Neurocomputing xxx (xxxx) xxx

Contents lists available at ScienceDirect

Neurocomputing

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

Multi-scale feature fusion residual network for Single Image Super-Resolution

Jinghui Qin, Yongjie Huang, Wushao Wen*

School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China

ARTICLE INFO

Article history:
Received 23 May 2019
Revised 20 October 2019
Accepted 25 October 2019
Available online xxx

Communicated by Gaofeng MENG

Keywords:
Single Image Super-Resolution
Multi-scale feature fusion
Residual network

ABSTRACT

We have witnessed great success of Single Image Super-Resolution (SISR) with convolutional neural networks (CNNs) in recent years. However, most existing Super-Resolution (SR) networks fail to utilize the multi-scale features of low-resolution (LR) images to further improve the representation capability for more accurate SR. In addition, most of them do not exploit the hierarchical features across networks for the final reconstruction. In this paper, we propose a novel multi-scale feature fusion residual network (MSFFRN) to fully exploit image features for SISR. Based on the residual learning, we propose a multi-scale feature fusion residual block (MSFFRB) with multiple intertwined paths to adaptively detect and fuse image features at different scales. Furthermore, the outputs of each MSFFRB and the shallow features are used as the hierarchical features for global feature fusion. Finally, we recover the high-resolution image based on the fused global features. Extensive experiments on four standard benchmarks demonstrate that our MSFFRN achieves better accuracy and visually pleasing than the current state-of-the-art methods.

1. Introduction

Single Image Super-Resolution (SISR) [1], which aims at recovering a visually clear high-resolution (HR) image from a downgraded low-resolution (LR) image, is a classical low-level computer vision task. It not only has abundant application scenarios in medical imaging [2], security and surveillance imaging [3], and remote sensing imagery [4–6], but also can serve as a built-in module to improve the performance of other computer vision tasks, such as image restoration [7], image recognition [8,9] and object detection [10], etc. However, the SISR is an ill-posed inverse problem because we can get the same LR image from an infinite number of HR images by downsampling. To mitigate this problem, researchers have proposed numerous solutions which can be divided into three categories: interpolation-based methods [11], reconstruction-based methods [12] and learning-based methods [13–20]. Due to the poor performance of the former kinds of SR methods with large upscaling factors, most of the recent state-of-the-art SR methods apply deep convolutional neural networks (CNNs) to learn a mapping from LR to HR images to achieve superior SR performance.

Recently, deep convolutional neural network (CNN) has been shown superior performance improvements over conventional SR methods in the SISR problem and has become the dominant technology for SISR task. Dong et al. [13] proposed SRCNN, a three-layer CNN, to make the first successful attempt to solve SISR problem by learning a non-linear mapping between LR and HR without requiring any artificial feature engineering. From then on, more and more studies concentrate on how to design a more effective neural network to improve the performance of SISR. VDSR [15] achieved significant improvements over SRCNN by increasing the network depth and using residual learning. EDSR [18] enhanced SRResNet [21] by removing the batch normalization layers and using deeper and wider network structure. RDN [19] utilized residual dense blocks to fully exploit the hierarchical features from all the convolutional layers. RCAN [20] built a deeper network by utilizing residual in residual (RIR) structure and channel attention mechanism to mitigate the training difficulty of very deep SR network and improve the representation ability of SR network. While these models achieved predominant objective performance in terms of peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [22] in SISR problem, all of them tend to construct deeper and more complex network structures, leading to greater training difficulty. What is worse, they ignore how to utilize multi-scale features of each LR image to further improve the SISR performance effectively.

* Corresponding author.
E-mail addresses: qinjingh@mail2.sysu.edu.cn (J. Qin), wenwsh@mail.sysu.edu.cn (W. Wen).

https://doi.org/10.1016/j.neucom.2019.10.076
0925-2312/

To fully exploit the multi-scale image features of LR images, we propose a novel multi-scale feature fusion residual network (MSFFRN) for SISR. In addition, a multi-scale feature fusion residual block (MSFFRB) is proposed as the basic building block for MSFFRN. First, at different scales, we build MSFFRB to extract and fuse the image features as local fused multi-scale features. Second, the outputs of all MSFFRBs and the shallow feature extraction module are concatenated for global hierarchical feature fusion. The global hierarchical feature fusion will adaptively merge and refine all local features, providing a more effective feature representation for more accurate SR reconstruction. Finally, the effective and robust global feature representation will be used to reconstruct high-resolution images via an upscaling module which consists of some sub-pixel convolutions.

In summary, our main contributions are four folds:

- We propose a novel multi-scale feature fusion residual network (MSFFRN) for accurate SISR. The proposed network makes full use of all the local fused multi-scale features and global hierarchical features from the original LR image.
- We propose multi-scale feature fusion residual block (MSFFRB), which can effectively extract multi-scale features and fuse them via multiple intertwined paths for accurate local feature representation.
- We take full advantage of the hierarchical feature maps from all MSFFRB blocks and shallow feature extraction module for more accurate reconstruction. This is proved to be conducive to improve the model performance significantly.
- We experimentally show that our model can outperform most of state-of-the-art models on several standard benchmarks.

The remainder of the paper is organized as follows. In Section 2 we review related works. Then, we describe and analyze the proposed MSFFRN in Section 3. After that, we demonstrate the superior performance of our proposed MSFFRN with extensive experimental results in Section 4. Finally, Section 5 ends this paper with conclusions.

2. Related works

2.1. Deep learning based image super-resolution

Recently, deep learning based SR methods have achieved dramatic advantages against conventional SR methods. Dong et al. proposed SRCNN [13] which is the first SR method applying a three-layer CNN to establish an end-to-end mapping between the interpolated LR images and their corresponding HR images. SRCNN makes a remarkable improvement compared to conventional SR methods via jointly optimizing the feature extraction, non-linear mapping, and image reconstruction stages in an end-to-end manner. VDSR [15] achieved significant improvements over SRCNN by stacking more convolutional layers with residual learning. DRCN [17] is the first SR method to introduce recursive learning in a very deep network for parameter sharing. Tai et al. introduced recursive blocks in DRRN [23] and memory blocks in Memnet [24] for deeper network. However, all of these methods need to interpolate the original LR images to the desired size before applying them into the networks. This pre-processing step increases computation complexity quadratically [14,25], impeding further improvement in speed. To address the above problem, Dong et al. [14] adopted deconvolution, smaller filter sizes, and more convolution layers to realize real-time SR process. In addition, Shi et al. proposed ESPCN [25], where an efficient sub-pixel convolution layer was introduced to upscale the final LR feature maps into the HR output. The efficient sub-pixel convolution layer was then adopted in almost all subsequent SR methods, e.g., SRResNet [21], EDSR [18], etc. All of these methods extract features in the LR space and upscale the final features with transposed or sub-pixel convolution layer. With the help of deconvolution and sub-pixel convolution layer, these networks can either achieve real-time performance or be built to be very deep or wide. However, these methods simply stack building blocks in a chain way and ignore fully exploiting the hierarchical information from each building block. This way might prevent further performance improvement.

2.2. Efficient feature extraction block design

Recently, apart from novel SR networks, many feature extraction blocks for SR have been proposed as well, such as residual block [8], dense block [26], and inception block [27]. Haris et al. [28] introduced inception block to exploit multiple features from low-resolution images. Lim et al. [18] proposed an enhanced residual block to mitigate the difficulty of network training and achieved state-of-the-art performance. Li et al. [29] proposed a dilated residual block with a gated selective mechanism to adaptively learn more high-frequency information and filter the low-frequency information. Wang et al. [30] proposed multi-memory residual block which embeds convolutional long short-term memory into the residual block to progressively extract and retain inter-frame temporal correlations between the consecutive LR frames. Tong et al. [31] introduced dense block to provide an effective way to combine the low-level features and high-level features to boost the reconstruction performance. Zhang et al. [19] proposed residual dense block which combines the advantages of residual block and dense block to fully exploit the hierarchical features of LR images. However, both residual block and dense block have some limitations. Residual block and dense block use only a single size of convolution kernel, which makes it difficult to detect image features at different scale. In addition, the computational complexity of dense block increases at a high growth rate. To solve the above drawbacks, Li et al. [32] proposed a multi-scale residual block (MSRB) by introducing convolution kernels of different sizes based on the residual structure. In MSRB, convolution kernels of different sizes are designed for detecting the features of images at different scales adaptively. Meanwhile, a skip connection is applied between different scale features for feature sharing and reuse. In addition, a 1×1 convolution layer at the end of the block is used as bottleneck layer for feature fusion and computational complexity reduction. Although MSRB can detect image feature at different scale, large kernel size applied in MSRB lead to the increase of computational complexity. In addition, the two bypasses of MSRB with same depth cannot fully exploit shallow and deep local image features at the same time. Therefore, we propose a novel multi-scale feature fusion block (MSFFRB) with multiple intertwined paths to fully exploit shallow and deep local image feature at different scale to solve the above mentioned deficiencies. In our MSFFRB, different paths with same convolutional kernel size will detect image features at different scale adaptively. Different paths will be intertwined with each other in a simple rule inspired by FractalNet [33] for feature fusion and reuse.

3. Proposed method

3.1. Problem overview

Given a high-resolution image I^{HR} and its corresponding low-resolution image I^{LR} obtained by some kind of downsampling operations, the goal of the SISR task is to reconstruct a high-resolution image I^{SR} from I^{LR} as close as possible to the ground-truth high-resolution image I^{HR}. We suppose an m-layer deep convolutional neural network $\mathcal{F}_\theta$ parameterized by weights and biases θ is applied to learn an end-to-end mapping function between the I^{LR} and I^{HR}. Given a training dataset $\{I_i^{LR}, I_i^{HR}\}_{i=1}^{N}$,

Please cite this article as: J. Qin, Y. Huang and W. Wen, Multi-scale feature fusion residual network for Single Image Super-Resolution, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.10.076

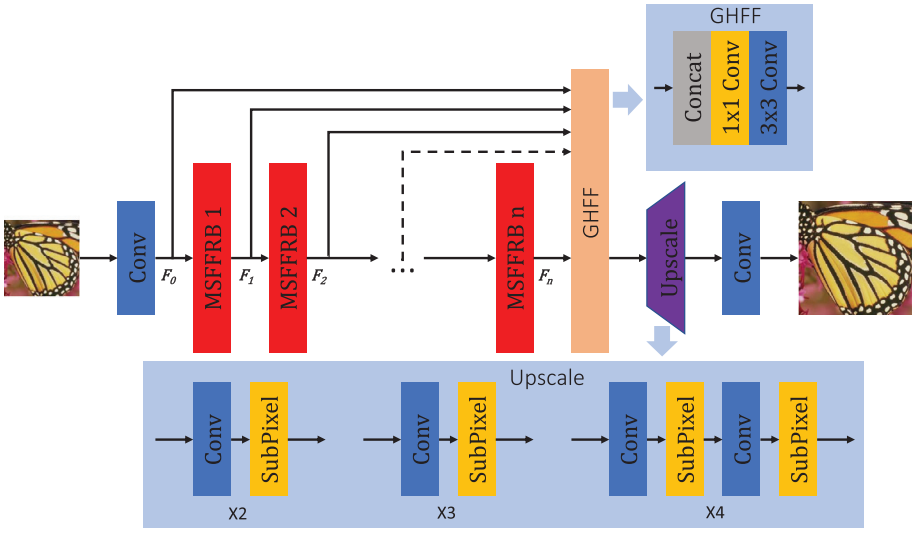

Fig. 1. Network architecture of our multi-scale feature fusion residual network (MSFFRN).

the SISR problem can be transformed to find parameters $\hat{\theta}$ to minimize the following equation:

$$\hat{\theta} = arg\min_{\theta} \frac{1}{N}\sum_{j=1}^{N} \mathcal{L}^{SR}(\mathcal{F}_{\theta}(I_i^{LR}), I_i^{HR}), \tag{1}$$

where $\theta = \{W_1, W_2, W_3, \ldots, W_m, b_1, b_2, b_3, \ldots, b_m\}$ denotes the weights and biases of the m-layer deep convolutional neural network adopted to solve the above equation. $\mathcal{F}_{\theta}(I_i^{LR})$ denotes the SR image I_i^{SR} obtained by forwarding I_i^{LR} to the deep convolutional neural network. $\mathcal{L}^{SR}$ denotes the loss function used to minimize the difference between I_i^{SR} and I_i^{HR}. Based on the above model, we detail our network design in Section 3.2 and describe the loss function we choose in Section 3.4.

3.2. Network architecture

As shown in Fig 1, our proposed MSFFRN mainly consists of five parts: shallow feature extraction, deep feature extraction based on our MSFFRBs, global feature fusion, upscale module, and reconstruction. Let's denote I_{LR} and I_{SR} as the input and output of our MSFFRN. Inspired by [18,21], we use only one convolution layer to extract the shallow feature F_0 from the LR input. The procedure can be expressed as follows:

$$F_0 = H_{sf}(LR), \tag{2}$$

where $H_{sf}(\cdot)$ denotes the convolution operation applied to shallow feature extraction. F_0 is then used for deep feature extraction with our cascaded MSFFRBs. Thus, we define the procedure of deep feature extraction as follows:

$$F_{df} = H_{df}(F_0), \tag{3}$$

where $H_{df}(\cdot)$ denotes our cascaded MSFFRBs. Suppose that we cascade n MSFFRBs, then F_{df} is a set of feature maps from all n MSFFRBs:

$$F_{df} = \{F_1, F_2, \ldots, F_n\}, \tag{4}$$

The global feature fusion structure (GFFS) described in Section 3.3 will fuse all the shallow and deep features to output a global feature representation F_{gf}. The global feature representation F_{gf} is then upscaled via a upscale module:

$$F_{up} = H_{up}(F_{gf}), \tag{5}$$

where $H_{up}(\cdot)$ and F_{up} denote the upscale module and the upscaled features respectively. There are some previously proposed upscale structures we can adopt, such as deconvolution/transposed convolution [14], nearest-neighbor upsampling + convolution [34], and sub-pixel convolution [25]. In this work, we adopt sub-pixel convolution as our upscale structure because it can achieve better feature representation and reduce checkerboard artifact in SR images [35]. Finally, the upscaled features F_{up} is reconstructed via a convolutional layer:

$$I_{SR} = H_{rec}(F_{up}) = H_{MSFFRN}(I_{LR}), \tag{6}$$

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

3.2.1. Multi-scale feature fusion residual block (MSFFRB)

In order to adaptively detect and fuse image features at different scale, we propose multi-scale feature fusion residual block (MSFFRB), which has multiple intertwined paths with fusion operation. In this section, we will provide a detailed description of our MSFFRB. As shown in Fig. 2, our MSFFRB consists of two parts: multi-scale feature fusion and local residual learning.

Multi-scale feature fusion: different from previous works, we construct a intertwined multi-path network. In the network, the convolutional kernels used in different paths have the same size, but the number of convolution operations, denoted as network depth in this study, of each path is different. It means that different paths use convolutional kernels with the same size but different network depth. During the process of multi-scale feature extraction, features extracted from different paths will be fused not only at the tail of the network, but also during the feed-forward process of the network. In this way, the network has the following advantages: First, the information between different paths can be shared with each other so that the network can adaptively detect image features at different scales. Second, each path can benefit from the gradients from different paths to mitigate vanishing gradient problem during training. This will result in a more powerful representation learning for more accurate SR. As shown in Fig. 3, our intertwined multi-path network for multi-scale feature fusion follows a simple expansion rule and can be extended to arbitrary number of paths in recursion, but it must be emphasized that all parameters in MSFFRB are unshared. Let P denote the number of paths of our intertwined multi-path network and $f_P(\cdot)$ denote our network's structure, connections and layer types. We suppose a

Please cite this article as: J. Qin, Y. Huang and W. Wen, Multi-scale feature fusion residual network for Single Image Super-Resolution, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.10.076

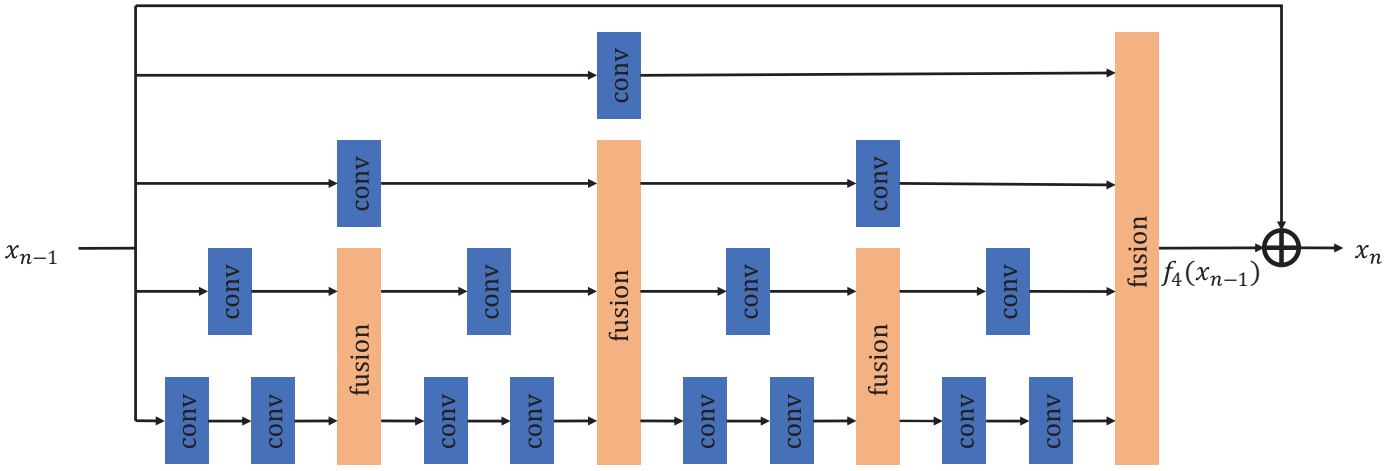

Fig. 2. The structure of Multi-scale Feature Fusion Residual Block (MSFFRB) with 4 intertwined paths feature fusion. We use this block as the default block in our model.

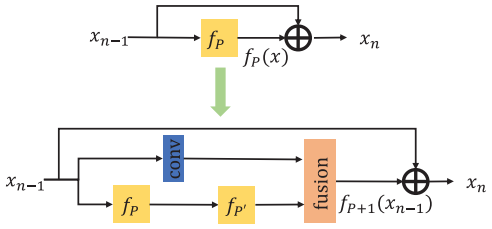

Fig. 3. A simple expansion rule generates a multi-scale feature fusion residual architecture with P intertwined paths.

single path (P = 1) is the base case of our network as follows:

$$f_1(x) = conv(x) \tag{7}$$

Then, we can define our network recursively:

$$f_{P+1}(x) = [(f_P \circ f_{P'})(x)] \oplus [conv(x)] \tag{8}$$

where $\circ$ denotes cascade operation and $\oplus$ denotes fusion operation. For clarification, in Eq. (8), f_P and $f_{P'}$ have same network structure, but they do not share their parameters. The depth of network is defined to be the number of *conv* layers on the longest path between input and output, scales as 2^{P-1}. There are several choices can serve as fusion operation, such as element-wise mean, element-wise weighted mean, and 1×1 convolution, etc. We empirically choose 1×1 convolution as the fusion operation based on our experiments. The fusion operation $\oplus$ will merge all input feature maps from different paths into a fused output feature maps. The channel count of the fused output feature maps corresponds to the size of the filter set in the preceding *conv* layer. We set P = 4 as our default path number of our MSFFRB, as shown in Fig. 2. In MSFFRB, each *conv* is followed by the ReLU activation function, the default kernel size is 3×3, and the channel counts of input and output are all 64.

Local residual learning: in order to make the information flow more efficiently in the network and mitigate the difficulty of network training, we adopt residual learning to each MSFFRB. Formally, we describe a multi-scale feature fusion residual block (MSFFRB) as:

$$x_n = f_p(x_{n-1}) + x_{n-1}, \tag{9}$$

where x_{n-1} and x_n represent the input and output of MSFFRB, respectively. The operation + is performed by a shortcut connection and element-wise addition.

3.3. Global feature fusion structure (GFFS)

For SISR problem, although LR images have lost some high-frequency information contained in their corresponding HR images, the LR images and SR output images are still highly correlated. Thus, how to fully exploit all features extracted from an input image to reconstruct a high-quality SR image is crucial. In this work, a simple global feature fusion structure is proposed. We first concatenate all outputs of the MSFFRBs and the shallow feature module. Then, we introduce a convolutional layer with 1×1 kernel size as bottleneck layer to adaptively extract useful information from these local features for high-quality image reconstruction. Following the bottleneck layer, we add a convolutional layer with 3×3 kernel size to refine the compressed information from the bottleneck layer. The refinement can provide more powerful representation for better SR reconstruction. Thus, the output of the global feature fusion structure (GFFS) can be formulated as:

$$F_{gf} = H_{GFF}([F_0, F_1, \ldots, F_n]), \tag{10}$$

where $[F_0, F_1, \ldots, F_n]$ represent the concatenation of the feature maps of the first convolutional layer and all n MSFFRBs. H_{GFF} denotes the global fusion structure which contains a 1×1 convolutional layer and a 3×3 convolutional layer in turn. The 1×1 convolutional layer adaptively fuses a range of local features in different hierarchical levels and extracts useful information for high-quality image reconstruction while the 3×3 convolutional layer refines the compressed information from the bottleneck layer for more powerful and robust feature representation.

3.4. Loss function

A superior loss function can improve network performance to a great extent. Thus, many researchers also focus on designing a superior objective optimization function for SISR. In SISR problem, the most widely-used loss functions are the Mean Square Error (MSE) function and the L2 loss function. Although these methods can obtain relatively high PSNR/SSIM, solutions using MSE or L2 optimization often produce excessively smooth textures, leading to a blurry visual effect. Recently, a variety of loss functions such as content loss function based on VGG [36], Charbonnier Penalty function (a differentiable variant of L_1 norm) [16], and a combination of L1 function and SSIM function [37], have been proposed. However, with the increased computation complexity, their performance improvements are only marginal. For simplicity, we adopt L1 as our loss function. Thus, our loss function is defined as:

$$\mathcal{L}^{SR}(\mathcal{F}_\theta(I_i^{LR}), I_i^{HR}) = \left\| (\mathcal{F}_\theta(I_i^{LR}) - I_i^{HR}) \right\|_1. \tag{11}$$

Please cite this article as: J. Qin, Y. Huang and W. Wen, Multi-scale feature fusion residual network for Single Image Super-Resolution, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.10.076

Table 1
Investigations on the number of MSFFRBs. We observe the best PSNR (dB) and SSIM values on Set5 ($\times 2$), Set14 ($\times 2$), B100 ($\times 2$), and Urban100 ($\times 2$). We use to bold label the first place.

	D = 8 PSNR/SSIM	D = 9 PSNR/SSIM	D = 10 PSNR/SSIM
Set5	38.11/0.9609	38.12/**0.9610**	**38.15/0.9610**
Set14	33.77/0.9187	33.85/**0.9195**	**33.88/0.9195**
B100	32.24/0.9005	32.27/0.9007	**32.29/0.9010**
Urban100	32.43/0.9311	32.58/0.9324	**32.60/0.9326**

Table 3
Investigations on different fusion operations. We observe the best PSNR (dB) and SSIM values on Set5 ($\times 2$), Set14 ($\times 2$), B100 ($\times 2$), and Urban100 ($\times 2$). We use bold to label the first place.

	Arithmetic mean PSNR/SSIM	Learnable weighted mean PSNR/SSIM	1×1 conv PSNR/SSIM
Set5	38.10/0.9608	38.08/0.9608	**38.15/0.9610**
Set14	33.82/0.9197	33.81/0.9192	**33.88/0.9195**
B100	32.25/0.9005	32.24/0.9004	**32.29/0.9010**
Urban100	32.46/0.9314	32.49/0.9316	**32.60/0.9326**

where I_i^{LR} and I_i^{HR} are an image pair in training set.

4. Experiments

In this section, we evaluate the performance of our model using several standard benchmarks. We first introduce the dataset used for training and testing, then we describe the implementation details. Next, we conduct a series of ablation studies to explore the properties of our model. Finally, we compare our model with several state-of-the-art models.

4.1. Datasets

We choose DIV2K [38] as the training dataset. DIV2K is a high-quality (2K resolution) image dataset for image super-resolution task. The DIV2K dataset consists of 800 training images, 100 validation images, and 100 test images. We use the 800 training images for training and the first 10 validation images for validating our model. We do not adopt the testing dataset of DIV2K as our benchmark since the ground truth of the test dataset is not public. We evaluate the performance of our model on four standard benchmark datasets: Set5 [39], Set14 [40], B100 [41], and Urban100 [42]. These datasets contain a wide variety of images that can fully evaluate the effectiveness of any SR model. Therefore, they are suitable for evaluating our model.

4.2. Implementation and training details

The implementation details of our proposed MSFFRN is as following. We set the number of MSFFRB as D = 10 in the MSFFRN. In each MSFFRB, we set the number of intertwined paths as P = 4. We set the kernel size of all convolutional layers as 3×3 except that in the fusion operations where we set the kernel size to 1×1. For convolutional layers with kernel size 3×3, zero-padding strategy is used to keep the size of output feature map fixed. All convolutional layers in shallow feature extraction and MSFFRBs have C = 64 filters. For upscaling module $H_{up}(\cdot)$, following [18], we use sub-pixel conv [25] to upscale the coarse resolution features to fine ones. The final convolutional layer has 3 filters, as we output color images.

Following [18,20], we use the RGB input patches of size 48×48 randomly cropped from LR images and the corresponding HR patches for training. All training data are randomly rotated by 90°, 180°, 270° and flipped horizontally. We preprocess all the images by subtracting the mean RGB value of the DIV2K dataset. We train our model with ADAM optimizer [43] by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We set minibatch size as 16 and an epoch as 1000 iterations of back-propagation. The learning rate is initialized as $1e{-}4$ and halved at every 200 epoch. We implement MSFFRN with the Pytorch 1.0 framework [44] and train it using NVIDIA RTX 2080Ti GPU and Cuda 10.0. We use the default weight initialization method in Pytorch 1.0. Our x2 model is trained from scratch. After the model converges, we use its weights to initialize models for other scales.

4.3. Ablation study

4.3.1. Study of the number of MSFFRBs

We study the effect of the number of MSFFRBs by training three models with a different number of MSFFRBs: 8, 9, and 10 with scaling factor 2. We use D = 8, D = 9, and D = 10 to represent the model with different number of MSFFRBs. In Table 1, it is shown that as we add more MSFFRB blocks into our network, the performance of our network become better. Since the SR speed is not the primary consideration in this work, we choose D = 10 as default number of blocks in our final model.

4.3.2. Study of the number of intertwined paths

To investigate the effect of the number of intertwined paths in MSFFRB, we first train three models with 2, 3, and 4 different number of intertwined paths with scaling factor 2. In these three models, the number of filters (c) in MSFFRBs are all 64. We use P = 2 (c = 64), P = 3 (c = 64), and P = 4 (c = 64) to represent the models with different number of intertwined paths and c = 64 to stand for the number of filters in MSFFRBs. As shown in Table 2, the more intertwined paths are used to extract multi-scale local image features, the better is the performance (PSNR and SSIM) of our model. Considering the number of parameters of the above models are different, to exclude the effect of different number of parameters, we adjust the number of filters (c) in the models with different number of intertwined paths and let the number of parameters among them have the same magnitude. To let the above three models with same number of parameters, we increase c to 128 for P = 2 (c = 64) and 90 for P = 3 (c = 64). We use P = 2 (c = 128) and P = 3 (c = 90) to represent the two new models with the almost same number of parameters as P = 4 (c = 64). We train these two new models and evaluate them on standard benchmarks. As shown in Table 2, we can observe that when the model have the same number of parameters, the more the number of intertwined paths, the better the performance (PSNR and SSIM).

Table 2
Investigations on the number of intertwined paths (P). We observe the best PSNR (dB) and SSIM values on Set5 ($\times 2$), Set14($\times 2$), B100 ($\times 2$), and Urban100 ($\times 2$). We use bold to label the first place. In the table, c stands for the number of filters in our MSFFRBs.

	P = 2 (c = 64) PSNR/SSIM	P = 2 (c = 128) PSNR/SSIM	P = 3 (c = 64) PSNR/SSIM	P = 3 (c = 90) PSNR/SSIM	P = 4 (c = 64) PSNR/SSIM
Set5	37.97/0.9604	38.09/0.9608	38.09/0.9607	38.12/0.9609	**38.15/0.9610**
Set14	33.54/0.9168	33.72/0.9185	33.69/0.9184	33.78/0.9189	**33.88/0.9195**
B100	32.14/0.8992	32.24/0.9004	32.22/0.9001	32.26/0.9007	**32.29/0.9010**
Urban100	31.97/0.9267	32.36/0.9305	32.29/0.9300	32.51/0.9316	**32.60/0.9326**

Please cite this article as: J. Qin, Y. Huang and W. Wen, Multi-scale feature fusion residual network for Single Image Super-Resolution, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.10.076

Table 4
The PSNR and SSIM results of different methods on Set5, Set14, B100, and Urban100 with down-sampling factor $\times 2$, $\times 3$, and $\times 4$. We use bold and italic to label first and second place, respectively.

Algorithm	Scale	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	URBAN100 PSNR/SSIM
Bicubic [45]	x2	33.69/0.9284	30.34/0.8675	29.57/0.8434	26.88/0.8438
A+ [46]	x2	36.60/0.9542	32.42/0.9059	31.24/0.8870	29.25/0.8955
SelfExSR [47]	x2	36.60/0.9537	32.46/0.9051	31.20/0.8863	29.55/0.8983
SRCNN [13]	x2	36.71/0.9536	32.32/0.9052	31.36/0.8880	29.54/0.8962
FSRCNN [14]	x2	37.06/0.9554	32.76/0.9078	31.53/0.8912	29.88/0.9024
VDSR [15]	x2	37.53/0.9583	33.05/0.9107	31.92/0.8965	30.79/0.9157
DRCN [17]	x2	37.63/0.9584	33.06/0.9108	31.85/0.8947	30.76/0.9147
LapSRN [16]	x2	37.52/0.9581	33.08/0.9109	31.80/0.8949	30.41/0.9112
CARN [48]	x2	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256
MSRN [32]	x2	38.08/0.9605	33.74/0.9170	32.23/0.9002	32.22/0.9326
MSFFRN (Ours)	x2	*38.15/0.9610*	*33.88/0.9195*	*32.29/0.9010*	*32.60/0.9326*
MSFFRN+ (Ours)	x2	**38.22/0.9613**	**33.93/0.9202**	**32.34/0.9015**	**32.77/0.9339**
Bicubic [45]	x3	30.41/0.8655	27.64/0.7722	27.21/0.7344	24.46/0.7411
A+ [46]	x3	32.63/0.9085	29.25/0.8194	28.31/0.7828	26.05/0.8019
SelfExSR [47]	x3	32.66/0.9089	29.34/0.8222	28.30/0.7839	26.45/0.8124
SRCNN [13]	x3	32.47/0.9067	29.23/0.8201	28.31/0.7832	26.25/0.8028
FSRCNN [14]	x3	33.20/0.9149	29.54/0.8277	28.55/0.7945	26.48/0.8175
VDSR [15]	x3	33.68/0.9201	29.86/0.8312	28.83/0.7966	27.15/0.8315
DRCN [17]	x3	33.85/0.9215	29.89/0.8317	28.81/0.7954	27.16/0.8311
LapSRN [16]	x3	33.82/0.9207	29.89/0.8304	28.82/0.7950	27.07/0.8298
CARN [48]	x3	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493
MSRN [32]	x3	34.38/0.9262	30.34/0.8395	29.08/0.8041	28.08/0.8554
MSFFRN (Ours)	x3	*34.65/0.9292*	*30.53/0.8463*	*29.23/0.8086*	*28.65/0.8619*
MSFFRN+ (Ours)	x3	**34.76/0.9299**	**30.65/0.8476**	**29.28/0.8096**	**28.82/0.8646**
Bicubic [45]	x4	28.43/0.8022	26.10/0.6936	25.97/0.6517	23.14/0.6599
A+ [46]	x4	30.33/0.8565	27.44/0.7450	26.83/0.6999	24.34/0.7211
SelfExSR [47]	x4	30.34/0.8593	27.55/0.7511	26.84/0.7032	24.83/0.7403
SRCNN [13]	x4	30.50/0.8573	27.62/0.7453	26.91/0.6994	24.53/0.7236
FSRCNN [14]	x4	30.73/0.8601	27.71/0.7488	26.98/0.7029	24.62/0.7272
VDSR [15]	x4	31.36/0.8796	28.11/0.7624	27.29/0.7167	25.18/0.7543
DRCN [17]	x4	31.56/0.8810	28.15/0.7627	27.24/0.7150	25.15/0.7530
LapSRN [16]	x4	31.54/0.8811	28.19/0.7635	27.32/0.7162	25.21/0.7564
CARN [48]	x4	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837
MSRN [32]	x4	32.07/0.8903	28.60/0.7751	27.52/0.7273	26.04/0.7896
MSFFRN (Ours)	x4	*32.44/0.8978*	*28.76/0.7860*	*27.67/0.7400*	*26.47/0.7980*
MSFFRN+ (Ours)	x4	**32.59/0.8996**	**28.89/0.7884**	**27.74/0.7417**	**26.67/0.8020**

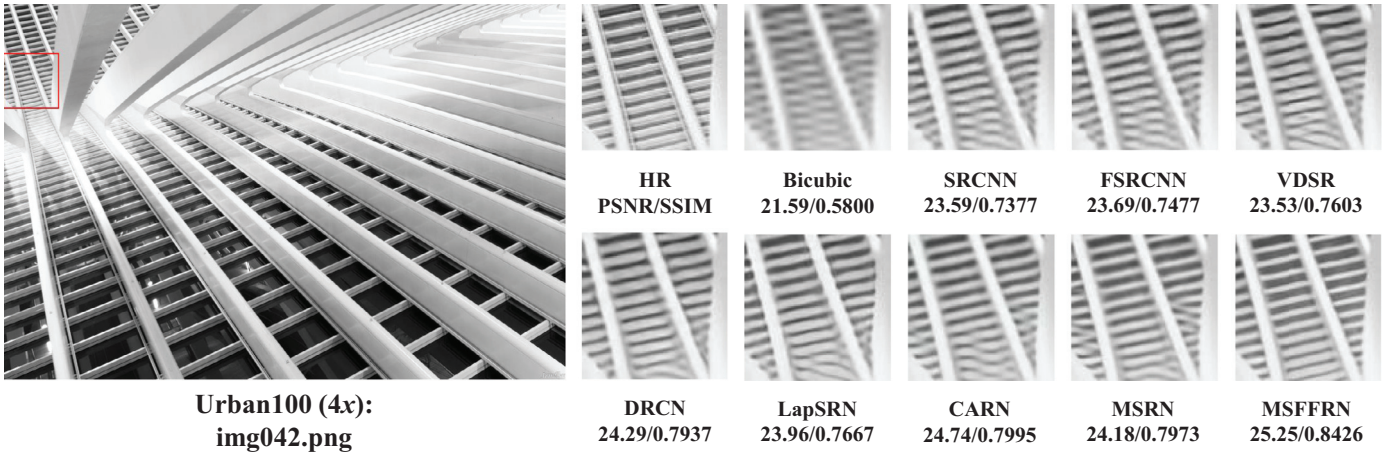

Fig. 4. The img042 image from the Urban100 dataset with an upscaling factor 4. (Zoom in for best view).

Since the model P = 4 (c = 64) outperforms other models by a large margin on all datasets, we set 4 as the default number of intertwined paths in MSFFRB.

4.3.3. Study of the fusion operation

To study the effect of different fusion operations in MSFFRB, we trained three models with different fusion operations: arithmetic mean, learnable weighted mean, and 1×1 conv with scaling factor 2. The performances of the models with different fusion operations on different datasets are shown in Table 3. From Table 3, we can see that the 1×1 conv operation achieves the best performance and outperforms other two fusion operations. Specifically, the 1×1 conv operation outperforms the arithmetic mean operation and learnable weighted mean operation by 0.14 dB and 0.11 dB on Urban100, respectively. Since the excellent performance of 1×1 conv fusion operation, we choose the 1×1 conv as the default fusion operation of our model.

Please cite this article as: J. Qin, Y. Huang and W. Wen, Multi-scale feature fusion residual network for Single Image Super-Resolution, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.10.076

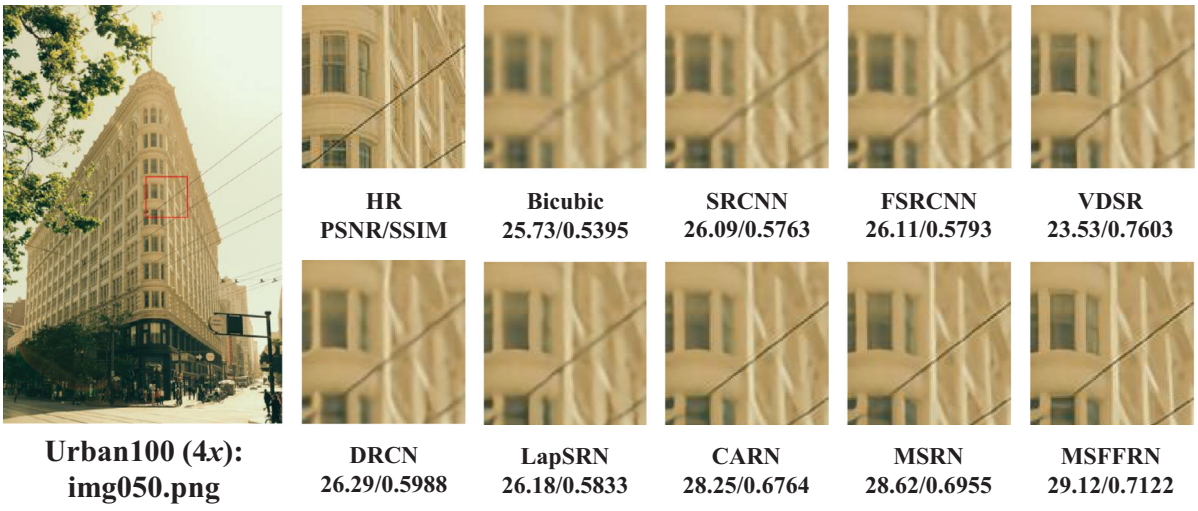

Fig. 5. The img050 image from the Urban100 dataset with an upscaling factor 4. (Zoom in for best view).

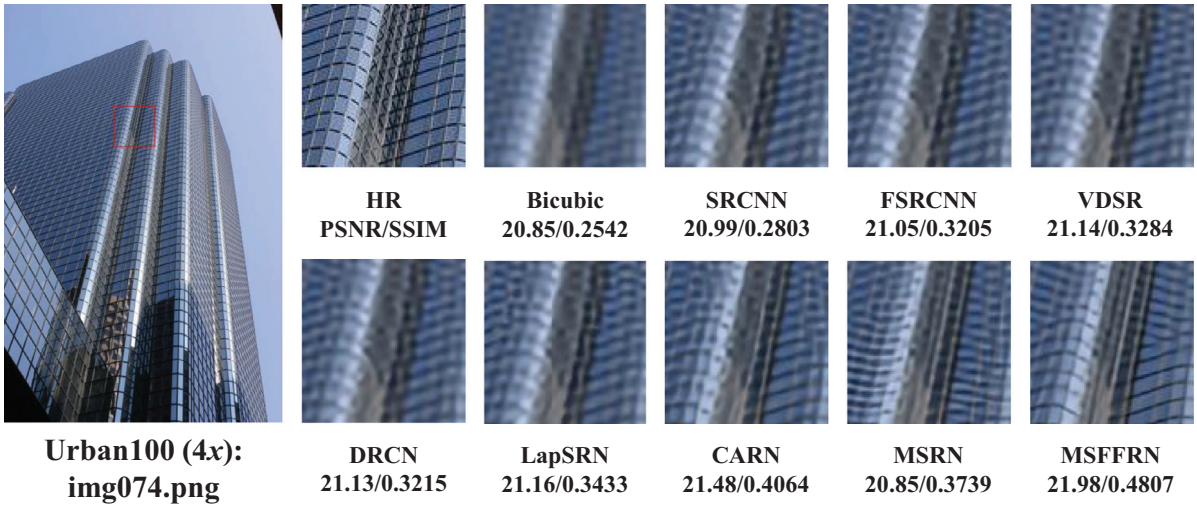

Fig. 6. The img074 image from the Urban100 dataset with an upscaling factor 4. (Zoom in for best view).

4.4. Comparisons with state-of-the-art methods

To verify the capability of our network, we perform a comprehensive comparison between our model and other 10 state-of-the-art SR methods, including Bicubic [45], A+ [46], SelfExSR [47], SRCNN [13], FSRCNN [14], VDSR [15], DRCN [17], LapSRN [16], CARN [48],and MSRN [32]. The specifications of EDSR [18], RDN [19], and RCAN [20] are much larger than our method. Specially, the number of parameters of EDSR (43M), RDN (23M), and RCAN (16M) are 6.85, 3.65, and 2.5 times greater than ours (6.4M), respectively. Thus, we do not compare our method with these three models. There are two reasons that we choose MSRN as our baseline. First, MSRN is a state-of-the-art SR model that uses multi-scale image feature for image super-resolution. Second, the number of parameters of our method (6.4M) and MSRN (6.3M) have the same order of magnitude. For these existing methods, we use the SR results publicly released by the authors. In addition, we also introduce self-ensemble strategy to further improve our MSFFRN and denote the self-ensembled one as MSFFRN+. For a fair comparison, the SR results are evaluated with PSNR and SSIM [22] on the Y channel (i.e., luminance) of transformed YCbCr space. All the reported PSNR/SSIM measures are calculated after removing M-pixel from each border (M means the upscaling factor).

The evaluation results of the SR methods including our model and 10 state-of-the-art methods for $\times 2$, $\times 3$, and $\times 4$ SR are demonstrated in Table 4. Compared with all previous methods, our MSFFRN+ performs the best on all the datasets with all scaling factors. Even without self-ensemble strategy, our MSFFRN outperforms other compared methods. From Table 4, we can see the proposed network outperforms MSRN [32] by a considerable margin in all scales which demonstrates the effectiveness of our proposed network. This also shows that our multi-scale feature fusion block is more effective than MSRB in MSRN [32]. Specifically, our approach MSFFRN outperforms MSRN with 0.43 dB and 0.37 dB on scale $\times 4$ on Urban100 and Set 5, respectively. Moreover, our model surpasses MSRN with 0.57 dB and 0.19 dB on Urban100 and Set14 on factor $\times 3$. Although our model is not designed for real-time super-resolution, we can process about 26 images and 6 images per second for upscaling factor $\times 3$ on B100 and Urban100 datasets with NVIDIA GTX 2080Ti GPU which is comparable to MSRN. To further analyze the proposed MSFFRN against other state-of-the-art SR methods in a qualitative manner, we also present several visual examples of SR images with $\times 4$ upscaling among different SR approaches. In Figs. 4–6, we show visual comparisons on scale $\times 4$. We can conclude that our proposed method can reconstruct the lines, the contours and the details better while

Please cite this article as: J. Qin, Y. Huang and W. Wen, Multi-scale feature fusion residual network for Single Image Super-Resolution, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.10.076

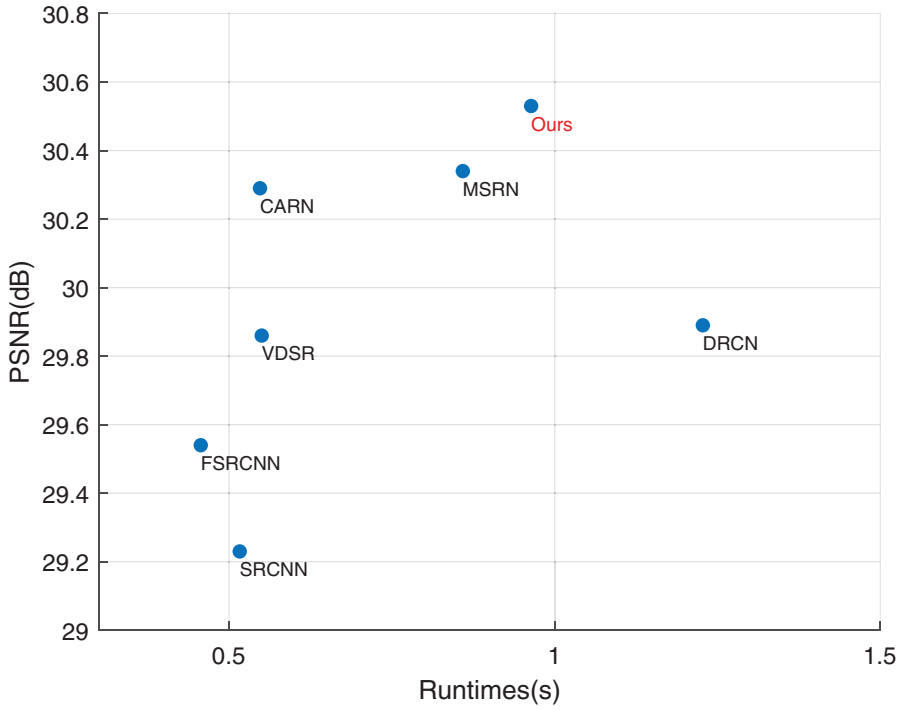

Fig. 7. Comparisons of accuracy and speed on Set14 with scale factor 3 for different approaches.

other methods suffer from blurring artifacts. There are two reasons for the performance improvement. Firstly, most of the other methods do not take full advantage of multi-scale image features to improve the representation ability of their network, except for MSRN [32]. Although MSRN can adaptively detect multi-scale image features, our multi-scale feature fusion residual block is more effective. In addition, we fuse hierarchical features from MSFFRBs and shallow feature extraction module for the final reconstruction.

4.5. Efficiency analysis

To verify the efficiency of our network, we evaluate running time of some representation methods, including SRCNN [13], FSRCNN [14], VDSR [15], DRCN [17], CARN [48], MSRN [32], and our method MSFFRN (Ours). We assessed the running time of all involved methods on Set14 for upscaling factor of $\times 3$ on NVIDIA RTX 2080Ti GPU and Cuda 10.0. The plot of accuracy and speed is shown in Fig. 7, where the results represent the mean PSNR and running time over all images of Set14 on our GPU. It is obvious that our method can achieve better trade-off. Our method is relatively fast while obtaining the best reconstruction performance among the representation methods we compared.

5. Conclusion

In this paper, we proposed an effective multi-scale feature fusion residual block (MSFFRB), which is used to adaptively extract and fuse the image features at different scales for more powerful representation. Based on MSFFRB, we proposed a novel multi-scale feature fusion residual network (MSFFRN) for SISR. The proposed network consists of five parts: shallow feature extraction module, deep feature extraction module based on our MSFFRB, global feature fusion module, upscale module, and reconstruction module. MSFFRN is a simple but effective SR model that can fully exploit the local multi-scale features and the hierarchical features to generate accurate SR images. Extensive evaluations on several standard benchmark datasets demonstrate that our network achieves superior performance than state-of-the-art methods we compared in terms of image quality metrics and visual quality.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research has been supported by grant 2017YFB0202502 and 2017YFB1001603 in the National Key R&D of China, by Project U1711264 supported by National Natural Science Foundation of China. The authors would like to thank the anonymous reviewers for their helpful and constructive comments and suggestions regarding this manuscript.

References

[1] W.T. Freeman, E.C. Pasztor, O.T. Carmichael, Learning low-level vision, Int. J. Comput. Vis. 40 (1) (2000) 25–47.
[2] W. Shi, J. Caballero, C. Ledig, X. Zhuang, W. Bai, K. Bhatia, A.M.S.M. de Marvao, T. Dawes, D. ORegan, D. Rueckert, Cardiac image super-resolution with global correspondence using multi-atlas patchmatch, in: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2013, pp. 9–16.
[3] W.W. Zou, P.C. Yuen, Very low resolution face recognition problem, IEEE Trans. Image Process. 21 (1) (2012) 327–340.
[4] K. Jiang, Z. Wang, P. Yi, J. Jiang, J. Xiao, Y. Yao, Deep distillation recursive network for remote sensing imagery super-resolution, Remote Sens. (Basel) 10 (11) (2018) 1700.
[5] K. Jiang, Z. Wang, P. Yi, J. Jiang, A progressively enhanced network for video satellite imagery superresolution, IEEE Signal Process Lett. 25 (11) (2018) 1630–1634.
[6] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, J. Jiang, Edge-enhanced GAN for remote sensing image superresolution, IEEE Trans. Geosci. Remote Sens. 57 (8) (2019) 5799–5812.
[7] X. Wang, K. Yu, C. Dong, et al., Recovering Realistic Texture in Image Super-Resolution by Deep Spatial Feature Transform, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 606–615.
[8] S.R. Kaiming He, X. Wang, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 770–778.
[9] L. Zhou, Z. Wang, Y. Luo, Z. Xiong, Separability and compactness network for image recognition and superresolution, IEEE Trans Neural Netw. Learn. Syst. (2019).

Please cite this article as: J. Qin, Y. Huang and W. Wen, Multi-scale feature fusion residual network for Single Image Super-Resolution, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.10.076

[10] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, P. Torr, Deeply supervised salient object detection with short connections, in: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 5300–5309.
[11] L. Zhang, X. Wu, An edge-guided image interpolation algorithm via directional filtering and data fusion, IEEE Trans. Image Process. 15 (8) (2006) 2226–2238.
[12] K. Zhang, X. Gao, D. Tao, X. Li, Single image super-resolution with non-local means and steering kernel regression, IEEE Trans. Image Process. 21 (11) (2012) 4544–4556.
[13] 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, Springer, 2014, pp. 184–199.
[14] C. Dong, C.C. Loy, X. Tang, Accelerating the super-resolution convolutional neural network, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 391–407.
[15] J. Kim, J. Kwon Lee, K. Mu 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.
[16] W.-S. Lai, J.-B. Huang, N. Ahuja, M.-H. Yang, Deep Laplacian pyramid networks for fast and accurate superresolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2, 2017, p. 5.
[17] J. Kim, J. Kwon Lee, K. Mu 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.
[18] 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 (CVPR) Workshops, 1, 2017, p. 4.
[19] 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 (CVPR), 2018.
[20] 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. 286–301.
[21] C. Ledig, Z. Wang, W. Shi, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2017, pp. 4681–4690.
[22] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[23] Y. Tai, J. Yang, X. Liu, Image super-resolution via deep recursive residual network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2017, pp. 2790–2798.
[24] 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, IEEE, 2017, pp. 4549–4557.
[25] W. Shi, J. Caballero, F. Huszár, 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.
[26] 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. 4700–4708.
[27] C. Szegedy, W. Liu, Y. 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.
[28] M. Haris, M.R. Widyanto, H. Nobuhara, Inception learning super-resolution, Appl. Opt. 56 (22) (2017) 6043–6048.
[29] F. Li, H. Bai, Y. Zhao, Filternet: adaptive information filtering network for accurate and fast image super-resolution, IEEE Trans. Circuits Syst. Video Technol. (2019) In press, doi:10.1109/TCSVT.2019.2906428.
[30] 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) (2018) 2530–2544.
[31] T. Tong, G. Li, X. Liu, Q. Gao, Image super-resolution using dense skip connections, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4799–4807.
[32] J. Li, F. Fang, K. Mei, G. Zhang, Multi-scale residual network for image super-resolution, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 517–532.
[33] G. Larsson, M. Maire, G. Shakhnarovich, Fractalnet: ultra-deep neural networks without residuals, in: Proceedings of International Conference on Learning Representations, 2017.
[34] V. Dumoulin, J. Shlens, M. Kudlur, A learned representation for artistic style, Proceedings of ICLR 2(2017).
[35] A. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, W. Shi, Checkerboard artifact free sub-pixel convolution: a note on sub-pixel convolution, resize convolution and convolution resize, arXiv:1707.02937 (2017).
[36] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of International Conference on Learning Representations, 2015.
[37] H. Zhao, O. Gallo, I. Frosio, J. Kautz, Loss functions for image restoration with neural networks, IEEE Trans. Comput. Imaging 3 (1) (2017) 47–57.
[38] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, 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, 2017, pp. 114–125.
[39] M. Bevilacqua, A. Roumy, C. Guillemot, M.L. Alberi-Morel, Low-complexity single-image super-resolution based on nonnegative neighbor embedding, in: Proceedings of British Machine Vision Conference, BMVA Press, 2012, pp. 1–10.
[40] R. Zeyde, M. Elad, M. Protter, On single image scale-up using sparse-representations, in: Proceedings of the International Conference on Curves and Surfaces, Springer, 2010, pp. 711–730.
[41] 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. ICCV 2001, 2, IEEE, 2001, pp. 416–423.
[42] J.-B. 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.
[43] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proceedings of the ICLR, 2015.
[44] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, Advances in Neural Information Processing Systems Autodiff Workshop (2017).
[45] R. Keys, Cubic convolution interpolation for digital image processing, IEEE Trans. Acoust. 29 (6) (1981) 1153–1160.
[46] R. Timofte, V. De, L.V. Gool, Anchored neighborhood regression for fast example-based super-resolution, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1920–1927.
[47] J.B. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: Proceedings of the Computer Vision and Pattern Recognition, 2015, pp. 5197–5206.
[48] N. Ahn, B. Kang, K.-A. Sohn, Fast, accurate, and lightweight super-resolution with cascading residual network, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 252–268.

Jinghui Qin received his B.E. and M.E. degree in software engineering from Sun Yat-Sen University, Guangzhou, China in 2012 and 2014. He is currently working towards a Ph.D. degree at the School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China. His research interests include computer vision, machine learning, and deep learning.

Yongjie Huang received his B.E. degree in software engineering from Sun Yat-Sen University, Guangzhou, China in 2017. He is currently working towards a M.E. degree at the School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China. His research interests include computer vision and machine learning.

Wushao Wen is a professor in the School of Data and Computer Science,Sun Yat-Sen University, P.R. China. He received the B.S. degree from the University of Science and Technology of China in 1993, the M.S. and Ph.D. degrees from the University of California, Davis, in 1999 and 2001, respectively. He was an Engineer and Project Manager with China Telecommunication, Inc., from 1993 to 1997. He held various leading engineer and technical management positions in Cisco System, Ciena Corporation, Juniper Networks from 2000 to 2010 in Silicon Valley, USA, all in networking and security areas. He was appointed as the Chief Director of the Networks and Information Center, the Director of Shared Experimentation Teaching Center, Sun Yat-Sen University, P.R. China in 2013, 2014 respectively. He is currently doing research in cloud computing, network security, network architectures, machine learning, and deep learning.

Please cite this article as: J. Qin, Y. Huang and W. Wen, Multi-scale feature fusion residual network for Single Image Super-Resolution, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.10.076