ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200416

Mid-Weight Image Super-Resolution with Bypass Connection Attention Network

Hao Shen and **Zhong-Qiu Zhao*** [1, 2]

Abstract. Deeper networks have limited improvements for image super-resolution (SR), and are much more difficult to train. The main reason is that these networks consist of many stacked building blocks which can produce many redundant features. Besides, most of SR methods neglect the fact that different features contain various types of information with varying degrees of contributions to image reconstruction, and thus lack sufficient representational capability. Taking these issues into account, we propose a mid-weight bypass connection attention network (BCAN) with more powerful representational capability but fewer parameters. In detail, we design a novel bypass connection attention module (BCAM), which consists of several bypass connection attention blocks (BCABs), enhancing high contribution information and suppressing redundant information. Further, we embed a mixed residual attention unit (MRAU) in each BCAB, which is composed of a channel attention unit and a spatial attention unit. After obtaining all hierarchical features, we propose an adaptive feature fusion module (AFFM), which can effectively combine hierarchical features based on different contributions of each BCAM. Experiments on benchmark datasets with various degradation models show that our BCAN can achieve better performance than existing state-of-the-art methods.

1 Introduction

Single image super-resolution (SISR), which refers to the process of recovering a high-resolution (HR) image from its low-resolution (LR) image, is used in various real-world computer vision tasks such as medical imaging [15], surveillance and security [27]. It can not only improve image perpetual quality but also serve as an auxiliary task for other computer vision applications [35]. However, SISR is very challenging and ill-posed since there are multiple HR images corresponding to a single LR image. In spite of such difficulty, numerous learning-based methods have been proposed to learn maps between LR and HR images.

Recently, convolutional neural network (CNN) based methods [4, 8, 22, 38] have been actively explored and achieve state-of-the-art performance on various datasets of SR. Among them, SRCNN [4] firstly introduced a three-layer CNN for image SR, with a significant improvement over conventional methods. VDSR [17] increased the depth of the network by using skip connections to ease the difficulty of training deep network and achieved a notable improvement over SRCNN. Inspired by ResNet [10], the strategy of residual learning was adopted by deep CNN-based image SR methods to build deeper networks. Furthermore, dense connections proposed in DenseNet [13] allowed direct connections between any two layers within the same dense block, providing an effective way to combine low-level and high-level features to boost the reconstruction performance. The SRDenseNet [30] and RDN [38] employed dense block and residual dense block to extract hierarchical features which are beneficial to feature reconstruction. However, these networks have some drawbacks: (1) the extreme connectivity pattern not only hinders their scalability to large width or high depth but also produces redundant computation; (2) these networks ignore the fact that the contribution of hierarchical features is different. Hence, how to obtain effective hierarchical features by utilizing fewer parameters and adaptively fuse them is worth exploring.

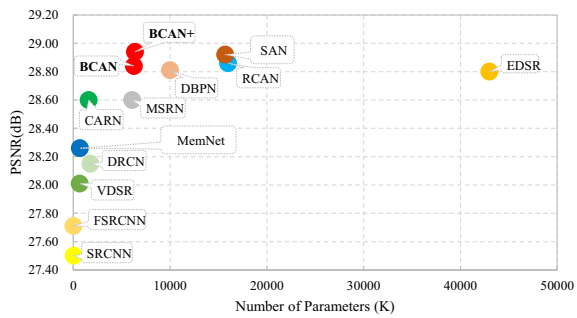

Figure 1. Performance vs number of parameters. The results are evaluated on Set14 with scaling factor $\times 4$.

In addition, most of the CNN-based methods inherently treat all types of features equally, so they can not effectively distinguish the detailed characteristics of images (e.g., low-frequency and high-frequency information). In order to get more discriminative features, RCAN [37] and SAN [8] adopted the channel attention mechanism to design a very deep network and pushed the state-of-the-art performance of image SR forward. However, the information contained in feature maps is also diverse over spatial positions. For example, the edge or texture regions usually contain more high-frequency information while the smooth areas have more low-frequency information. The high-frequency information needs to be extracted by complex filters, the low-frequency information needs to be extracted by relatively less detailed filters. Most of the previous networks [8, 12, 37] only focus on the relation of various channels without considering the positional relation (spatial relation) of each channel. How to utilize both channel attention and spatial attention and effectively combine them in SR networks still is an open issue.

To solve these problems, we propose a mid-weight bypass connection attention network (BCAN). To change extreme dense con-

[1] Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology.
[2] School of Computer Science and Information Engineering, Hefei University of Technology.
Emails: haoshen@mail.hfut.edu.cn, z.zhao@hfut.edu.cn.
* Corresponding author: Zhong-Qiu Zhao.

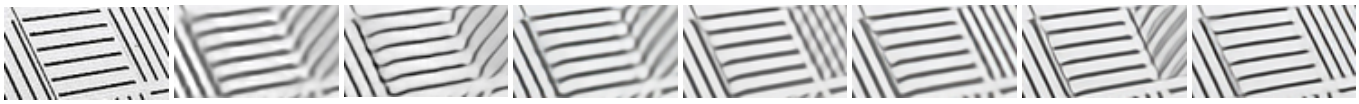

Figure 2. The comparisons of our proposed BCAN with existing state-of-the-art methods on "img_092" from Urban100 ($\times 4$). Our proposed BCAN network generates more realistic visual result.

nection mode and obtain hierarchical features, a bypass connection attention module (BCAM) is developed as the basic building module for the BCAN. Each BCAM contains multiple bypass connection attention blocks (BCABs), extracting local shallow features and local deep features. All convolutional features in the current BCAB are concatenated, and these features will be fed into later BCABs to fully utilize all information. In order to improve the representational capability, we combine channel attention and spatial attention to construct a mixed residual attention unit (MRAU) and embed the MRAU into the tail of each BCAB. After extracting multi hierarchical features, we construct an adaptive feature fusion module (AFFM) that combines hierarchical features by the different contributions of each BCAM. As shown in Figure 2, our BCAN achieves better visual performance compared with other state-of-the-art SR methods.

In summary, the main contributions of this paper can be summarized as follows:

1. We propose a novel mid-weight bypass connection attention network (BCAN) for high-quality image SR with different degradation models. Extensive experiments on five public datasets demonstrate our BCAN has better SR performance using fewer parameters compared with other state-of-the-art methods.
2. We propose a bypass connection attention module (BCAM) to extract the hierarchical features, which serves as a basic module for the whole network. Besides, we construct an adaptive feature fusion module (AFFM) to fuse these hierarchical features effectively.
3. We combine channel attention (CA) and spatial attention (SA) to construct a mixed residual attention unit (MRAU), which is embedded in the BCAM and improves the representational capability of the network.

2 Related Work

2.1 Deep Learning Based Image Super-Resolution

In recent years, deep learning-based SR methods have been actively explored and achieve great progress. Among them, SRCNN [4] firstly applied CNN to image SR and improved SR performance over traditional methods. The baseline was further improved by increasing the depth of the network in VDSR [17]. To reduce network parameters, DRCN [16] and DRRN [33] adopted recursive learning to achieve parameter sharing. However, these methods, which adopt the strategy of pre-resample, not only increase the computation complexity of the network but also make the image losing some detailed information. FSRCNN [5] and ESPCN [28] added a deconvolution layer and sub-pixel convolution layer to the tail of the network to obtain HR image respectively, which reduced the amount of computation. In the past few years, with the continuous improvement of computing power, deep CNNs [8, 22, 38] have been explored for image SR. However, these deep CNNs based SR methods not only need more training skills but also require more time consumption, which result in poor reproducibility.

2.2 Bypass Connections

Recently, skip connections [7, 13, 16, 22, 38, 39] have been applied to many image SR networks in order to ease the problem of gradient vanishing and information flow weakened. In VDSR [17], a skip connection was utilized to link the input and reconstruction layer, which constructed a 20 layer network and achieved competitive results. SRDenseNet [30] introduced dense skip connections to build a dense block, where each layer had direct connections to all subsequent layers. This dense block is used as the basis of the whole network and effectively boosts reconstruction performance. RDN [38] combined dense skip connections and local residual learning to build a residual dense block (RDB). Compared with dense block, RDB has a larger growth rate and utilizes local residual learning to extract richer hierarchical features. All these networks show that it is essential to build many skip connections to train a network. However, frequent connections in each Conv layer not only produce many redundant features, but also increase a large amount of computation. Therefore, we adopt the manner of bypass connections to construct a novel bypass connection attention block (BCAB), and use bypass connections to combine several BCABs to construct a concise bypass connection attention module (BCAM).

2.3 Attention Mechanism

The aim of attention mechanism is to recalibrate the extracted feature maps so that more discriminative and effective features are obtained. Many computer vision tasks, such as image generation [23], object detection [40], image captioning [6] and visual question answering [32], have employed attention mechanism in deep networks. A few recent SR methods also embed the attention mechanism to their networks. RCAN [37] employed channel attention (CA) to network and pushed the state-of-the-art performance of image SR forward. SAN [8] changed the method of pooling, adopting covariance pooling to obtain a novel second-order channel attention, and achieved better performance over RCAN. In this paper, considering that there are different types of information in inter-channel and intra-channel, we utilize both channel attention and spatial attention to obtain CA and SA maps respectively. Then, we fuse the CA and SA maps by the element-wise operation.

3 Proposed Method

3.1 Network Architecture

As shown in Figure 3, our proposed BCAN mainly contains four parts: shallow feature extraction net (SFENet), hierarchical feature extraction net (HFENet), upscale net (UpNet) and reconstruction net (RecNet). Similar to RDN [38], we use the strategy of residual learning (RL) to train our network. We denote I_{LR} and I_{SR} as the input and output of BCAN respectively.

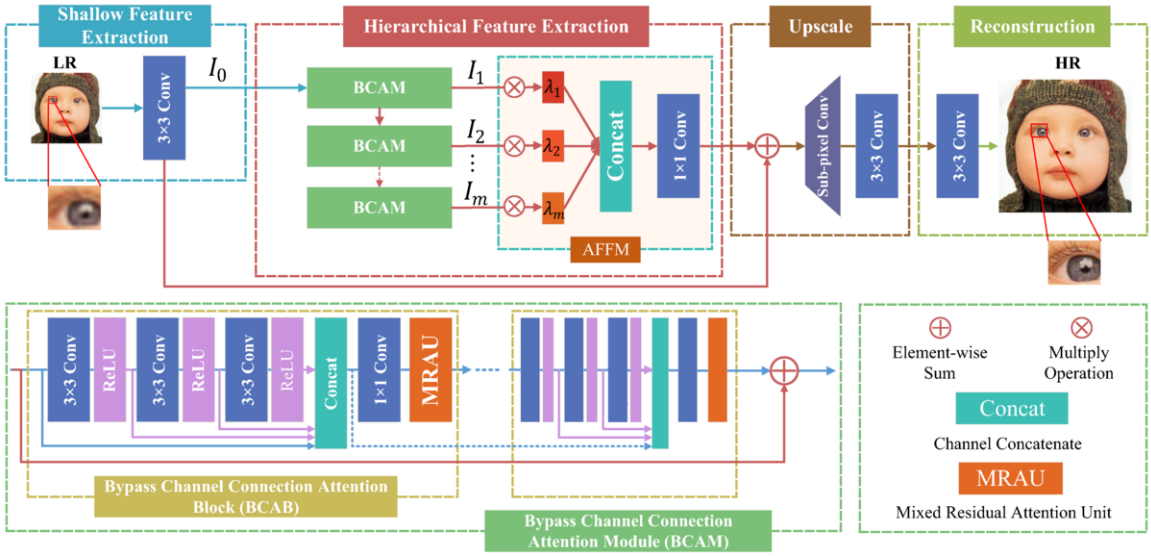

Figure 3. The architecture of the proposed bypass connection attention network (BCAN) ant its sub-modules.

In the SFENet, we utilize one convolutional (Conv) layer to extract the features from the I_{LR} image.

$$I_0 = H_{SFENet}(I_{LR}), \tag{1}$$

where $H_{SFENet}(\cdot)$ denotes the function of SFENet, I_0 denotes the output of SFENet and serves as the input of HFENet. The HFENet includes a sequence of bypass connection attention modules (BCAMs) and an adaptive feature fusion module (AFFM). The output I_m of the m-th BCAM can be obtained by

$$I_m = H_m(I_{m-1}) = H_m(H_{m-1}(\cdots H_1(I_0)\cdots)), \tag{2}$$

where $H_m(\cdot)$ denotes the function of the m-th BCAM. The HFENet totally contains M BCAMs and all features obtained by one BCAM then sequentially fed into next module. We can adjust the depth of the network by changing the numbers of the BCAM. More details about BCAM will be introduced in the next subsection. After extracting the hierarchical features with a set of BCAMs, we adopt AFFM to fuse all features. We define the output of HFENet as:

$$\begin{aligned} I_{HF} &= H_{HFENet}(I_0) \\ &= H_{AFFM}(I_1, I_2, \cdots, I_M) + I_0 \\ &= W_{AFF}[\lambda_1 I_1, \lambda_2 I_2, \cdots, \lambda_M I_M] + I_0, \end{aligned} \tag{3}$$

where $HFENet(\cdot)$ denotes the function of HFENet, $H_{AFFM}(\cdot)$ denotes the function of adaptive feature fusion module, λ_M is the adaptive weight factor for the M-th BCAM, $[\cdots]$ denotes channel concatenation operation, and W_{AFF} represents the weight sets of 1×1 Conv layer. In the AFFM, different from MSRN [20], we deploy a learnable weight factor after each BCAM, which can further explore model representational capacity. Eventually, we obtain the output I_{HF} and transmit the output into the UpNet for mapping transformation. We define the output of UpNet as:

$$I_{UP} = H_{UpNet}(I_{HF}), \tag{4}$$

where $H_{UpNet}(\cdot)$ denotes the function of UpNet. Specifically, the UpNet is composed of a sub-pixel Conv [28] layer followed by a Conv layer for converting LR features to HR features. Finally, the upscaled features are reconstructed via one Conv layer.

$$I_{SR} = H_{RecNet}(I_{UP}) = H_{BCAN}(I_{LR}), \tag{5}$$

where $H_{RecNet}(\cdot)$ and $H_{BCAN}(\cdot)$ denote the functions of RecNet and BCAN, respectively. I_{SR} represents the reconstructed image via BCAN.

Our BCAN is optimized by minimizing the difference between the super-resolved image I_{SR} and the corresponding ground-truth I_{HR}. As done in previous works [20,22,37,38], we adopt L_1 loss function to measure the difference. The loss function can be defined as:

$$L(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \|H_{BCAN}(I_{LR}^i) - I_{HR}^i\|_1, \tag{6}$$

where Θ represents the total parameters set of network. I_{LR}^i and I_{HR}^i denote the i-th LR and HR patch pair of total N patch pairs.

3.2 Bypass Connection Attention Module (BCAM)

The BCAM contains multiple bypass connection attention blocks (BCABs) as shown in Figure 3. Each BCAB contains a sequence of Conv layers and a mixed residual attention unit (MRAU).

Different from previous work [30, 38], each Conv layer in each BCAB is not connected to all subsequent Conv layers, which enables the network more concise and obtain more hierarchical features at the same time. Then we adopt a 1×1 Conv layer to control the output features. The specific operation is

$$\begin{aligned} X_{m,b} =& W_{m,b}[I_{m,1}^1, \cdots, I_{m,1}^c, \cdots, I_{m,b}^1, \cdots, I_{m,b}^c, \\ & \cdots, I_{m-1}], \end{aligned} \tag{7}$$

where $W_{m,b}$ is weight sets of the 1×1 Conv layer, I_{m-1} denotes the input of the m-th BCAM, $I_{m,b}^c$ is the c-th Conv layer of the b-th BCAB in the m-th BCAM. We assume that each BCAB consists of C Conv layers and each Conv layer has G feature maps. Thereby, $G \times ((C \times b) + 1)$ feature maps will be concatenated, and then we adopt a 1×1 Conv layer to reduce redundant information. The output $X_{m,b}$ of BCAB will be fed into the mixed residual attention unit (MRAU).

$$I_{m,b} = R_{m,b}(X_{m,b}), \tag{8}$$

where $R_{m,b}(\cdot)$ is the function corresponding to MRAU, $I_{m,b}$ denotes the output of the b-th BCAB in the m-th BCAM. The details of MRAU will be discussed in next section.

Finally, we adopt the residual learning (RL) strategy to train our network steadily. The final output of m-th BCAM can be obtained by

$$I_m = I_{m,B} + I_{m-1}, \tag{9}$$

where I_{m-1} and I_m represent the input and output of m-th BCAM, respectively. $I_{m,B}$ represents the output of last BCAB in the m-th BCAM.

3.3 Mixed Residual Attention Unit (MRAU)

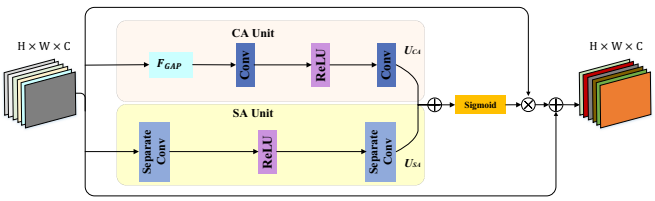

Figure 4. Our proposed mixed residual attention unit (MRAU).

The features generated by CNNs contain different types of information across channels and spatial regions which have different contributions for the recovery of high-frequency details. In order to improve the representational capability of the network, we embed a mixed residual attention unit (MRAU) in the tail of each BCAB. The detailed structure of MRAU is illustrated in Figure 4.

In the channel attention (CA) unit, let $x = [x_1, x_2, \cdots, x_C]$ serves as input of MRAU, which has C feature maps with size of $H \times W$, then we obtain the channel-wise output statistic z adopting global average pooling. The c-th element of z is computed by

$$\begin{aligned} z_c &= F_{GAP}(x) \\ &= \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j), \end{aligned} \tag{10}$$

where $x_c(i,j)$ denotes the value at position (i,j) of the c-th feature. The excitation and scaling processes are performed in the same way as in [37]. Then the CA maps can be obtained by

$$U_{CA} = W_{CA}^2(\delta(W_{CA}^1 z)), \tag{11}$$

where $\delta(\cdot)$ denotes the function of ReLU [26], $W_{CA}^1 \in \mathbb{R}^{\frac{C}{r} \times C \times 1 \times 1}$ is weight sets of the first Conv layer which is followed by ReLU activation and used to decrease the number of channels of z by reduction ratio r. Then the number of channels is increased back to the initial amount via second Conv layer with weight of $W_{CA}^2 \in \mathbb{R}^{C \times \frac{C}{r} \times 1 \times 1}$, the bias term is omitted for simplicity.

In addition, the information contained in feature maps is also diverse over spatial positions. For example, the edge or texture regions usually contain more high-frequency information while the smooth areas have more low-frequency information. In other words, different regions of the image need to obtain different attention.

Therefore, in the spatial attention (SA) unit, we explore a complementary form of attention, spatial attention, to improve the representational capability of the network. The SA maps can be obtained by

$$U_{SA} = W_{SA}^2(\delta(W_{SA}^1 x)), \tag{12}$$

where $\{W_{SA}^i\}_{i=1}^2 \in \mathbb{R}^{C \times C \times H \times W}$ is the weight of two depth-wise convolutions [11].

CA and SA units exploit the relationship of inter-channel and intra-channel, respectively. To make full use of both attention, we combine these two units by adopting element-wise sum operation. To assign different attention to different types of feature maps, we employ a gating mechanism with a sigmoid activation after fusing these two attention units. Then we rescale the input x as follows.

$$\hat{x} = \sigma(U_{CA} \oplus U_{SA}) \otimes x, \tag{13}$$

where $\hat{x}$ denotes the recalibrated features, $\sigma(\cdot)$ denotes the sigmoid activation function, $\oplus$ and $\otimes$ denotes element-wise sum and element-wise product, respectively.

Considering the low-level features are more important for image reconstruction, we adopt a simple yet more suitable residual attention learning method by combining input features x and recalibrated features $\hat{x}$ directly.

$$y = x + \hat{x}, \tag{14}$$

3.4 Implementation

In our BCAN, we set the number of BCAM as M = 10. In each BCAM, we set the BCAB number as 4 and each BCAB has 3 Conv layers. Except 1×1 Conv layer, we use 3×3 as the kernel size of all other Conv layers. We use 64 filters in all Conv layers except for the final reconstructed layer with 3 filters producing color images. In AFFM, all learnable weight factors are initialized as 1. For upscale net $H_{UpNet}(\cdot)$, we use ESPCNN [28] to upscale the coarse resolution features to fine ones.

4 Experimental Results and Analyses

4.1 Settings

We choose 800 training images from DIV2K dataset [29] as training data. For testing, we evaluate our results under peak signal noise ratios PSNR and SSIM [31] on five standard datasets: Set5 [3], Set14 [34], BSD100 [24], Urban100 [14] and Manga109 [25]. To keep consistent with previous works, quantitative results are only evaluated on luminance channel of transformed YCbCr space.

Following the work in [22], we randomly crop 16 patches of size 48×48 from the LR images as input for each training mini-batch. We randomly augment the patches by flipping horizontally or vertically and rotating 90°. Our model is trained with ADAM optimizer [18] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. The initial learning rate is set to 10^{-3} by using weight normalization and then decreases to half every 2×10^5 iterations of back-propagation.

In order to demonstrate the effectiveness of our BCAN, we adopt three different degradation models to obtain LR images. The first one is bicubic downsampling (denoted as BI), which is used in most previous methods. We evaluate the results with scaling factor $\times 2$, $\times 3$ and $\times 4$, respectively. The second one firstly blurs HR images by Gaussian kernel of size 7×7 with standard deviation 1.6, then obtains LR images via downsampling blurred images (denoted as BD). The third is a very challenging one, which first bicubic-downsamples HR image with scaling factor $\times 3$ and then adds Gaussian noise of level 30 (denote as DN). In the latter two cases, we only evaluate the results with the scaling factor $\times 3$ in order to keep consistent with previous work [38].

4.2 Study of B, C and M

In this subsection, we explore the influence of the number of BCAB (denoted as B), the number of Conv layer in each BCAB (denoted

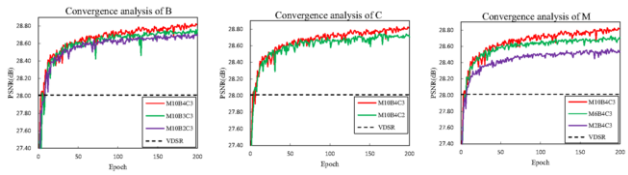

Figure 5. Convergence analysis of B, C and M. The curves for each combination are based on the PSNR value on Set14 with scaling factor $\times 4$ in 200 epochs.

as C) and the number of BCAM (denoted as M). As shown in Figure 5(a) and 5(b), larger B and C would lead to higher performance. The main reason lies in that the network becomes deeper and wider with larger B and C. On the other hand, we embed an MRAU in each BCAB, which further improves the representational capability of the network. We then investigate the influence of M by fixing B and C to 4 and 3, respectively. It can be observed from Figure 5(c) that the reconstruction performance is significantly improved with the increasing M. In conclusion, choosing larger B, C or M contributes to better results. It should be noticed that small B, C and M, our BCAN also has better results than VDSR [17]. Eventually, considering the trade-off between network performance and model complexity, we adopt M = 10, B = 4 and C = 3 in our BCAN model.

4.3 Ablation Investigation

To verify the effectiveness of spatial attention (SA), channel attention (CA) and adaptive feature fusion module (AFFM), we conduct a series of ablation studies. The specific performance is listed in Table 1. $Base$ (PSNR = 28.62dB) refers to a baseline model which is obtained without SA, CA or AFFM.

Table 1. Investigation of spatial attention (SA), channel attention (CA), and adaptive feature fusion module (AFFM). The table shows the best PSNR value on Set14 with scaling factor $\times 4$ in 200 epochs.

Models	$Base$	R_1	R_2	R_3	R_4	R_5
SA	✗	✓	✗	✗	✓	✓
CA	✗	✗	✓	✗	✓	✓
AFFM	✗	✗	✗	✓	✗	✓
PSNR	28.62	28.72	28.71	28.73	28.76	28.80

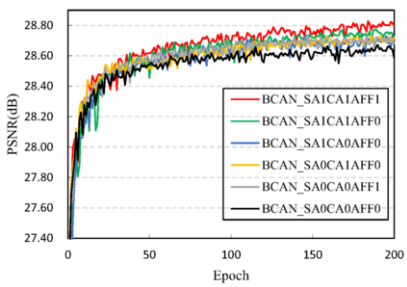

Figure 6. Convergence analysis on SA, CA, and AFFM. These results are evaluated on Set14 with scaling factor $\times 4$ in 200 epochs.

To demonstrate the effect of our proposed mixed residual attention unit (MRAU), we first add a component of CA or SA to the $Base$ model. We can observe from Table 1 that R_1 (PSNR = 28.72dB) and R_2 (PSNR = 27.71dB) models obtain better PSNR than $Base$ model. More importantly, when we combine CA and SA to R_4 (PSNR = 28.76dB) or R_5 (PSNR = 28.80dB) model, the PSNR further improved compared with R_1 and R_2 models. This not only shows that CA and SA play an important role in improving the SR performance, but also demonstrates the effectiveness of combining the two attention methods. We further show the effect of AFFM from the results of R_3 (PSNR = 28.73dB) and R_5 model. We can observe that the results are improved obviously, no matter CA and SA are used or not, which fully demonstrates the importance of adaptively fusing hierarchical features.

We also visualize the convergence process of these six combinations in Figure 6. The curves are consistent with the analysis above, these experiments and visual analyses strongly demonstrate the effectiveness of our proposed SA, CA and AFFM.

4.4 Results with BI Degradation Model

For BI degradation model, we compare our method with other state-of-the-art SR methods such as VDSR [17], LapSRN [19], SRFBN [21], EDSR [22], D-DBPN [9], MSRN [20], RDN [38] and SAN [8]. The PSNR/SSIM comparing results are shown in Table 2, most of which are re-evaluated from the corresponding public codes. In particular, followed by D-DBPN [9], we increase the Flickr2K [1] datasets to train our BCAN in order to further improve the performance (denoted as BCAN+). It can be seen that our BCAN outperforms most of the other methods, and our BCAN+ outperforms almost all comparative methods, especially on the Urban100 [14] and Manga109 [25] datasets. In addition, it is worth mentioning that EDSR utilizes much more number of filters (256 vs. 64), D-DBPN employs more training images (DIV2K+Flickr2K+ImageNet vs. DIV2K), SAN has more than twice the parameters of our BCAN (15.7M vs. 6M), and RDN has much more parameters than ours (22M vs. 6M). Hence, these observations indicate that our BCAN has great advantages in terms of SR performance and network parameters.

Figure 7 shows visual comparisons with scaling factor $\times 4$ on "img_004" and "img_073" from Urban100 dataset, from which we observe that other methods produce more blurred lattices or edges, and in contrast, our BCAN alleviates the blurring artifacts and recovers more details. Similar observations are shown in "Yumeiro-Cooking", our BCAN recovers clearer line which is very close to the ground truth. Such comparisons demonstrate that our BCAN owns more powerful representational capability for complex features compared with other state-of-the-art methods.

4.5 Results with BD and DN Degradation Models

For BD and DN degradation models, we compare our method with other state-of-the-art methods such as SRCNN [4], FSRCNN [5], VDSR [17], IRCNN [36], RDN [38], SRFBN [21] and SAN [8]. Table 3 shows the comparing results on PNSR and SSIM, from which we can observe that our BCAN is superior to most state-of-the-art methods, but our BCAN+ achieves the best performance with fewer parameters.

Figure 8 shows the visual comparisons for BD and DN degradation models. We pick out images from Urban100 dataset for comparison because this dataset is very challenging. From images on the top row in the figure, we observe that for details in "img_024", most methods suffer from heavy blurring artifacts, while our BCAN produces a recovering image close to ground truth. From the bottom row in

Table 2. Average PSNR/SSIM of various SR methods for scaling factor $\times 2$, $\times 3$ and $\times 4$ with BI degradation model. The best results are **highlighted**.

Datasets	Scale	VDSR [17] $\approx 665K$	LapSRN [19] ≈ 813	SRFBN [21] $\approx 3.5M$	EDSR [22] $\approx 43M$	D-DBPN [9] $\approx 10M$	MSRN [20] $\approx 6.5M$	RDN [38] $\approx 22M$	SAN [8] $\approx 15.7M$	BCAN(Ours) $\approx$ **6M**	BCAN+(Ours) $\approx$ **6M**
Set5	$\times 2$	37.53/0.9587	37.52/0.9581	38.11/0.9609	38.11/0.9602	38.09/0.9600	38.08/0.9605	38.24/0.9614	**38.31/0.9620**	38.24/0.9613	38.25/0.9614
	$\times 3$	33.66/0.9213	33.82/0.9227	34.70/0.9292	34.65/0.9280	-/-	34.38/0.9262	34.71/0.9296	34.75/**0.9300**	34.71/0.9297	**34.78/0.9300**
	$\times 4$	31.35/0.8838	31.54/0.8850	32.47/0.8983	32.46/0.8968	32.47/0.8980	32.07/0.8903	32.47/0.8990	**32.64/0.9003**	32.51/0.8989	32.55/0.8995
Set14	$\times 2$	33.03/0.9124	33.08/0.9109	33.82/0.9196	33.92/0.9195	33.85/0.9190	33.74/0.9170	34.01/0.9212	34.07/0.9213	33.99/0.9209	**34.14/0.9231**
	$\times 3$	29.77/0.8314	29.79/0.8320	30.51/0.8461	30.52/0.8462	-/-	30.34/0.8395	30.57/0.8468	30.59/0.8476	30.57/0.8468	**30.65/0.8481**
	$\times 4$	28.01/0.7674	28.19/0.7720	28.81/0.7868	28.80/0.7876	28.82/0.7860	28.60/0.7751	28.81/0.7871	**28.92**/0.7888	28.85/0.7877	**28.92/0.7897**
BSD100	$\times 2$	31.92/0.8965	31.80/0.8949	32.29/0.9010	32.32/0.9013	32.27/0.9000	32.23/0.9013	32.34/0.9017	**32.42/0.9028**	32.35/0.9018	32.38/0.9021
	$\times 3$	28.83/0.7966	28.82/0.7973	29.24/0.8084	29.25/0.8093	-/-	29.08/0.8041	29.26/0.8093	**29.33/0.8112**	29.26/0.8091	**29.33**/0.8106
	$\times 4$	27.29/0.7167	27.32/0.7280	27.72/0.7409	27.71/0.7420	27.72/0.7400	27.52/0.7273	27.72/0.7419	27.78/**0.7436**	27.74/0.7416	**27.79**/0.7435
Urban100	$\times 2$	30.76/0.914	30.41/0.9112	32.62/0.9328	32.93/0.9351	32.55/0.9324	32.22/0.9326	32.89/0.9353	**33.10/0.9370**	32.97/0.9355	33.08/0.9363
	$\times 3$	27.14/0.8279	27.07/0.8272	28.73/0.8641	28.80/0.8653	-/-	28.08/0.8554	28.80/0.8653	28.93/0.8671	28.85/0.8657	**29.07/0.8696**
	$\times 4$	25.18/0.7524	25.21/0.7560	26.60/0.8015	26.64/0.8033	26.38/0.7946	26.04/0.7896	26.61/0.8028	26.79/0.8068	26.70/0.8038	**26.87/0.8090**
Manga109	$\times 2$	37.22/0.9729	37.27/0.9855	39.08/0.9779	39.10/0.9773	38.89/0.9775	38.82/0.9768	39.18/0.9780	39.32/**0.9792**	39.26/0.9778	**39.52**/0.9789
	$\times 3$	32.01/0.9310	32.19/0.9334	34.18/0.9481	34.17/0.9476	-/-	33.44/0.9427	34.13/0.9484	34.30/0.9494	34.30/0.9488	**34.64/0.9503**
	$\times 4$	28.83/0.8809	29.09/0.8900	34.15/0.9160	31.02/0.9148	30.91/0.9137	30.17/0.9034	31.00/0.9151	31.18/0.9169	31.13/0.9167	**31.40/0.9186**

Table 3. Average PSNR/SSIM of various SR methods for scaling factor $\times 3$ with BD and DN degradation models. The best results are **highlighted**.

Dataset	Model	Bicubic	SRCNN [4]	FSRCNN [5]	VDSR [17]	IRCNN-G [36]	RDN [38]	SRFBN [21]	SAN [8]	BCAN(Ours)	BCAN+(Ours)
Set5	BD	28.78/0.8308	32.21/0.9001	26.23/0.8124	33.25/0.9150	33.38/0.9182	34.58/0.9280	34.66/0.9283	34.75/0.9290	34.71/0.9290	**34.77/0.9292**
	DN	24.01/0.5369	25.01/0.6950	24.18/0.6932	25.20/0.7183	25.70/0.7379	28.47/0.8151	28.53/0.8182	-/-	28.61/0.8198	**28.65/0.8213**
Set14	BD	26.38/0.7271	28.89/0.8105	24.44/0.7106	29.46/0.8244	29.63/0.8281	30.53/0.8447	30.48/0.8439	30.68/0.8466	30.63/0.8458	**30.69/0.8473**
	DN	22.87/0.4724	23.78/0.5898	23.02/0.5856	24.00/0.6112	24.45/0.6305	26.60/0.7101	26.60/0.7144	-/-	26.68/0.7144	**26.73/0.7161**
BSD100	BD	26.33/0.6918	28.13/0.7740	24.86/0.6832	28.57/0.7893	28.65/0.7922	29.23/0.8079	29.21/0.8069	29.33/**0.8101**	29.28/0.8085	**29.35**/0.8098
	DN	22.92/0.4449	23.76/0.5538	23.41/0.5556	24.00/0.5749	24.28/0.5900	25.93/0.6573	25.95/0.6625	-/-	25.99/0.6623	**26.01/0.6635**
Urban100	BD	23.52/0.6862	25.84/0.7856	22.04/0.6745	26.61/0.8136	26.77/0.8154	28.46/0.8582	28.48/0.8581	28.83/0.8646	28.75/0.8625	**28.93/0.8659**
	DN	21.63/0.4687	21.90/0.5737	21.15/0.5682	22.22/0.6096	22.90/0.6429	24.92/0.7364	24.99/0.7424	-/-	25.17/0.7463	**25.25/0.7491**
Manga109	BD	25.46/0.8149	29.64/0.9003	23.04/0.7927	31.06/0.9234	31.15/0.9245	33.97/0.9465	34.07/0.9466	34.46/0.9487	34.45/0.9484	**34.73/0.9498**
	DN	23.01/0.5381	23.75/0.7148	22.39/0.7111	24.20/0.7525	24.88/0.7765	28.00/0.8591	28.02/0.8618	-/-	28.19/0.8641	**28.35/0.8669**

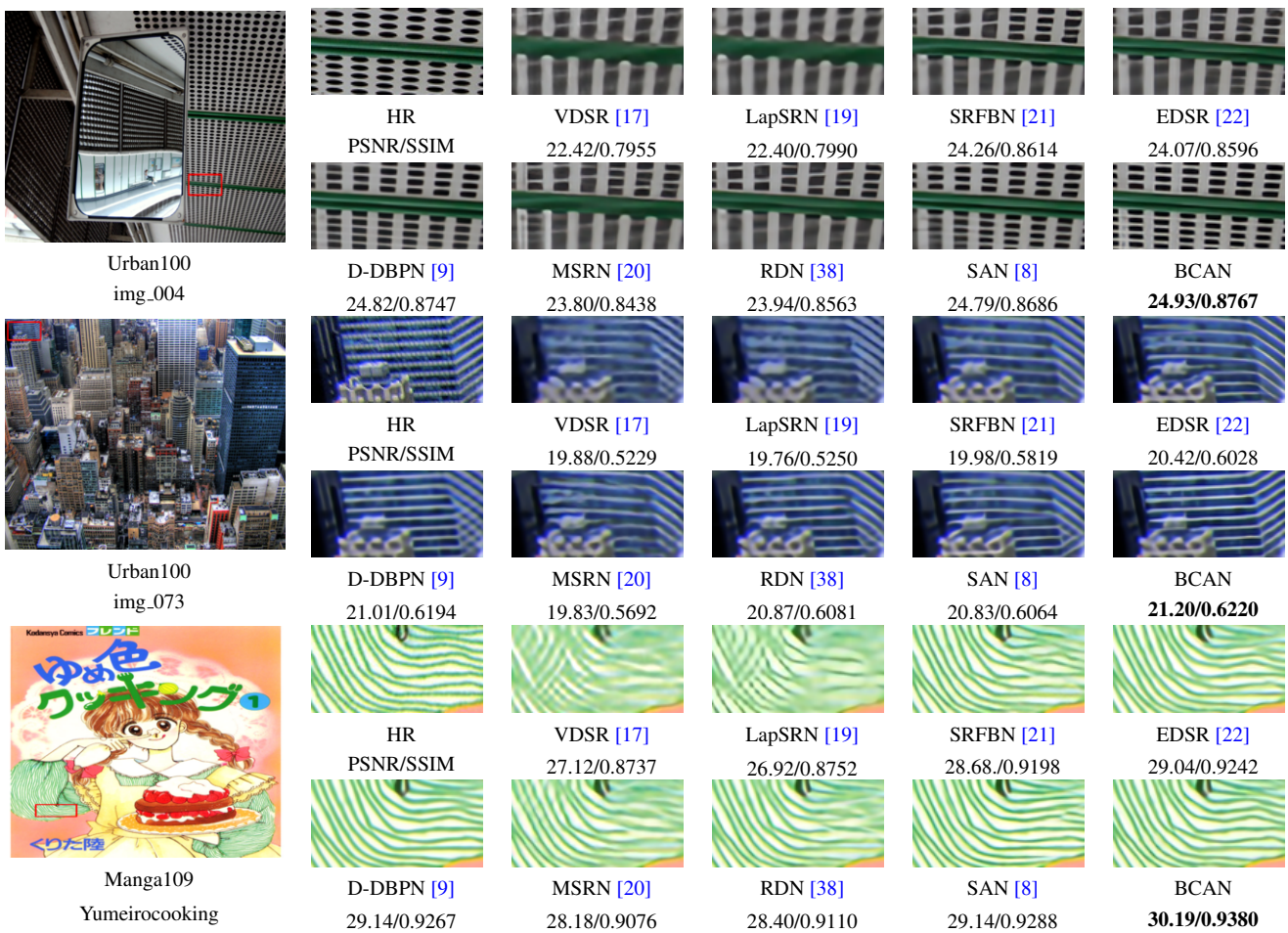

Figure 7. Visual comparison for $\times 4$ SR with BI degradation model. The best results are **highlighted**.

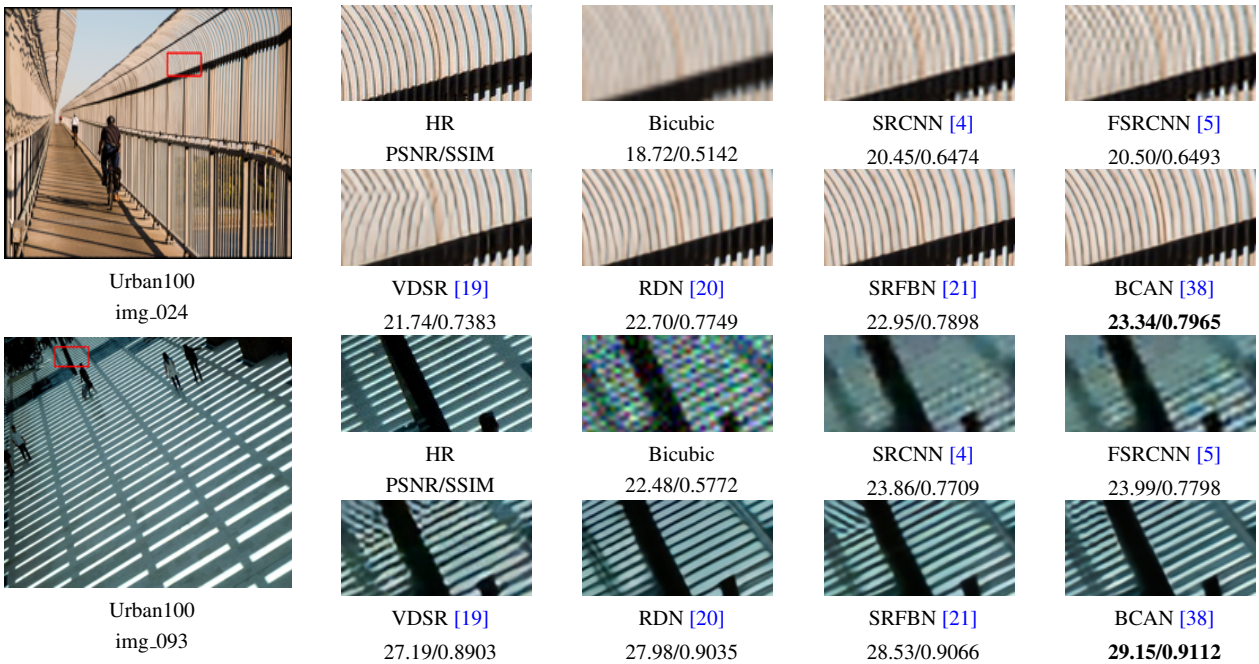

Figure 8. Visual comparison for $\times 3$ SR with BD and DN degradation models. The best results are **highlighted**.

the figure, we observe that for "img_093", which is corrupted with noise and loses some details, our method clearly reconstructs the zebra crossing and the stripe patterns, while other methods result in severe distortions or noticeable artifacts. These visual comparisons also demonstrate the powerful representational capability of our BCAN.

4.6 Model Complexity Analysis

4.6.1 Model Size Comparison

We study the trade-off between the SR performance and the number of network parameters of our BCAN, BCAN+ and existing state-of-the-art networks. Figure 1 shows the PSNR performances of various CNN-based methods versus the number of parameters, where the results are evaluated on the Set14 dataset with scaling factor $\times 4$. In comparison with the deep networks, such as RCAN [37], SAN [8] and RDN [38], our BCAN can achieve competitive SR performances, while only needs the 40% and 20% parameters of RCAN and RDN, respectively. However, our BCAN+ can obtain the best SR performance.

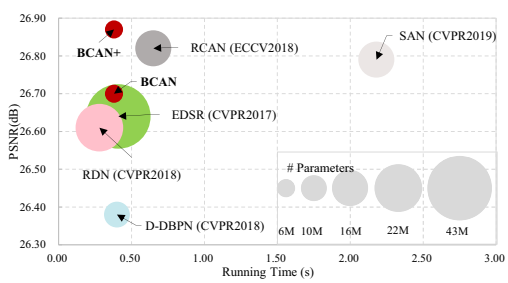

Figure 9. Trade-off between performance vs. number of parameters and running time on Urban100 dataset with scaling factor $\times 4$. The $x-$axis and $y-$axis denote the running time and PSNR values, and the size of the circle denotes the number of the network parameters.

4.6.2 Running Time Comparison

In addition, we compare running time of our proposed BCAN and RCAN+ with five deep networks: EDSR [22], D-DBPN [9], RDN [35], RCAN [37] and SAN [8]. The running time of all networks is evaluated on the same machine with 3.6GHz Intel i7 CPU (64G RAM) and an NVIDIA 2080Ti GPU using their official codes. We can observe from Figure 9 that our BCAN has a great tradeoff between running time and PSNR values. RCAN and SAN have slightly higher PSNR values but spend more running time. This is because they focus on the deeper network (about 400 Conv layers) in pursuit of higher PSNR results. However, our BCAN+ achieves the best SR performance but with fewer running time.

5 Conclusions

This paper proposes a mid-weight bypass connection attention network (BCAN) which adopts bypass connections and attention mechanism for image SR. The results evaluated on five different datasets show that our BCAN outperforms other state-of-the-art methods with fewer network parameters. The reasons for the improvement of our BCAN are as follows. Firstly, our proposed bypass connection attention module (BCAM) changes the mode of dense connections to reduce redundant features. Secondly, we combine channel attention with spatial attention to construct a mixed residual attention unit (MRAU) and then embed it into BCAM to obtain more effective hierarchical features. Finally, we propose an adaptive feature fusion module (AFFM) to combine these hierarchical features.

6 Acknowledgments

This research was supported by the National Natural Science Foundation of China (Nos.61672203, 61976079 &U1836102) and Anhui Natural Science Funds for Distinguished Young Scholar (No.170808J08).

REFERENCES

[1] Eirikur Agustsson and Radu Timofte, 'Ntire 2017 challenge on single image super-resolution: Dataset and study', in *CVPRW*, (2017).

[2] Namhyuk Ahn, Byungkon Kang, and Kyung Ah Sohn, 'Fast, accurate, and lightweight super-resolution with cascading residual network', in *ECCV*, (2018).

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel, 'Low-complexity single-image super-resolution based on nonnegative neighbor embedding', in *BMVC*, (2012).

[4] Dong Chao, Change Loy Chen, Kaiming He, and Xiaoou Tang, 'Learning a deep convolutional network for image super-resolution', in *ECCV*, (2014).

[5] Dong Chao, Change Loy Chen, and Xiaoou Tang, 'Accelerating the super-resolution convolutional neural network', in *ECCV*, (2016).

[6] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua, 'Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning', in *CVPR*, (2017).

[7] Rong Chen, Yanyun Qu, Kun Zeng, Jinkang Guo, and Yuan Xie, 'Persistent memory residual network for single image super resolution', in *CVPR*, (2018).

[8] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang, 'Second-order attention network for single image super-resolution', in *CVPR*, (2019).

[9] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita, 'Deep back-projection networks for super-resolution', in *CVPR*, (2018).

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', in *CVPR*, (2016).

[11] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, 'Mobilenets: Efficient convolutional neural networks for mobile vision applications', *arXiv preprint arXiv:1704.04861*, (2017).

[12] Jie Hu, Li Shen, and Gang Sun, 'Squeeze-and-excitation networks', in *CVPR*, (2018).

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, 'Densely connected convolutional networks', in *CVPR*, (2017).

[14] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja, 'Single image super-resolution from transformed self-exemplars', in *CVPR*, (2015).

[15] Yawen Huang, Ling Shao, and Alejandro F Frangi, 'Simultaneous super-resolution and cross-modality synthesis of 3d medical images using weakly-supervised joint convolutional sparse coding', in *CVPR*, (2016).

[16] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, 'Deeply-recursive convolutional network for image super-resolution', in *CVPR*, (2016).

[17] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, 'Accurate image super-resolution using very deep convolutional networks', in *CVPR*, (2016).

[18] Diederik P Kingma and Jimmy Ba, 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980*, (2014).

[19] Wei Sheng Lai, Jia Bin Huang, Narendra Ahuja, and Ming Hsuan Yang, 'Deep laplacian pyramid networks for fast and accurate super-resolution', in *CVPR*, (2017).

[20] Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang, 'Multi-scale residual network for image super-resolution', in *ECCV*, (2018).

[21] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu, 'Feedback network for image super-resolution', in *CVPR*, (2019).

[22] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee, 'Enhanced deep residual networks for single image super-resolution', in *CVPR*, (2017).

[23] Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov, 'Generating images from captions with attention', *Computer Science*, (2016).

[24] David Martin, Charless Fowlkes, Doron Tal, Jitendra Malik, et al., 'A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics', in *ICCV*, (2001).

[25] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa, 'Sketch-based manga retrieval using manga109 dataset', *Multimedia Tools and Applications*, (2017).

[26] Vinod Nair and Geoffrey E Hinton, 'Rectified linear units improve restricted boltzmann machines', in *ICML*, (2010).

[27] Pejman Rasti, Tõnis Uiboupin, Sergio Escalera, and Gholamreza Anbarjafari, 'Convolutional neural network super resolution for face recognition in surveillance monitoring', in *AMDO*, (2016).

[28] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang, 'Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network', in *CVPR*, (2016).

[29] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang, 'Ntire 2017 challenge on single image super-resolution: Methods and results', in *CVPRW*, (2017).

[30] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao, 'Image super-resolution using dense skip connections', in *ICCV*, (2017).

[31] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al., 'Image quality assessment: from error visibility to structural similarity', *IEEE transactions on image processing*, **13**(4), 600–612, (2004).

[32] Huijuan Xu and Kate Saenko, 'Ask, attend and answer: Exploring question-guided spatial attention for visual question answering', in *ECCV*, (2016).

[33] Tai Ying, Yang Jian, and Xiaoming Liu, 'Image super-resolution via deep recursive residual network', in *CVPR*, (2017).

[34] Roman Zeyde, Michael Elad, and Matan Protter, 'On single image scale-up using sparse-representations', in *Curves and Surfaces*, (2010).

[35] Haochen Zhang, Liu Dong, and Zhiwei Xiong, 'Convolutional neural network-based video super-resolution for action recognition', in *FG*, (2018).

[36] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang, 'Learning deep cnn denoiser prior for image restoration', in *CVPR*, (2017).

[37] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu, 'Image super-resolution using very deep residual channel attention networks', in *ECCV*, (2018).

[38] Yulun Zhang, Yapeng Tian, Kong Yu, Bineng Zhong, and Fu Yun, 'Residual dense network for image super-resolution', in *CVPR*, (2018).

[39] Zhong-Qiu Zhao, Jian Hu, Weidong Tian, and Ning Ling, 'Cooperative adversarial network for accurate super resolution', in *ACCV*, (2018).

[40] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu, 'Object detection with deep learning: A review', *IEEE transactions on neural networks and learning systems*, **30**(11), 3212–3232, (2019).