Neurocomputing 382 (2020) 116–126

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Wavelet-based residual attention network for image super-resolution

Shengke Xue*, Wenyuan Qiu, Fan Liu, Xinyu Jin

College of Information Science and Electronic Engineering, Zhejiang University, No. 38 Zheda Road, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 20 July 2019
Revised 2 October 2019
Accepted 10 November 2019
Available online 4 December 2019

Communicated by Dr C Chen

Keywords:
Super-resolution
Wavelet transform
Multi-kernel convolution
Channel attention
Spatial attention

ABSTRACT

Image super-resolution (SR) is a fundamental technique in the field of image processing and computer vision. Recently, deep learning has witnessed remarkable progress in many super-resolution approaches. However, we observe that most studies focus on designing deeper and wider architectures to improve the quality of image SR at the cost of computational burden and speed. Few researches adopt lightweight but effective modules to improve the efficiency of SR without compromising its performance. In this paper, we propose the Wavelet-based residual attention network (WRAN) for image SR. Specifically, the input and label of our network are four coefficients generated by the two-dimensional (2D) Wavelet transform, which reduces the training difficulty of our network by explicitly separating low-frequency and high-frequency details into four channels. We propose the multi-kernel convolutional layers as basic modules in our network, which can adaptively aggregate features from various sized receptive fields. We adopt the residual attention block (RAB) that contains channel attention and spatial attention modules. Thus, our method can focus on more crucial underlying patterns in both channel and spatial dimensions in a lightweight manner. Extensive experiments validate that our WRAN is computationally efficient and demonstrate competitive results against state-of-the-art SR methods.

1. Introduction

Single image super-resolution (SR), defined as restoring high-resolution (HR) images from corresponding low-resolution (LR) versions, is an important technique of image processing in computer vision community. Deep learning-based image SR has accumulated substantial attention in recent years [1–7]. It has been extended to various real-world applications, e.g., medical imaging [8–12], video surveillance [13–17], remote sensing [18,19], prerequisite for image classification [20,21], detection [22], recognition [23,24], and denoising [25].

Image SR is notoriously challenging, known as inherently ill-posed, because a specific LR image corresponds to multiple HR counterparts. This leads to the solution to SR being usually intractable. In practice, the exact LR image of each HR image generally does not exist. Thus, we assume that the downsampled HR image by the bicubic interpolation [26] is considered as the LR version. From this perspective, some fine details are inevitably lost and image SR is regarded as its inverse process.

Many SR methods take the pre-upsampling scheme, i.e., LR images are interpolated to the same size as HR images before passed to networks for training [27,28]. The advantage of this manner is that upsampling can be done by traditional methods (e.g., bicubic interpolation) at the beginning of networks. This can significantly alleviate the training difficulty. In addition, these models can process input images with arbitrary (even fractional) scale factors, without modifying their structures. However, the pre-upsampling scheme has some drawbacks, e.g., noise amplification and distortion. Because most operations are executed in HR space, the computational burden (time and memory) is much heavier than those post-upsampling methods that use specific learnable layers (e.g., deconvolution or sub-pixel convolution [29]) at the end of their networks [30–32].

To remedy this defect, we further apply the Wavelet transform, which is considered as an efficient operator by decomposing images into high-frequency (HF) texture details and low-frequency (LF) information. Specifically, an LR image is first upscaled to a coarse HR image with desired dimension by using bicubic interpolation. Then we exploit the Wavelet transform to obtain four coefficients. In this manner, the HF and LF components are explicitly separated to four channels, which is helpful to train our model. In addition, given that $\boldsymbol{I}_{\text{bic}} \in \mathbb{R}^{h \times w}$, after the Wavelet transform, the results are $\boldsymbol{I}_{\text{w}} \in \mathbb{R}^{\frac{h}{2} \times \frac{w}{2} \times 4}$, which means that the image is halved in both directions, then grouped into four channels. Empirically, four half-sized images are easier to train than a large image. Note that the Wavelet transform and its inverse operation are both

* Corresponding author.
E-mail addresses: xueshengke@zju.edu.cn (S. Xue), qiuwenyuan@zju.edu.cn (W. Qiu), flyingliufan@zju.edu.cn (F. Liu), jinxinyuzju@gmail.com (X. Jin).

https://doi.org/10.1016/j.neucom.2019.11.044
0925-2312/

invertible, leading to no information loss. [33–35] have already introduced the Wavelet transform to image SR problems.

Generally, convolutional neural networks (CNNs) are able to effectively extract hierarchical features from input data. Inspired by the inception framework [36], we propose a new type structure for multi-scale feature extraction, namely multi-kernel convolutional layer, as the basic module in our model. We use different kernel sizes (1×1, 3×3, and 5×5) for convolution to extract features through multiple paths simultaneously. Then features from these paths are further aggregated, so that diverse patterns from different receptive fields can be fused to further improve SR performance.

Recently, some studies focus on the inter-channel relationship in CNNs. Hu et al. [37] proposed "squeeze-and-excitation" (SE) block that adjusted channel-wise feature representations by explicitly modeling interactions between channels. Due to its substantial performance advance in terms of classification accuracy with little extra computational cost, Zhang et al. [38] introduced the SE block to image SR and remarkably enhanced the representation ability of their model, leading to impressive improvement of the SR results. Inspired by the SE block [37], Woo et al. proposed CBAM [39] that not only upgraded the SE block to the channel attention but further suggested the spatial attention. In this manner, CBAM plays a crucial role in deciding "which channel" and "where" in features to focus on. However, to the best of our knowledge, CBAM achieved significant progresses only on detection and classification tasks yet.

In this paper, we first integrate this framework to image SR tasks. We refer it as the residual attention block (RAB). Similar to SE block and CBAM, our RAB is also lightweight; i.e., it adds negligible computational burden to our model. To this end, our approach can adaptively enhance desired features and suppress unnecessary ones in both channel and spatial dimensions. This can effectively improve the reconstruction quality of image SR.

The major contributions of this paper include:

- We use the four coefficients generated by the 2D Wavelet transform as the input, so that the coarse contents and sharp details are separated explicitly before training. This can help alleviate the learning difficulty of our network without information loss. Correspondingly, our model generates four channels, followed by the 2D inverse Wavelet transform that produces the intact residual image.
- We adopt the multi-kernel convolutional layer as a basic module. Specifically, feature maps go through multiple paths that have convolutional layers with different kernel sizes, to extract diverse underlying patterns from various sized receptive fields. Then they are concatenated along the channel dimension and aggregated to the same width as the input of this module.
- We exploit the residual attention block, which includes channel attention and spatial attention modules, for adaptive feature refinement in channel and spatial dimensions. They can be seamlessly integrated to general CNNs and add trivial computational cost. In addition, residual attention blocks are quite suitable for Wavelet sub-bands, since they inherently contain various types of features.

2. Related work

2.1. Wavelet transform for super-resolution

As a traditional image processing technique, Wavelet transform has been widely used for image SR. By using multi-frames, Ji and co-workers [40,41] predicted missing details in Wavelet coefficients to achieve super-resolution. Anbarjafari and Demirel [42] directly adopted bicubic interpolation to upsample Wavelet sub-bands to generate SR results. Since partial Wavelet coefficients are usually sparse, it is proper to integrate sparse coding methods to refine image details [43,44]. However, due to limited training data and inextensible mode sizes, the methods above cannot produce state-of-the-art SR results, especially compared to those deep learning-related methods that make full use of massive data and sophisticated architectures.

Guo et al. proposed DWSR [33], which combined Wavelet transform with ResNet [45]. It predicted the residual Wavelet sub-bands rather than HR counterparts, since the sparse output would help stable training and robust convergence. Huang et al. proposed Wavelet-SRNet [34] for face super-resolution. It developed a unified framework for multiple scale factors during training and presented Wavelet prediction loss for extra supervision. Inspired by U-Net [46], Liu et al. proposed MWCNN [35], where pooling (unpooling) operations were replaced by (inverse) 2D Wavelet transform. Since Wavelet transform is invertible, it is guaranteed that no information loss occurs after this operator. However, a large number of Wavelet transform operations excessively decompose feature maps, leading to numerous redundant channels, which may not facilitate gradient backpropagation during training. Detailed discussion will be presented in Section 3.6.

2.2. Deep learning-based super-resolution

Deep learning has revealed remarkable progress in image super-resolution. Dong et al. first proposed SRCNN [27], which was a simple three-layer CNN trained in an end-to-end manner. It outperforms traditional SR methods remarkably in terms of the quality of reconstructed images. Empirically, increasing the depth and width of a CNN will improve its representation power. Kim et al. proposed VDSR [28], using 20 convolutional layers to obtain larger receptive fields. Residual learning and gradient clipping strategies were adopted to help stable training and avoid gradient vanishing issues. DRCN [47] also developed a deep structure with recursive learning, sharing parameters by 16 times, to effectively extract more high-level contextual information. To reduce computation burden, Lai et al. proposed LapSRN [30] that adopted the Laplacian pyramid structure to gradually upscale intermediate images with additional supervision. Based on DenseNet [48], MemNet [31] replaced normal convolutions with recursive nets. Tai et al. declared that local connections in a block indicated short-term memory, and incoming connections from previous blocks indicated long-term memory. To simplify the structure of SR networks, EDSR [49] discarded batch normalization layers. Lim et al. reported that batch normalization impeded the quality of image SR, though it has been proven to be effective in other high-level vision tasks, e.g., classification, detection, and segmentation. Thus, Lim et al. designed a 64-layer network with 256 filters to further boost the learning capability of EDSR. In addition, to reuse computation, the $\times 3/\times 4$ scale model adopted the pre-trained $\times 2$ model as initialization rather than training it from scratch. Haris et al. proposed DBPN [50], which consisted of pairs of iterative up-/down-sampling layers. They creatively offered a feedback scheme that allowed projection errors to transfer between LR and HR spaces. To make full use of intermediate feature maps, RDN [32] integrated residual learning and dense connection strategies in local and global manners. With multiple skip connections inside the network, hierarchical features could be adaptively fused and gradient information could sufficiently propagate through the network. Except SRCNN, these aforementioned methods inevitably possess a huge amount of parameters and require substantial memory and time to train them.

2.3. Attention mechanism

Similar to human perception [51,52], attention mechanism plays a significant role in computer vision tasks. One crucial

characteristic of human visual system is that one does not deal with the entire view at once. In contrast, humans exploit a series of incomplete glimpses to selectively capture salient visual information better [53]. Generally, attention is considered as a guidance to reallocate available computation resources to the most informative parts of an input [37].

Recently, several studies incorporated attention mechanisms to improve the performance of CNNs in high-level vision tasks. Wang et al. first proposed the residual attention network [54] which adopted an encoder–decoder style attention scheme, for classification. The bottom-up/top-down architecture created a soft mask to multiply, thus adaptively enhancing or suppressing features. Hu et al. proposed the SE block [37], where the primitive channel-wise attention was used after residual blocks to improve classification accuracy. To consider where to concentrate on feature maps in spatial dimension, Woo et al. proposed CBAM [39], which presented channel and spatial attention modules that could be easily incorporated to mainstream CNNs. Their model was quite effective for image detection tasks on large-scale datasets.

However, only a few studies adopted attention structures for image SR. Zhang et al. proposed RCAN [38] that put channel attention modules inside residual blocks, so that abundant high-frequency features were captured and could be adaptively rescaled according to the inter-dependencies among channels. Though acquiring positive effect, RCAN contains a large number of channel attention modules, which were used excessively in their model. In addition, RCAN lacks successive convolutional layers to extract hierarchical features, which is considered as the basic function in general CNNs.

3. Proposed method

3.1. Wavelet transform

As a traditional image processing technique, Wavelet transform is widely used for image analysis. Fig. 2 shows the basic principle of the 2D discrete Wavelet transform. Image $\boldsymbol{X}$ is first passed through a low-pass filter G_{L} and a high-pass filter G_{H}, then half downsampled along columns. After that, two paths are both passed to low-pass and high-pass filters respectively, and further half downsampled along rows. Finally, the output are four coefficients, denoted as {$\boldsymbol{A}$, $\boldsymbol{V}$, $\boldsymbol{H}$, $\boldsymbol{D}$}. In this paper, we use the "Haar" kernel throughout our experiments.

Fig. 3 illustrates an example of 2D (inverse) Wavelet transform with "Haar" kernel. As we can see, image $\boldsymbol{X}$ is decomposed to four sub-bands: $\boldsymbol{A}$, $\boldsymbol{V}$, $\boldsymbol{H}$, and $\boldsymbol{D}$, which comprise average, vertical, horizontal, and diagonal information details from the original image. In addition, each sub-band is half the size of $\boldsymbol{X}$. Note that the Wavelet transform and its inverse operation are both invertible, leading to no information loss. Thus, our model can easily generate the intact residual image by using the inverse Wavelet transform.

With the help of the Wavelet transform, our model learns to predict four half-sized channels, which approximate four coefficients produced by the Wavelet transform on the residual SR image. Since diverse underlying patterns are preserved in four channels instead of one big image, the learning difficulty of our model can be significantly reduced. In Section 3.6, Fig. 7 reveals that it is not necessary to use multiple Wavelet transform operations in our network.

3.2. Network structure

Fig. 1 shows the overall structure of our network. The input of our model $\boldsymbol{I}^{\mathrm{W}}_{\mathrm{bic}} \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times 4}$ are the four coefficients of the 2D Wavelet transform applied on the bicubic LR image $\boldsymbol{I}_{\mathrm{bic}} \in \mathbb{R}^{h\times w}$. As mentioned earlier, they are half sized in both row and column dimensions, and are grouped into four channels before training. First, we use a convolutional layer to extract shallow features from the input:

$$\boldsymbol{F}_0 = f_{\mathrm{ext}}(\boldsymbol{I}^{\mathrm{W}}_{\mathrm{bic}}) = \sigma(C(\boldsymbol{I}^{\mathrm{W}}_{\mathrm{bic}}, 5\times 5, c), \alpha) \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times c}, \tag{1}$$

where $C(\cdot)$ denotes a convolutional layer, in which 5×5 is the kernel size and channel $c = 64$; $\sigma(\cdot)$ denotes a leaky rectified linear unit (ReLU) layer. Since $\boldsymbol{I}^{\mathrm{W}}_{\mathrm{bic}}$ inherently includes negative pixels due to the property of the Wavelet transform, we adopt leaky ReLU (negative slope $\alpha = 0.1$) for non-linear activation. Note that bias is omitted for brief notations.

The main body of our model consists of L successive identical blocks, each of which contains a multi-kernel convolutional layer, a channel attention module, and a spatial attention module. In each block, we use local skip connections to help the information flow. Thus, we have

$$\begin{aligned}\boldsymbol{F}_{i+1} &= H_{i+1}(\boldsymbol{F}_i)\\ &= f_{\mathrm{spa}}(f_{\mathrm{chn}}(f_{\mathrm{conv}}(\boldsymbol{F}_i))) + \boldsymbol{F}_i,\ \ i = 0, 1, \ldots, L-1,\end{aligned} \tag{2}$$

where $f_{\mathrm{conv}}(\cdot)$, $f_{\mathrm{chn}}(\cdot)$, and $f_{\mathrm{spa}}(\cdot)$ denote the function of multi-kernel convolution, channel attention, and spatial attention, respectively. Note that the output of each block has the same dimension as its input.

To overcome the ubiquitous gradient vanishing issue in deep network structures, we concatenate all outputs of these blocks along channel dimension, i.e.,

$$\boldsymbol{F}_{\mathrm{cat}} = [\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_L] \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times cL}. \tag{3}$$

To this end, feature maps from shallow layers to deep layers are sufficiently exploited in forward computation, and gradient information can be effectively back-propagated to the front of the network. Empirically, this framework can achieve better convergence in training.

After fusing features from several blocks, we further shrink these feature maps to a compact size by using the bottleneck structure, which includes two 3×3 convolutional layers and one leaky ReLU activation layer, as shown in Fig. 1:

$$\boldsymbol{F}_{\mathrm{w}} = C(\sigma(C(\boldsymbol{F}_{\mathrm{cat}}, 3\times 3, c), \alpha), 3\times 3, 4) \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times 4}. \tag{4}$$

The goal of training our network is that the output $\boldsymbol{F}_{\mathrm{w}}$ can well approximate the four coefficients of the Wavelet transform on the real residual image $\boldsymbol{I}_{\mathrm{HR}} - \boldsymbol{I}_{\mathrm{bic}}$. Equivalently, we have

$$\boldsymbol{I}_{\mathrm{HR}} \simeq \mathrm{idWT}(\boldsymbol{F}_{\mathrm{w}}) + \boldsymbol{I}_{\mathrm{bic}}, \tag{5}$$

where $\mathrm{idWT}(\cdot)$ denotes the inverse discrete Wavelet transform function. Note that we adopt global residual learning by adding a long skip connection from $\boldsymbol{I}_{\mathrm{bic}}$ to the end of our model, so that our network learns to predict residual components rather than the HR image directly. As common strategies, this helps robust training and fast convergence.

3.3. Multi-kernel convolutional layer

Inspired by the inception structure [36], we adopt the multi-kernel convolutional layers as our basic modules, as shown in Fig. 4. Specifically, the input goes through four paths, which are convolutional layers with different kernel sizes (1×1, 3×3, 5×5, and {3×3, 5×5}):

$$\begin{cases}\boldsymbol{F}^1_{\mathrm{in}} = \sigma(C(\boldsymbol{F}_{\mathrm{in}}, 1\times 1, c), \alpha),\\ \boldsymbol{F}^2_{\mathrm{in}} = \sigma(C(\boldsymbol{F}_{\mathrm{in}}, 3\times 3, c), \alpha),\\ \boldsymbol{F}^3_{\mathrm{in}} = \sigma(C(\boldsymbol{F}_{\mathrm{in}}, 5\times 5, c), \alpha),\\ \boldsymbol{F}^4_{\mathrm{in}} = \sigma(C(\sigma(C(\boldsymbol{F}_{\mathrm{in}}, 3\times 3, c), \alpha), 5\times 5, c), \alpha).\end{cases} \tag{6}$$

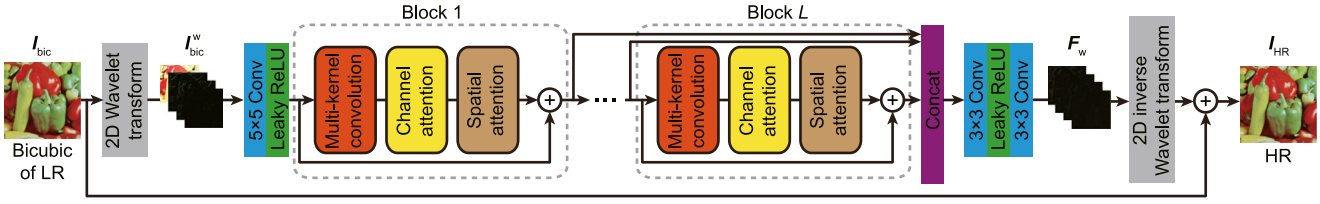

Fig. 1. Overall structure of our Wavelet-based residual attention network.

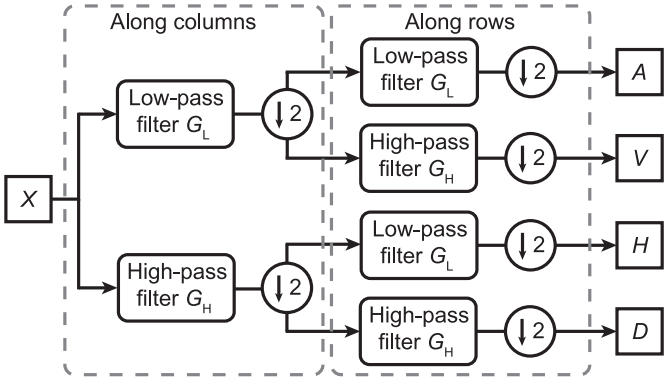

Fig. 2. Process of the 2D discrete Wavelet transform.

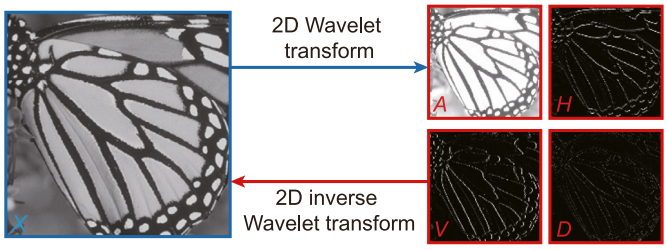

Fig. 3. Example of the 2D discrete Wavelet transform. ***X*** is decomposed into four sub-bands: ***A*** (average); ***H*** (horizontal); ***V*** (vertical); ***D*** (diagonal). They are half-sized than ***X*** in both row and column direction.

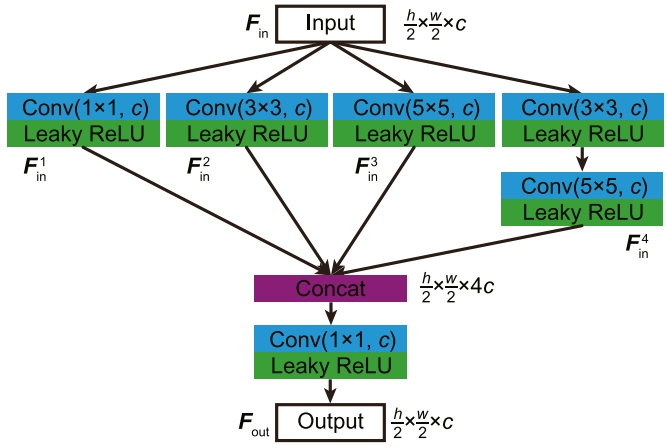

Fig. 4. Multi-kernel convolutional layer.

Then four parts are concatenated along the channel dimension, followed by a 1×1 convolutional layer that aggregates features to the same width as the input:

$$\boldsymbol{F}_{\text{out}} = \sigma(C([\boldsymbol{F}^1_{\text{in}}, \boldsymbol{F}^2_{\text{in}}, \boldsymbol{F}^3_{\text{in}}, \boldsymbol{F}^4_{\text{in}}], 1 \times 1, c), \alpha). \tag{7}$$

Each convolutional layer is accompanied by a leaky ReLU activation layer to keep non-linearity. Similar to the Inception structure [55–57] that uses the split-transform-merge strategy, our network has strong feature extraction power by using different sized receptive fields and multiple paths. Though multi-kernel learning has been widely adopted in high-level vision problems (e.g., classification, segmentation, and detection), it is seldom adopted in low-level vision issues, such as super-resolution.

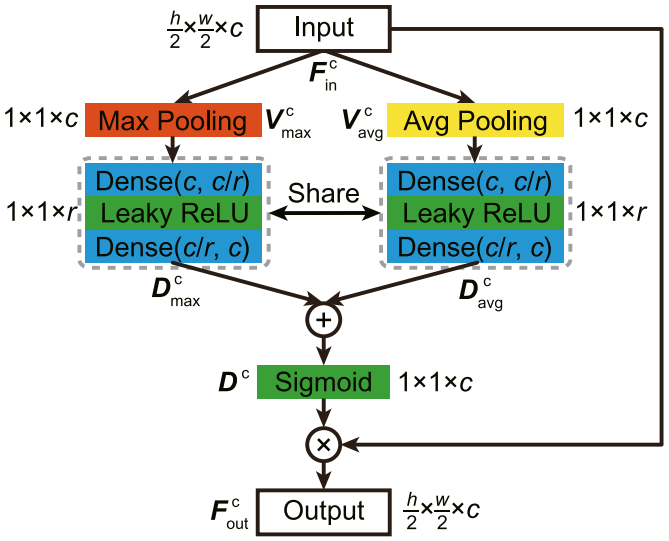

Fig. 5. Channel attention module.

3.4. Channel attention module

Fig. 5 shows the structure of the channel attention module [39], which exploits the channel inter-dependencies of feature maps. This module focuses on which channels are important in computation. Input feature $\boldsymbol{F}^c_{\text{in}}$ is squeezed by max pooling and average pooling operations respectively to aggregate contextual information in spatial dimension, generating two vectors:

$$\begin{cases} \boldsymbol{V}^c_{\max} = P(\boldsymbol{F}^c_{\text{in}}, \text{'max'}, \text{ axis=[0,1]}) \in \mathbb{R}^{1\times1\times c}, \\ \boldsymbol{V}^c_{\text{avg}} = P(\boldsymbol{F}^c_{\text{in}}, \text{'avg'}, \text{ axis=[0,1]}) \in \mathbb{R}^{1\times1\times c}, \end{cases} \tag{8}$$

where 'max' and 'avg' denote the max pooling and average pooling respectively, and axis = [0,1] means pooling is executed in the first two dimensions of feature $\boldsymbol{F}^c_{\text{in}}$. Then two vectors are forwarded to two parameter-shared fully-connected layers to acquire two feature vectors, respectively. Each value in a vector can be considered as a descriptor of its corresponding channel:

$$\begin{cases} \boldsymbol{D}^c_{\max} = W_2(\sigma(W_1(\boldsymbol{V}^c_{\max}, c/r), \alpha), c) \in \mathbb{R}^{1\times1\times c}, \\ \boldsymbol{D}^c_{\text{avg}} = W_2(\sigma(W_1(\boldsymbol{V}^c_{\text{avg}}, c/r), \alpha), c) \in \mathbb{R}^{1\times1\times c}. \end{cases} \tag{9}$$

Here, $W_1(\cdot)$ and $W_2(\cdot)$ are shared for both feature vectors. The hidden layer size is set to c/r to reduce parameter overhead, where r is the reduction ratio. In this scheme, relationships between channels can be leveraged with trivial computation. Description vectors $\boldsymbol{D}^c_{\max}$ and $\boldsymbol{D}^c_{\text{avg}}$ are merged by element-wise sum, followed by a sigmoid activation layer:

$$\boldsymbol{D}^c = \text{sig}(\boldsymbol{D}^c_{\max} + \boldsymbol{D}^c_{\text{avg}}) \in \mathbb{R}^{1\times1\times c}. \tag{10}$$

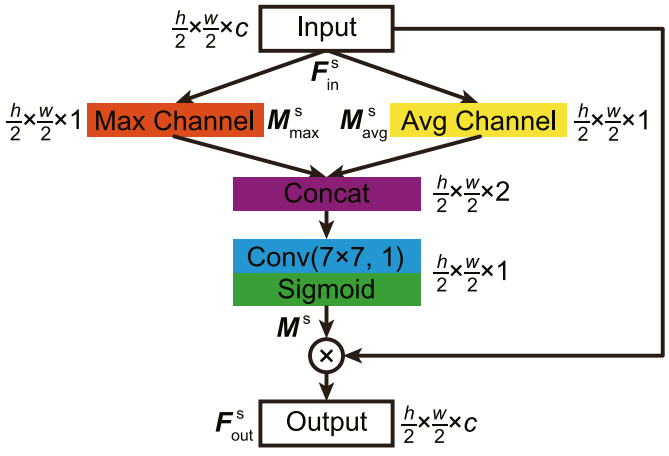

Fig. 6. Spatial attention module.

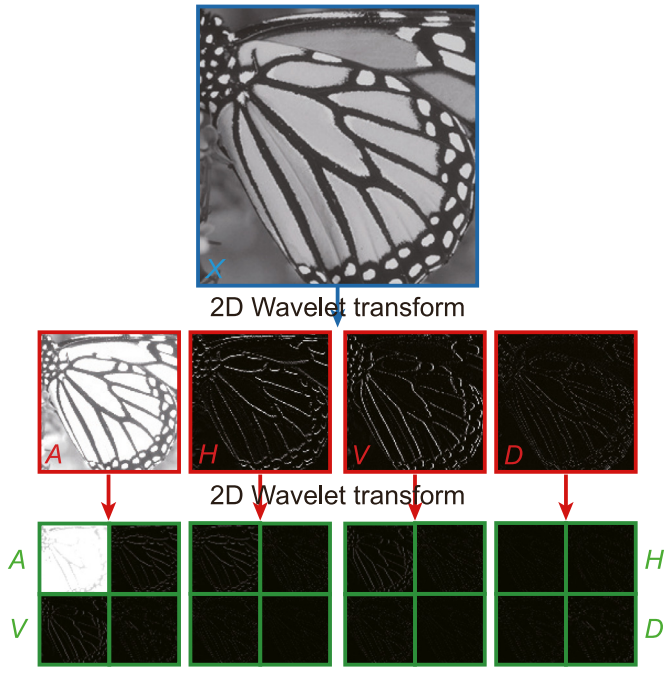

Fig. 7. Example of two Wavelet transform operations on one image. The second transform is not effective to separate images.

Finally, the description vector $\boldsymbol{D}^{\mathrm{c}}$ is applied to the input of this module by element-wise product; i.e., each descriptor multiplies one feature map, written as

$$\boldsymbol{F}^{\mathrm{c}}_{\mathrm{out}} = \boldsymbol{D}^{\mathrm{c}} \circ \boldsymbol{F}^{\mathrm{c}}_{\mathrm{in}} \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times c}, \tag{11}$$

where $\circ$ denotes the element-wise product. Note that the input and output have the same dimension. Thus, this module can be easily integrated to general CNNs.

3.5. Spatial attention module

Fig. 6 shows the structure of the spatial attention module [39], which uses the inter-spatial relationships of features. In contrast to the channel attention, spatial attention focuses on "where" is the informative part in feature maps. Input feature $\boldsymbol{F}^{\mathrm{s}}_{\mathrm{in}}$ is squeezed by max pooling and average pooling operations respectively along the channel axis, generating two 2D attention maps:

$$\begin{cases} \boldsymbol{M}^{\mathrm{s}}_{\mathrm{max}} = P(\boldsymbol{F}^{\mathrm{s}}_{\mathrm{in}}, \text{'max'}, \text{ axis=2}) \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times 1}, \\ \boldsymbol{M}^{\mathrm{s}}_{\mathrm{avg}} = P(\boldsymbol{F}^{\mathrm{s}}_{\mathrm{in}}, \text{'avg'}, \text{ axis=2}) \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times 1}. \end{cases} \tag{12}$$

Then they are concatenated and fused by a convolutional layer with 7×7 kernel size. Sigmoid function is applied to introduce non-linearity and normalize the attention map to [0,1]:

$$M^{\mathrm{s}} = \mathrm{sig}(C([\boldsymbol{M}^{\mathrm{s}}_{\mathrm{max}}, \boldsymbol{M}^{\mathrm{s}}_{\mathrm{avg}}], 7\times 7, 1)) \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times 1}. \tag{13}$$

Similar to channel attention, this attention map is finally multiplied element-wisely with the input of this module; i.e., each value in a map multiplies the elements at corresponding positions of all feature maps:

$$\boldsymbol{F}^{\mathrm{s}}_{\mathrm{out}} = \boldsymbol{M}^{\mathrm{s}} \circ \boldsymbol{F}^{\mathrm{s}}_{\mathrm{in}} \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times c}. \tag{14}$$

Note that the input and output have the same dimension. Thus, this module can be easily integrated to general CNNs.

3.6. Discussions

Many approaches adopt the mean squared error (ℓ_2 loss) as the cost function to make the SR results approximate to ground-truth HR images. However, these approaches usually produce blurry or over-smoothed outputs and miss some textual details. Compared to ℓ_2 loss that is more tolerant to small errors, mean absolute error (MAE) (ℓ_1 loss) can penalize small values efficiently and holds better convergence in the whole training stage. Thus, we adopt ℓ_1 loss to train our network:

$$\mathrm{loss} = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{N_i}\left\|\boldsymbol{I}^{(i)}_{\mathrm{SR}} - \left(\boldsymbol{I}^{(i)}_{\mathrm{HR}} - \boldsymbol{I}^{(i)}_{\mathrm{bic}}\right)\right\|_1, \tag{15}$$

where $\boldsymbol{I}^{(i)}_{\mathrm{SR}}$ and $\boldsymbol{I}^{(i)}_{\mathrm{HR}} - \boldsymbol{I}^{(i)}_{\mathrm{bic}}$ denote the i^{th} output residual image and ground-truth residual image respectively, N_i denotes the total number of pixels in the ith image, and M denotes the number of images in one training batch.

Fig. 7 illustrates an image processed by the Wavelet transform two times. Obviously, after the first Wavelet transform, the original image is apparently separated into four channels, which indicate average, vertical, horizontal, and diagonal details, respectively. However, the second Wavelet transform is not effective, since the same splitting process has been executed in the last step. Most sub-channels are close or equal to zero, which cannot help training further.

This is contrary to [35] that uses multi-level Wavelet-CNN (MWCNN). We argue that MWCNN mainly adopts the Wavelet transform to replace the pooling and unpooling operators in their fully-convolutional network, because the (inverse) Wavelet transform is intrinsically invertible (no information loss), and halves (doubles) the dimension of images directly. In our network, we only use one (inverse) Wavelet transform, and we take this as preprocess to input data (labels). Since the Wavelet transform has no trainable parameter, we do not need to compute them in the training stage.

Note that it is crucial to arrange the order three modules: multi-kernel convolution, channel attention, and spatial attention, in one block. According to CBAM [39], arranging three modules above sequentially is better than doing in parallel. In addition, they claimed that the channel-first order shows slightly better performance than the spatial-first order. Since their structure is effective only for high-vision tasks (e.g., detection and classification), we need to verify its effect to image SR. In Section 4.4, we will conduct experiments to validate the contribution of three modules, respectively.

Batch normalization (BN) proposed by Sergey et al. [58] is an effective technique to decrease inter-covariate shift in networks. Specifically, it performs on each min-batch and introduces extra parameters to retain representation capability. Because BN can re-calibrate intermediate features and alleviate the gradient vanishing

issue, it is feasible to accelerate training with higher learning rates and model is less sensitive to initialization.

However, Lim et al. [49] claimed that BN harms the scale information of images and constrains range flexibility of the model. By removing the BN layer, EDSR cuts down substantial memory and designs a large model, to improve the SR performance. In this paper, we adopt this prior experience and get rid of the BN layer in our network.

4. Experiments

Our source code is implemented by Tensorflow [59] with Keras and is available online[1]. All experiments are executed on a CentOS 7 Linux Server, equipped with two Intel Xeon(R) E5 CPUs, 256 GB memory, and two NVIDIA Tesla P4 (8 GB) GPUs.

4.1. Datasets and metrics

We use the DIV2K dataset [60] which comprises abundant 2K-resolution high quality images. To be concrete, it has been grouped into 800 training images and 100 validation images. We use total 800 images for training our network but select only 10 images for validation for speed consideration. Data argumentation scheme is adopted to prevent over-fitting. Training images are randomly rotated 90°, 180°, or 270°, and flipped horizontally or vertically. At the testing stage, we choose four commonly used benchmark datasets: Set5 [61], Set14 [62], B100 [63], and Urban100 [64]. All training images are cropped to 96 × 96 size with no overlap. The corresponding downsampled images are 48 × 48, 32 × 32, and 24 × 24, for scale factors 2, 3, and 4, respectively. After rescaled to the original size by bicubic interpolation, they are processed by the 2D Wavelet transform as the input to our network. Similarly, the outcome of the 2D Wavelet transform on the residual image (5) is used as the ground-truth for training. Note that 2D (inverse) Wavelet transform operator has no trainable parameter, so they are not included in our training process. We just prepare training data and label pairs $(\boldsymbol{I}^{\mathrm{w}}_{\mathrm{bic}}, \boldsymbol{F}_{\mathrm{w}})$ before training. By adopting smaller image size, our model is able to converge fast and requires less memory during the training stage.

As we know, some methods deal with images in RGB color space, while others operate images in YCbCr space. Currently, there is no universally accepted consensus that which scheme is better. In this paper, we execute all images on the Y channel of YCbCr space for fair comparison. Cb and Cr channels are directly upscaled by the bicubic interpolation. Combined with Y channel from SR results, we can obtain color images for better visualization.

In this paper, we use the peak signal-to-noise ratio (PSNR), which is a commonly used metric to evaluate the quality of the reconstructed image. Mathematically, PNSR is defined as follows:

$$\mathrm{MSE} = \frac{1}{N}\|\boldsymbol{I}_{\mathrm{SR}} - \boldsymbol{I}_{\mathrm{HR}}\|_{\mathrm{F}}^2, \tag{16}$$

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{P_{\max}^2}{\mathrm{MSE}}\right) \mathrm{dB}, \tag{17}$$

where $P_{\max}$ denotes the maximum pixel value (usually equals to 1.0) in an image. In addition, we adopt the structural similarity index (SSIM) [65] to measure the quality of super-resolution. It ranges from 0 to 1, where higher value is better.

[1] https://github.com/xueshengke/WRANSR-keras.

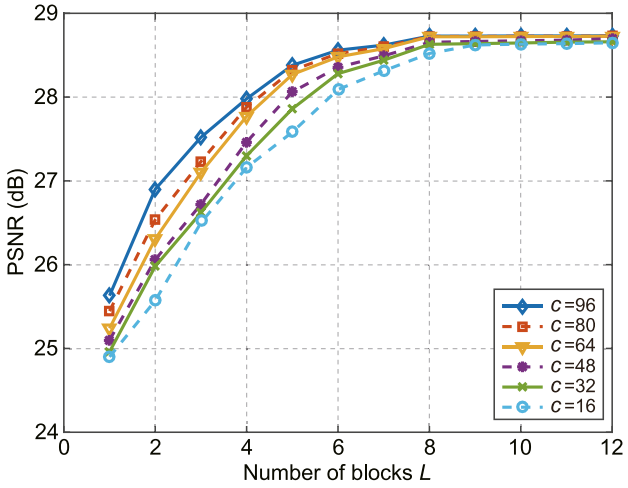

Fig. 8. Performance of our method with different values of number of blocks L and channel width c on Set14 dataset under scale factor 4.

4.2. Training details

Some training configurations are given as follows. Batch size is set to 64. The number of epochs is 200. In each epoch, 500 iterations are required to generate batches from argumentation data. We set the initial learning rate $\lambda = 1\mathrm{e}{-3}$, and it will multiply 0.1 after every 40 epochs until it reaches minimum 1e-8. Adam [66] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1\mathrm{e}{-8}$ is adopted. In addition, we deploy the early stopping paradigm; i.e., training process will terminate in advance if no improvement in terms of PSNR in validation is detected during 20 continuous epochs.

The kernel size of each convolution has been described in the previous section. Zero padding is adopted in each convolution operation to keep the sizes of feature maps of the input and the output identical. We set the number of blocks $L = 8$, the channel width $c = 64$, the negative slope of leaky ReLU $\alpha = 0.1$, and the reduction ratio $r = 4$ for the best performance. Detailed parameter selection will be reported in Section 4.3.

More hyper parameter settings can be found in our source code at Github.

4.3. Parameter selection

Fig. 8 shows the SR performance of our method with various values of number of blocks L and channel width c on Set14 under scale factor 4. As we can see, by increasing the number of blocks L, our model performs better. This is consistent with our intuition that wider and deeper networks possess stronger learning capability and therefore obtain better performance. However, when $L > 8$, PSNR becomes steady, since over deep structures are quite difficult to train and do not necessarily boost the performance. In addition, as c becomes larger, PSNR increases gradually. But setting a large c means that the number of parameters and computational burden will noticeably grow. To balance the trade-off between model size and performance, we choose $c = 64$.

Fig. 9 (a) and (b) describes the training loss curves of MAE versus epoch under various values of negative slop α and reduction ratio r, respectively. Fig. 9(a) shows that as α increases, the convergence becomes slower and the final MAE will be larger. This is because the ReLU ($\alpha = 0$) inherently has sparse effect. But as α grows, the ReLU approximates to identity function $f(x) = x$, which means the sparse effect vanishes gradually. However, we discover

Table 1
Effect of different combinations of modules in our network. Number of parameters and PSNR are given under scale factor 2 on Set5 and Set14 datasets.

Description	Different types of combinations										
Module	1	2	3	4	5	6	7	8	9	10	11
Multi-kernel convolution $\rightarrow M$	$[M]$	$\begin{bmatrix} M \\ C \end{bmatrix}$	$\begin{bmatrix} C \\ M \end{bmatrix}$	$\begin{bmatrix} M \\ S \end{bmatrix}$	$\begin{bmatrix} S \\ M \end{bmatrix}$	$\begin{bmatrix} M \\ C \\ S \end{bmatrix}$	$\begin{bmatrix} M \\ S \\ C \end{bmatrix}$	$\begin{bmatrix} C \\ S \\ M \end{bmatrix}$	$\begin{bmatrix} S \\ C \\ M \end{bmatrix}$	$\begin{bmatrix} C \\ M \\ S \end{bmatrix}$	$\begin{bmatrix} S \\ M \\ C \end{bmatrix}$
Channel attention $\rightarrow C$ Spatial attention $\rightarrow S$											
# parameters	299,392	301,520		299,491		301,619					
PSNR/dB (Set5)	36.85	37.65	37.61	37.51	37.49	**38.32**	38.26	38.29	38.19	38.13	38.10
PSNR/dB (Set14)	32.78	33.59	33.62	33.64	33.67	**34.21**	34.13	34.18	34.07	34.01	34.02

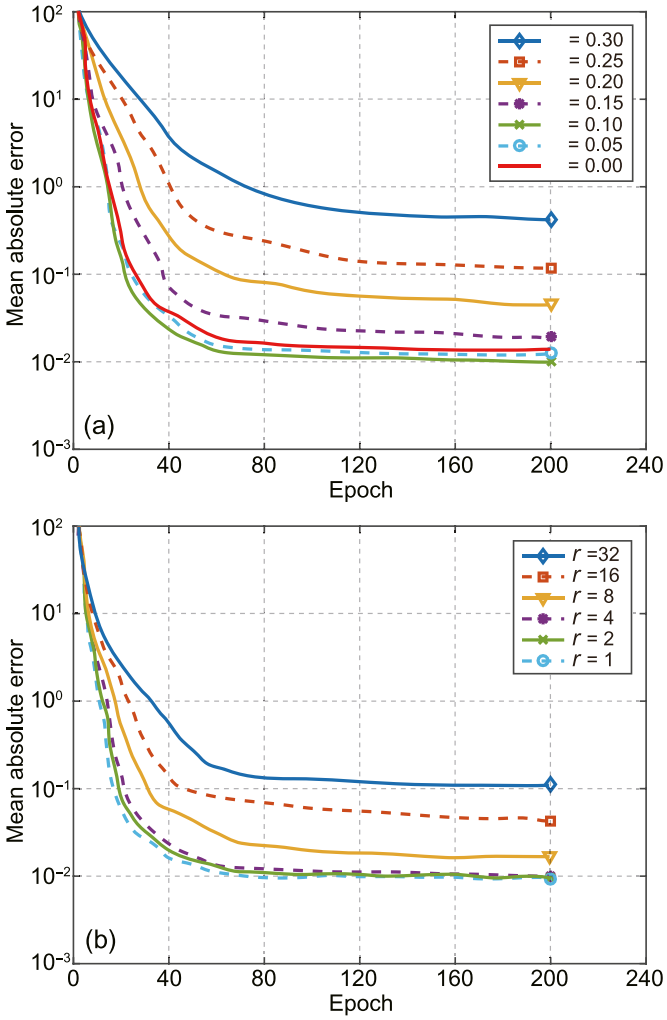

Fig. 9. Training curve of our method under different values of negative slope α (a) and reduction ratio r (b) on Set14 dataset under scale factor 4.

that a small value of α helps our model converge better. This is because our input images are processed by the Wavelet transform, resulting in some negative gradient pixels naturally included. Thus, the tiny value $\alpha = 0.1$ is beneficial to keep negative information flow effectively in forward pass and backpropagation.

Fig. 9 (b) shows that different values of reduction ratio r influence the convergence of our model. Note that $r = 1$ means no dimension reduction on description vectors in channel attention modules. When r grows, i.e., the length of a description vector is compressed, the convergence becomes worse, since some information of channel importance has been lost in this period. We choose $r = 4$ to keep the best performance, since the number of parameters saved in the channel attention module is trivial.

4.4. Ablation study

Table 1 compares the effect of different combinations of three modules in our network. We report the number of parameters and PSNR on Set5 and Set14 datasets under scale factor 2. Obviously, the additional number of parameters introduced by channel attention and spatial attention modules are negligible. From module 1 to 5, it can be seen that channel attention and spatial attention modules both significantly improve the SR performance with trivial extra computational cost. In module combination 6 to 11, we adopt all three modules with different orders. It is observed that setting the multi-kernel convolution as the first part in one block leads to the best PSNR. Similarly, combining channel attention and spatial attention modules (8 and 9) instead of separating them (10 and 11) produces superior performance. Because they are both lightweight structures (Figs. 5 and 6), linking them is helpful to information flow in forward pass and backpropagation, but splitting them will break this advantage. In addition, we discover that putting the channel attention module before the spatial attention module will obtain slightly better outcomes. Thus, we adopt the structure as module 6 throughout our experiments.

Table 2 compares the effects of different structures of several modules in our network. To validate the effectiveness of the Wavelet transform, we compare three types of input data: 1. original images; 2. LF ($\boldsymbol{A}$) and HF ($[\boldsymbol{H} + \boldsymbol{V} + \boldsymbol{D}]$) parts; 3. Wavelet sub-bands ($\boldsymbol{A}$, $\boldsymbol{H}$, $\boldsymbol{V}$, $\boldsymbol{D}$), respectively. It can be seen that the PSNR increases substantially as the input data are split to more channels before training. By separating the input data to average, horizontal, vertical, and diagonal sub-bands, Wavelet type achieves the best PSNR that other combinations. As the basic function in our network, we prove the efficacy of various kernel sizes for SR. We compare three trials: 3×3, 5×5, and multi-kernel (Fig. 4), with the same number of channels. The results indicate that the larger kernel size is, the better performance our method will obtain, which conforms with our expertise that a large receptive field in neural networks is beneficial for extracting high-level features and learning the representative capability. To verify the influence of the attention modules, we compare three types of pooling: max, average, and max & average. By using the max pooling or average pooling individually, we can see that they reveal similar results in terms of PSNR, and are consistently slightly worse than using both pooling operations. It shows that adopting both max pooling and average pooling is robust to train a huge amount of images and hence obtains better performance.

Table 3 summarizes the model sizes of several SR methods. The numbers of parameters and layers are used for comparison. Intuitively, SRCNN is the simplest methods, because it only contains three convolutional layers. To improve the performance, DWSR, VDSR, and LapSRN consistently increase their numbers of

Table 2
Effect of different structures of modules in our network. PSNR is given under scale factor 3 on Set5 and Set14 datasets.

Description		Different types of combinations								
Module		1	2	3	4	5	6	7	8	9
Input	Non-Wavelet	✓								
	Wavelet (***A***, [***H*** + ***V*** + ***D***])		✓							
	Wavelet (***A***, ***H***, ***V***, ***D***)			✓	✓	✓	✓	✓	✓	✓
Convolution	Single-kernel (3 × 3)				✓					
	Single-kernel (5 × 5)					✓				
	Multi-kernel	✓	✓	✓			✓	✓	✓	✓
Attention	'Max' only							✓		
	'Avg' only								✓	
	'Max' & 'Avg'	✓	✓	✓	✓	✓	✓			✓
PSNR/dB (Set5)		33.56	34.17	**34.79**	33.89	34.11	**34.79**	34.49	34.51	**34.79**
PSNR/dB (Set14)		29.64	30.25	**30.71**	29.96	30.19	**30.71**	30.45	30.48	**30.71**

Table 3
Comparison of model sizes of different super-resolution methods.

Method	Number of parameters	Number of layers
SRCNN [27]	57K	3
DWSR [33]	300K	10
VDSR [28]	665K	20
MemNet [31]	675K	80
LapSRN [30]	812K	24
DRCN [47]	1.78M	20
WRAN [ours]	2.71M	51
DBPN [50]	10.24M	46
RCAN [38]	15.60M	213
MWCNN [35]	16.14M	24
RDN [32]	21.88M	149
EDSR [49]	43.19M	69

parameters to 10^5 order and deepen their networks to at most 24 layers. As two special cases, MemNet holds similar number of parameters than three above, but has a quite deep structure with 80 layers; DRCN comprises 20 layers but requires 1.78 million parameters. Though our model contains 51 layers and has over 2.7 million parameters, WRAN is relatively smaller than DBPN and MWCNN. Beneficial from the channel attention structure, RCAN stacks over 200 layers but its parameters are not substantially increased. Both RDN and EDSR construct large-scale models, resulting in a huge number of parameters to learn. Thus, the model size of our WRAN is considered as moderate compared to other SR methods.

4.5. Comparison against state-of-the-art SR methods

Table 4 shows the quantitative results (PSNR/SSIM) of several SR methods on four datasets under scale factors 2, 3, and 4. Bicubic interpolation and A+ reveal the worst results, since they are conventional approaches and do not need to train their models. As the first deep learning-based method, SRCNN slightly improves PSNR in all cases. By designing deep and wide structures, VDSR and LapSRN achieve superior performance than SRCNN, because they adopt numerous parameters in training, to effectively explore the latent patterns in feature maps. Similar to our method, DWSR and MWCNN also use the Wavelet transform as the preprocess for their inputs. With the advantage of separating LF and HF components explicitly, these two methods obtain relatively better performance than previous methods. However, they adopt normal CNN structures in their networks, which leads to limited advance in terms of PSNR and SSIM. Since RCAN and our WRAN both use the channel attention module, two methods achieve the best results against others. In most cases, our WRAN reports the best PSNR, because we integrate the merits of the Wavelet transform, multi-kernel convolution, and spatial attention. However, RCAN performs

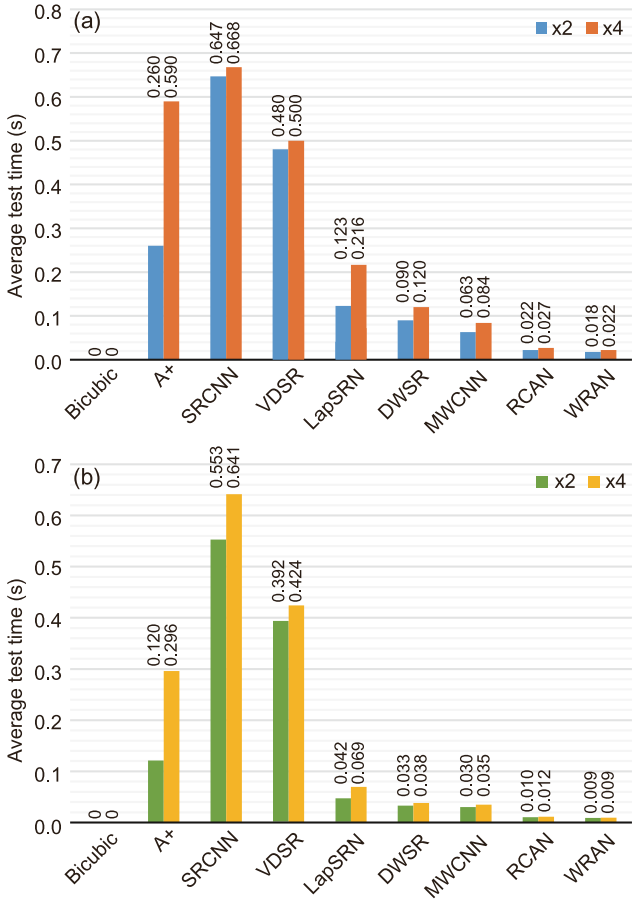

Fig. 10. Average test time per image of different methods: (a) B100 dataset, scale × 2 and × 4; (b) Urban100 dataset, scale × 2 and × 4.

slightly better in large scale factor × 4 cases, since they build an extremely deep network and over 5 times parameter numbers (Table 3) than our method. In scale factor × 4 cases, the bicubic input inevitably lacks much HF details, resulting in ineffectively separating features by the Wavelet transform. Thus, our method beats other SR methods but is slightly inferior than RCAN.

Fig. 10 shows the average test time per image of different methods on B100 and Urban100 datasets under scale factors × 2 and × 4. Note that the bicubic interpolation is widely used as the basic operation for image SR. It is simple and requires no parameter. Thus, we consider that bicubic interpolation takes no test time and will not discuss it here. Obviously, SRCNN consumes the most time than other methods, since it contains multiple convolutional

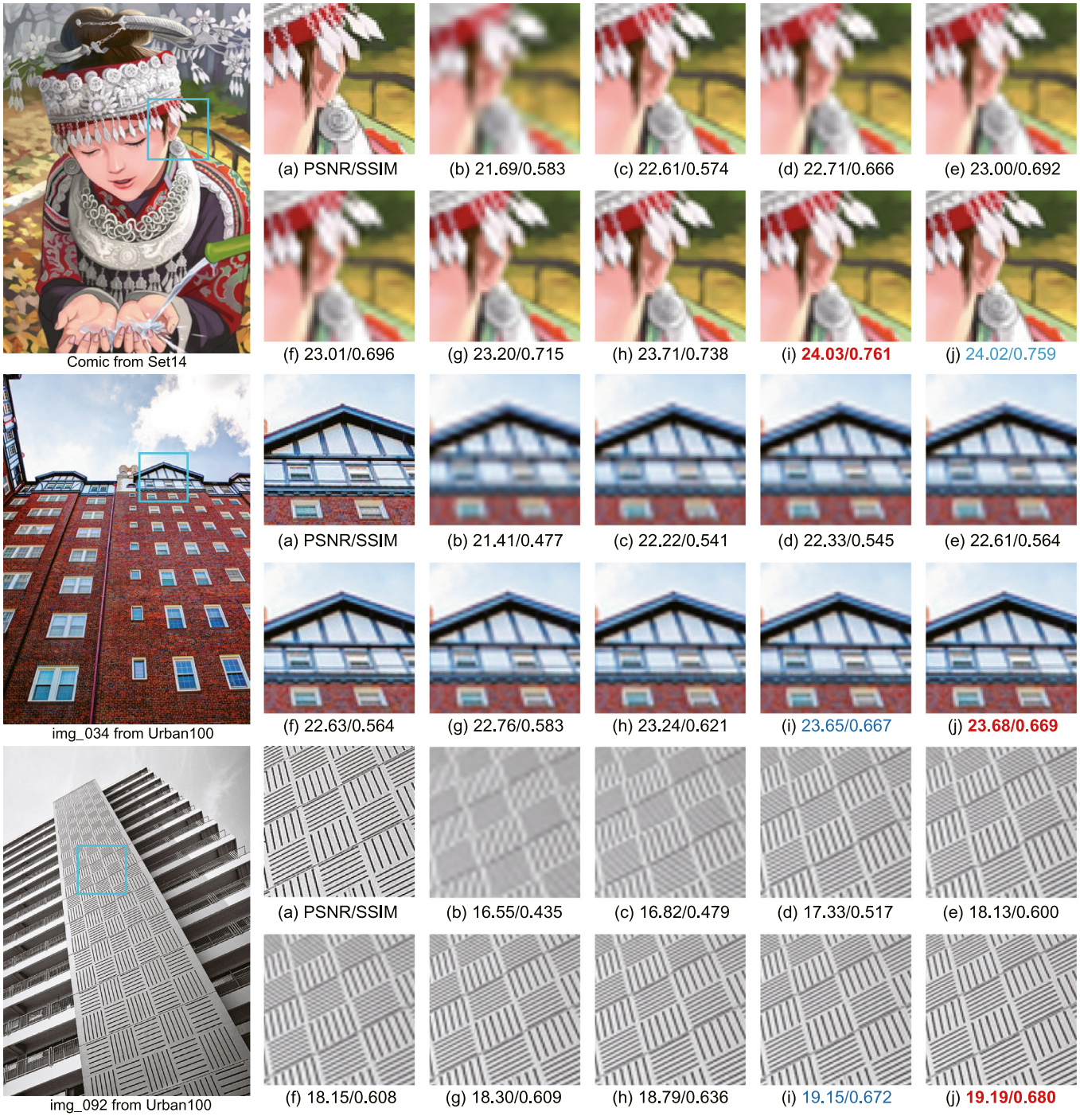

Fig. 11. Visual comparison of different methods under scale factor × 4: (a) high-resolution; (b) bicubic; (c) A+; (d) SRCNN; (e) VDSR; (f) LapSRN; (g) DWSR; (h) MWCNN; (i) RCAN; (j) WRAN

layers that require much computational cost, though SRCNN has only a few parameters. VDSR also requires considerable test time, since it has 20 convolutional layers and runs in the large image space as SRCNN adopts. A+ needs much test time that varies dramatically under scale factors × 2 and × 4. Because A+ involves dictionary learning and regression that entail processing time proportional to the image size, this method hereby runs apparently faster under the small scale factor. By processing images gradually in a pyramid manner from small scales to large scales, LapSRN performs several times faster than previous methods. With the advantage of the Wavelet transform, DWSR and MWCNN both require less time than LapSRN. This proves the effectiveness of the Wavelet transform that separates images into small image spaces. Beneficial from the attention blocks, RCAN and our WRAN are the most fastest methods than others (12 ms at most), which validates that spatial and channel attention modules save substantial computational cost.

Fig. 11 illustrates three examples from Set14 and Urban100 datasets, generated by several SR methods under scale factor × 4. Intuitively, Fig. 11(b) and (c) show relatively poor reconstructed images, since they contain visually blurry textures. By exploiting CNN, SRCNN (Fig. 11(d)) reveals clearer images than previous two methods. As model size becomes larger and more complicated, the reconstruction quality of these methods (Fig. 11(e)–(h)) improves

Table 4
Quantitative results of our method compared with other super-resolution approaches on four datasets (**bold** numbers and underlined numbers denote the best results and the second best results, respectively).

Dataset	Scale	PSNR (dB)/SSIM								
		Bicubic	A+ [67]	SRCNN [27]	VDSR [28]	LapSRN [30]	DWSR [33]	MWCNN [35]	RCAN [38]	WRAN [ours]
Set5 [61]	× 2	33.65/0.928	36.53/0.952	36.65/0.952	37.52/0.956	37.24/0.955	37.43/0.956	37.91/0.960	*38.27/0.961*	**38.32/0.963**
	× 3	30.38/0.865	32.57/0.905	32.74/0.906	33.65/0.918	34.05/0.921	33.82/0.921	34.18/0.927	*34.74/0.929*	**34.79/0.931**
	× 4	28.40/0.806	30.26/0.856	30.46/0.858	31.33/0.879	31.31/0.817	31.39/0.883	32.12/0.894	**32.63/0.900**	*32.36/0.899*
Set14 [62]	× 2	30.23/0.866	32.27/0.903	32.41/0.904	33.02/0.910	32.95/0.908	33.07/0.910	33.70/0.918	*34.12/0.921*	**34.21/0.922**
	× 3	27.54/0.771	29.12/0.816	29.27/0.817	29.76/0.828	29.96/0.834	29.83/0.830	30.16/0.841	*30.65/0.848*	**30.71/0.852**
	× 4	25.98/0.698	27.29/0.745	27.47/0.746	27.99/0.763	28.04/0.764	28.04/0.766	28.41/0.781	**28.87/0.788**	*28.60/0.786*
B100 [63]	× 2	29.55/0.841	31.20/0.884	31.35/0.885	31.88/0.894	31.67/0.890	31.80/0.894	32.23/0.899	*32.41/0.902*	**32.57/0.907**
	× 3	27.20/0.735	28.28/0.780	28.40/0.783	28.80/0.794	28.91/0.799	– / –	29.12/0.806	*29.32/0.811*	**29.36/0.819**
	× 4	25.94/0.663	26.80/0.704	26.88/0.706	27.26/0.721	27.20/0.720	27.25/0.724	27.62/0.735	***27.77/0.743***	*27.71/0.742*
Urban100 [64]	× 2	26.87/0.838	29.22/0.892	29.49/0.892	30.75/0.912	30.40/0.908	30.46/0.916	32.30/0.929	*33.34/0.938*	**33.47/0.940**
	× 3	24.45/0.731	26.02/0.794	26.23/0.795	27.12/0.824	27.05/0.824	– / –	28.13/0.851	**29.09/0.870**	*28.99/0.869*
	× 4	23.12/0.653	24.32/0.716	24.50/0.718	25.15/0.748	25.19/0.751	25.26/0.754	26.27/0.789	**26.82/0.808**	*26.74/0.803*

progressively. Intuitively, Fig. 11(i) and (j) illustrate the best SR results than other approaches. As we can observe, some fine details are successfully recovered by RCAN and our WRAN, because we use the Wavelet transform to split LF and HF components in advance, and adopt channel attention and spatial attention modules to adaptively readjust features during training.

5. Conclusions

In this paper, we have proposed the Wavelet-based residual attention network (WRAN) for image SR. Instead of stacking large-scale models with a large amount of computation and enormous parameters, we adopt lightweight but effective modules: channel attention and spatial attention, to improve the efficiency of image SR without sacrificing much performance. We use the 2D Wavelet transform to explicitly separate low-frequency and high-frequency details from one image to four channels before training. Thus, the learning difficulty of our network can be mitigated. Inspired by the Inception structure, we propose the multi-kernel convolutional layers as basic modules to adaptively aggregate features from different sized receptive fields. It is proved that combining basic modules, channel attention, and spatial attention properly can further enhance the SR outcomes. Extensive experiments show that our WRAN requires relatively fewer parameters and generates competitive results against state-of-the-art SR methods quantitatively and qualitatively.

Declaration of Competing Interest

None.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 81873894).

References

[1] Z. Feng, J. Lai, X. Xie, J. Zhu, Image super-resolution via a densely connected recursive network, Neurocomputing 316 (2018) 270–276, doi:10.1016/j.neucom.2018.07.076.
[2] F. Zhou, X. Li, Z. Li, High-frequency details enhancing densenet for super-resolution, Neurocomputing 290 (2018) 34–42, doi:10.1016/j.neucom.2018.02.027.
[3] Z. Li, Q. Li, W. Wu, J. Yang, Z. Li, X. Yang, Deep recursive up–down sampling networks for single image super-resolution, Neurocomputing (1) (2019) 1–12, doi:10.1016/j.neucom.2019.04.004.
[4] L. Zhu, S. Zhan, H. Zhang, Stacked u-shape networks with channel-wise attention for image super-resolution, Neurocomputing 345 (2019) 58–66, doi:10.1016/j.neucom.2018.12.077.
[5] Y. Li, V. Tsiminaki, R. Timofte, M. Pollefeys, L. van Gool, 3D appearance super-resolution with deep learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9671–9680.
[6] S. Zhang, Y. Lin, H. Sheng, Residual networks for light field image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11046–11055.
[7] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, W. Wu, Feedback network for image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3867–3876.
[8] Y. Chen, Y. Xie, Z. Zhou, F. Shi, A.G. Christodoulou, D. Li, Brain MRI super resolution using 3D deep densely connected neural networks, in: Proceedings of the IEEE Fifteenth International Symposium on Biomedical Imaging, 2018, pp. 739–742, doi:10.1109/ISBI.2018.8363679.
[9] F. Shi, J. Cheng, L. Wang, P.T. Yap, D. Shen, LRTV: MR Image super-resolution with low-rank and total variation regularizations, IEEE Trans. Med. Imag. 34 (12) (2015) 2459–2466, doi:10.1109/TMI.2015.2437894.
[10] O. Oktay, W. Bai, M. Lee, R. Guerrero, et al., Multi-input cardiac image super-resolution using convolutional neural networks, in: Proceedings of the International Conference onMedical Image Computing and Computer-Assisted Intervention, 2016, pp. 246–254, doi:10.1007/978-3-319-46726-9_29.
[11] Y. Huang, L. Shao, A.F. Frangi, Simultaneous super-resolution and cross-modality synthesis of 3D medical images using weakly-supervised joint convolutional sparse coding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5787–5796, doi:10.1109/CVPR.2017.613.
[12] J. Du, Z. He, L. Wang, A. Gholipour, Z. Zhou, D. Chen, Y. Jia, Super-resolution reconstruction of single anisotropic 3D MR images using residual convolutional neural network, Neurocomputing (1) (2019) 1–12, doi:10.1016/j.neucom.2018.10.102.
[13] P. Rasti, T. Uiboupin, S. Escalera, G. Anbarjafari, Convolutional neural network super resolution for face recognition in surveillance monitoring, in: F.J. Perales, J. Kittler (Eds.), Proceedings of the International Conference on Articulated Motion and Deformable Objects, 2016, pp. 175–184, doi:10.1007/978-3-319-41778-3_18.
[14] S. Xue, W. Qiu, F. Liu, X. Jin, Low-rank tensor completion by truncated nuclear norm regularization, in: Proceedings of the Twenty-forth International Conference on Pattern Recognition (ICPR), 2018, pp. 2600–2605, doi:10.1109/ICPR.2018.8546008.
[15] J. Jiang, R. Hu, Z. Wang, Z. Han, Noise robust face hallucination via locality-constrained representation, IEEE Trans. Multim. 16 (5) (2014) 1268–1281, doi:10.1109/TMM.2014.2311320.
[16] J. Jiang, Y. Yu, S. Tang, J. Ma, A. Aizawa, K. Aizawa, Context-patch face hallucination based on thresholding locality-constrained representation and reproducing learning, IEEE Trans. Cybern. (2018) 1–14, doi:10.1109/TCYB.2018.2868891.
[17] Z. Wang, P. Yi, K. Jiang, J. Jiang, Z. Han, T. Lu, J. Ma, Multi-memory convolutional neural network for video super-resolution, IEEE Trans. Image Process. 28 (5) (2019) 2530–2544, doi:10.1109/TIP.2018.2887017.
[18] K. Jiang, Z. Wang, P. Yi, J. Jiang, J. Xiao, Y. Yao, Deep distillation recursive network for remote sensing imagery super-resolution, Remote Sens. 10 (11) (2018) 1700, doi:10.3390/rs10111700.
[19] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, J. Jiang, Edge-enhanced GAN for remote sensing image super-resolution, IEEE Trans. Geosc. Remote Sens. 57 (8) (2019) 5799–5812, doi:10.1109/TGRS.2019.2902431.
[20] D. Dai, Y. Wang, Y. Chen, L. Van Gool, Is image super-resolution helpful for other vision tasks? in: Proceedings of the IEEE Winter Conference on Applications of Comput. Vis., 2016, pp. 1–9, doi:10.1109/WACV.2016.7477613.
[21] S. Xue, X. Jin, Robust classwise and projective low-rank representation for image classification, Signal Image Video Process. 12 (1) (2018) 107–115, doi:10.1007/s11760-017-1136-1.
[22] M. Haris, G. Shakhnarovich, N. Ukita, Task-driven super resolution: Object detection in low-resolution images (2018) 1–26.

[23] H. Zhang, D. Liu, Z. Xiong, Convolutional neural network-based video super-resolution for action recognition, in: Proceedings of the Thirteenth IEEE International Conference on Automatic Face and Gesture Recognition, 2018, pp. 746–750, doi:10.1109/FG.2018.00117.
[24] L. Zhou, Z. Wang, Y. Luo, Z. Xiong, Separability and compactness network for image recognition and super-resolution, IEEE Trans. on Neur. Netw. Learn. Syst. (2019) 1–12, doi:10.1109/TNNLS.2018.2890550.
[25] K. Zhang, W. Zuo, L. Zhang, Learning a single convolutional super-resolution network for multiple degradations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3262–3271, doi:10.1109/CVPR.2018.00344.
[26] R. Keys, Cubic convolution interpolation for digital image processing, IEEE Trans. Acoust Speech Signal Process 6 (29) (1981) 1153–1160, doi:10.1109/TASSP.1981.1163711.
[27] C. Dong, C.C. Loy, K. He, X. Tang, Learning a deep convolutional network for image super-resolution, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 184–199, doi:10.1007/978-3-319-10593-2_13.
[28] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1646–1654, doi:10.1109/CVPR.2016.182.
[29] W. Shi, J. Caballero, F. Huszar, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, Z. Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1874–1883, doi:10.1109/CVPR.2016.207.
[30] W. Lai, J. Huang, N. Ahuja, 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, 2017, pp. 5835–5843, doi:10.1109/CVPR.2017.618.
[31] Y. Tai, J. Yang, X. Liu, C. Xu, MemNet: a persistent memory network for image restoration, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4549–4557, doi:10.1109/ICCV.2017.486.
[32] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2472–2481, doi:10.1109/CVPR.2018.00262.
[33] T. Guo, H.S. Mousavi, T.H. Vu, V. Monga, Deep Wavelet prediction for image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 1100–1109, doi:10.1109/CVPRW.2017.148.
[34] H. Huang, R. He, Z. Sun, T. Tan, Wavelet-SRNet: a wavelet-based CNN for multi-scale face super resolution, in: Proceedings of the IEEE International Conference on Comput. Vis., 2017, pp. 1698–1706, doi:10.1109/ICCV.2017.187.
[35] P. Liu, H. Zhang, K. Zhang, L. Lin, W. Zuo, Multi-level Wavelet-CNN for image restoration, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 886–895, doi:10.1109/CVPRW.2018.00121.
[36] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1800–1807, doi:10.1109/CVPR.2017.195.
[37] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141, doi:10.1109/CVPR.2018.00745.
[38] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, Image super-resolution using very deep residual channel attention networks, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 294–310, doi:10.1007/978-3-030-01234-2_18.
[39] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, CBAM: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 3–19, doi:10.1007/978-3-030-01234-2_1.
[40] H. Ji, C. Fermüller, Robust wavelet-based super-resolution reconstruction: theory and algorithm, IEEE Trans. Patt. Anal. Mach. Intell. 31 (4) (2009) 649–660, doi:10.1109/TPAMI.2008.103.
[41] M.D. Robinson, C.A. Toth, J.Y. Lo, S. Farsiu, Efficient fourier-wavelet super-resolution, IEEE Trans. Image Process. 19 (10) (2010) 2669–2681, doi:10.1109/TIP.2010.2050107.
[42] G. Anbarjafari, H. Demirel, Image super resolution based on interpolation of wavelet domain high frequency subbands and the spatial domain input image, ETRI J. 32 (3) (2010) 390–394, doi:10.4218/etrij.10.0109.0303.
[43] S. Mallat, G. Yu, Super-resolution with sparse mixing estimators, IEEE Trans. Image Process. 19 (11) (2010) 2889–2900, doi:10.1109/TIP.2010.2049927.
[44] W. Dong, L. Zhang, G. Shi, X. Wu, Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization, IEEE Trans. Image Process. 20 (7) (2011) 1838–1857, doi:10.1109/TIP.2011.2108306.
[45] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778, doi:10.1109/CVPR.2016.90.
[46] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W.M. Wells, A.F. Frangi (Eds.), Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Cham, 2015, pp. 234–241, doi:10.1007/978-3-319-24574-4_28.
[47] J. Kim, J.K. Lee, K.M. Lee, Deeply-recursive convolutional network for image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1637–1645, doi:10.1109/CVPR.2016.181.
[48] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2261–2269, doi:10.1109/CVPR.2017.243.
[49] B. Lim, S. Son, H. Kim, S. Nah, K.M. Lee, Enhanced deep residual networks for single image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 1132–1140, doi:10.1109/CVPRW.2017.151.
[50] M. Haris, G. Shakhnarovich, N. Ukita, Deep back-projection networks for super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1664–1673, doi:10.1109/CVPR.2018.00179.
[51] R.A. Rensink, The dynamic representation of scenes, Vis. Cogn. 7 (1–3) (2000) 17–42, doi:10.1080/135062800394667.
[52] M. Corbetta, G.L. Shulman, Control of goal-directed and stimulus-driven attention in the brain, Nature Rev. Neurosc. 3 (3) (2002) 201–215, doi:10.1038/nrn755.
[53] H. Larochelle, G.E. Hinton, Learning to combine foveal glimpses with a third-order Boltzmann machine, in: J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, et al. (Eds.), Proceedings of the Advances in Neural Information Processing Systems, 2010, pp. 1243–1251.
[54] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, Residual attention network for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6450–6458, doi:10.1109/CVPR.2017.683.
[55] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9, doi:10.1109/CVPR.2015.7298594.
[56] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for comput. vis., in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826, doi:10.1109/CVPR.2016.308.
[57] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, Inception-ResNet and the impact of residual connections on learning, in: Proceedings of the Thirty-first AAAI Conference on Artificial Intelligence, 2017, pp. 1–7.
[58] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd International Conference on Machine Learning, 2015, pp. 448–456.
[59] M. Abadi, P. Barham, J. Chen, et al., Tensorflow: a system for large-scale machine learning, in: Proceedings of the Twelfth USENIX Symposium on Operating Systems Design and Implementation, 2016, pp. 265–283.
[60] E. Agustsson, R. Timofte, NTIRE 2017 challenge on single image super-resolution: dataset and study, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 1122–1131, doi:10.1109/CVPRW.2017.150.
[61] M. Bevilacqua, A. Roumy, C. Guillemot, M.-L.A. Morel, Low-complexity single-image super-resolution based on nonnegative neighbor embedding, in: Proceedings of the British Machine Vision Conference, 2012, pp. 135.1–135.10, doi:10.5244/C.26.135.
[62] R. Zeyde, M. Elad, M. Protter, On single image scale-up using sparse-representations, in: Proceedings of the International Conference on Curves and Surfaces, 2012, pp. 711–730, doi:10.1007/978-3-642-27413-8_47.
[63] D. Martin, C. Fowlkes, D. Tal, 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, 2001, pp. 416–423, doi:10.1109/ICCV.2001.937655.
[64] J. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5197–5206, doi:10.1109/CVPR.2015.7299156.
[65] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612, doi:10.1109/TIP.2003.819861.
[66] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the International Conference for Learning Representations, 2015.
[67] R. Timofte, V. de Smet, L. van Gool, A+: adjusted anchored neighborhood regression for fast super-resolution, in: Proceedings of the Asian Conference on Computer Vision, 2015, pp. 111–126, doi:10.1007/978-3-319-16817-3_8.

Shengke Xue received the B.S. degree in communication engineering, from the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China, in 2015. Currently, He is pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. At present, his research interests include pattern recognition, image processing, and machine learning.