Image Super-Resolution Using Very Deep Residual Channel Attention Networks

Yulun Zhang[1], Kunpeng Li[1], Kai Li[1], Lichen Wang[1],
Bineng Zhong[1], and Yun Fu[1,2]

[1]Department of ECE, Northeastern University, Boston, USA
[2]College of Computer and Information Science, Northeastern University, Boston, USA
{yulun100,li.kai.gml,wanglichenxj}@gmail.com,
bnzhong@hqu.edu.cn, {kunpengli,yunfu}@ece.neu.edu

Abstract. Convolutional neural network (CNN) depth is of crucial importance for image super-resolution (SR). However, we observe that deeper networks for image SR are more difficult to train. The low-resolution inputs and features contain abundant low-frequency information, which is treated equally across channels, hence hindering the representational ability of CNNs. To solve these problems, we propose the very deep residual channel attention networks (RCAN). Specifically, we propose a residual in residual (RIR) structure to form very deep network, which consists of several residual groups with long skip connections. Each residual group contains some residual blocks with short skip connections. Meanwhile, RIR allows abundant low-frequency information to be bypassed through multiple skip connections, making the main network focus on learning high-frequency information. Furthermore, we propose a channel attention mechanism to adaptively rescale channel-wise features by considering interdependencies among channels. Extensive experiments show that our RCAN achieves better accuracy and visual improvements against state-of-the-art methods.

Keywords: Super-Resolution, Residual in Residual, Channel Attention

1 Introduction

We address the problem of reconstructing an accurate high-resolution (HR) image given its low-resolution (LR) counterpart, usually referred as single image super-resolution (SR) [12]. Image SR is used in various computer vision applications, ranging from security and surveillance imaging [13], medical imaging [14] to object recognition [8]. However, image SR is an ill-posed problem, since there exists multiple solutions for any LR input. To tackle such an inverse problem, numerous learning based methods have been proposed to learn mappings between LR and HR image pairs.

Recently, deep convolutional neural network (CNN) based methods [1–11, 15–17] have achieved significant improvements over conventional SR methods. Among them, Dong et al. [18] proposed SRCNN by firstly introducing a three-layer CNN for image SR. Kim et al. increased the network depth to 20 in

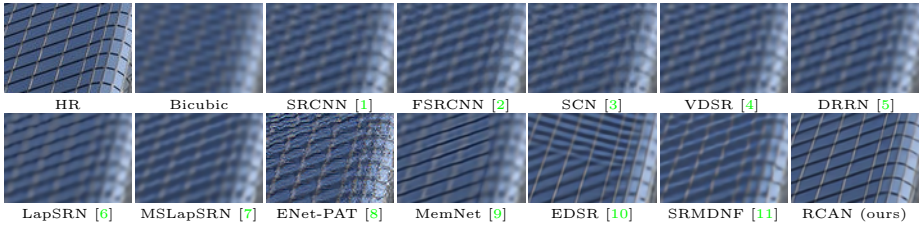


Fig. 1. Visual results with Bicubic (BI) degradation (4×) on "img_074" from Urban100

VDSR [4] and DRCN [19], achieving notable improvements over SRCNN. Network depth was demonstrated to be of central importance for many visual recognition tasks, especially when He at al. [20] proposed residual net (ResNet), which reaches 1,000 layers with residual blocks. Such effective residual learning strategy was then introduced in many other CNN-based image SR methods [5, 8–10, 21]. Lim et al. [10] built a very wide network EDSR and a very deep one MDSR (about 165 layers) by using simplified residual blocks. The great improvements on performance of EDSR and MDSR indicate that the depth of representation is of crucial importance for image SR. However, to the best of our knowledge, simply stacking residual blocks to construct deeper networks can hardly obtain better improvements. Whether deeper networks can further contribute to image SR and how to construct very deep trainable networks remains to be explored.

On the other hand, most recent CNN-based methods [1–11] treat channel-wise features equally, which lacks flexibility in dealing with different types of information (e.g., low- and high-frequency information). Image SR can be viewed as a process, where we try to recover as more high-frequency information as possible. The LR images contain most low-frequency information, which can directly forwarded to the final HR outputs and don't need too much computation. While, the leading CNN-based methods (e.g., EDSR [10]) would extract features from the original LR inputs and treat each channel-wise feature equally. Such process would wastes unnecessary computations for abundant low-frequency features, lacks discriminative learning ability across feature channels, and finally hinders the representational power of deep networks.

To practically resolve these problems, we propose a residual channel attention network (RCAN) to obtain very deep trainable network and adaptively learn more useful channel-wise features simultaneously. To ease the training of very deep networks (e.g., over 400 layers), we propose residual in residual (RIR) structure, where the residual group (RG) serves as the basic module and long skip connection (LSC) allows residual learning in a coarse level. In each RG module, we stack several simplified residual block [10] with short skip connection (SSC). The long and short skip connection as well as the short-cut in residual block allow abundant low-frequency information to be bypassed through these identity-based skip connections, which can ease the flow of information. To make a further step, we propose channel attention (CA) mechanism to adaptively rescale each channel-wise feature by modeling the interdependencies across feature channels. Such CA mechanism allows our proposed network to concentrate on more useful

channels and enhance discriminative learning ability. As shown in Figure 1, our RCAN achieves better visual SR result compared with state-of-the-art methods.

Overall, our contributions are three-fold: (1) We propose the very deep residual channel attention networks (RCAN) for highly accurate image SR. Our RCAN can reach much deeper than previous CNN-based methods and obtains much better SR performance. (2) We propose residual in residual (RIR) structure to construct very deep trainable networks. The long and short skip connections in RIR help to bypass abundant low-frequency information and make the main network learn more effective information. (3) We propose channel attention (CA) mechanism to adaptively rescale features by considering interdependencies among feature channels. Such CA mechanism further improves the representational ability of the network.

2 Related Work

Numerous image SR methods have been studied in the computer vision community [1–11, 22]. Attention mechanism is popular in high-level vision tasks, but is seldom investigated in low-level vision applications [23]. Due to space limitation, here we focus on works related to CNN-based methods and attention mechanism.

Deep CNN for SR. The pioneer work was done by Dong et al. [18], who proposed SRCNN for image SR and achieved superior performance against previous works. By introducing residual learning to ease the training difficulty, Kim et al. proposed VDSR [4] and DRCN [19] with 20 layers and achieved significant improvement in accuracy. Tai et al. later introduced recursive blocks in DRRN [5] and memory block in MemNet [9]. These methods would have to first interpolate the LR inputs to the desired size, which inevitably loses some details and increases computation greatly.

Extracting features from the original LR inputs and upscaling spatial resolution at the network tail then became the main choice for deep architecture. A faster network structure FSRCNN [2] was proposed to accelerate the training and testing of SRCNN. Ledig et al. [21] introduced ResNet [20] to construct a deeper network, SRResNet, for image SR. They also proposed SRGAN with perceptual losses [24] and generative adversarial network (GAN) [25] for photo-realistic SR. Such GAN based model was then introduced in EnhanceNet [8], which combines automated texture synthesis and perceptual loss. Although SRGAN and Enhancenet can alleviate the blurring and oversmoothing artifacts to some degree, their predicted results may not be faithfully reconstructed and produce unpleasing artifacts. By removing unnecessary modules in conventional residual networks, Lim et al. [10] proposed EDSR and MDSR, which achieve significant improvement. However, most of these methods have limited network depth, which has demonstrated to be very important in visual recognition tasks [20] and can reach to about 1,000 layers. Simply stacking residual blocks in MDSR [10], very deep networks can hardly achieved improvements. Furthermore, most of these methods treat the channel-wise features equally, hindering better discriminative ability for different types of features.

Attention mechanism. Generally, attention can be viewed as a guidance to bias the allocation of available processing resources towards the most informative

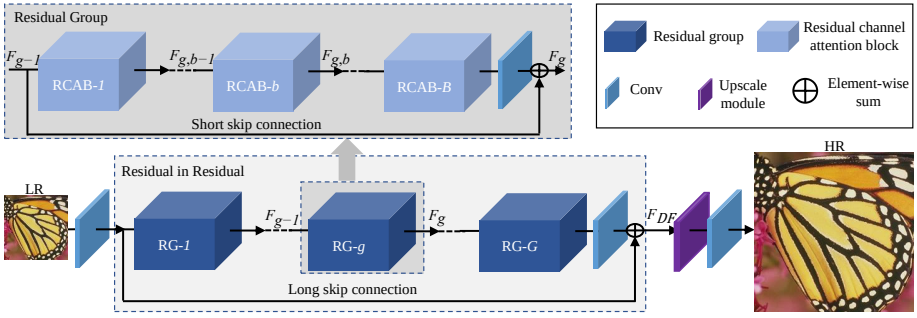


Fig. 2. Network architecture of our residual channel attention network (RCAN)

components of an input [23]. Recently, tentative works have been proposed to apply attention into deep neural networks [23, 26, 27], ranging from localization and understanding in images [28, 29] to sequence-based networks [30, 31]. It's usually combined with a gating function (e.g., sigmoid) to rescale the feature maps. Wang et al. [26] proposed residual attention network for image classification with a trunk-and-mask attention mechanism. Hu et al. [23] proposed squeeze-and-excitation (SE) block to model channel-wise relationships to obtain significant performance improvement for image classification. However, few works have been proposed to investigate the effect of attention for low-level vision tasks (e.g., image SR).

In image SR, high-frequency channel-wise features are more informative for HR reconstruction. If our network pays more attention to such channel-wise features, it should be promising to obtain improvements. To investigate such mechanism in very deep CNN, we propose very deep residual channel attention networks (RCAN), which we will detail in next section.

3 Residual Channel Attention Network (RCAN)

3.1 Network Architecture

As shown in Figure 2, our RCAN mainly consists four parts: shallow feature extraction, residual in residual (RIR) deep feature extraction, upscale module, and reconstruction part. Let's denote I_{LR} and I_{SR} as the input and output of RCAN. As investigated in [10, 21], we use only one convolutional layer (Conv) to extract the shallow feature F_0 from the LR input

$$F_0 = H_{SF}\left(I_{LR}\right), \tag{1}$$

where $H_{SF}\left(\cdot\right)$ denotes convolution operation. F_0 is then used for deep feature extraction with RIR module. So we can further have

$$F_{DF} = H_{RIR}\left(F_0\right), \tag{2}$$

where $H_{RIR}\left(\cdot\right)$ denotes our proposed very deep residual in residual structure, which contains G residual groups (RG). To the best of our knowledge, our proposed RIR achieves the largest depth so far and provides very large receptive

field size. So we treat its output as deep feature, which is then upscaled via a upscale module

$$F_{UP} = H_{UP}\left(F_{DF}\right), \tag{3}$$

where $H_{UP}(\cdot)$ and F_{UP} denote a upscale module and upscaled feature respectively.

There're several choices to serve as upscale modules, such as deconvolution layer (also known as transposed convolution) [2], nearest-neighbor upsampling + convolution [32], and ESPCN [33]. Such post-upscaling strategy has been demonstrated to be more efficient for both computation complexity and achieve higher performance than pre-upscaling SR methods (e.g., DRRN [5] and MemNet [9]). The upscaled feature is then reconstructed via one Conv layer

$$I_{SR} = H_{REC}\left(F_{UP}\right) = H_{RCAN}\left(I_{LR}\right), \tag{4}$$

where $H_{REC}(\cdot)$ and $H_{RCAN}(\cdot)$ denote the reconstruction layer and the function of our RCAN respectively.

Then RCAN is optimized with loss function. Several loss functions have been investigated, such as L_2 [1–5,8,9,11,16], L_1 [6,7,10,17], perceptual and adversarial losses [8,21]. To show the effectiveness of our RCAN, we choose to optimize same loss function as previous works (e.g., L_1 loss function). Given a training set $\left\{I^i_{LR}, I^i_{HR}\right\}_{i=1}^{N}$, which contains N LR inputs and their HR counterparts. The goal of training RCAN is to minimize the L_1 loss function

$$L\left(\Theta\right) = \frac{1}{N}\sum_{i=1}^{N}\left\|H_{RCAN}\left(I^i_{LR}\right) - I^i_{HR}\right\|_1, \tag{5}$$

where Θ denotes the parameter set of our network. The loss function is optimized by using stochastic gradient descent. More details of training would be shown in Section 4.1. As we choose the shallow feature extraction $H_{SF}(\cdot)$, upscaling module $H_{UP}(\cdot)$, and reconstruction part $H_{UP}(\cdot)$ as similar as previous works (e.g., EDSR [10] and RDN [17]), we pay more attention to our proposed RIR, CA, and the basic module RCAB.

3.2 Residual in Residual (RIR)

We now give more details about our proposed RIR structure (see Figure 2), which contains G residual groups (RG) and long skip connection (LSC). Each RG further contains B residual channel attention blocks (RCAB) with short skip connection (SSC). Such residual in residual structure allows to train very deep CNN (over 400 layers) for image SR with high performance.

It has been demonstrated that stacked residual blocks and LSC can be used to construct deep CNN in [10]. In visual recognition, residual blocks [20] can be stacked to achieve more than 1,000-layer trainable networks. However, in image SR, very deep network built in such way would suffer from training difficulty

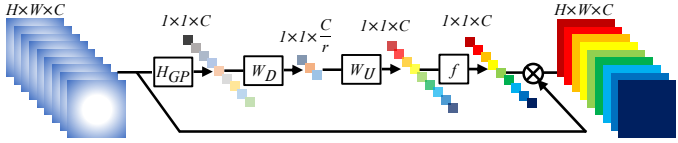


Fig. 3. Channel attention (CA). $\otimes$ denotes element-wise product

and can hardly achieve more performance gain. Inspired by previous works in SRRestNet [21] and EDSR [10], we proposed residual group (RG) as the basic module for deeper networks. A RG in the g-th group is formulated as

$$F_g = H_g\left(F_{g-1}\right) = H_g\left(H_{g-1}\left(\cdots H_1\left(F_0\right)\cdots\right)\right), \tag{6}$$

where H_g denotes the function of g-th RG. F_{g-1} and F_g are the input and output for g-th RG. We observe that simply stacking many RGs would fail to achieve better performance. To solve the problem, the long skip connection (LSC) is further introduced in RIR to stabilize the training of very deep network. LSC also makes better performance possible with residual learning via

$$F_{DF} = F_0 + W_{LSC}F_G = F_0 + W_{LSC}H_g\left(H_{g-1}\left(\cdots H_1\left(F_0\right)\cdots\right)\right), \tag{7}$$

where W_{LSC} is the weight set to the Conv layer at the tail of RIR. The bias term is omitted for simplicity. LSC can not only ease the flow of information across RGs, but only make it possible for RIR to learning residual information in a coarse level.

As discussed in Section 1, there are lots of abundant information in the LR inputs and features and the goal of SR network is to recover more useful information. The abundant low-frequency information can be bypassed through identity-based skip connection. To make a further step towards residual learning, we stack B residual channel attention blocks in each RG. The b-th residual channel attention block (RCAB) in g-th RG can be formulated as

$$F_{g,b} = H_{g,b}\left(F_{g,b-1}\right) = H_{g,b}\left(H_{g,b-1}\left(\cdots H_{g,1}\left(F_{g-1}\right)\cdots\right)\right), \tag{8}$$

where $F_{g,b-1}$ and $F_{g,b}$ are the input and output of the b-th RCAB in g-th RG. The corresponding function is denoted with $H_{g,b}$. To make the main network pay more attention to more informative features, a short skip connection (SSC) is introduced to obtain the block output via

$$F_g = F_{g-1} + W_g F_{g,B} = F_{g-1} + W_g H_{g,B}\left(H_{g,B-1}\left(\cdots H_{g,1}\left(F_{g-1}\right)\cdots\right)\right), \tag{9}$$

where W_g is the weight set to the Conv layer at the tail of g-th RG. The SSC further allows the main parts of network to learn residual information. With LSC and SSC, more abundant low-frequency information is easier bypassed in the training process. To make a further step towards more discriminative learning, we pay more attention to channel-wise feature rescaling with channel attention.

3.3 Channel Attention (CA)

Previous CNN-based SR methods treat LR channel-wise features equally, which is not flexible for the real cases. In order to make the network focus on more informative features, we exploit the interdependencies among feature channels, resulting in a channel attention (CA) mechanism (see Figure 3).

How to generate different attention for each channel-wise feature is a key step. Here we mainly have two concerns: First, information in the LR space has abundant low-frequency and valuable high-frequency components. The low-frequency parts seem to be more complanate. The high-frequency components would usually be regions, being full of edges, texture, and other details. On the other hand, each filter in Conv layer operates with a local receptive field. Consequently, the output after convolution is unable to exploit contextual information outside of the local region.

Based on these analyses, we take the channel-wise global spatial information into a channel descriptor by using global average pooling. As shown in Figure 3, let $X = [x_1, \cdots, x_c, \cdots, x_C]$ be an input, which has C feature maps with size of $H \times W$. The channel-wise statistic $z \in \mathbb{R}^C$ can be obtained by shrinking X through spatial dimensions $H \times W$. Then the c-th element of z is determined by

$$z_c = H_{GP}(x_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j), \tag{10}$$

where $x_c(i, j)$ is the value at position (i, j) of c-th feature x_c. $H_{GP}(\cdot)$ denotes the global pooling function. Such channel statistic can be viewed as a collection of the local descriptors, whose statistics contribute to express the whole image [23]. Except for global average pooling, more sophisticated aggregation techniques could also be introduced here.

To fully capture channel-wise dependencies from the aggregated information by global average pooling, we introduce a gating mechanism. As discussed in [23], the gating mechanism should meet two criteria: First, it must be able to learn nonlinear interactions between channels. Second, as multiple channel-wise features can be emphasized opposed to one-hot activation, it must learn a non-mututually-exclusive relationship. Here, we opt to exploit simple gating mechanism with sigmoid function

$$s = f(W_U \delta(W_D z)), \tag{11}$$

where $f(\cdot)$ and $\delta(\cdot)$ denote the sigmoid gating and ReLU [34] function, respectively. W_D is the weight set of a Conv layer, which acts as channel-downscaling with reduction ratio r. After being activated by ReLU, the low-dimension signal is then increased with ratio r by a channel-upscaling layer, whose weight set is W_U. Then we obtain the final channel statistics s, which is used to rescale the input x_c

$$\widehat{x}_c = s_c \cdot x_c, \tag{12}$$

where s_c and x_c are the scaling factor and feature map in the c-th channel. With channel attention, the residual component in the RCAB is adaptively rescaled.

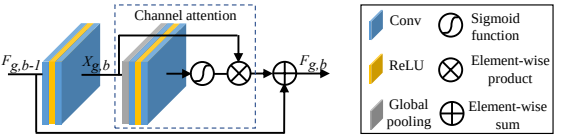


Fig. 4. Residual channel attention block (RCAB)

3.4 Residual Channel Attention Block (RCAB)

As discussed above, residual groups and long skip connection allow the main parts of network to focus on more informative components of the LR features. Channel attention extracts the channel statistic among channels to further enhance the discriminative ability of the network.

At the same time, inspired by the success of residual blocks (RB) in [10], we integrate CA into RB and propose residual channel attention block (RCAB) (see Figure 4). For the b-th RB in g-th RG, we have

$$F_{g,b} = F_{g,b-1} + R_{g,b}\left(X_{g,b}\right) \cdot X_{g,b}, \tag{13}$$

where $R_{g,b}$ denotes the function of channel attention. $F_{g,b}$ and $F_{g,b-1}$ are the input and output of RCAB, which learns the residual $X_{g,b}$ from the input. The residual component is mainly obtained by two stacked Conv layers

$$X_{g,b} = W^2_{g,b}\delta\left(W^1_{g,b}F_{g,b-1}\right), \tag{14}$$

where $W^1_{g,b}$ and $W^2_{g,b}$ are weight sets the two stacked Conv layers in RCAB.

We further show the relationships between our proposed RCAB and residual block (RB) in [10]. We find that the RBs used in MDSR and EDSR [10] can be viewed as special cases of our RCAB. For RB in MDSR, there is no rescaling operation. It is the same as RCAB, where we set $R_{g,b}\left(\cdot\right)$ as constant 1. For RB with constant rescaling (e.g., 0.1) in EDSR, it is the same as RCAB with $R_{g,b}\left(\cdot\right)$ set to be 0.1. Although the channel-wise feature rescaling is introduced to train a very wide network, the interdependencies among channels are not considered in EDSR. In these cases, the CA is not considered.

Based on residual channel attention block (RCAB) and RIR structure, we construct a very deep RCAN for highly accurate image SR and achieve notable performance improvements over previous leading methods. More discussions about the effects of each proposed component are shown in Section 4.2.

3.5 Implementation Details

Now we specify the implementation details of our proposed RCAN. We set RG number as G=10 in the RIR structure. In each RG, we set RCAB number as 20. We set 3×3 as the size of all Conv layers except for that in the channel-downscaling and channel-upscaling, whose kernel size is 1×1. For Conv layers with kernel size 3×3, zero-padding strategy is used to keep size fixed. Conv layers in shallow feature extraction and RIR structure have C=64 filters, except for that in the channel-downscaling. Conv layer in channel-downscaling has $\frac{C}{r}$=4 filters, where the reduction ratio r is set as 16. For upscaling module $H_{UP}\left(\cdot\right)$, we follow [10,17,33] and use ESPCNN [33] to upscale the coarse resolution features to fine ones. The final Conv layer has 3 filters, as we output color images. While, our network can also process gray images.

Table 1. Investigations of RIR (including LSC and SSC) and CA. We observe the best PSNR (dB) values on Set5 (2×) in 5×10^4 iterations

Residual in Residual (RIR)	LSC	✗	✓	✗	✓	✗	✓	✗	✓
	SSC	✗	✗	✓	✓	✗	✗	✓	✓
Channel attention (CA)		✗	✗	✗	✗	✓	✓	✓	✓
PSNR on Set5 (2×)		37.45	37.77	37.81	37.87	37.52	37.85	37.86	37.90

4 Experiments

4.1 Settings

We clarify the experimental settings about datasets, degradation models, evaluation metric, and training settings.

Datasets and degradation models. Following [10,11,17,35], we use 800 training images from DIV2K dataset [35] as training set. For testing, we use five standard benchmark datasets: Set5 [36], Set14 [37], B100 [38], Urban100 [22], and Manga109 [39]. We conduct experiments with Bicubic (BI) and blur-downscale (BD) degradation models [11,15,17].

Evaluation metrics. The SR results are evaluated with PSNR and SSIM [40] on Y channel (i.e., luminance) of transformed YCbCr space. We also provide performance (e.g., top-1 and top-5 recognition errors) comparisons on object recognition by several leading SR methods.

Training settings. Data augmentation is performed on the 800 training images, which are randomly rotated by 90°, 180°, 270° and flipped horizontally. In each training batch, 16 LR color patches with the size of 48×48 are extracted as inputs. Our model is trained by ADAM optimizor [41] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The initial leaning rate is set to 10^{-4} and then decreases to half every 2×10^5 iterations of back-propagation. We use PyTorch [42] to implement our models with a Titan Xp GPU.[1]

4.2 Effects of RIR and CA

We study the effects of residual in residual (RIR) and channel attention (CA).

Residual in residual (RIR). To demonstrate the effect of our proposed residual in residual structure, we remove long skip connection (LSC) or/and short skip connection (SSC) from very deep networks. Specifically, we set the number of residual block as 200, namely 10 residual groups, resulting in very deep networks with over 400 Conv layers. In Table 1, when both LSC and SSC are removed, the PSNR value on Set5 (×2) is relatively low, no matter channel attention (CA) is used or not. For example, in the first column, the PSNR is 37.45 dB. After adding RIR, the performance reaches 37.87 dB. When CA is added, the performance can be improved from 37.52 dB to 37.90 dB by using RIR. This indicates that simply stacking residual blocks is not applicable to achieve very deep and powerful networks for image SR. The performance would increase with LSC or SSC and can obtain better results by using both of them. These comparisons show that LSC and SSC are essential for very deep networks. They also demonstrate the effectiveness of our proposed residual in residual (RIR) structure for very deep networks.

[1] The RCAN source code is available at https://github.com/yulunzhang/RCAN.

Channel attention (CA). We further show the effect of channel attention (CA) based on the observations and discussions above. When we compare the results of first 4 columns and last 4 columns, we find that networks with CA would perform better than those without CA. Benefitting from very large network depth, the very deep trainable networks can achieve a very high performance. It's hard to obtain further improvements from such deep networks, but we obtain improvements with CA. Even without RIR, CA can improve the performance from 37.45 dB to 37.52 dB. These comparisons firmly demonstrate the effectiveness of CA and indicate adaptive attentions to channel-wise features really improves the performance.

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

Method	Scale	Set5		Set14		B100		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	0.9339
SRCNN [1]	×2	36.66	0.9542	32.45	0.9067	31.36	0.8879	29.50	0.8946	35.60	0.9663
FSRCNN [2]	×2	37.05	0.9560	32.66	0.9090	31.53	0.8920	29.88	0.9020	36.67	0.9710
VDSR [4]	×2	37.53	0.9590	33.05	0.9130	31.90	0.8960	30.77	0.9140	37.22	0.9750
LapSRN [6]	×2	37.52	0.9591	33.08	0.9130	31.08	0.8950	30.41	0.9101	37.27	0.9740
MemNet [9]	×2	37.78	0.9597	33.28	0.9142	32.08	0.8978	31.31	0.9195	37.72	0.9740
EDSR [10]	×2	38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351	39.10	0.9773
SRMDNF [11]	×2	37.79	0.9601	33.32	0.9159	32.05	0.8985	31.33	0.9204	38.07	0.9761
D-DBPN [16]	×2	38.09	0.9600	33.85	0.9190	32.27	0.9000	32.55	0.9324	38.89	0.9775
RDN [17]	×2	38.24	0.9614	34.01	0.9212	32.34	0.9017	32.89	0.9353	39.18	0.9780
RCAN (ours)	×2	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.34	0.9384	39.44	0.9786
RCAN+ (ours)	×2	**38.33**	**0.9617**	**34.23**	**0.9225**	**32.46**	**0.9031**	**33.54**	**0.9399**	**39.61**	**0.9788**
Bicubic	×3	30.39	0.8682	27.55	0.7742	27.21	0.7385	24.46	0.7349	26.95	0.8556
SRCNN [1]	×3	32.75	0.9090	29.30	0.8215	28.41	0.7863	26.24	0.7989	30.48	0.9117
FSRCNN [2]	×3	33.18	0.9140	29.37	0.8240	28.53	0.7910	26.43	0.8080	31.10	0.9210
VDSR [4]	×3	33.67	0.9210	29.78	0.8320	28.83	0.7990	27.14	0.8290	32.01	0.9340
LapSRN [6]	×3	33.82	0.9227	29.87	0.8320	28.82	0.7980	27.07	0.8280	32.21	0.9350
MemNet [9]	×3	34.09	0.9248	30.00	0.8350	28.96	0.8001	27.56	0.8376	32.51	0.9369
EDSR [10]	×3	34.65	0.9280	30.52	0.8462	29.25	0.8093	28.80	0.8653	34.17	0.9476
SRMDNF [11]	×3	34.12	0.9254	30.04	0.8382	28.97	0.8025	27.57	0.8398	33.00	0.9403
RDN [17]	×3	34.71	0.9296	30.57	0.8468	29.26	0.8093	28.80	0.8653	34.13	0.9484
RCAN (ours)	×3	34.74	0.9299	30.65	0.8482	29.32	0.8111	29.09	0.8702	34.44	0.9499
RCAN+ (ours)	×3	**34.85**	**0.9305**	**30.76**	**0.8494**	**29.39**	**0.8122**	**29.31**	**0.8736**	**34.76**	**0.9513**
Bicubic	×4	28.42	0.8104	26.00	0.7027	25.96	0.6675	23.14	0.6577	24.89	0.7866
SRCNN [1]	×4	30.48	0.8628	27.50	0.7513	26.90	0.7101	24.52	0.7221	27.58	0.8555
FSRCNN [2]	×4	30.72	0.8660	27.61	0.7550	26.98	0.7150	24.62	0.7280	27.90	0.8610
VDSR [4]	×4	31.35	0.8830	28.02	0.7680	27.29	0.0726	25.18	0.7540	28.83	0.8870
LapSRN [6]	×4	31.54	0.8850	28.19	0.7720	27.32	0.7270	25.21	0.7560	29.09	0.8900
MemNet [9]	×4	31.74	0.8893	28.26	0.7723	27.40	0.7281	25.50	0.7630	29.42	0.8942
EDSR [10]	×4	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033	31.02	0.9148
SRMDNF [11]	×4	31.96	0.8925	28.35	0.7787	27.49	0.7337	25.68	0.7731	30.09	0.9024
D-DBPN [16]	×4	32.47	0.8980	28.82	0.7860	27.72	0.7400	26.38	0.7946	30.91	0.9137
RDN [17]	×4	32.47	0.8990	28.81	0.7871	27.72	0.7419	26.61	0.8028	31.00	0.9151
RCAN (ours)	×4	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087	31.22	0.9173
RCAN+ (ours)	×4	**32.73**	**0.9013**	**28.98**	**0.7910**	**27.85**	**0.7455**	**27.10**	**0.8142**	**31.65**	**0.9208**
Bicubic	×8	24.40	0.6580	23.10	0.5660	23.67	0.5480	20.74	0.5160	21.47	0.6500
SRCNN [1]	×8	25.33	0.6900	23.76	0.5910	24.13	0.5660	21.29	0.5440	22.46	0.6950
FSRCNN [2]	×8	20.13	0.5520	19.75	0.4820	24.21	0.5680	21.32	0.5380	22.39	0.6730
SCN [3]	×8	25.59	0.7071	24.02	0.6028	24.30	0.5698	21.52	0.5571	22.68	0.6963
VDSR [4]	×8	25.93	0.7240	24.26	0.6140	24.49	0.5830	21.70	0.5710	23.16	0.7250
LapSRN [6]	×8	26.15	0.7380	24.35	0.6200	24.54	0.5860	21.81	0.5810	23.39	0.7350
MemNet [9]	×8	26.16	0.7414	24.38	0.6199	24.58	0.5842	21.89	0.5825	23.56	0.7387
MSLapSRN [7]	×8	26.34	0.7558	24.57	0.6273	24.65	0.5895	22.06	0.5963	23.90	0.7564
EDSR [10]	×8	26.96	0.7762	24.91	0.6420	24.81	0.5985	22.51	0.6221	24.69	0.7841
D-DBPN [16]	×8	27.21	0.7840	25.13	0.6480	24.88	0.6010	22.73	0.6312	25.14	0.7987
RCAN (ours)	×8	27.31	0.7878	25.23	0.6511	24.98	0.6058	23.00	0.6452	25.24	0.8029
RCAN+ (ours)	×8	**27.47**	**0.7913**	**25.40**	**0.6553**	**25.05**	**0.6077**	**23.22**	**0.6524**	**25.58**	**0.8092**

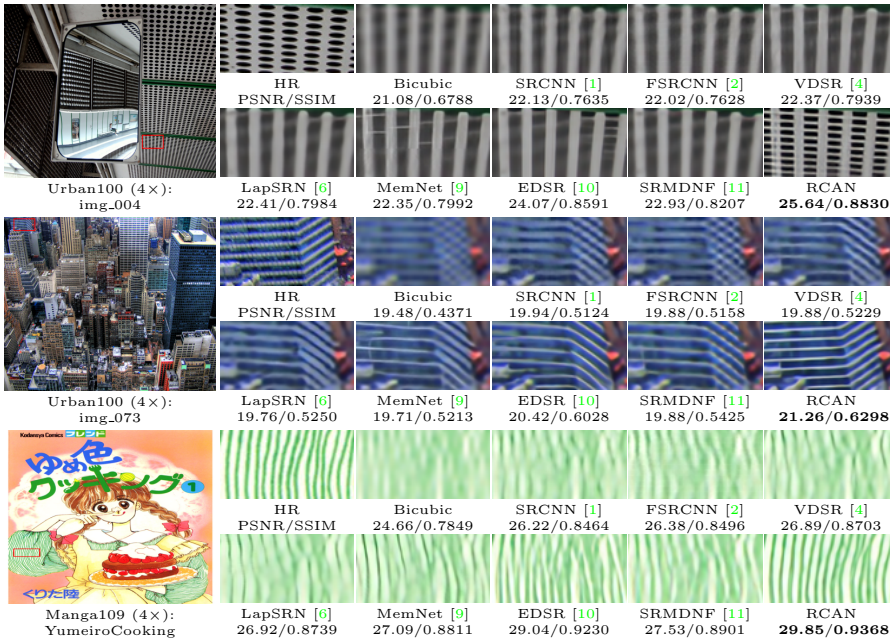


Fig. 5. Visual comparison for 4× SR with BI model on Urban100 and Manga109 datasets. The best results are **highlighted**

4.3 Results with Bicubic (BI) Degradation Model

We compare our method with 11 state-of-the-art methods: SRCNN [1], FSRCNN [2], SCN [3], VDSR [4], LapSRN [6], MemNet [9], EDSR [10], SRMDNF [11], D-DBPN [16], and RDN [17]. Similar to [10, 17, 43], we also introduce self-ensemble strategy to further improve our RCAN and denote the self-ensembled one as RCAN+. More comparisons are provided in supplementary material.

Quantitative results by PSNR/SSIM. Table 2 shows quantitative comparisons for ×2, ×3, ×4, and ×8 SR. The results of D-DBPN [16] are cited from their paper. When compared with all previous methods, our RCAN+ performs the best on all the datasets with all scaling factors. Even without self-ensemble, our RCAN also outperforms other compared methods.

On the other hand, when the scaling factor become larger (e.g., 8), the gains of our RCAN over EDSR also becomes larger. For Urban100 and Manga109, the PSNR gains of RCAN over EDSR are 0.49 dB and 0.55 dB. EDSR has much larger number of parameters (43 M) than ours (16 M), but our RCAN obtains much better performance. Instead of constantly rescaling the features in EDSR, our RCAN adaptively rescales features with channel attention (CA). CA allows our network to further focus on more informative features. This observation indicates that very large network depth and CA improve the performance.

Visual results. In Figure 5, we show visual comparisons on scale ×4. For image "img_004", we observe that most of the compared methods cannot recover the lattices and would suffer from blurring artifacts. In contrast, our RCAN can alleviate the blurring artifacts better and recover more details. For image

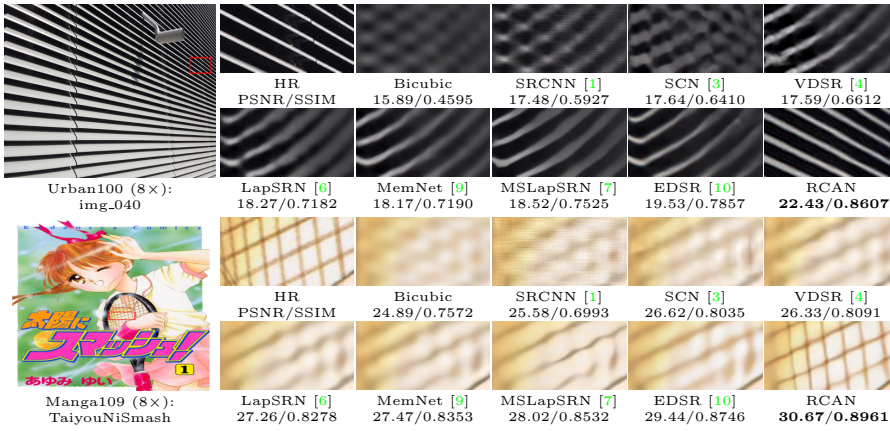


Fig. 6. Visual comparison for 8× SR with BI model on Urban100 and Manga109 datasets. The best results are **highlighted**

"img_073", most of the compared methods produce blurring artifacts along the horizontal lines. What's worse, for the right parts of the cropped images, FSRCNN cannot recover lines. Other methods would generate some lines with wrong directions. Only our RCAN produces more faithful results. For image "YumeiroCooking", the cropped part is full of textures. As we can see, all the compared methods suffer from heavy blurring artifacts, failing to recover more details. While, our RCAN can recover them obviously, being more faithful to the ground truth. Such obvious comparisons demonstrate that networks with more powerful representational ability can extract more sophisticated features from the LR space.

To further illustrate the analyses above, we show visual comparisons for 8× SR in Figure 6. For image "img_040", due to very large scaling factor, the result by Bicubic would lose the structures and produce different structures. This wrong pre-scaling result would also lead some state-of-the-art methods (e.g., SRCNN, VDSR, and MemNet) to generate totally wrong structures. Even starting from the original LR input, other methods cannot recover the right structure either. While, our RCAN can recover them correctly. For smaller details, like the net in image "TaiyouNiSmash", the tiny lines can be lost in the LR image. When the scaling factor is very large (e.g., 8), LR images contain very limited information for SR. Losing most high-frequency information makes it very difficult for SR methods to reconstruct informative results. Most of compared methods cannot achieve this goal and produce serious blurring artifacts. However, our RCAN can obtain more useful information and produce finer results.

As we have discussed above, in BI degradation model, the reconstruction of high-frequency information is very important and difficult, especially with large scaling factor (e.g., 8). Our proposed RIR structure makes the main network learn residual information. Channel attention (CA) is further used to enhance the representational ability of the network by adaptively rescaling channel-wise features.

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

Method	Scale	Set5		Set14		B100		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 [44]	×3	32.21	0.9001	28.89	0.8105	28.13	0.7740	25.84	0.7856	29.64	0.9003
SRCNN [1]	×3	32.05	0.8944	28.80	0.8074	28.13	0.7736	25.70	0.7770	29.47	0.8924
FSRCNN [2]	×3	26.23	0.8124	24.44	0.7106	24.86	0.6832	22.04	0.6745	23.04	0.7927
VDSR [4]	×3	33.25	0.9150	29.46	0.8244	28.57	0.7893	26.61	0.8136	31.06	0.9234
IRCNN [15]	×3	33.38	0.9182	29.63	0.8281	28.65	0.7922	26.77	0.8154	31.15	0.9245
SRMDNF [11]	×3	34.01	0.9242	30.11	0.8364	28.98	0.8009	27.50	0.8370	32.97	0.9391
RDN [17]	×3	34.58	0.9280	30.53	0.8447	29.23	0.8079	28.46	0.8582	33.97	0.9465
RCAN (ours)	×3	34.70	0.9288	30.63	0.8462	29.32	0.8093	28.81	0.8647	34.38	0.9483
RCAN+ (ours)	×3	**34.83**	**0.9296**	**30.76**	**0.8479**	**29.39**	**0.8106**	**29.04**	**0.8682**	**34.76**	**0.9502**

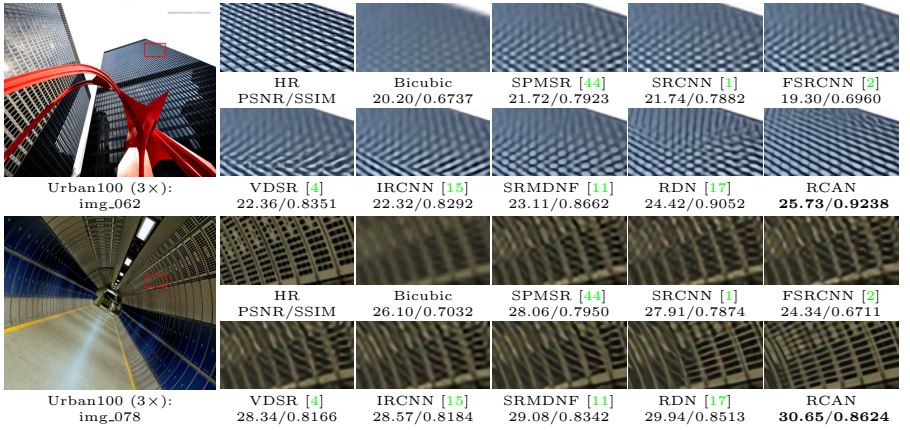


Fig. 7. Visual comparison for 3× SR with BD model on Urban100 dataset. The best results are **highlighted**

4.4 Results with Blur-downscale (BD) Degradation Model

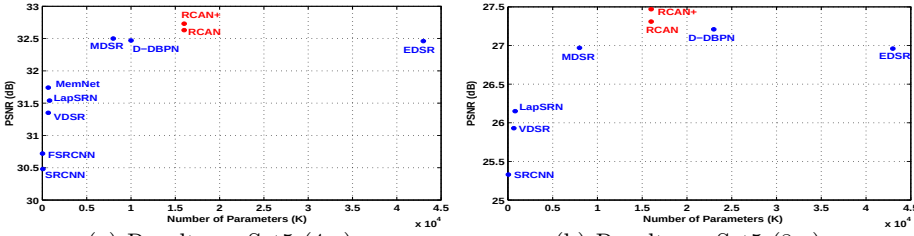
We further apply our method to super-resolve images with blur-down (BD) degradation model, which is also commonly used recently [11, 15, 17].

Quantitative results by PSNR/SSIM. Here, we compare 3× SR results with 7 state-of-the-art methods: SPMSR [44], SRCNN [1], FSRCNN [2], VDSR [4], IRCNN [15], SRMDNF [11], and RDN [17]. As shown in Table 3, RDN has achieved very high performance on each dataset. While, our RCAN can obtain notable gains over RDN. Using self-ensemble, RCAN+ achieves even better results. Compared with fully using hierarchical features in RDN, a much deeper network with channel attention in RCAN achieves better performance. This comparison also indicates that there has promising potential to investigate much deeper networks for image SR.

Visual Results. We also show visual comparisons in Figure 7. For challenging details in images "img_062" and "img_078", most methods suffer from heavy blurring artifacts. RDN alleviates it to some degree and can recover more details. In contrast, our RCAN obtains much better results by recovering more informative components. These comparisons indicate that very deep channel attention guided network would alleviate the blurring artifacts. It also demonstrates the strong ability of RCAN for BD degradation model.

Table 4. ResNet object recognition performance. The best results are **highlighted**

Evaluation	Bicubic	DRCN [19]	FSRCNN [2]	PSyCo [45]	ENet-E [8]	RCAN	Baseline
Top-1 error	0.506	0.477	0.437	0.454	0.449	**0.393**	0.260
Top-5 error	0.266	0.242	0.196	0.224	0.214	**0.167**	0.072



(a) Results on Set5 (4×) (b) Results on Set5 (8×)

Fig. 8. Performance and number of parameters. Results are evaluated on Set5

4.5 Object Recognition Performance

Image SR also serves as pre-processing step for high-level visual tasks (e.g., object recognition). We evaluate the object recognition performance to further demonstrate the effectiveness of our RCAN.

Here we use the same settings as ENet [8]. We use ResNet-50 [20] as the evaluation model and use the first 1,000 images from ImageNet CLS-LOC validation dataset for evaluation. The original cropped 224×224 images are used for baseline and downscaled to 56×56 for SR methods. We use 4 stat-of-the-art methods (e.g., DRCN [19], FSRCNN [2], PSyCo [45], and ENet-E [8]) to upscale the LR images and then calculate their accuracies. As shown in Table 4, our RCAN achieves the lowest top-1 and top-5 errors. These comparisons further demonstrate the highly powerful representational ability of our RCAN.

4.6 Model Size Analyses

We show comparisons about model size and performance in Figure 8. Although our RCAN is the deepest network, it has less parameter number than that of EDSR and RDN. Our RCAN and RCAN+ achieve higher performance, having a better tradeoff between model size and performance. It also indicates that deeper networks may be easier to achieve better performance than wider networks.

5 Conclusions

We propose very deep residual channel attention networks (RCAN) for highly accurate image SR. Specifically, the residual in residual (RIR) structure allows RCAN to reach very large depth with LSC and SSC. Meanwhile, RIR allows abundant low-frequency information to be bypassed through multiple skip connections, making the main network focus on learning high-frequency information. Furthermore, to improve ability of the network, we propose channel attention (CA) mechanism to adaptively rescale channel-wise features by considering interdependencies among channels. Extensive experiments on SR with BI and BD models demonstrate the effectiveness of our proposed RCAN. RCAN also shows promissing results for object recognition.

Acknowledgements: This research is supported in part by the NSF IIS award 1651902, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Award W911NF-17-1-0367.

References

1. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. TPAMI (2016)
2. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: ECCV. (2016)
3. Wang, Z., Liu, D., Yang, J., Han, W., Huang, T.: Deep networks for image super-resolution with sparse prior. In: ICCV. (2015)
4. Kim, J., Kwon Lee, J., Mu Lee, K.: Accurate image super-resolution using very deep convolutional networks. In: CVPR. (2016)
5. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR. (2017)
6. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR. (2017)
7. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Fast and accurate image super-resolution with deep laplacian pyramid networks. arXiv:1710.01992 (2017)
8. Sajjadi, M.S., Schölkopf, B., Hirsch, M.: Enhancenet: Single image super-resolution through automated texture synthesis. In: ICCV. (2017)
9. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. In: ICCV. (2017)
10. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: CVPRW. (2017)
11. Zhang, K., Zuo, W., Zhang, L.: Learning a single convolutional super-resolution network for multiple degradations. In: CVPR. (2018)
12. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision. IJCV (2000)
13. Zou, W.W., Yuen, P.C.: Very low resolution face recognition problem. TIP (2012)
14. Shi, W., Caballero, J., Ledig, C., Zhuang, X., Bai, W., Bhatia, K., de Marvao, A.M.S.M., Dawes, T., ORegan, D., Rueckert, D.: Cardiac image super-resolution with global correspondence using multi-atlas patchmatch. In: MICCAI. (2013)
15. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. In: CVPR. (2017)
16. Haris, M., Shakhnarovich, G., Ukita, N.: Deep back-projection networks for super-resolution. In: CVPR. (2018)
17. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: CVPR. (2018)
18. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: ECCV. (2014)
19. Kim, J., Kwon Lee, J., Mu Lee, K.: Deeply-recursive convolutional network for image super-resolution. In: CVPR. (2016)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)
21. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR. (2017)
22. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR. (2015)
23. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. arXiv preprint arXiv:1709.01507 (2017)

24. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV. (2016)
25. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014)
26. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification. In: CVPR. (2017)
27. Li, K., Wu, Z., Peng, K.C., Ernst, J., Fu, Y.: Tell me where to look: Guided attention inference network. In: CVPR. (2018)
28. Cao, C., Liu, X., Yang, Y., Yu, Y., Wang, J., Wang, Z., Huang, Y., Wang, L., Huang, C., Xu, W., Ramanan, D., Huang, T.S.: Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In: ICCV. (2015)
29. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: NIPS. (2015)
30. Bluche, T.: Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. In: NIPS. (2016)
31. Miech, A., Laptev, I., Sivic, J.: Learnable pooling with context gating for video classification. arXiv preprint arXiv:1706.06905 (2017)
32. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. In: ICLR. (2017)
33. 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)
34. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: ICML. (2010)
35. Timofte, R., Agustsson, E., Van Gool, L., Yang, M.H., Zhang, L., Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M., et al.: Ntire 2017 challenge on single image super-resolution: Methods and results. In: CVPRW. (2017)
36. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: BMVC. (2012)
37. Zeyde, R., Elad, M., Protter, M.: On single image scale-up using sparse-representations. In: Proc. 7th Int. Conf. Curves Surf. (2010)
38. 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)
39. 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 (2017)
40. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP (2004)
41. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR. (2014)
42. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. (2017)
43. Timofte, R., Rothe, R., Van Gool, L.: Seven ways to improve example-based single image super resolution. In: CVPR. (2016)
44. Peleg, T., Elad, M.: A statistical prediction model based on sparse representations for single image super-resolution. TIP (2014)
45. Pérez-Pellitero, E., Salvador, J., Ruiz-Hidalgo, J., Rosenhahn, B.: Psyco: Manifold span reduction for super resolution. In: CVPR. (2016)