# DenseNet with Deep Residual Channel-Attention Blocks for Single Image Super Resolution

Dong-Won Jang and Rae-Hong Park
Department of Electronic Engineering, Sogang University
Seoul 04107, South Korea
{javelot87, rhpark}@sogang.ac.kr

## Abstract

*This paper proposes a DenseNet with deep Residual Channel Attention (DRCA) for single image super resolution. Recent works have shown that skip connections between layers improve the performance of the convolutional neural network such as ResNet and DenseNet. We have interpreted the role of ResNet (feature value refinement by addition) and DenseNet (feature value memory by concatenation). The contribution of the proposed network is dense connections between residual groups rather than convolution layers. In terms of feature value refinement and memory, the proposed method refines the feature values sufficiently (by residual group) and memorizes the refined feature values intermittently (by dense connections between residual groups). Experimental results show that the proposed DRCA (14.2M) achieved better performance than the state-of-the-art methods with fewer parameters.*

## 1. Introduction

Convolutional neural network (CNN) based methods have achieved impressive performance in various image applications such as classification and single image super resolution (SISR). ResNet [5] and DenseNet [7] have the advantages in back-propagation with strong gradients using skip connections and give good performances on classifications. Also, in SISR, these blocks are successfully used and improve the performance in terms of the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [25]. After the first CNN-based SISR method (SRCNN [3]) was introduced, deep network [10], Laplacian pyramid [13], residual network [14], dense network [22], and back-projection network [4] based on back-projection algorithm [2] have been successfully applied to SISR.

Recent networks have been deeper and deeper, and thus some problems arise. Szegedy *et al*. [20] addressed that deep network can die early in the training and this phenomenon can be prevented by adding scaling layer after convolution layer. EDSR [15], the winner of NTIRE2017 Super-Resolution Challenge [21], used the deeper and wider architecture of the SRResNet with scaling layer [20], in which the scaling factor was set to 0.1.

Channel attention (CA) computes channel weights from 0 to 1 and successfully increases the performance of classification networks such as ResNet, VGG16, and Inception [6]. Because CA works as the scaling layer with trainable scaling factor, CA can be well performed for SISR network. Zhang *et al*. [28] applied CA to SISR with residual-in-residual learning (RCAN) and achieved the state-of-the art performance.

Recently, combining ResNet and DenseNet has been used in SISR. Zhang *et al*. proposed RDN [29] that consists of residual dense blocks (RDB). Wen *et al*. [26] also combined ResNet and DensNet (DRNet) that consists of densely connected residual blocks which are similar to those of RDN. RRDB [23] combined residual-in-residual learning [28] and dense connection [7]. However, these hybrid methods gave worse performance than RCAN [28] that uses ResNet only.

We propose one of combining methods of ResNet and DenseNet, however, we focus on the role of ResNet and DenseNet. We interpret that ResNet refines the feature value by addition and DenseNet memorizes the feature value by concatenation. Previous combining methods (RDN, DRNet, and RRDB) densely connect convolution layers, whereas the proposed method densely connects residual groups rather than convolution layers. The proposed DenseNet with Residual Channel Attention is called DRCA. Using our interpretation, the proposed DRCA sufficiently updates the feature values by residual blocks in the residual group and memorizes the refined feature values intermittently by dense connections between residual groups. Although the proposed method is one of combinations of ResNet, DenseNet, and channel-attention, this small difference achieved better performance than the state-of-the-art methods.

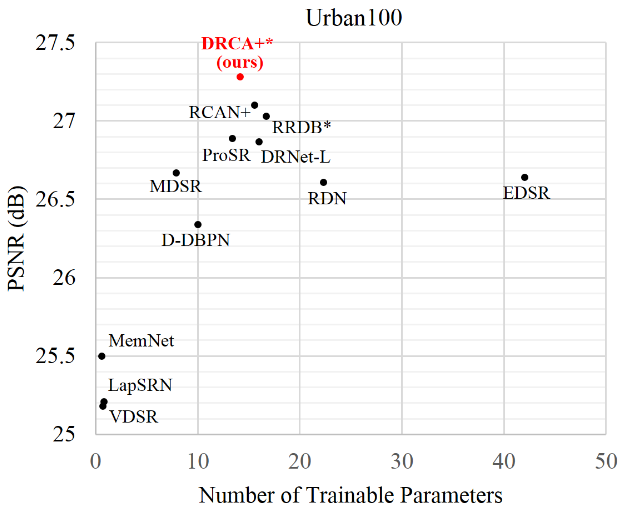

Figure 1. Performance comparison in terms of the PSNR versus the number of trainable parameters for scale factor 4 (+ and * indicate self-ensemble [15] and different train dataset (DF2K), respectively).

Fig. 1 shows the performance comparison of the proposed and state-of-the-art methods in terms of the PSNR versus the number of parameters on Urban100 dataset [8], where the proposed DRCA+* achieved better performance than the state-of-the-art methods (+ and * indicate self-ensemble [15] and different train dataset (DF2K dataset), respectively).

Our contributions are as follows:

- We propose DRCA by effectively combining the DenseNet architecture with residual learning.
- The proposed DRCA (14.2M) achieves better performance with fewer parameters and faster run-time than the state-of-the art method (RCAN, 15.8M).

## 2. Related work

Because the proposed network combines ResNet and DenseNet with CA, we briefly review DenseNet, ResNet, and CA. Most SISR methods consist of three steps: initial feature extraction, non-linear mapping, and reconstruction. The key contribution of existing SISR methods is how to construct non-linear mapping: e.g., by residual learning (SRResNet and EDSR), dense connectivity (SRDenseNet), back-projection unit (D-DBPN), residual dense block (DRNet and RDN), and residual in residual (RCAN). The proposed method performed non-linear mapping by dense connections between residual groups.

Let $I_{LR}$ and $I_{SR}$ be the low-resolution (LR) input image and the output SR image, respectively. The desired high-resolution (HR) image (ground truth) is denoted by $I_{HR}$. Most SISR methods was trained by supervised learning which minimized the error between $I_{SR}$ and $I_{HR}$. The initial feature $\mathbf{f}_1$ extracted from input $I_{LR}$ is given by $\mathbf{f}_1 = F_1(I_{LR})$ where $F_1(\cdot)$ represents the first convolution layer.

### 2.1. DenseNet

Dense connectivity ensures maximum information paths between layers by connecting all layers [7]. The output features of all convolution layers in the dense blocks are concatenated along the channel axis. Let $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ be the input and output of the$i$-th densely connected convolution layer, respectively. The output $\mathbf{x}_i$ of the $i$-th convolution layer $F_i$ can be obtained by

$$\mathbf{x}_i = F_i([\mathbf{x}_{i-1}, \mathbf{x}_{i-2}, \cdots, \mathbf{x}_0]) \tag{1}$$

where $[\cdot]$ represents feature concatenation along the channel axis and the input $\mathbf{x}_0$ is the initial feature $\mathbf{f}_1$. Because the input of the convolution layer is repeatedly concatenated, the number of channels of the inputs for the next convolution layer consistently increases with growth rate $k$.

### 2.2. ResNet

He *et al.* [5] introduced residual learning. Let $\mathbf{x}$ and $F(\mathbf{x})$ be the network input and output, respectively, then desired underlying mapping $H(\mathbf{x})$ is to be learned. Residual learning trains $H(\mathbf{x}) - \mathbf{x} = F(\mathbf{x})$ rather than $H(\mathbf{x})$ as in most CNNs. The desired mapping $H(\mathbf{x})$ is cast into $H(\mathbf{x}) = F(\mathbf{x}) + \mathbf{x}$ which can be implemented with skip connections.

### 2.3. Channel-attention

CA was introduced as *squeeze and excitation* block for classification [6]. In CA, a weight is multiplied to each channel and the weights are trainable parameters. In classification, residual learning with CA gives better performance. Zhang *et al.* [28] also argued that CA improves the performance of SISR.

Fig. 2 shows the block diagram of CA. CA consists of squeeze section and excitation section. Squeeze works as global pooling: formally, input feature $\mathbf{x} \in R^{H \times W \times C}$ is averaged channel-wise to generate $\mathbf{z} \in R^{1 \times 1 \times C}$, e.g., at the $c$-th channel:

$$z^c = F_{gp}(\mathbf{x}^c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x^c(i,j), 1 \leq c \leq C \tag{2}$$

where $F_{gp}(\cdot)$ represents global pooling, and $H$, $W$, and $C$ denote the height, width, and the number of channels of the input feature $\mathbf{x}$, respectively. Then, channel-wise weights $\mathbf{s}$ are computed by

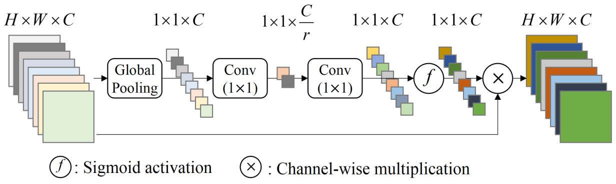

Figure 2. Block diagram of CA.

$$\mathbf{s} = f(F_{sq2}(\delta(F_{sq1}((z))))), \tag{3}$$

where $f$ denotes the sigmoid activation, $\delta$ indicates the rectified linear unit (ReLU) [12], and $F_{sq1}(\cdot)$ and $F_{sq2}(\cdot)$ are $1 \times 1$ convolution layers with channel reduction parameter $r$. Finally, the input and the obtained weights, $\mathbf{s} = [s^1, s^2, \cdots, s^C]$, are multiplied channel-wise to give

$$\widetilde{\mathbf{x}}^c = s^c \mathbf{x}^c, 1 \leq c \leq C \tag{4}$$

where the output feature $\widetilde{\mathbf{x}} = [\widetilde{\mathbf{x}}^1, \widetilde{\mathbf{x}}^2, \cdots, \widetilde{\mathbf{x}}^C]$ of CA has the same dimension as the input feature $\mathbf{x} \in R^{H\times W\times C}$.

## 3. Proposed method

In DenseNet, an increased number of dense connections indicates an increase of the number of channels (growth rate $k$). Huang *et al.* [7] argued that direct connections from any layers to all subsequent layers improve the information flow between layers. However, too large channel dimension due to many dense connections degrades the performance of the network if there is no bottleneck layer [7]. Furthermore, the transition layer is located between dense blocks to reduce the number of channels. Tong *et al.* [22] argued that the skip connections between dense blocks improve the performance of network in terms of the PSNR for SISR.

Fig. 3 shows the overall framework of the proposed DRCA. The proposed DRCA densely connects residual groups rather than convolution layers. Because the bottleneck layers or the transition layers can be overloaded by many dense connections, we construct the proposed DRCA with a small number of residual groups (five or six residual groups). First, the proposed DRCA extracts the initial feature with one convolution layer and this feature is densely connected to all residual groups and the last projection layer ($1 \times 1$ convolution layer). Similar to most ResNet-based SISR methods, we use long skip connection. Upsample block consists of convolution layer and pixel shuffle layer (PSL) [19]. We use PSL with a scale factor of 2. For ×4 enlargement, two upsample blocks are used sequentially, which is the same as in EDSR and RCAN.

### 3.1. Residual group

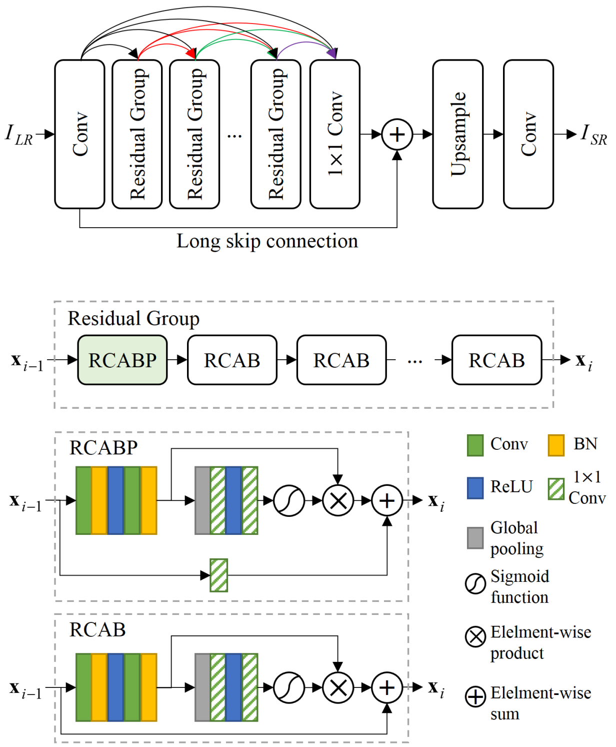

Figure 3. Block diagram of the proposed network with composite blocks.

Residual group consists of residual channel-attention blocks (RCAB). In the proposed network, because residual groups are densely connected, the first RCAB include projection unit (RCABP) to reduce the number of channels in the residual group. For example, let $k$ be the number of channels of the initial feature $\mathbf{f}_1$ and also the growth rate of dense connection between residual groups. The number of input channels of the $i$-th residual group is $i \times k$ and each residual group generates $k$ channel features. To the element-wise sum with different number of channels between input and output, RCABP reduces the number of channels from $i \times k$ to $k$ by using $1 \times 1$ convolution layer at the residual connection and at the first convolution layer in the residual block. RCABP is located at the first in each residual group and works as a bottleneck layer [7] as shown in Fig. 3. Details of RCAB and RCABP are also shown in Fig. 3.

### 3.2. Difference from previous methods

**Difference from RDN [29], RRDB [23], and DRNet [26].** Our method, RDN, RRDB, and DRNet combine ResNet and DenseNet for SISR. Fig. 4 shows basic blocks of RDN, RRDB, and DRNet. The basic block of RRDB consists of three dense blocks (Fig. 4(b)) with residual in residual connections [28]. For simplicity, we show the details of dense block in RRDB only. All of previous combining methods densely connect convolution layers, whereas

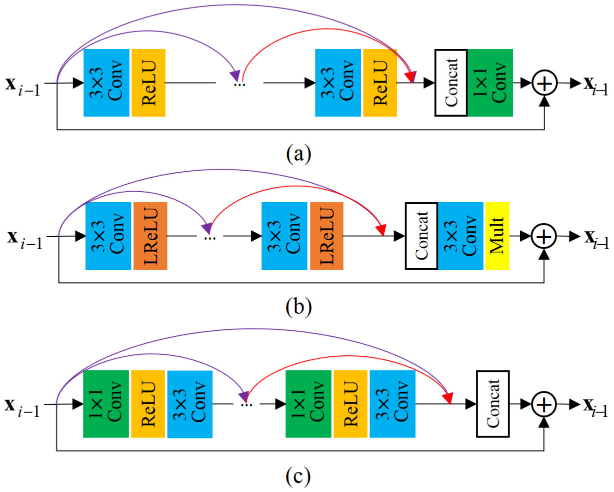

Figure 4. Basic units of RDN, RRDB, and DRNet. (a) RDB in RDN [29]. (b) Dense block in RRDB [23]. (c) DRBlock in DRNet [26].

the proposed method densely connects residual groups. As shown in Figs. 4(a) and 4(b), RDN and RRDB reduce the number of channels at the last of the basic block by using the projection layer to pass the features to the next basic block. In other words, these methods have the transition layer in every basic block, which may degrade the performance as reported in SRDenseNet [22]. DRNet reduces the number of channels at the bottleneck of each convolution layer as shown in Fig. 4(c).

Furthermore, the proposed method uses global dense connections with local residual connections, whereas the other hybrid methods use global residual connections with local dense connections. Dense connections realize a contiguous memory mechanism by passing the concatenated features of the previous layers to the next layer [29]. Because RDN, RRDB, and DRNet densely connect the convolution layers only inside the basic block, the contiguous memory mechanism is only used inside the basic block. However, the proposed method densely connects from initial feature extraction to residual groups and the last projection layer, and thus most parts of the proposed method realize the contiguous memory mechanism. Moreover, the proposed method concatenates the features after a sufficient update by many convolution layers in the residual group, whereas the previous methods concatenate the features after one update by one convolution layer.

**Difference from RCAN [28].** Our method and RCAN use residual groups. The key difference is connections between residual groups. Our residual groups are connected using dense connection whereas the residual groups in RCAN [28] are connected using residual connection, and thus the proposed method can be called residual in dense connections. Table 1 summarizes the differences between the proposed method and previous methods.

| Method | Connections between convolution layers | Connections between basic blocks | Use BN? |
|---|---|---|---|
| RDN | Dense | Residual | No |
| DRNet | Dense | Residual | No |
| RRDB | Dense | Residual | No |
| RCAN | Residual | Residual | No |
| Ours | Residual | **Dense** | Yes |

Table 1. Differences between the proposed method and the previous methods (RDN, DRNet, RRDB, and RCAN).

### 3.3. Implementation details

We set the number of residual groups to five and the number of RCABs to 36 (one RCABP and 35 RCABs) for each residual group. Then, the total number of RCABs is 180. The kernel size of all convolution layers except for the projection layer is set to 3 × 3.The size of zero padding is set to one to keep the same size between input and output features. At the first convolution layer of RCABP in the i-th residual group, the number of channels is reduced from $i \times k$ to $k$. The channel reduction ratio $r$ in CA is set to 16, which is the same as in RCAN [28]. The number of initial channels and the growth rate $k$ are set to 64. The number of channels in PSL [19] is set to $4k$, which is the same as in conventional SISR methods [14, 15, 28, 29]. Similar to EDSR and RCAN, we first train ×2 enlargement network and then use this network as pre-trained network for ×4 enlargement network. The ReLU is used as activation unit. In contrast to recent SISR methods [15, 23, 24, 28, 29], we use batch normalization (BN) [9].

## 4. Experimental results and discussions

We implemented the proposed networks with the PyTorch [18] framework and trained using NVIDIA GTX 1080 Ti. Training takes 10 days. The source code is available online: `https://github.com/dong-won-jang/DRCA`. We showed results with/without self-ensemble [15] and discussed pros and cons of self-ensemble.

### 4.1. Training

We trained the proposed networks using DIV2K [21]. The LR images were obtained by bicubic downsampling of HR images using MATLAB. The batch size and input image size were set to 16 and respectively. The input images were randomly cropped, rotated (90, 180, and 270 degrees), and flipped during training. The learning rate was set to $10^{-4}$ and decreased by a factor of 2 for every

$2 \times 10^5$ back-propagation iterations; training was terminated at $10^6$ iterations. We used $L_1$ loss as training loss and Adam optimizer [11] with $\beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 10^{-8}$. Furthermore, Wang *et al.* [23] reported the influence of training dataset. Although using the same network and the same training procedure, different training dataset can make a big performance difference (by 0.3dB in ×4 enlargement). Similar to Wang *et al.* s method, we trained the proposed DRCA on DIV2K dataset [21] only and on the merged dataset with DIV2K and Flickr2K [15], namely DF2K dataset. Similar to Wang *et al.*, the proposed network trained on DF2K dataset gave better performance than trained on DIV2K dataset. For fair comparisons, in the rest of the paper, results trained on DF2K dataset is represented as *.

### 4.2. Ablation study

We studied the influence of the number of residual groups with the total number of RCABs fixed. The first model consists of five residual groups with 36 RCABs, whereas the second model consists of six residual groups with 30 RCABs. Then, the total number of RCABs is 180 for both models. We use BN only for the first model. In this case, the number of parameters is approximately 14.18M. For the second model, we remove BN. In this case, the number of parameters is approximately 14.15M.

Table 2 shows quantitative comparisons of our two models using five datasets: Set5 [1], Set14 [27], BSD100 [16], Urban100 [8], and Manga109 [17]. These dataset images can be classified into natural scenes (Set5, Set14, and BSD100), urban scenes (Urban100), and Japanese manga (Manga109). Although the numbers of parameters of two models are similar, the performances of these models are different. Five residual groups with BN is better than six residual groups without BN for all datasets. In the rest of the paper, the results of DRCA are obtained by five residual groups with 36 RCABs.

| Dataset | DRCA($G$=5, $R$=36) | | DRCA($G$=6, $R$=30) | |
|---|---|---|---|---|
| | PSNR (dB) | SSIM | PSNR (dB) | SSIM |
| Set5 | **32.68** | **0.9009** | 32.61 | 0.8997 |
| Set14 | **28.91** | **0.7898** | 28.88 | 0.7886 |
| BSD100 | **27.80** | **0.7444** | 27.78 | 0.7434 |
| Urban100 | **26.94** | **0.8111** | 26.84 | 0.8082 |
| Manga109 | **31.37** | **0.9182** | 31.29 | 0.9179 |

Table 2. Quantitative comparison depending on the numbers of residual groups (*G*) and RCABs (*R*) (scale factor: 4).

### 4.3. Quantitative comparisons

For quantitative comparisons, the PSNR and SSIM were used. These metrics were computed on Y channel only in the transformed YCbCr color space. Similar to [4, 14, 15, 28, 29], we cropped the boundary by the same number of pixels as the scale factor before evaluation, e.g., four pixels in case of ×4 enlargement.

We compared the proposed method with 11 state-of-the-art methods for two enlargement cases: VDSR [10], LapSRN [13], SRResNet [14], SRDenseNet [22], EDSR [15], D-DBPN [4], RDN [29], DRNet-L [26], RCAN [28], ProSR [24], and RRDB [23]. The results of SRResNet and SRDenseNet were computed by re-implementation [26] for the same training dataset. We used self-ensemble [15] which was the average resulting image of 8 geometric transformations including original image during test phase. In the rest of the paper, self-ensemble was represented as +. The best and the second best were emphasized by bold and underline, respectively.

Table 3 shows the quantitative comparisons for ×2 and ×4 enlargement. Quantitative results not provided in the papers [24, 26] were indicated by hyphens. For ×2 enlargement, there is keen competition between RCAN+ and the proposed DRCA+ in terms of the PSNR and SSIM. Either the proposed DRCA+ or RCAN+ is usually the best. For ×4 enlargement, the proposed DRCA+* outperforms the others in terms of the PSNR and SSIM. Without self-ensemble, the proposed DRCA achieved the best performance in most datasets. Because Set5, Set14, and BSD100 usually consist of natural scenes with small height and width, the performance difference between the state-of-the-art methods and the proposed method on these datasets is fairly smaller than that of Urban100 and Manga109 datasets.

### 4.4. Qualitative comparisons

Both self-ensemble and DF2K dataset improve the quantitative performance in terms of the PSNR and SSIM, however, self-ensemble may generate aliasing problem. To discuss pros and cons of self-ensemble, the only recent state-of-the-art SISR methods were used for qualitative comparisons. To show which method is clearly better or worse, we show cropped and enlarged images.

Figs. 5 and 6 show high frequency regions in Urban 100 dataset. For image “img_076” of Urban100 for ×4 enlargement, EDSR, RDN, and DRNet generated blurred pattern in solid box and reconstructed reasonable structure in dashed box in HR image. RRDB and RCAN slightly restored structure in solid box, whereas generated undesirable structure in dashed box. The proposed DRCA well reconstructed reasonable structure in solid box, whereas our method also suffered from undesirable edges in dashed box. This artifact can be reduced by self-ensemble and extra train dataset such as DF2K. Self-ensemble is average resulting image of

| Method | Scale | Set5 |  | Set14 |  | BSD100 |  | Urban100 |  | Manga109 |  |
|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | PSNR (dB) | SSIM | PSNR (dB) | SSIM | PSNR (dB) | SSIM | PSNR (dB) | SSIM | PSNR (dB) | SSIM |
| Bicubic | 2 | 33.66 | 0.9299 | 30.24 | 0.8688 | 29.56 | 0.8431 | 26.88 | 0.8403 | 30.80 | 0.9339 |
| VDSR [10] | 2 | 37.53 | 0.9590 | 33.05 | 0.9130 | 31.90 | 0.8960 | 30.77 | 0.9140 | 37.22 | 0.9750 |
| LapSRN [13] | 2 | 37.52 | 0.9591 | 33.08 | 0.9130 | 31.08 | 0.8950 | 30.41 | 0.9101 | 37.27 | 0.9740 |
| EDSR [15] | 2 | 38.11 | 0.9602 | 33.92 | 0.9195 | 32.32 | 0.9013 | 32.93 | 0.9351 | 39.10 | 0.9773 |
| D-DBPN [4] | 2 | 38.09 | 0.9600 | 33.85 | 0.9190 | 32.27 | 0.9000 | 32.55 | 0.9324 | 38.89 | 0.9775 |
| RDN [29] | 2 | 38.24 | 0.9614 | 34.01 | 0.9212 | 32.34 | 0.9017 | 32.89 | 0.9353 | 39.18 | 0.9780 |
| DRNet-L [26] | 2 | 38.12 | 0.9604 | 34.07 | 0.9213 | 32.35 | 0.9016 | 33.09 | 0.9366 | - | - |
| ProSR [24] | 2 | - | - | 34.00 | - | 32.34 | - | 32.91 | - | - | - |
| RCAN [28] | 2 | 38.27 | 0.9614 | 34.12 | 0.9216 | 32.41 | 0.9027 | 33.34 | 0.9384 | 39.44 | 0.9786 |
| RCAN+ [28] | 2 | **38.33** | **0.9617** | 34.23 | 0.9225 | **32.46** | **0.9031** | **33.54** | **0.9399** | **39.61** | **0.9788** |
| DRCA (ours) | 2 | 38.28 | 0.9615 | 34.13 | 0.9230 | 32.39 | 0.9023 | 33.25 | 0.9382 | 39.40 | 0.9779 |
| DRCA+ (ours) | 2 | 38.32 | **0.9617** | **34.25** | **0.9235** | 32.43 | 0.9027 | 33.41 | 0.9392 | 39.55 | 0.9784 |
| Bicubic | 4 | 28.42 | 0.8104 | 26.00 | 0.7027 | 25.96 | 0.6675 | 23.14 | 0.6577 | 24.89 | 0.7866 |
| VDSR [10] | 4 | 31.35 | 0.8830 | 28.02 | 0.7680 | 27.29 | 0.7251 | 25.18 | 0.7540 | 28.83 | 0.8870 |
| LapSRN [13] | 4 | 31.54 | 0.8850 | 28.19 | 0.7720 | 27.32 | 0.7270 | 25.21 | 0.7560 | 29.09 | 0.8900 |
| SRResNet [14] | 4 | 32.07 | 0.8930 | 28.56 | 0.7809 | 27.57 | 0.7352 | 26.08 | 0.7839 | - | - |
| SRDenseNet [22] | 4 | 32.04 | 0.8927 | 28.52 | 0.7800 | 27.54 | 0.7346 | 26.05 | 0.7830 | - | - |
| EDSR [15] | 4 | 32.46 | 0.8968 | 28.80 | 0.7876 | 27.71 | 0.7420 | 26.64 | 0.8033 | 31.02 | 0.9148 |
| D-DBPN [4] | 4 | 32.47 | 0.8990 | 28.82 | 0.7860 | 27.72 | 0.7400 | 26.38 | 0.7946 | 30.91 | 0.9137 |
| RDN [29] | 4 | 32.47 | 0.8990 | 28.81 | 0.7871 | 27.72 | 0.7419 | 26.61 | 0.8028 | 31.00 | 0.9151 |
| DRNet-L [26] | 4 | 32.61 | 0.8993 | 28.96 | 0.7896 | 27.80 | 0.7426 | 26.87 | 0.8075 | - | - |
| ProSR [24] | 4 | - | - | 28.94 | - | 27.79 | - | 26.89 | - | - | - |
| RRDB [23] | 4 | 32.60 | 0.9002 | 28.88 | 0.7896 | 27.76 | 0.7432 | 26.73 | 0.8072 | 31.16 | 0.9164 |
| RRDB* [23] | 4 | 32.73 | 0.9011 | 28.99 | 0.7917 | 27.85 | 0.7455 | 27.03 | 0.8153 | 31.66 | 0.9196 |
| RCAN [28] | 4 | 32.63 | 0.9002 | 28.87 | 0.7889 | 27.77 | 0.7436 | 26.82 | 0.8087 | 31.22 | 0.9173 |
| RCAN+ [28] | 4 | 32.73 | 0.9013 | 28.98 | 0.7910 | 27.85 | 0.7455 | 27.10 | 0.8142 | 31.65 | 0.9208 |
| DRCA (ours) | 4 | 32.68 | 0.9009 | 28.91 | 0.7898 | 27.80 | 0.7444 | 26.94 | 0.8111 | 31.37 | 0.9182 |
| DRCA* (ours) | 4 | 32.73 | 0.9015 | 28.98 | 0.7913 | 27.86 | 0.7458 | 27.11 | 0.8156 | 31.74 | 0.9220 |
| DRCA+ (ours) | 4 | 32.75 | 0.9017 | 28.99 | 0.7913 | 27.86 | 0.7459 | 27.15 | 0.8156 | 31.71 | 0.9214 |
| DRCA+* (ours) | 4 | **32.78** | **0.9021** | **29.05** | **0.7926** | **27.90** | **0.7467** | **27.28** | **0.8190** | **31.95** | **0.9236** |

Table 3. Quantitative comparisons with the state-of-the-art methods for scale factor of 2 and 4 (**Best** and second best were highlighted). Results using DF2K dataset and self-ensemble were denoted with * and +, respectively.

8 different geometric transformations [15]. If there are no undesirable edges in the other transformation, undesirable structure can be reduced as shown in RCAN+ and DRCA+ of Fig. 5. Furthermore, extra train dataset improved the performance as shown in RRDB* and DRCA*. The proposed DRCA+* restored image with the best PSNR and SSIM and reasonable structure.

Self-ensemble can reduce undesirable structure by averaging. However, if some geometric transformation generates undesirable edges, self-ensemble may generate undesirable edges as shown in RCAN+ of Fig. 6. For image “img_073” as shown in Fig. 6, most existing methods suffered from aliasing and blurring artifacts in the left building (dashed box in HR image). RCAN produced no aliasing in the left building, however, RCAN+ generated undesirable edges. In contrast, the proposed DRCA recovered them faithfully and any artifact was observed by using self-ensemble as shown in DRCA+ and DRCA+*.

For image “253027” of BSD100 for ×4 enlargement as shown in Fig. 7, DRNet and RRDB generated aliasing in mane of the left zebra and most methods blurred cheek of the left zebra (solid box in HR image). Furthermore, most methods failed to recover stripe patterns in the right zebra (dashed box in HR image). Without self-ensemble, the proposed DRCA was not the best in terms of the PSNR, whereas it recovered well without blurred pattern and alias-

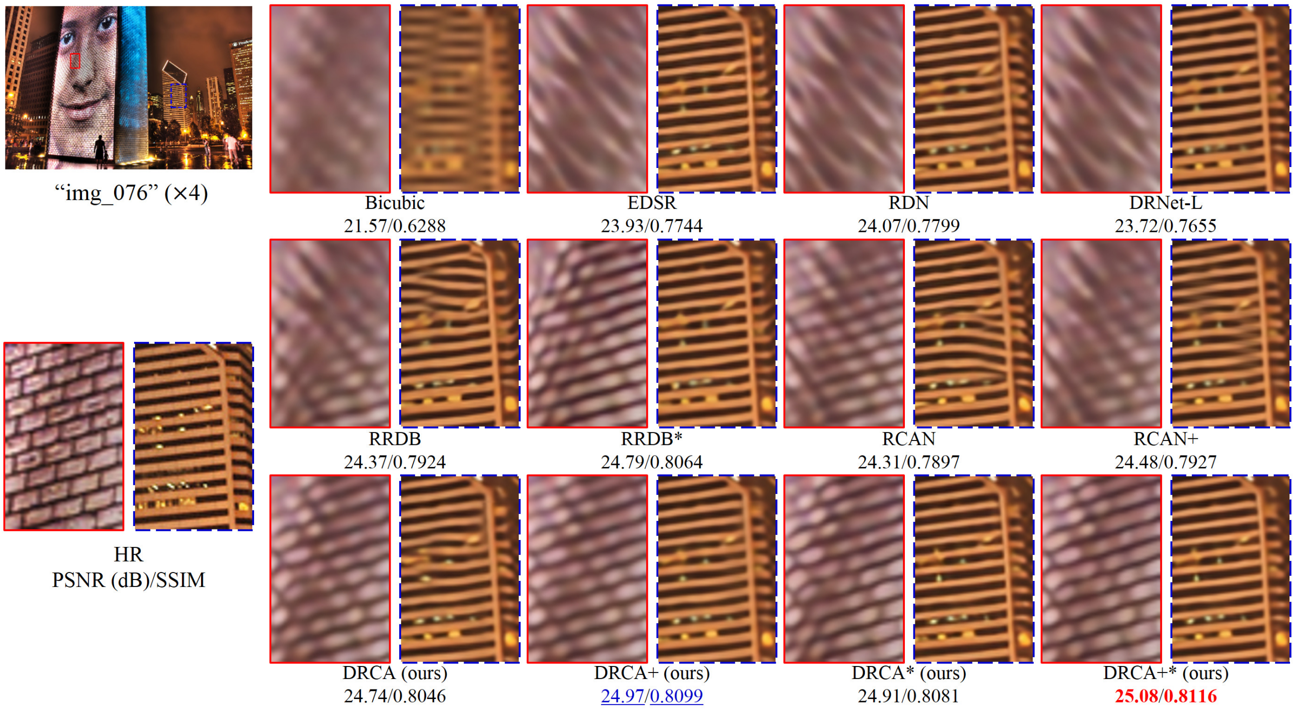

Figure 5. Qualitative comparisons with the state-of-the-art methods on "img_076" from Urban100 dataset.

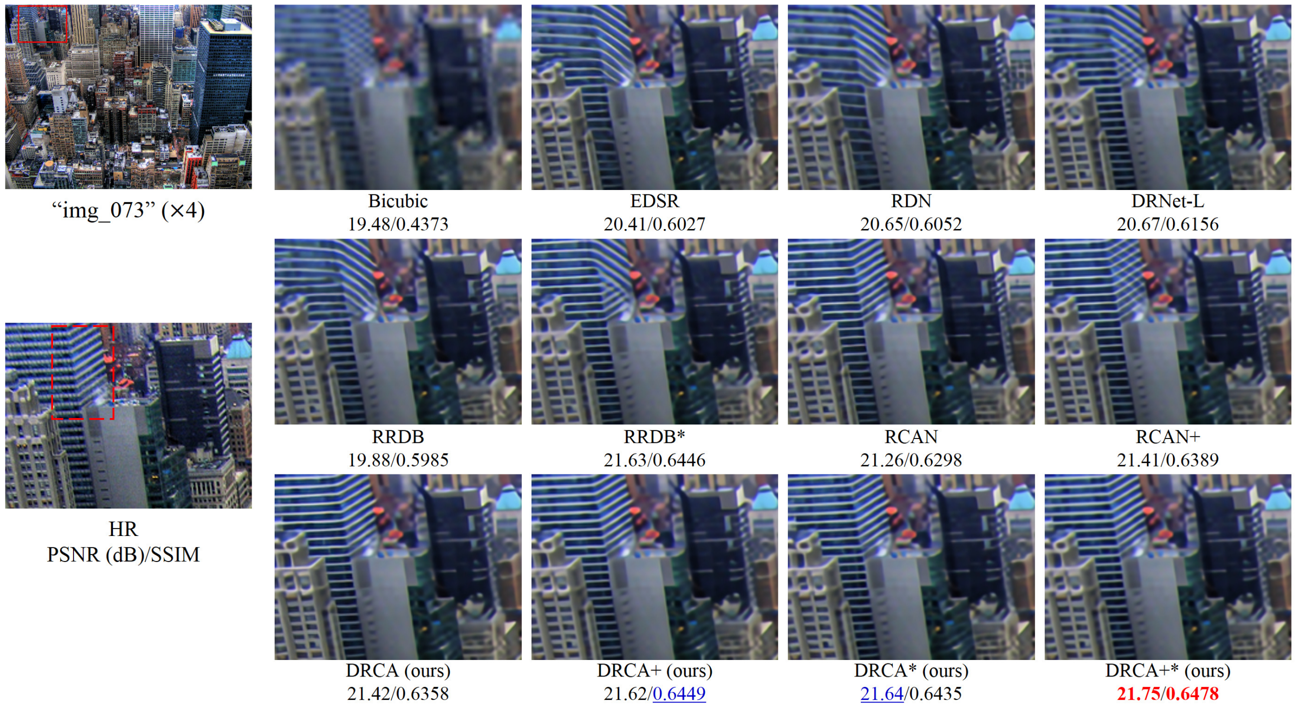

Figure 6. Qualitative comparisons with the state-of-the-art methods on "img_073" from Urban100 dataset.

ing. Furthermore, the proposed DRCA recovered stripe on abdomen faithfully in the right zebra. With self-ensemble, RCAN+ slightly recovered the stripe pattern on the back of the right zebra. DRCA+ slightly blurred the stripe pattern

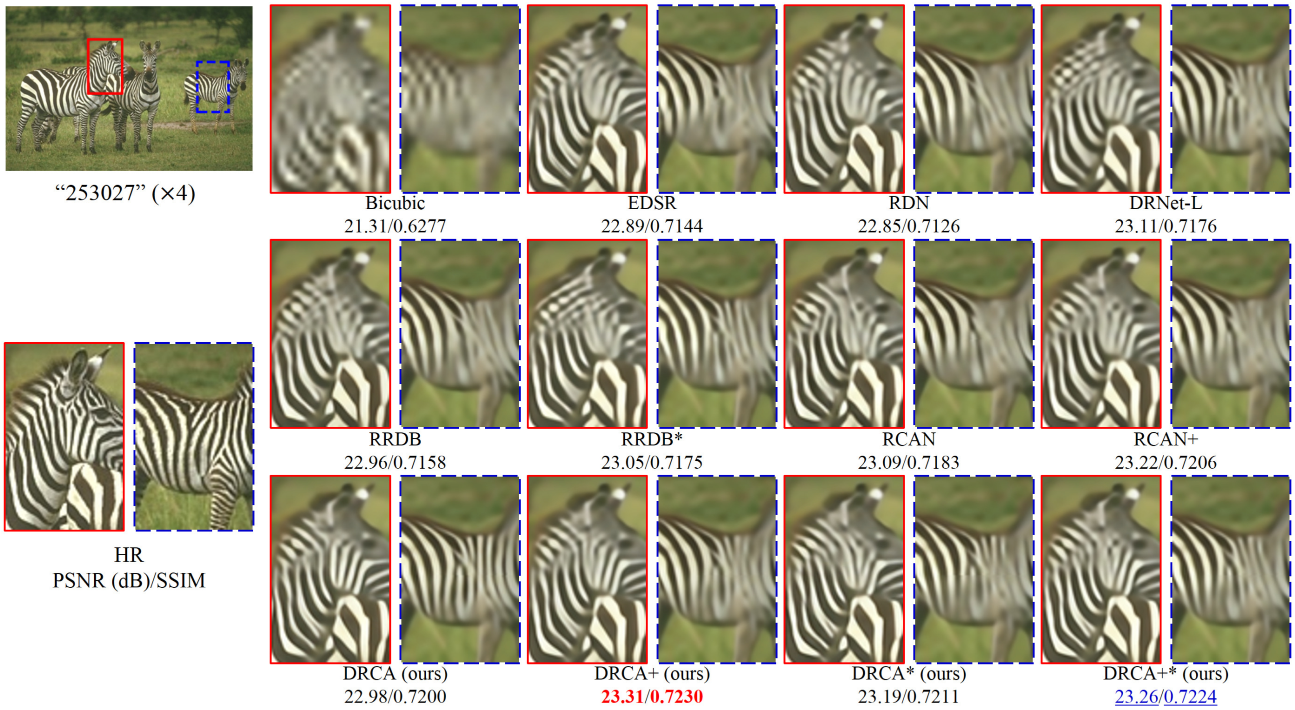

Figure 7. Qualitative comparisons with the state-of-the-art methods on "253027" from BSD100 dataset.

on the abdomen of the right zebra. However, RCAN+ and DRCA+ achieved higher PSNR and SSIM than RCAN and DRCA, respectively. Self-ensemble commonly increased the PSNR and SSIM even though self-ensemble blurred high-frequency regions. Unlike in Figs. 5 and 6, for image "253027", DRCA+ is better than DRCA+*.

### 4.5. Run-time

We measured average run-time of RCAN and the proposed DRCA using Urban100 dataset with scale factor 4. RCAN achieved 0.932s/image and the proposed method performed 0.892s/image with i5 CPU and GTX1080Ti. The proposed DRCA achieved better performance with fewer trainable parameters and less computation than RCAN.

## 5. Conclusion

We proposed SISR methods by combining DensNet and ResNet: densely connected residual groups rather than convolution layers. The deeper residual group extracted better features without increasing the number of channels of output features. Compared with the state-of-the-art methods using five datasets, the proposed network gave better results with fewer parameters and less computation. So, dense connections between deep residual groups improved network efficiency. We are careful to say that the proposed network is good for other applications. If the network learns the mapping between input image and output image with supervised learning, we expect the proposed DRCA can perform well in various applications, such as image denoising, image dehazing, and image contrast enhancement. Future work will focus on the detailed investigation of the role of ResNet and DenseNet with multiple streams or grouped convolutions and on selection of the ratio between the number of residual groups and the number of RCABs.

## Acknowledgement

This research was supported in part by the BK21 Plus Program. The authors would like to thank Y. Zhang [28,29], X. Wang [23], and R. Wen [26] for providing their codes, trained models, or resulting images.

## References

[1] M. Bevilacqua, A. Roumy, C. Guillemot, and M. Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proc. British Machine Vision Conference*, pages 1–10, 2012.

[2] S. Dai, M. Han, Y. Wu, and Y. Gong. Bilateral back-projection for single image super resolution. In *Proc. IEEE Int. Conf. Multimedia and Expo*, pages 1039–1042. IEEE, 2007.

[3] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016.

[4] M. Haris, G. Shakhnarovich, and N. Ukita. Deep back-projection networks for super-resolution. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 1664–1673, 2018.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[6] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[7] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[8] J. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 5197–5206, June 2015.

[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. Int. Conf. Machine Learning*, pages 448–456, 2016.

[10] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 1646–1654, 2016.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. Int. Conf. Learning Representations*, 2015.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105. 2012.

[13] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 624–632, 2017.

[14] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 4681–4690, 2017.

[15] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *Proc. IEEE Computer Vision and Pattern Recognition Workshops*, pages 136–144, 2017.

[16] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. volume 2, pages 416–423, 02 2001.

[17] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, Oct 2017.

[18] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *Neural Information Processing Systems Workshop*, 2017.

[19] W. Shi, J. Caballero, F. Huszar, J. Totz, A. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 1874–1883, 2016.

[20] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proc. Int. Conf. Learning Representations Workshop*, 2016.

[21] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proc. IEEE Computer Vision and Pattern Recognition Workshops*, pages 114–125, 2017.

[22] T. Tong, G. Li, X. Liu, and Q. Gao. Image super-resolution using dense skip connections. In *Proc. IEEE Int. Conf. Computer Vision*, pages 4799–4807, 2017.

[23] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proc. European Conference on Computer Vision Workshops*, September 2018.

[24] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers. A fully progressive approach to single-image super-resolution. In *Proc. IEEE Computer Vision and Pattern Recognition Workshops*, June 2018.

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

[26] R. Wen, K. Fu, H. Sun, X. Sun, and L. Wang. Image super-resolution using densely connected residual networks. *IEEE Signal Processing Letters*, 25(10):1565–1569, 2018.

[27] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Proc. Int. Conf. Curves and Surfaces*, pages 711–730, 2012.

[28] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. In *Proc. European Conference on Computer Vision*, pages 286–301, 2018.

[29] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 2472–2481, 2018.