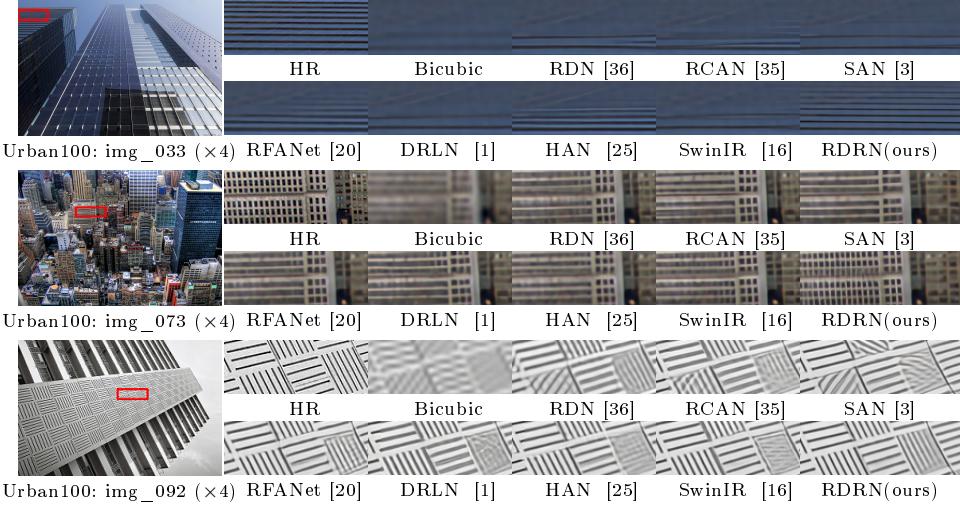
Following [36, 35, 20, 1, 25, 37], we provide results for blur-downscale degradation, where HR image is blurred by a  $7\times 7$  Gaussian kernel with standard deviation  $\sigma = 1.6$  and then downsampled by bicubic interpolation with scaling factor  $\times 3$ .

**Table 3.** Quantitative results with BI degradation model. The best and second best results are highlighted in **bold** and underlined

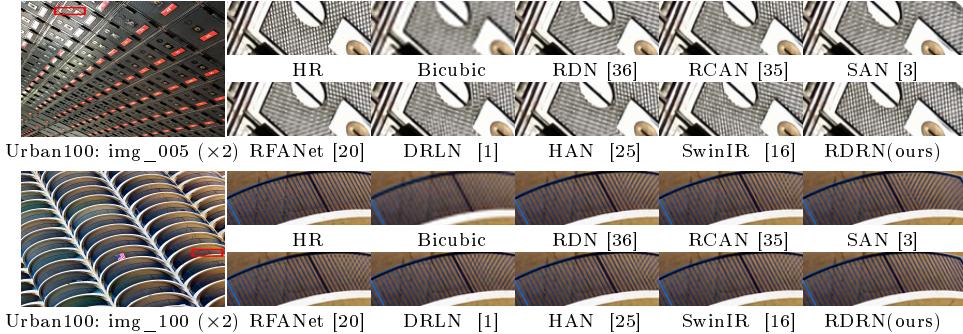
Methods	Scale	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM								
Bicubic	$\times 2$	33.66	0.9299	30.24	0.8688	29.56	0.8431	26.88	0.8403	30.80	0.9339
SRCNN [4]	$\times 2$	36.66	0.9542	32.45	0.9067	31.36	0.8879	29.50	0.8946	35.60	0.9663
RDN [36]	$\times 2$	38.24	0.9614	34.01	0.9212	32.34	0.9017	32.89	0.9353	39.18	0.9780
RCAN [35]	$\times 2$	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.34	0.9384	39.44	0.9786
SAN [3]	$\times 2$	38.31	0.9620	34.07	0.9213	32.42	0.9028	33.10	0.9370	39.32	0.9792
NLSN [24]	$\times 2$	38.34	0.9618	34.08	0.9231	32.43	0.9027	33.42	0.9394	39.59	0.9789
DRLN [1]	$\times 2$	38.27	0.9616	34.28	0.9231	32.44	0.9028	33.37	0.9390	39.58	0.9786
HAN [25]	$\times 2$	38.27	0.9614	34.16	0.9217	32.41	0.9027	33.35	0.9385	39.46	0.9785
RFANet[20]	$\times 2$	38.26	0.9615	34.16	0.9220	32.41	0.9026	33.33	0.9389	39.44	0.9783
CRAN [37]	$\times 2$	38.31	0.9617	34.22	0.9232	32.44	0.9029	33.43	0.9394	39.75	0.9793
SwinIR [16]	$\times 2$	38.42	0.9623	34.46	0.9250	32.53	0.9041	33.81	0.9427	39.92	0.9797
NLSN* [19]	$\times 2$	38.43	0.9622	34.40	0.9249	32.50	0.9036	33.78	0.9419	39.89	0.9798
RDRN(ours)	$\times 2$	38.54	0.9627	34.67	0.9261	32.53	0.9043	34.12	0.9442	40.35	0.9807
RDRN+ (ours)	$\times 2$	<b>38.59</b>	<b>0.9629</b>	<b>34.76</b>	<b>0.9265</b>	<b>32.56</b>	<b>0.9046</b>	<b>34.27</b>	<b>0.9452</b>	<b>40.48</b>	<b>0.9810</b>
Bicubic	$\times 3$	30.39	0.8682	27.55	0.7742	27.21	0.7385	24.46	0.7349	29.95	0.8556
SRCNN [4]	$\times 3$	32.75	0.9090	29.30	0.8215	28.41	0.7863	26.24	0.7989	30.48	0.9117
RDN [36]	$\times 3$	34.71	0.9296	30.57	0.8468	29.26	0.8093	28.80	0.8653	34.13	0.9484
RCAN [35]	$\times 3$	34.74	0.9299	30.65	0.8482	29.32	0.8111	29.09	0.8702	34.44	0.9499
SAN [3]	$\times 3$	34.75	0.9300	30.59	0.8476	29.33	0.8112	28.93	0.8671	34.30	0.9494
NLSN [24]	$\times 3$	34.85	0.9306	30.70	0.8485	29.34	0.8117	29.25	0.8726	34.57	0.9508
DRLN [1]	$\times 3$	34.78	0.9303	30.73	0.8488	29.36	0.8117	29.21	0.8722	34.71	0.9509
HAN [25]	$\times 3$	34.75	0.9299	30.67	0.8483	29.32	0.8110	29.10	0.8705	34.48	0.9500
RFANet[20]	$\times 3$	34.79	0.9300	30.67	0.8487	29.34	0.8115	29.15	0.8720	34.59	0.9506
CRAN [37]	$\times 3$	34.80	0.9304	30.73	0.8498	29.38	0.8124	29.33	0.8745	34.84	0.9515
SwinIR [16]	$\times 3$	34.97	0.9318	30.93	0.8534	29.46	0.8145	29.75	0.8826	35.12	0.9537
NLSN* [19]	$\times 3$	34.95	0.9316	30.86	0.8513	29.45	0.8141	29.77	0.8812	35.20	0.9534
RDRN(ours)	$\times 3$	<b>35.04</b>	<b>0.9322</b>	<b>30.99</b>	<b>0.8530</b>	<b>29.50</b>	<b>0.8152</b>	<b>29.87</b>	<b>0.8830</b>	<b>35.44</b>	<b>0.9543</b>
RDRN+ (ours)	$\times 3$	<b>35.10</b>	<b>0.9326</b>	<b>31.04</b>	<b>0.8539</b>	<b>29.53</b>	<b>0.8158</b>	<b>30.02</b>	<b>0.8848</b>	<b>35.58</b>	<b>0.9549</b>
Bicubic	$\times 4$	28.42	0.8104	26.00	0.7027	25.96	0.6675	23.14	0.6577	24.89	0.7866
SRCNN [4]	$\times 4$	30.48	0.8628	27.50	0.7513	26.90	0.7101	24.52	0.7221	27.58	0.8555
RDN [36]	$\times 4$	32.47	0.8990	28.81	0.7871	27.72	0.7419	26.61	0.8028	31.00	0.9151
RCAN [35]	$\times 4$	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087	31.22	0.9173
SAN [3]	$\times 4$	32.64	0.9003	28.92	0.7888	27.78	0.7436	26.79	0.8068	31.18	0.9169
NLSN [24]	$\times 4$	32.59	0.9000	28.87	0.7891	27.78	0.7444	26.96	0.8109	31.27	0.9184
DRLN [1]	$\times 4$	32.63	0.9002	28.94	0.7900	27.83	0.7444	26.98	0.8119	31.54	0.9196
HAN [25]	$\times 4$	32.64	0.9002	28.90	0.7890	27.80	0.7442	26.85	0.8094	31.42	0.9177
RFANet[20]	$\times 4$	32.66	0.9004	28.88	0.7894	27.79	0.7442	26.92	0.8112	31.41	0.9187
CRAN [37]	$\times 4$	32.72	0.9012	29.01	0.7918	27.86	0.7460	27.13	0.8167	31.75	0.9219
SwinIR [16]	$\times 4$	32.92	0.9044	29.09	0.7950	27.92	0.7489	27.45	<b>0.8254</b>	32.03	0.9260
NLSN* [19]	$\times 4$	32.86	0.9025	29.11	0.7940	27.92	0.7481	<b>27.49</b>	0.8247	32.09	0.9251
RDRN(ours)	$\times 4$	<b>32.94</b>	<b>0.9039</b>	<b>29.17</b>	<b>0.7951</b>	<b>27.96</b>	<b>0.7490</b>	<b>27.49</b>	0.8241	32.27	0.9259
RDRN+ (ours)	$\times 4$	<b>33.00</b>	<b>0.9046</b>	<b>29.24</b>	<b>0.7961</b>	<b>28.01</b>	<b>0.7499</b>	<b>27.63</b>	<b>0.8266</b>	<b>32.47</b>	<b>0.9273</b>

**Quantitative results.** In Table 4 we compare the proposed RDRN model with the following super-resolution methods: SPMSR [26], SRCNN [4], FSRCNN [6], VDSR [12], IRCNN [33], SRMDNF [34], RDN [36], RCAN [35], SRFBN [15], SAN [3], HAN [25], FRANet [20] and CRAN [37]. As shown, our solution achieves consistently better performance than other methods even without self-ensemble (i.e. RDRN+).

**Visual results.** We show visual comparison for  $\times 3$  SR with BD degradation in Figure 5. The proposed model can recover grid textures and stripes even under heavy blur conditions. In the provided examples, RDRN reconstructs all stripes in the correct direction. In contrast, compared models have problems with the stripes direction and blurred areas.



**Fig. 3.** Visual comparison for  $4\times$  SR with BI model on Urban100 dataset.



**Fig. 4.** Visual comparison for  $2\times$  SR with BI model on Urban100 dataset.

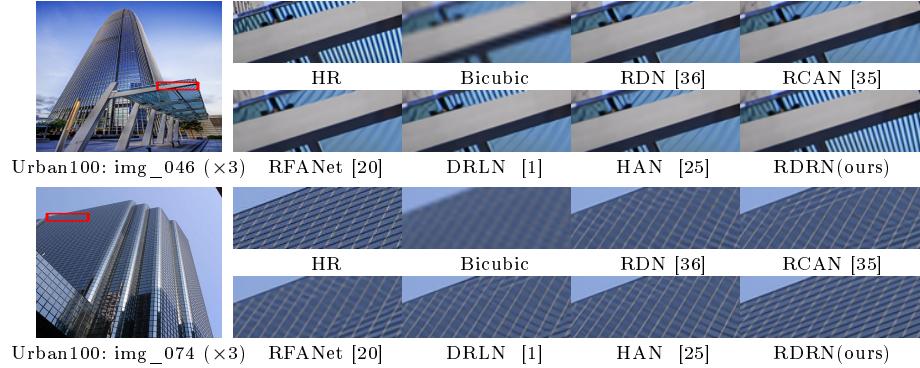
#### 4.5 Results with Bicubic Downscale-Noise (DN) Degradation Model

We apply our method to super-resolve images with the downscale-noise (DN) degradation model, which is widely used in recent SISR papers [4, 12, 6, 33, 33, 36, 37]. For DN degradation, HR image is first downsampled with scaling factor  $\times 3$ , after which Gaussian noise with noise level 30 is added to it.

**Quantitative results.** In Table 5 we compare the proposed RDRN model with the following super-resolution methods: SRCNN [4], FSRCNN [6], VDSR [12], IRCNN\_G [33], IRCNN\_C [33], RDN [36], CRAN [37]. As shown, our solution achieves consistently better performance than the other methods even without self-ensemble (i.e. RDRN+). For DN degradation we do not provide visual comparison, because the results of recent work is not available publicly.

**Table 4.** Quantitative results with BD degradation model. The best and second best results are highlighted in **bold** and underlined

Method	Scale	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM								
Bicubic	$\times 3$	28.78	0.8308	26.38	0.7271	26.33	0.6918	23.52	0.6862	25.46	0.8149
SPMSR [26]	$\times 3$	32.21	0.9001	28.89	0.8105	28.13	0.7740	25.84	0.7856	29.64	0.9003
SRCNN [4]	$\times 3$	32.05	0.8944	28.80	0.8074	28.13	0.7736	25.70	0.7770	29.47	0.8924
FSRCNN [6]	$\times 3$	26.23	0.8124	24.44	0.7106	24.86	0.6832	22.04	0.6745	23.04	0.7927
VDSR [12]	$\times 3$	33.25	0.9150	29.46	0.8244	28.57	0.7893	26.61	0.8136	31.06	0.9234
IRCNN [33]	$\times 3$	33.38	0.9182	29.63	0.8281	28.65	0.7922	26.77	0.8154	31.15	0.9245
SRMDNF [34]	$\times 3$	34.01	0.9242	30.11	0.8364	28.98	0.8009	27.50	0.8370	32.97	0.9391
RDN [36]	$\times 3$	34.58	0.9280	30.53	0.8447	29.23	0.8079	28.46	0.8582	33.97	0.9465
RCAN [35]	$\times 3$	34.70	0.9288	30.63	0.8462	29.32	0.8093	28.81	0.8647	34.38	0.9483
SRFBN [15]	$\times 3$	34.66	0.9283	30.48	0.8439	29.21	0.8069	28.48	0.8581	34.07	0.9466
SAN [3]	$\times 3$	34.75	0.9290	30.68	0.8466	29.33	0.8101	28.83	0.8646	34.46	0.9487
HAN [25]	$\times 3$	34.76	0.9294	30.70	0.8475	29.34	0.8106	28.99	0.8676	34.56	0.9494
RFANet [20]	$\times 3$	34.77	0.9292	30.68	0.8473	29.34	0.8104	28.89	0.8661	34.49	0.9492
CRAN [37]	$\times 3$	34.90	0.9302	30.79	0.8485	29.40	0.8115	29.17	0.8706	34.97	0.9512
RDRN(ours)	$\times 3$	35.07	0.9317	31.07	0.8524	29.54	0.8152	29.72	0.8792	35.53	0.9538
RDRN+(ours)	$\times 3$	<b>35.12</b>	<b>0.9320</b>	<b>31.15</b>	<b>0.8533</b>	<b>29.57</b>	<b>0.8157</b>	<b>29.86</b>	<b>0.8812</b>	<b>35.66</b>	<b>0.9543</b>



**Fig. 5.** Visual comparison for 3 $\times$  SR with BD model on Urban100 dataset.

#### 4.6 Model Complexity Analyses

Fig. 1 and Table 6 demonstrate comparison with recent SR works in terms of model size, FLOPs and performance on Manga109 dataset. Compared to NLSN, the number of parameters of our RDRN<sub>5</sub> is reduced by **18%**.

**Recursion Depth Analysis.** We show comparison of the proposed models with different recursion depth  $T = 4, 5, 6$ . The smaller network RDRN<sub>4</sub> outperforms most of the recent methods on Manga109 dataset with comparable complexity and number of parameters.

## 5 Conclusion

In this paper, we propose a recursively defined residual network (RDRN) for highly accurate image SR. Specifically, recursively defined residual block (RDRB)

**Table 5.** Quantitative results with DN degradation model. The best and second best results are highlighted in **bold** and underlined

Method	Scale	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM								
Bicubic	$\times 3$	24.01	0.5369	22.87	0.4724	22.92	0.4449	21.63	0.4687	23.01	0.5381
SRCNN [4]	$\times 3$	25.01	0.6950	23.78	0.5898	23.76	0.5538	21.90	0.5737	23.75	0.7148
FSRCNN [6]	$\times 3$	24.18	0.6932	23.02	0.5856	23.41	0.5556	21.15	0.5682	22.39	0.7111
VDSR [12]	$\times 3$	25.20	0.7183	24.00	0.6112	24.00	0.5749	22.22	0.6096	24.20	0.7525
IRCNN <sup>—</sup> G [33]	$\times 3$	25.70	0.7379	24.45	0.6308	24.28	0.5900	22.90	0.6429	24.88	0.7765
IRCNN <sup>—</sup> C [33]	$\times 3$	27.48	0.7925	25.92	0.6932	25.55	0.6481	23.93	0.6950	26.07	0.8253
RDN [36]	$\times 3$	28.47	0.8151	26.60	0.7101	25.93	0.6573	24.92	0.7354	28.00	0.8591
CRAN [37]	$\times 3$	28.74	0.8235	26.77	0.7178	26.04	0.6647	25.43	0.7566	28.44	0.8692
RDRN(ours)	$\times 3$	<b>28.81</b>	<u>0.8244</u>	<b>26.87</b>	<u>0.7201</u>	<b>26.11</b>	<u>0.6674</u>	<b>25.73</b>	<u>0.7654</u>	<b>28.80</b>	<u>0.8739</u>
RDRN+(ours)	$\times 3$	<b>28.84</b>	<u>0.8251</u>	<b>26.90</b>	<u>0.7205</u>	<b>26.12</b>	<u>0.6678</u>	<b>25.82</b>	<u>0.7678</u>	<b>28.88</b>	<u>0.8750</u>

**Table 6.** Number of parameters, FLOPs (for  $3 \times 64 \times 64$  input) and performance on Manga109 with upscale factor  $\times 4$  (BI model)

Model	Parameters, M	FLOPs, G	PSNR, dB
SwinIR	11.9	54	32.03
RCAN	15.6	65	31.21
SAN	15.9	67	31.18
HAN	16.9	67	31.42
RDRN <sub>4</sub> (ours)	18.5	62	32.09
NLSN*	44.2	222	32.09
NLSN	44.2	222	31.27
RDRN <sub>5</sub> (ours)	36.4	123	32.27
RDRN <sub>6</sub> (ours)	72.2	241	32.39

allows us to build and train a large and powerful network. To stabilize the training and further improve the quality we apply intermediate supervision (IS) loss function. Training with IS allows the network to learn more informative features for more accurate reconstruction. RDRN achieves superior SISR results under different degradation models, such as bicubic interpolation (BI), blur-downscale (BD) and downscale-noise (DN). Extensive experiments demonstrate that the proposed model outperforms recent state-of-the-art solutions in terms of accuracy and visual quality. The proposed network architecture is general and could be applied to other low-level computer vision tasks. The architecture of the core RDRB block can be simply described by two schemes: basic block and the recursive block. We implement only simple ideas and believe that the proposed approach could be significantly improved either using manual or automatic search.

**Acknowledgements** Authors would like to thank Ivan Mazurenko, Li Jieming and Liao Guiming from Huawei for fruitful discussions, guidance and support.

## References

1. Anwar, S., Barnes, N.: Densely residual laplacian super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2020)

2. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: BMVC (2012)
3. Dai, T., Cai, J., Zhang, Y., Xia, S.T., Zhang, L.: Second-order attention network for single image super-resolution. In: CVPR (2019)
4. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: ECCV (2014)
5. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. TPAMI **38**(2), 295–307 (2015)
6. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: ECCV (2016)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
9. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
10. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR (2015)
11. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
12. Kim, J., Kwon Lee, J., Mu Lee, K.: Accurate image super-resolution using very deep convolutional networks. In: CVPR (2016)
13. Kim, J., Kwon Lee, J., Mu Lee, K.: Deeply-recursive convolutional network for image super-resolution. In: CVPR (2016)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
15. Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., Wu, W.: Feedback network for image super-resolution. In: CVPR (2019)
16. Liang, J., Cao, J., Sun, G., Zhang, K., Gool, L.V., Timofte, R.: Swinir: Image restoration using swin transformer. arXiv preprint arXiv:2108.10257 (2021)
17. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: CVPR (2017)
18. Liu, J., Tang, J., Wu, G.: Residual feature distillation network for lightweight image super-resolution. European Conference on Computer Vision Workshops (2020)
19. Liu, J., Tang, J., Wu, G.: Adadm: Enabling normalization for image super-resolution. arXiv preprint arXiv:2111.13905 (2021)
20. 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)
21. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
22. Martin, D., Fowlkes, 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 (2001)
23. Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., Aizawa, K.: Sketch-based manga retrieval using manga109 dataset. Multimedia Tools and Applications **76**(20), 21811–21838 (2017)

24. Mei, Y., Fan, Y., Zhou, Y.: Image super-resolution with non-local sparse attention. In: CVPR. pp. 3517–3526. Computer Vision Foundation / IEEE (2021)
25. Niu, B., Wen, W., Ren, W., Zhang, X., Yang, L., Wang, S., Zhang, K., Cao, X., Shen, H.: Single image super-resolution via a holistic attention network. In: ECCV (12). Lecture Notes in Computer Science, vol. 12357, pp. 191–207. Springer (2020)
26. Peleg, T., Elad, M.: A statistical prediction model based on sparse representations for single image super-resolution. TIP **23**(6), 2569–2582 (2014)
27. Shi, W., Caballero, J., Huszár, 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 (2016)
28. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR (2017)
29. Timofte, R., Agustsson, E., Van Gool, L., Yang, M.H., Zhang, L.: Ntire 2017 challenge on single image super-resolution: Methods and results. In: CVPRW (2017)
30. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: arXiv preprint arXiv:1706.03762 (2017)
31. 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)
32. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: International conference on curves and surfaces (2010)
33. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. In: CVPR (2017)
34. Zhang, K., Zuo, W., Zhang, L.: Learning a single convolutional super-resolution network for multiple degradations. In: CVPR (2018)
35. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: ECCV (2018)
36. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: CVPR (2018)
37. Zhang, Y., Wei, D., Qin, C., Wang, H., Pfister, H., Fu, Y.: Context reasoning attention network for image super-resolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4278–4287 (October 2021)
38. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: CVPR (2019)