

Research Article

Adaptive Residual Channel Attention Network for Single Image Super-Resolution

Kerang Cao,¹ Yuqing Liu ,² Lini Duan,¹ and Tian Xie ²

¹Shenyang University of Chemical Technology, Shenyang 110000, China

²School of Software, Dalian University of Technology, Dalian 116620, China

Correspondence should be addressed to Yuqing Liu; liuyuqing@mail.dlut.edu.cn

Received 6 May 2020; Revised 8 July 2020; Accepted 23 July 2020; Published 28 August 2020

Academic Editor: Roberto Natella

Copyright © 2020 Kerang Cao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Single image super-resolution (SISR) is a traditional image restoration problem. Given an image with low resolution (LR), the task of SISR is to find the homologous high-resolution (HR) image. As an ill-posed problem, there are works for SISR problem from different points of view. Recently, deep learning has shown its amazing performance in different image processing tasks. There are works for image super-resolution based on convolutional neural network (CNN). In this paper, we propose an adaptive residual channel attention network for image super-resolution. We first analyze the limitation of residual connection structure and propose an adaptive design for suitable feature fusion. Besides the adaptive connection, channel attention is proposed to adjust the importance distribution among different channels. A novel adaptive residual channel attention block (ARCB) is proposed in this paper with channel attention and adaptive connection. Then, a simple but effective upscale block design is proposed for different scales. We build our adaptive residual channel attention network (ARCNN) with proposed ARCBs and upscale block. Experimental results show that our network could not only achieve better PSNR/SSIM performances on several testing benchmarks but also recover structural textures more effectively.

1. Introduction

Super-resolution (SR) is an important issue in the image restoration area. The task of single image super-resolution (SISR) is to find high-resolution (HR) images from the low-resolution (LR) images. Since it is an ill-posed problem, there are potential high-resolution images corresponding to an identical image with low resolution. There are practical applications with SISR methods, such as video quality enhancement, remote sensing image processing, and MRI analysis. To find the most suitable HR images, there are various methods for SISR problem and other image restoration tasks.

Deep learning has shown its amazing performance in various tasks [1–5]. Nowadays, there are convolutional neural network- (CNN-) based works focusing on SISR problem. As far as we know, SRCNN [6] is the first work using a three-layer CNN for image super-resolution. After SRCNN, Dong et al. proposed FSRCNN [7] with a deeper but narrower network and achieved better performance with

less time cost. Usually, when the network is deeper, the performance will be better. VDSR [8] proposed by Kim et al. used a very deep network design with global residual learning. Inspired by VDSR and residual connections, EDSR [9] proposed by Lim et al. applied an enhanced deep network for SISR problem with residual blocks. Besides EDSR, MDSR [9] designs an upscale module for different scaling factors jointly. Motivated by the structure of Laplacian pyramid, Lai et al. proposed LapSRN [10] with a progressive structure. Similar to MDSR, the progressive LapSRN could upscale images with different scaling factors concurrently. Recursive structures could effectively enlarge the receptive fields and make full use of sharing parameters. DRCN [11] proposed by Ghifary et al. used shared convolution layers to increase the receptive fields. DRRN [12] proposed by Tai et al. combined residual and recursive structures and achieved good performances.

ResNet [13] proposed by He et al. has been proved to be a success for network design. In ResNet, a residual block was

proposed for image classification and beyond with effective gradient flow. There are works for super-resolution with residual blocks. EDSR stacked residual blocks with a global shortcut to build a very deep network. RDN [14] proposed by Zhang et al. introduced a residual dense block (RDB) with feature fusion, achieving good performance. Besides RDN, RCAN [15] proposed by Zhang et al. designed a residual-in-residual structure to build the network deeper. There is a shortcut in residual block: the addition of feature maps before and after processing is regarded as the final result. In fact, the above methods apply a fixed ratio balancing the two parts, which does not distinguish the different importance.

Attention is used for human brain simulation. When watching pictures, human's brain usually focuses on more important area. There are attention methods for image processing tasks. SENet [16] proposed by Hu et al. introduced a channel attention method for importance distribution among channels. SENet requires few parameters, which is flexible for different network designs. To the best of our knowledge, RCAN [15] is the first image super-resolution work with channel attention mechanism. After RCAN, IMDN [17] proposed by Hui et al. modified the vanilla channel attention layers and achieved good performances with few parameters. SAN [18] proposed by Dai et al. introduced a second-order attention mechanism with channel and nonlocal attentions.

In this paper, we propose a novel adaptive residual channel attention block (ARCB) for image super-resolution. Different from vanilla residual blocks, an adaptive weight is learned from paired data for combining information of main path and shortcut. Considering the different importance of channels in residual blocks, channel attention is introduced in ARCB for weight distribution on channels. Besides block designs, recent works design special upscale modules for different scaling factors. In this paper, we introduce a simple but effective general upscale block design for different factors. The adaptive residual channel attention network (ARCNN) is designed based on ARCBs and proposed upscale block. Experiments are performed on several testing benchmarks. The results show that our ARCNN could not only achieve better performance on PSNR/SSIM comparison but also recover complex structural textures more effectively.

The contributions of this paper could be demonstrated as follows:

- (1) We propose a novel block named ARCB with channel attention mechanism. In ARCB, we propose an adaptive residual connection with learned weights. The weight factors could find suitable ratios for combination information from different paths. Channel attention mechanism in ARCB distributes different weights on channels for concentrating more on important information.
- (2) We propose a tiny but effective upscale block design method. With the proposed design, our network could be flexibly analogized for different scaling factors.

- (3) Experimental results show that our proposed ARCNN achieves better PSNR/SSIM results on several testing benchmarks and recovers more complex structural textures than other methods.

2. Related Works

2.1. Single Image Super-Resolution. Let I^{HR} and I^{LR} denote LR and HR images separately; the observation model of degradation step could be described as

$$I^{\text{LR}} = I^{\text{HR}} \downarrow + n, \quad (1)$$

where $(\cdot)\downarrow$ denotes the degradation and n denotes the noise. Usually the degradation models are chosen as bicubic-down with different scaling factors. Given I^{LR} , the task of single image super-resolution (SISR) is to find corresponding I^{HR} . However, there are several potential HR images degraded to a same LR image. Since SISR is an ill-posed problem, it is challenging to find the solution.

Convolutional neural network (CNN) has been proved as an effective tool for image restoration [1–5]. Recently, there are CNN-based works for SISR problem [19, 20]. To the best of our knowledge, SRCNN [6] proposed by Dong et al. is the first deep learning work for SISR. There are three convolution layers in SRCNN, corresponding to the three steps of sparse coding method: feature extraction, nonlinear mapping, and restoration. After SRCNN, FSRCNN [7] proposed by Dong et al. applied a deeper network for SISR problem. Different from SRCNN, FSRCNN used a deconvolution layer to upscale the feature map. By using shrinking and expanding modules, FSRCNN decreased the number of parameters and built the network deeper and faster. ESPCN [21] proposed by Shi et al. introduced a pixel shuffle layer to substitute the deconvolution layer for upscaling. Similar to FSRCNN, there are several convolution layers processing the LR feature maps. At the end of ESPCN, a convolution layer changed the channel number of feature maps, and the pixel shuffle layer performed the spatial transformation. From the practical experience, a deeper network usually results in better performance. VDSR [8] proposed by Kim et al. applied a very deep network with twenty convolution layers and global residual connection. Besides, batch normalization (BN) was used in VDSR to improve the performance. To preserve the resolution of feature maps, zero padding strategy was applied in VDSR. RED [22] proposed by Mao et al. introduced a symmetrical encoder-decoder structure with convolution and deconvolution layers. To transmit information to the bottom layer, residual connections were designed between blocks.

A deeper network will cause a large amount of parameters. Recursive design with sharing parameters is one of the choices to build lightweight networks. There are recursive networks for SISR problem. DRCN [11] proposed by Ghifary et al. applied sharing convolution layers to enlarge the receptive field with limited parameters. Similar to SRCNN, there are three modules in DRCN. Embedding network extracted the feature maps from input images. After feature extraction, the inference network with sharing

parameters performed the nonlinear mapping. Finally, reconstruction network restored the HR images from feature maps. To increase the network performance, there is a skip connection in DRCN to learn the residual information. DRRN [12] proposed by Tai et al. designed a recursive residual connection with sharing parameters to prevent gradient disappearance. By using recursive design and sharing parameters, DRRN built a 52-layer network with few parameters and performed better than VDSR.

Recently, there are works with good performances focusing on different block designs and network pipelines [23–27]. Dense connection has been proved effective for image super-resolution [23, 25, 26]. SRDenseNet [28] proposed by Tong et al. introduced dense connection to SISR problem and concatenated all feature maps as the final output. There are four components in SRDenseNet. Firstly, one convolution layer was used to extract low-level features. After extraction, several dense blocks were used to extract high-level features. Deconvolution layers were used to upscale the feature maps. Finally, a convolution layer was applied to HR image production. SRResNet [29] used residual blocks with skip connection to build a deep network for better performance. EDSR [9] proposed by Lim et al. removed redundant batch normalization layers from SRResNet, building a deeper network. Residual connection has been proved as an effective design for better network performance. RDN [14] proposed by Zhang et al. introduced a residual dense block named RDB combining two kinds of structures. By stacking RDBs with global feature fusion, the deep network RDN achieved good performance. MSRN [30] proposed by Li et al. introduced a multiscale residual block combining residual and inception blocks. Global fusion structure was applied in MSRN for feature extraction. To build a deeper network, a novel residual-in-residual structure was proposed in RCAN [15], which turned out to be a success.

2.2. Attention Mechanism. Attention mechanism was firstly proposed by human brain simulation. When watching an image or reading a sentence, the important areas will be paid more attention. There are different attention methods used in image processing. There are four kinds of attention mechanisms: item-wise soft attention, item-wise hard attention, location-wise soft attention, and location-wise hard attention. The difference between item-wise attention and location-wise attention is input form. Special sequential items are required for item-wise attention, while location-wise attention needs a single feature map. From another point of view, attentions could be separated as soft and hard attention. Soft attention focuses more on different areas and channels. After training, soft attentions will be generated by networks. Besides soft attention, hard attention concentrates more on different pixels. Hard attention is a random prediction procedure, which is usually implemented by reinforcement learning.

Spatial transformer network (STN) [31] is an attention method in spatial domain. In STN, information from origin images was transformed into another space with keypoints. The authors proposed a spatial transformer module for the

transformation. There are also works for channel domain attention. SENet [16] proposed by Hu et al. introduced a channel attention method for concentrating more on important channels. In SENet, a squeeze-and-excitation (SE) module is proposed to automatically learn the importance of different channels. In SE module, squeeze operation is firstly introduced to get the global channel features. After squeezing, excitation module is used to learn the relations among channels. There are two full connection layers with a ReLU activation in excitation module. Finally, scale module was applied after excitation for reweighting the feature maps. SENet focuses on the importation of channels, regarding different areas of feature maps equally. To consider the global information of feature maps, nonlocal neural networks [32] introduced a long-range dependencies attention block for better performance. However, the proposed nonlocal blocks required more memory cost and high computation complexity.

3. Method

In this section, we will describe the proposed ARCN. In ARCN, an adaptive residual channel attention block named ARCB is proposed to compose the network. Adaptive factors in ARCB for different information importance are learned while training. After adaptive residual connection, channel attention mechanism distributes the weights on channels, which considers the importance from another point of view. The main body of ARCN is composed of several ARCBs and a padding structure. A global skip connection is introduced to ARCN for residual learning. After the main body, an effective and tiny upscale module is designed for changing the scaling factors flexibly. We will introduce proposed ARCN in the following manner: Firstly, the network design will be described in general. After description, the details of ARCB will be discussed with channel attention. Detailed introduction of flexible upscale block will follow the description of ARCB. Finally, some comparisons will be done with other SISR works.

3.1. Network Design. The entire network structure is shown in Figure 1.

There are three modules in the proposed ARCN. Firstly, feature extraction module extracts feature maps from input LR images. After feature extraction, nonlinear mapping module processes the feature maps from LR space into HR space. A skip connection is applied to nonlinear mapping module for global residual learning. Finally, restoration module with a flexible upscale block restores the HR images from proposed feature maps.

There is one convolution layer in feature extraction module. The layer extracts low-level features from LR image and builds the feature maps. Let $f^{\text{FEM}}(\cdot)$ be the feature extraction module; then the operation could be demonstrated as

$$H_0 = f^{\text{FEM}}(I^{\text{LR}}), \quad (2)$$

where I^{LR} denotes the input LR images.

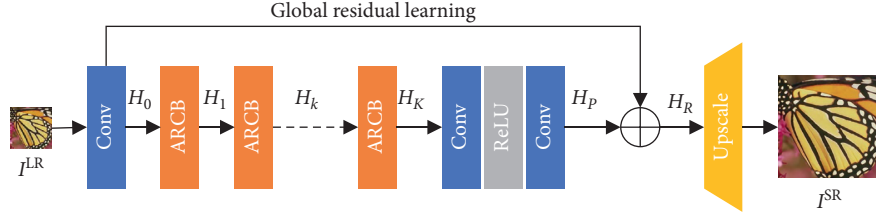


FIGURE 1: The architecture of our proposed adaptive residual channel attention network (ARCEN).

After feature extraction, several ARCBs are applied in nonlinear mapping modules for mapping feature maps from LR space to HR space. Let us denote H_k as the output of k -th ARCB; then there is

$$H_{k+1} = f_k^{\text{ARCB}}(H_k), \quad (3)$$

where $f_k^{\text{ARCB}}(\cdot)$ denotes the operation of k -th ARCB.

After K blocks, there is a padding structure composed of two convolution layers with ReLU activation. The padding structure is used to increase the network depth and weight the information from main path for global residual learning. The operation of padding structure and global residual learning could be demonstrated as

$$H_R = f^{\text{PAD}}(H_K) + H_0, \quad (4)$$

where $f^{\text{PAD}}(\cdot)$ denotes the padding structure and H_R denotes the feature maps after padding.

Finally, an effective upscale block is applied in restoration module. In restoration module, the final HR image I^{SR} is restored from processed feature maps. The operation of restoration module could be demonstrated as

$$I^{\text{SR}} = f^{\text{UP}}(H_R), \quad (5)$$

where $f^{\text{UP}}(\cdot)$ denotes the upscale block.

3.2. Adaptive Residual Connection Block. ARCB is introduced to the network with adaptive residual connection and channel attention. An illustration of proposed ARCB is shown in Figure 2(b).

There are two convolution layers with ReLU activation in ARCB. Different from ResBlock, which is used in most of SISR works, a channel attention layer is designed after the convolution layers. The attention layer weights information from different channels. After that, learned adaptive factor W_α is used to scale the processed feature maps.

Let $f^{\text{RES}}(\cdot)$ and $f^{\text{CA}}(\cdot)$ denote the main processing path and attention layer, respectively; then the operation of k -th ARCB could be demonstrated as

$$H_{k+1} = W_\alpha f^{\text{CA}}(f^{\text{RES}}(H_k)) + H_k, \quad (6)$$

where W_α denotes the adaptive factor learned while training.

The adaptive factor W_α is one of the main differences between ResBlock and ARCB. In vanilla ResBlock, the ratio of information mixture from two paths is fixed. However, it does not distinguish the importance of different information. In the proposed ARCB, weight factor W_α is a learnable parameter. In

other words, the ratio will be adjusted due to the training data, which is more suitable for information fusion.

Another main difference between vanilla ResBlock and ARCB is the channel attention. Convolution layers treat information from different channels equally. To concentrate more on important channels, channel attention is introduced to ARCB. The structure of channel attention is shown in Figure 3.

From Figure 3, global average pooling is firstly applied to information estimation. There is a hypothesis that when the feature maps are more complex, the information will be more important. From this point of view, the global average pooling operation could extract the information fast and effectively. After the feature extraction, a squeeze-and-excitation design is introduced for nonlinear mapping. In squeezing step, the channel number shrinks for information distillation. The most important information will be weighted after squeezing. Then, the excitation module preserves the channel number the same as the origin feature maps. Finally, a Sigmoid activation and a dot multiplication are adopted to distribute different importance among channels.

Let us denote by x_{in} and x_{out} the input and output of channel attention. The operation of channel attention mechanism could be demonstrated as

$$x_{\text{out}} = \sigma(\text{FE}(\text{ReLU}(\text{FS}(\text{AvgPool}(x_{\text{in}})))))) * x_{\text{in}}, \quad (7)$$

where $\sigma(\cdot)$ denotes the Sigmoid activation; $\text{FE}(\cdot)$ and $\text{FS}(\cdot)$ denote the squeeze and excitation modules. There is a ReLU activation between squeeze and excitation modules. The two modules are made of full connection layers.

There is a Sigmoid activation between the excitation and multiplication. On one hand, it will be helpful for nonlinearity. On the other hand, Sigmoid activation will convert weights to be negative. Since there is no negative for human vision system, it is designed to fit the biological process.

3.3. Flexible Upscale Block. Upscale block is widely used in various works for SISR problem, which increases the resolution of feature maps and restores the final HR image. There are different upscale block designs for different scaling factors without a unified pattern. In this paper, we proposed a flexible upscale block design pattern. With the proposed design, the structure could be easily modified for different scaling factors. The structure of our proposed flexible upscale blocks is shown in Figure 4.

As shown in Figure 4, there are three traditional upscale block designs: A, B, and C [30]. In A, a nonlinear upscale design is introduced with ReLU activation. Design B

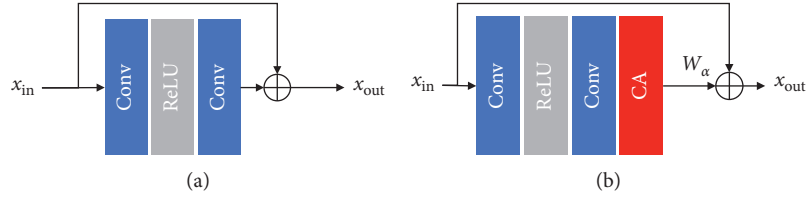


FIGURE 2: Structure of different blocks. (a) Res blocks. (b) Proposed ARCB.

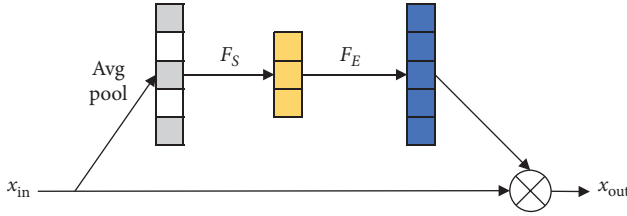


FIGURE 3: Structure of channel attention.

removes the ReLU activations but keeps the convolution and pixel-shuffle layers. Besides, design C uses deconvolution to substitute the pixel-shuffle layers. Different from other works, there is only one convolution layer with a pixel-shuffle layer in the proposed upscale block.

There are two benefits of using the flexible upscale block design. On one hand, there is only one convolution layer in the block, which saves the parameters and decreases the computation complexity. On the other hand, when the scaling factor is changed, the only modification of the block is the channel number of convolution layers. After changing, the main body of the network could be find-tuned for a new factor with few iterations.

There is a main difference between the proposed block and others. In other designs, there is a convolution layer after the last pixel-shuffle or deconvolution layer. Usually, it is used to restore the HR images with 3 channels from feature maps. However, in our proposed block, the restoration is proposed by the only convolution layer. On one hand, it is corresponding to the feature extraction module, which is also composed of only one convolution layer.

To introduce the design in detail, there are examples of different scaling factors. The special configurations are shown in Table 1. Notice that when the scaling factor is changed, the only modification is the channel number. From this point of view, the proposed upscale block is flexible for different factors.

4. Discussion

- (1) Difference from DRDN [33]: In DRDN, there are dense block (DB) structures for feature exploitation. The entire DRDN holds a global residual dense connection design to efficiently process the features. In ARCN, global and local residual connections are utilized to process the features. There is no dense connection in ARCN, which is for shrinking the channel numbers.

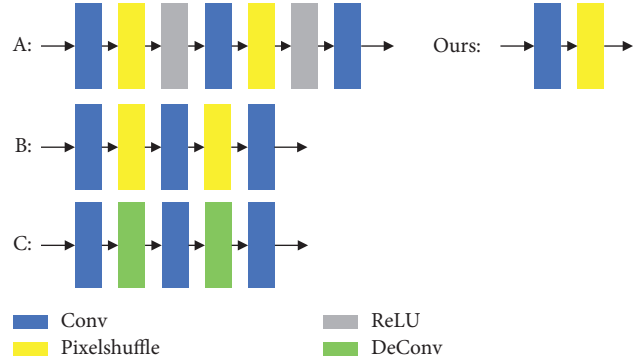
FIGURE 4: Structure of proposed flexible upscale block with scaling factor $\times 4$.

TABLE 1: Upscale block configurations for different scaling factors.

Scale	Layers	Input channels	Output channels	Kernel size
$\times 2$	Conv	64	$64 \times 2 \times 2$	3×3
	PS	$64 \times 2 \times 2$	64	—
$\times 3$	Conv	64	$64 \times 3 \times 3$	3×3
	PS	$64 \times 3 \times 3$	64	—
$\times 4$	Conv	64	$64 \times 4 \times 4$	3×3
	PS	$64 \times 4 \times 4$	64	—
$\times 8$	Conv	64	$64 \times 8 \times 8$	3×3
	PS	$64 \times 8 \times 8$	64	—

There is no channel attention and there are no adaptive weights in DRDN. In ARCN, the two components are utilized to exploit the features more effectively.

- (2) Motivation on global and local residual learning: In ARCN, global and local residual learning strategies are jointly applied for feature exploration. The residual connections can effectively solve the gradient vanishing problem, which make the network deeper. The local residual connection in ARCB ensures the gradient flow, while the global residual learning in ARCN guarantees the identical information transmission, which improves the network capacity and representation.

4.1. Experiment

4.1.1. Implementation Details. In proposed ARCN, there are $K=16$ ARCBs stacked as the main path of the nonlinear mapping module. The squeeze step in channel attention shrinks the channel number as $n=16$. For all convolution layers expect for upscale block, there are $c=64$ filters.

TABLE 2: Quantitative PSNR/SSIM comparison for scaling factors $\times 2$, $\times 3$, and $\times 4$, on testing benchmarks Set5, Set14, B100, Urban100, and Manga109. Our performance is shown in bold.

Scale	Method	Set5 [35] PSNR/ SSIM	Set14 [36] PSNR/ SSIM	B100 [37] PSNR/ SSIM	Urban100 [38] PSNR/ SSIM	Manga109 [39] PSNR/ SSIM
$\times 2$	Bicubic	33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403	30.80/0.9339
	SRCNN [6]	36.66/0.9542	32.45/0.9067	31.36/0.8879	29.50/0.8946	35.60/0.9750
	VDSR [8]	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140	37.22/0.9750
	FSRCNN [7]	37.00/0.9558	32.63/0.9088	31.53/0.8920	29.88/0.9020	36.67/0.9710
	DRCN [11]	37.63/0.9588	33.04/0.9118	31.85/0.8942	30.75/0.9133	37.55/0.9732
	LapSRN [10]	37.52/0.9591	32.99/0.9124	31.80/0.8952	30.41/0.9103	37.27/0.9740
	DRRN [12]	37.74/0.9591	33.23/0.9542	32.05/0.8973	31.23/0.9188	37.88/0.9749
	MemNet [40]	37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195	37.72/0.9542
	IDN [41]	37.83/0.9600	33.30/0.9148	32.08/0.8985	31.27/0.9196	38.01/0.9740
	EDSR (B) [9]	37.99/0.9604	33.57/0.9175	32.16/0.8994	31.92/0.9272	38.54/0.9749
	SRMDNF [42]	37.79/0.9601	33.32/0.9159	32.05/0.8985	31.33/0.9204	38.07/0.9769
	CARN [43]	37.72/0.9590	33.52/0.9166	36.66/0.8978	31.92/0.9256	38.36/0.9761
	Ms-LapSRN [44]	37.62/0.9600	33.13/0.9130	31.93/0.8970	30.82/0.0150	37.38/0.9765
Ours	38.006/0.9605	33.54/0.9173	32.15/0.8992	32.13/0.9276	38.70/0.9750	
$\times 3$	Bicubic	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349	26.95/0.8556
	SRCNN [6]	32.75/0.9090	29.30/0.8215	28.41/0.7863	26.24/0.7989	30.48/0.9117
	FSRCNN [7]	33.18/0.9140	29.37/0.8240	28.53/0.7910	26.43/0.8080	31.10/0.9210
	VDSR [8]	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279	32.01/0.9340
	DRCN [11]	33.82/0.9226	29.96/0.8311	28.80/0.7963	27.53/0.8276	32.66/0.9343
	LapSRN [10]	33.81/0.9220	29.79/0.8325	28.92/0.7980	27.07/0.8275	32.21/0.9350
	DRRN [12]	34.03/0.9244	29.99/0.8349	28.95/0.8004	27.53/0.8378	32.71/0.9379
	MemNet [40]	34.09/0.9248	30.00/0.8350	28.96/0.8001	27.56/0.8376	32.51/0.9369
	IDN [41]	34.11/0.9253	29.99/0.8354	28.95/0.8013	27.42/0.8359	32.71/0.9381
	EDSR (B) [9]	34.37/0.9270	30.28/0.8417	29.09/0.8052	28.15/0.8527	33.45/0.9439
	SRMDNF [42]	34.12/0.9542	30.04/0.8382	28.97/0.8025	27.57/0.8398	33.00/0.9403
	CARN [43]	34.29/0.9542	30.29/0.8407	29.06/0.8034	28.06/0.8493	33.50/0.9440
	Ms-LapSRN [44]	33.88/0.9230	29.89/0.8340	28.87/0.8000	27.23/0.8310	32.28/0.9360
Ours	34.36/0.9542	30.30/0.8412	29.07/0.8045	28.14/0.8514	33.50/0.9439	
$\times 4$	Bicubic	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577	24.89/0.7866
	SRCNN [6]	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221	27.58/0.8555
	FSRCNN [7]	30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280	27.90/0.8610
	VDSR [8]	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524	28.83/0.8870
	DRCN [11]	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510	28.93/0.8854
	LapSRN [10]	31.54/0.8842	28.09/0.7700	27.32/0.7275	25.21/0.7562	29.09/0.8900
	DRRN [12]	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638	29.45/0.8946
	MemNet [40]	31.74/0.8893	38.26/0.7723	27.40/0.7281	25.50/0.7630	29.42/0.8942
	IDN [41]	31.82/0.8903	28.25/0.7730	27.41/0.7297	25.41/0.7632	29.41/0.8942
	EDSR (B) [9]	32.09/0.8938	28.58/0.7813	27.57/0.7357	26.04/0.7849	30.35/0.9067
	SRMDNF [42]	31.96/0.8925	28.35/0.7787	27.49/0.7337	25.68/0.7731	30.09/0.9024
	CARN [43]	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	30.47/0.9084
	Ms-LapSRN [44]	31.62/0.8870	28.16/0.7720	27.36/0.7290	25.32/0.7600	29.18/0.8920
Ours	32.13/0.8941	28.58/0.7816	27.56/0.7356	26.09/0.7859	30.42/0.9074	

Our ARCNet is trained on DIV2K [34] dataset. DIV2K dataset is a novel dataset for SISR problem. There are 1000 images with up to 2K resolution from real world. In DIV2K, there are 800 images for training, 100 images for validation, and 100 images for testing. In this paper, we train our ARCNet with 800 training images and validate the model with 5 images. The paired training data are cropped with resolution of $48 * 48$ for LR patches. The batch size for training is set as 20. The model is updated for 1000 epochs by Adam optimizer. The learning rate of Adam optimizer is set as $lr = 10^{-4}$ and halved for every 200 iterations.

Five testing benchmarks are used to evaluate the network performance. They are Set5 [35], Set14 [36], B100 [37], Urban100 [38], and Manga109 [39]. The evaluation indexes are chosen as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM).

4.1.2. Experiment Results. We compare our ARCNet with some SISR works: SRCNN [6] and FSRCNN [7]. The quantitative PSNR/SSIM comparisons are shown in Table 2.

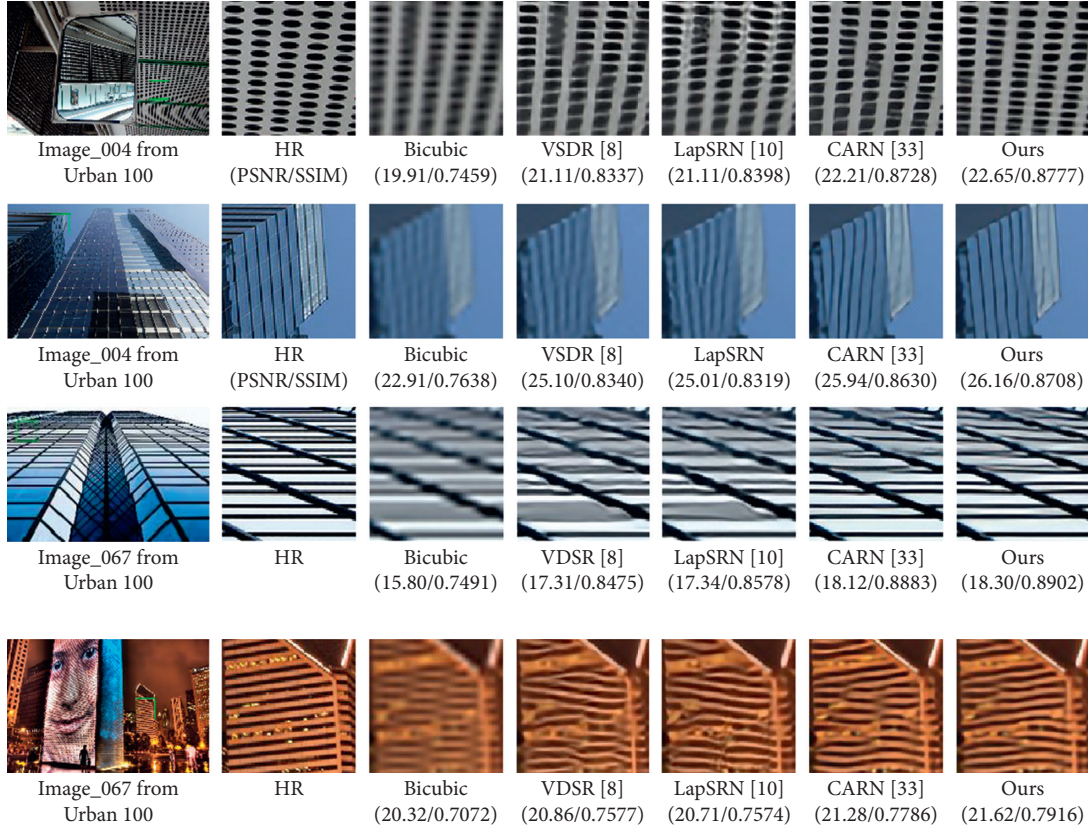


FIGURE 5: Visual quality comparisons of different methods with $B1 \times 4$ degradation. The images are chosen from Urban100 dataset. From the results, ARCN could recover the structural information more effectively.

TABLE 3: Quantitative comparison on parameters and performance for scaling factor $\times 4$. Our results are shown in bold.

Model	Param	Set5	Set14	B100	Urban100	Manga109
CARN [43]	1.592 M	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837	30.47/0.9084
EDSR (B) [9]	1.518 M	32.09/0.8938	28.58/0.7813	27.57/0.7357	26.04/0.7849	30.35/0.9067
SRMDNF [42]	1.552 M	31.96/0.8925	28.35/0.7787	27.49/0.7337	25.68/0.7731	30.09/0.9024
DRCN [11]	1.774 M	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510	28.93/0.8854
Ours	1.294 M	32.13/0.8941	28.58/0.7816	27.56/0.7356	26.09/0.7859	30.42/0.9074

From Table 2, our model has achieved competitive or better performance on five benchmarks compared to other works. For Urban100 and Manga109, our ARCN achieves better performance than the others. There are high-resolution images from real world in Urban100, and Manga109 is composed of the comic book covers. From this point of view, our ARCN could recover the complex structural textures more effectively.

Visualization comparisons are shown in Figure 5. There are four images chosen from Urban100 testing benchmark to compare the performance. From the visualization, our ARCN performs better than the others on structure texture recovery.

4.2. Model Analysis

- (1) Study on parameters: From the design, our proposed flexible upscale block could save the parameters. To show the comparison on parameter and performance,

we test the model on five benchmarks. The quantitative results are shown in Table 3. We compare our ARCN with several recent lightweight works for SISR problem: CARN [43], EDSR-baseline [9], SRMDNF [42], and DRCN [11]. The results show that our ARCN could achieve competitive or better performance with fewer parameters. There are around 18.7% parameters off in our ARCN with similar performance.

- (2) Study on adaptive factors: To demonstrate the effect of adaptive factors, we illustrate the learned features from two different parts of ARCB. As shown in Figure 6, (a) denotes the features processed from main path; (b) denotes the input feature from shortcut. The shortcut contains the origin information of input features, while the processed feature concentrates on the high-frequency information on features. After the adaptive fusion, the high-frequency information will be enhanced by aggregation.

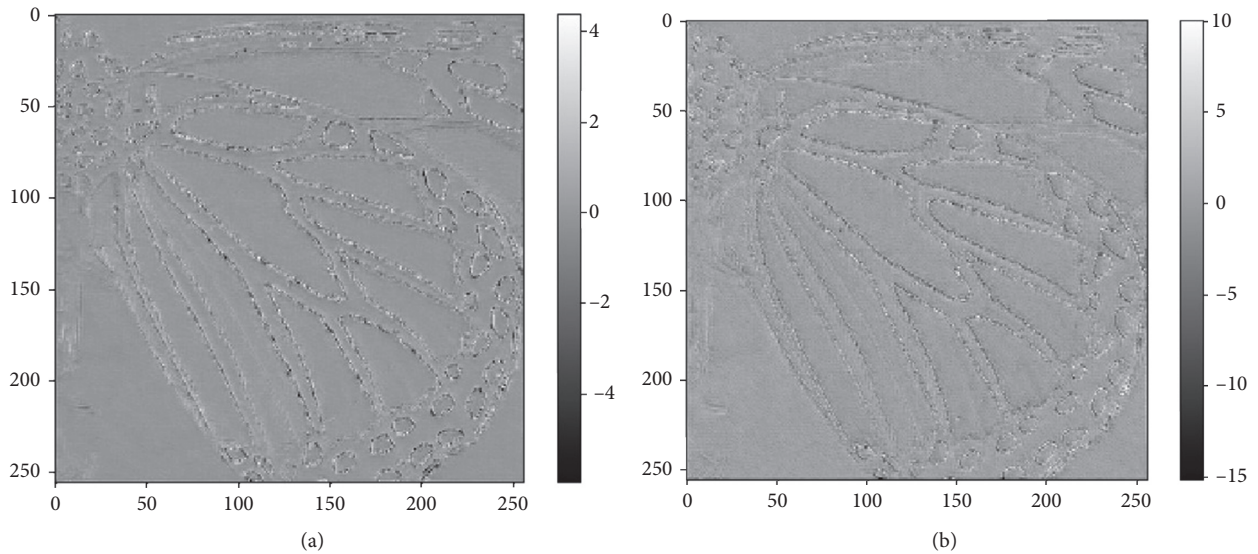


FIGURE 6: Illustration of different features from ARCB. (a) Processed feature. (b) Shortcut.

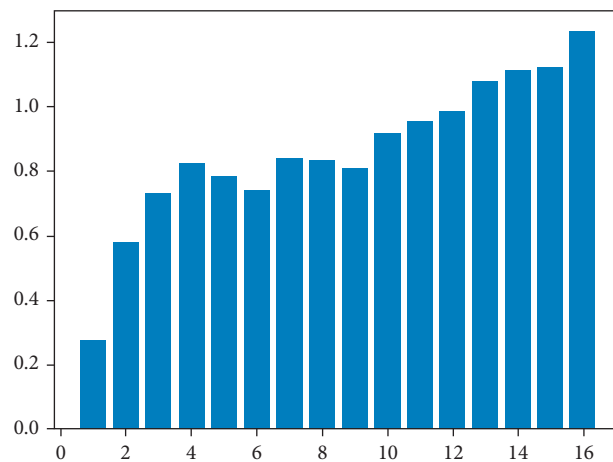


FIGURE 7: Illustration of learned W_α of different ARCBs.

TABLE 4: Quantitative PSNR/SSIM comparison on adaptive weights for scaling factor $\times 4$. Our results are shown in bold.

Weight	Set5	Set14	B100	Urban100	Manga109
w/o	32.13/0.8941	28.58/0.7816	27.56/0.7356	26.09/0.7859	30.42/0.9074
W	32.09/0.8931	28.48/0.7794	27.50/0.7337	25.88/0.7801	30.20/0.9044

TABLE 5: Comparisons on parameters of two upscale blocks with different scaling factors.

Scaling factor	$\times 2$	$\times 3$	$\times 4$	$\times 5$	$\times 6$	$\times 7$	$\times 8$
Cascading	119443	334083	297115	924931	481795	1811203	44867
Proposed	6924	11579	27696	43275	62316	84819	110784

Furthermore, to demonstrate the learned weights of different ARCBs, we show W_α in Figure 7. The x -axis denotes the depth of ARCBs. The y -axis denotes the learned weights. From the figures, with the increase of network depth, the exploited high-frequency information will be more important with larger W_α . This demonstrates the capacity of adaptive residual

learning, which learns a flexible weight for different level of information. A quantitative comparison of PSNR/SSIM with and without W_α is shown in Table 4. From the results, W_α acts as an important role for performance improvement.

- (3) Study on upscale block: In this paper, we propose an efficient upscale block for arbitrary upscaling factors.

As a substitution, the vanilla cascading design and proposed efficient one hold competitive performance. However, the proposed upscale block holds a much smaller number of parameters. A comparison of two upscale blocks with different scaling factors is shown in Table 5. From the table, proposed blocks have much fewer parameters with less computation cost for upscaling the images.

5. Conclusion

In this paper, we proposed a novel adaptive residual channel attention network named ARCN for single image super-resolution (SISR) problem. In the proposed ARCN, adaptive residual channel attention block (ARCB) was designed for better performance. Mixture factors in ARCB were learned while training, which weighted the information from two paths in blocks adaptively. Channel attention mechanism was introduced to ARCB for distributing the importance among different channels. Besides ARCB, a tiny but flexible upscale block design was proposed for different scaling factors. Experimental results showed that our proposed ARCN could not only achieve competitive or better performance with fewer parameters than other lightweight works but also recover the complex structural textures more effectively.

In the future, more reference-free perceptual assessments will be performed to demonstrate the network performance. Furthermore, more experiments will be conducted on real-world datasets.

Data Availability

The image and quantitative comparison data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was supported by the Science Foundation of Shenyang University of Chemical Technology (LQ2020020).

References

- [1] J.-e. Liu and F.-P. An, "Image classification algorithm based on deep learning-kernel function," *Scientific Programming*, vol. 2020, Article ID 7607612, 2020.
- [2] E. Torti, M. Musci, F. Guareschi, F. Leporati, and M. Piastra, "Deep recurrent neural networks for edge monitoring of personal risk and warning situations," *Scientific Programming*, vol. 2019, Article ID 9135196, 2019.
- [3] H.-F. Ke, C.-B. Lu, X.-B. Li, G.-Y. Zhang, Y. Mei, and X.-W. Shen, "An incremental optimal weight learning machine of single-layer neural networks," *Scientific Programming*, vol. 2018, Article ID 3732120, 2018.
- [4] W. Tong, "A hybrid algorithm framework with learning and complementary fusion features for whale optimization algorithm," *Scientific Programming*, vol. 2020, Article ID 5684939, 2020.
- [5] X. Xu, Y. Lei, and F. Yang, "Railway subgrade defect automatic recognition method based on improved faster r-cnn," *Scientific Programming*, vol. 2018, Article ID 4832972, 2018.
- [6] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proceedings of the European Conference on Computer Vision*, Springer, Munich, Germany, pp. 184–199, September 2014.
- [7] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proceedings of the European Conference on Computer Vision*, Springer, Munich, Germany, pp. 391–407, September 2016.
- [8] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, June 2016.
- [9] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 136–144, Chongqing, China, November 2017.
- [10] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 624–632, Honolulu, HI, USA, July 2017.
- [11] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proceedings of the European Conference on Computer Vision*, Springer, Munich, Germany, pp. 597–613, September 2016.
- [12] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3147–3155, Las Vegas, NV, USA, June 2017.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [14] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2472–2481, Salt Lake City, UT, USA, June 2018.
- [15] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 286–301, Munich, Germany, September 2018.
- [16] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.
- [17] Z. Hui, X. Gao, Y. Yang, and X. Wang, "Lightweight image super-resolution with information multi-distillation network," in *Proceedings of the 27th ACM International Conference on Multimedia*, New York, NY, USA, October 2019.
- [18] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11065–11074, Long Beach, CA, USA, June 2019.
- [19] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, and J. Jiang, "Edge-enhanced gan for remote sensing image super-resolution,"

- IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5799–5812, 2019.
- [20] Z. Shao, L. Wang, Z. Wang, and J. Deng, “Remote sensing image super-resolution using sparse representation and coupled sparse autoencoder,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 8, pp. 2663–2674, 2019.
- [21] W. Shi, J. Caballero, F. Huszar et al., “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.
- [22] X. Mao, C. Shen, and Y.-B. Yang, “Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections,” in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., Curran Associates, Inc., New York, NY, USA, 2016.
- [23] K. Jiang, Z. Wang, P. Yi, and J. Jiang, “Hierarchical dense recursive network for image super-resolution,” *Pattern Recognition*, vol. 107, p. 107475, 2020.
- [24] P. Yi, Z. Wang, K. Jiang, Z. Shao, and J. Ma, “Multi-temporal ultra dense memory network for video super-resolution,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2503–2516, 2020.
- [25] Z. Wang, P. Yi, K. Jiang et al., “Multi-memory convolutional neural network for video super-resolution,” *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2530–2544, 2019.
- [26] J. Ma, X. Wang, and J. Jiang, “Image super-resolution via dense discriminative network,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 7, pp. 5687–5695, 2020.
- [27] L. Zhou, Z. Wang, Y. Luo, and Z. Xiong, “Separability and compactness network for image recognition and super-resolution,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3275–3286, 2019.
- [28] T. Tong, G. Li, X. Liu, and Q. Gao, “Image super-resolution using dense skip connections,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4799–4807, Venice, Italy, October 2017.
- [29] C. Ledig, L. Theis, F. Huszar et al., “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 105–114, Las Vegas, NV, USA, June 2017.
- [30] J. Li, F. Fang, K. Mei, and G. Zhang, “Multi-scale residual network for image super-resolution,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 517–532, Munich, Germany, September 2018.
- [31] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu, “Spatial transformer networks,” in *Advances in Neural Information Processing Systems*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., Curran Associates, Inc., 2015, Curran Associates, Inc., New York, NY, USA.
- [32] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, Las Vegas, NV, USA, June 2018.
- [33] W. Wei, J. Yongbin, L. Yanhong, L. Ji, W. Xin, and Z. Tong, “An advanced deep residual dense network (drdn) approach for image super-resolution,” *International Journal of Computational Intelligence Systems*, vol. 12, pp. 1592–1601, 2019.
- [34] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 114–125, Honolulu, HI, USA, July 2017.
- [35] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, *Low-complexity Single-image Super-resolution Based on Nonnegative Neighbor Embedding*, Marco Bevilacqua, Miami, FL, USA, 2012.
- [36] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” in *Proceedings of the International Conference on Curves and Surfaces*, Springer, Paris, France, pp. 711–730, April 2010.
- [37] 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,” in *Proceedings of the Eighth IEEE International Conference on Computer Vision. ICCV*, vol. 2, IEEE, British Columbia, Canada, pp. 416–423, July 2001.
- [38] J.-B. Huang, A. Singh, and N. Ahuja, “Single image super-resolution from transformed self-exemplars,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5197–5206, Las Vegas, NV, USA, June 2015.
- [39] Y. Matsui, K. Ito, Y. Aramaki et al., “Sketch-based manga retrieval using manga109 dataset,” *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21811–21838, 2017.
- [40] Y. Tai, J. Yang, X. Liu, and C. Xu, “Memnet: a persistent memory network for image restoration,” in *Proceedings of the IEEE international Conference on Computer Vision*, pp. 4539–4547, Venice, Italy, October 2017.
- [41] Z. Hui, X. Wang, and X. Gao, “Fast and accurate single image super-resolution via information distillation network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 723–731, Salt Lake City, UT, USA, June 2018.
- [42] K. Zhang, W. Zuo, and L. Zhang, “Learning a single convolutional super-resolution network for multiple degradations,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3262–3271, Salt Lake City, UT, USA, June 2018.
- [43] N. Ahn, B. Kang, and K.-A. Sohn, “Fast, accurate, and lightweight super-resolution with cascading residual network,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, September 2018.
- [44] W. Lai, J. Huang, N. Ahuja, and M. Yang, “Fast and accurate image super-resolution with deep laplacian pyramid networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2599–2613, 2019.