Signal Processing 179 (2021) 107831

Contents lists available at ScienceDirect

Signal Processing

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

Single image super-resolution with multi-scale information cross-fusion network

Yanting Hu[a], Xinbo Gao[b,*], Jie Li[c], Yuanfei Huang[c], Hanzi Wang[d]

[a] School of Medical Engineering and Technology, Xinjiang Medical University, Urumqi 830011, China
[b] The Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[c] Video and Image Processing System Laboratory, School of Electronic Engineering, Xidian University, Xi'an 710071, China
[d] Fujian Key Laboratory of Sensing and Computing for Smart City, School of Information Science and Engineering, Xiamen University, Xiamen 361005, China

ARTICLE INFO

Article history:
Received 5 November 2018
Revised 10 May 2020
Accepted 28 September 2020
Available online 6 October 2020

Keywords:
Multi-scale information
Residual learning
Cascaded networks
Single image super-resolution

ABSTRACT

The deep convolutional neural networks have achieved significant improvements in terms of effectiveness and efficiency for single image super-resolution. However, as the depth of neural network grows, the information flow is weakened. On the other hand, most of the models adopt a single-stream structure with which integrating complementary information under different receptive fields is difficult. To improve information flow and capture sufficient knowledge for reconstructing high-frequency details, we propose a multi-scale information cross-fusion network (MSICF) in which a sequence of subnetworks is cascaded to infer high resolution features in a coarse-to-fine manner. In each cascaded subnetwork, we stack multiple multi-scale cross (MSC) modules to fuse multi-scale complementary information in an efficient way as well as to improve information flow across the layers. Meanwhile, by introducing residual-feature learning in each stage, the relative information between features at different levels is fully utilized to further boost reconstruction performance. We train the proposed network with cascaded-supervision and then assemble the intermediate predictions of the cascade to achieve high quality image. Extensive quantitative and qualitative evaluations on benchmark datasets illustrate the superiority of our proposed method over state-of-the-art super-resolution methods.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Single image super-resolution (SR) aims to infer a high resolution (HR) image from its single low resolution (LR) counterpart, which has wide applications in video surveillance, remote sensing imaging, medical imaging and digital entertainment. Since the SR process is inherently an ill-posed inverse problem, exploring and enforcing strong prior information about HR image are necessary to guarantee the stability of SR process. Many traditional example-based SR methods have been devoted to resolving this problem via probabilistic graphical model [1], neighbour embedding [2,3], sparse coding [4] and linear or nonlinear regression [5–7].

More recently, deep neural networks have been utilized for image SR via modeling the mapping from LR to HR space and achieved impressive performances in terms of effectiveness and efficiency [8]. Dong et al. [9] first develop a deep convolutional neural network (CNN) with three convolutional layers (SRCNN) to predict the nonlinear relationship between interpolated middle resolution image and HR one. Since the deeper CNNs by stacking more convolutional layers have larger receptive fields to capture more contextual information, Kim et al. construct two deep neural networks of VDSR [10] and DRCN [11], which utilize global residual learning and recursive layers respectively to improve SR accuracy. To control the number of model parameters and fuse multi-level features, Tai et al. build recursive modules with global and local residual learning in Tai et al. [12] and construct memory blocks with a densely connected structure in Tai et al. [13]. Taking VDSR as a baseline architecture, Plötz et al. [14] insert their proposed neural nearest neighbors blocks to incorporate non-local operation into SR model for further performance improvement. Later, Dong et al. [15] extend SRCNN by introducing a deconvolution layer at the end of the network to directly learn the mapping from original LR image to HR one. Following this architecture, Ledig et al. [16] construct a deep residual network (SRResNet) with 16 residual blocks [17] to super-resolve LR image and combine it with adversarial learning to obtain photo-realistic images. Lim et al. [18] develop an enhanced deep super-resolution network by removing the batch normalization layers in SRResNet and win the NTIRE2017 super-resolution challenge [19]. And Hui et al. utilize the iden-

* Corresponding author.
E-mail address: gaoxb@cqupt.edu.cn (X. Gao).

https://doi.org/10.1016/j.sigpro.2020.107831
0165-1684/© 2020 Elsevier B.V. All rights reserved.

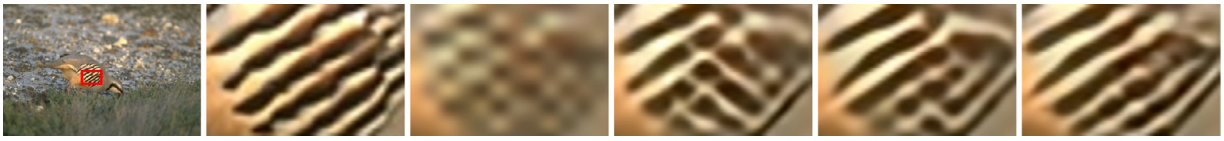
tity mapping of residual block [17] and the channel split operation to construct two lightweight distillation networks [20,21] (i.e., IDN and IMDN) for achieving a balance among reconstruction quality, inference speed, and memory occupation. In [22], He et al. propose to apply an ordinary differential equation (ODE)-inspired scheme to design CNN for relieving the dilemma between enhancing SR performance and reducing computation overhead. Inspired by attention mechanism, Zhang et al. integrate the channel-wise attention into the residual blocks in Zhang et al. [23] and further propose residual local and non-local attention learning in Zhang et al. [24] for feature recalibration. Meanwhile, Yang et al. [25] utilize attention mechanism to aggregate the features at different levels in a self-adaptive channel-wise convex weighted way and propose a lightweight feature fusion network (LFFN) to make a better trade-off between model size and SR performance. Chang et al. [26] also integrate channel-wise attention into the multi-path wide-activated residual blocks to well exploit multi-level features for detail recovery. Further, to conveniently pass information across several layers or modules, the densely connected structures [27] are also adopted in SR models. Tong et al. [28] utilize the blocks in DenseNet [27] as building modules to reuse learnt features and build dense skip-connections to fuse multi-level features. Zhang et al. [29] combine local dense connections and residual learning to construct a residual dense network to super-resolve LR image. Recently, Yang et al. [30] take advantage of ResNet's [17] parameter-economic feature re-usage and DenseNet's [27] distinguishing feature exploration to design a concise but efficient network, where an adaptive element-wise fusion strategy is combined with multi-supervised training process to further improve SR performance. In addition, [31,32] incorporate a feedback mechanism into network designs for exploiting both LR and HR signals jointly.

Assembling a set of subnetworks, adopted in some image SR methods, is also an effective solution to improve performance. Tang et al. [33] propose a joint residual network with three parallel subnetworks to learn the low-frequency information and high-frequency information for SR. Yamanaka et al. [34] integrate skip-connection layers and parallelized reconstruction network into a deep CNN architecture for achieving better reconstruction performance with less computational resources. Also, [35] discusses different subnetwork fusion schemes and proposes a context-wise network fusion approach which integrates the outputs of individual networks by additional convolutional layers. Recently, Wang et al. [36] build a controllable feature space network by entangling a main branch and a tuning branch for interactive image restoration in a fine-grained way. In addition to parallel network fusion, progressive network fusion or network-cascading structure is adopted in SR. Wang et al. [37] exploit the sparse representation prior and build a cascaded sparse coding network in which each subnetwork is trained for a small upscaling factor. Recently, the cascade of convolutional neural networks is also utilized in Cui et al. [38], Lai et al. [39], Wang et al. [40] to progressively upscale low-resolution images.

For image SR, the input and output as well as information among layers in networks are highly correlated. It is important for SR to combine the knowledge of features at different levels and different scales. Although previous SR approaches utilize the deeper networks to take more contextual information, the fusion of multi-scale complementary information under different receptive fields is still difficult due to adopting single-stream structure in their designed networks, such as SRCNN [9], FSRCNN [15], VDSR [10] and DRCN [11]. Besides, it is found in Zagoruyko and Komodakis [41], Zhao et al. [42] that for deep networks, as a way to increase the number of layers, increasing the width is more effective than increasing the depth. Therefore, in order to conveniently promote information integration for image SR, the multi-stream structure and network widening may be beneficial. On the other hand, applying the residual information learning and the cascaded or progressive structure into SR can simplify the difficulty of direct super-resolving images, which has been manifested in several SR methods. Taking the above into consideration, we propose a deep Multi-Scale Information Cross-Fusion network (MSICF) for SR (illustrated in Fig. 2), which consists of a feature extraction network, a set of cascaded subnetworks and a reconstruction network. A set of cascaded subnetworks is utilized to gradually reconstruct HR features. In each subnetwork (shown in Fig. 2(b)), we develop a multi-scale cross (MSC) module with two branches (depicted in Fig. 3(b)) for fusing multi-scale information under different receptive fields, and then stack several MSC modules to learn the residual information between input and output features. During training, the cascaded-supervision strategy is adopted to supervise all of the predictions from subnetworks and the final output from overall MSICF model. As illustrated in Fig. 1, compared with other SR methods (VDSR [10], MemNet [13]), our proposed MSICF network generates more realistic result. The major contributions of our method are summarized as follows:

(1) We develop a multi-scale cross (MSC) module where two branches having different receptive fields are assembled in parallel via averaging and adding. The MSC module can not only fuse complementary information under different receptive fields but also help information flow across the network.
(2) We construct a subnetwork via stacking multiple MSC modules and adopting residual-feature learning, which has a cross-fusion structure to capture a wide variety of representations for inferring high resolution features.
(3) We design an MSICF network by cascading several residual-feature learning subnetworks to reconstruct high resolution image in a coarse-to-fine manner, where the outputs from the cascaded subnetworks are fed into the corresponding reconstruction networks to obtain intermediate predictions, and then the cascaded-supervision as well as the aggregation of intermediate predictions are adopted to boost SR performance.

The remainder of this paper is organized as follows. Section 2 discusses the related single image SR methods and network architectures. Section 3 describes the proposed MSICF network in detail. Model analysis and experimental comparisons with other state-of-the-art methods are presented in Section 4,



"*8023*" image from BSD100	Ground truth PSNR / SSIM / IFC	Bicubic 28.50 / 0.8291 / 2.6513	VDSR 29.54 / 0.8651 / 3.3878	MemNet 30.14 / 0.8697 / 3.5634	MSICF (ours) **30.62 / 0.8721 / 3.6621**

Fig. 1. The comparisons of our method (MSICF) with existing methods on single image super-resolution for a scale factor of 4 ×. The proposed MSICF produces more faithful textures on the wing.

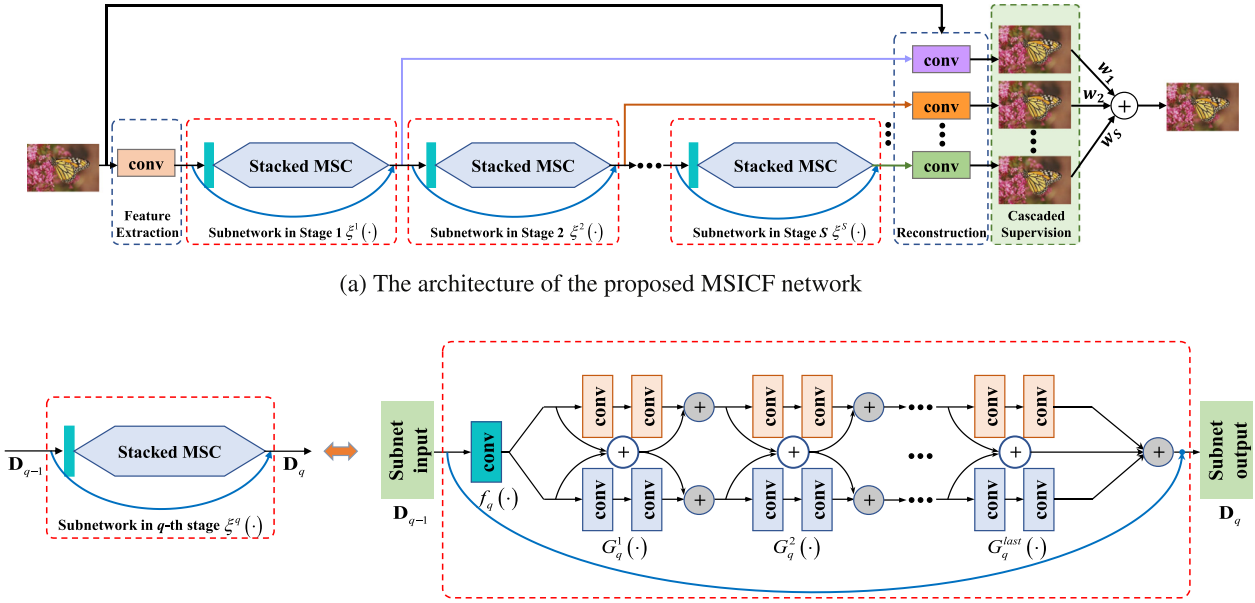


Fig. 2. The architectures of our proposed MSICF model and each cascaded subnetwork in MSICF model. (a) The overall architecture of the proposed MSICF model, which utilizes cascaded structure, cascaded-supervision and prediction assembling to boost SR performance. (b) The architecture of each subnetwork in (a), which applies a sequence of MSC modules to capture rich information for inferring HR features. The blue arrow denotes the skip-connection for residual-feature learning. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

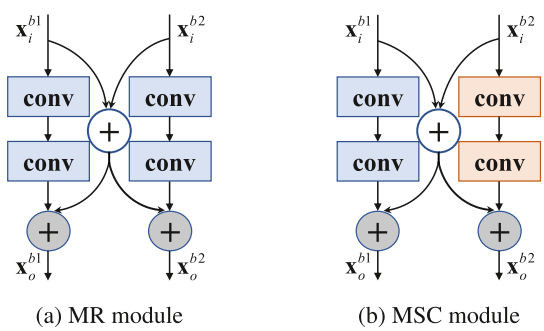


Fig. 3. The built modules: (a) A merge-and-run (MR) module, in which all convolutional filters have the same size. (b) A multi-scale cross (MSC) module, where the convolutional filters in two branches have different sizes and are denoted by different color rectangles for discrimination. The MSC module is designed to fuse information under different receptive fields and improve information flow. In (a) and (b), the "+" operators in white circles denote the average operation and in gray circles denote the element-wise addition. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and Section 5 concludes the paper with observations and discussions.

2. Related work

Numerous single image SR methods and different network architectures have been proposed in the literatures. Here, we focus our discussions on the approaches which are related to our method.

2.1. Multi-branch module

To improve information flow and make training easy, the networks consisting of multi-branch modules have been developed, such as highway networks [43], residual networks [17,44], and GoogLeNet [45,46]. Residual network is built by stacking a sequence of residual blocks which contain a residual mapping branch and an identity mapping branch. The inception module in GoogLeNet consists of parallel convolutional branches with different kernel sizes which are then concatenated at the end of the module for width increase and information fusion. The inception module is modified in Yang et al. [25], Chang et al. [26] to take advantage of the different types of information for SR performance improvement. Recently, Zhao et al. [42] propose a deep merge-and-run neural network, which contains a set of merge-and-run (MR) blocks (depicted in Fig. 3(a)). The MR block assembles the residual branches in parallel with a merge-and-run mapping which is shown as a linear idempotent function in Zhao et al. [42]. The idempotent mapping implies that the information from the early blocks can quickly flow to the later blocks and the gradient also can be quickly back-propagated to the early blocks from the later blocks, and thus the training difficulty is reduced. Although these different methods vary in network topology and training procedure, they all share a key characteristic: containing multiple branches in each block. On the other hand, multi-branch networks can be viewed as an ensemble of many subnetworks with different depths by which the performance could be boosted. Considering the advantages of multi-branch module, we design an MSC module via operating the convolutions with different kernel sizes on two parallel branches to fuse information at different scales for detail recovery.

2.2. Residual learning

Since the SR prediction is largely similar to the input, residual learning is widely adopted in SR methods to achieve faster convergence and better accuracy. In [47], residual feature patches between the estimated HR feature patches and the ground truth feature patches are estimated via a set of cascaded linear regressions. In deep learning-based SR, two networks of VDSR [10] and DRCN [11] are built to learn the residual image between HR and LR images by using a skip-connection from the input to the reconstruction layer. Later, the residual learning is extensively utilized in different SR networks [12,13,16,18,20–26,28–30,33,34]. Tai

et al. [12] term the above way of estimating the residual image by the skip-connection as global residual learning (GRL) (like in VDSR [10] and DRCN [11]) and introduce a multi-path mode for local residual learning which is combined with GRL to boost SR performance. Zhang et al. [23,29] also combine the local residual learning and global residual learning to improve the information flow and stabilize the training process. Taking the effectiveness of GRL in training deep networks into account, we also adopt GRL in our proposed method. Further, we introduce residual learning to the feature space, termed as residual-feature learning (RFL), which is performed in each stage of the cascaded process.

2.3. Cascaded structure

In image SR, reconstructing high-frequency details becomes very challenging when the upscaling factor is large. To simplify the difficulty of direct super-resolving the details, some progressive or cascaded structures are proposed. There are two fashions in cascaded structure for SR: stage-by-stage refining and stage-by-stage upscaling. In former manner, the output of the previous stage is taken as the input and the ground truth as target for each stage, where the input and output have the same size. Meanwhile, the cascade minimizes the prediction errors at each stage and thus the prediction is gradually close to the target. Hu et al. [47] develop a framework of cascaded linear regressions to refine the predicted feature patches in a coarse-to-fine fashion and then merge all predicted patches to generate an HR image. In stage-by-stage upscaling manner, one stage of the cascade is utilized to upscale LR image with a smaller scale factor, the output of which is further fed into the next stage until the desired image scale. With this manner, deep network cascade is proposed in Cui et al. [38] by using a cascade of collaborative local auto-encoders to gradually upscale low resolution image. Recently, Lai et al. [39] present a Laplacian pyramid super-resolution network (LapSRN) based on a cascade of convolutional neural networks, which can progressively predict the sub-band residuals of HR image at multiple pyramid levels. To supervise intermediate stages in LapSRN, different scales of reference images at corresponding levels need be generated by downsampling the ground truth HR image. In [40], an asymmetric structure trained via the curriculum learning is designed to incrementally upsample the input image, and the same progressive principle is further extended to the generative adversarial network for improving the perceptual quality of super-resolved image. Compared with the refining mode in which the cascade explicitly minimizes the prediction errors at each stage, this incremental approach has a loose control over the errors [48]. In the light of the above, we utilize the cascaded structure with a stage-by-stage refining fashion to gradually refine reconstructed image.

3. The proposed MSICF network

Our proposed MSICF model for image SR, outlined in Fig. 2, consists of a feature extraction network, a set of cascaded subnetworks and a reconstruction network. The feature extraction network is applied to represent the input image as a set of feature maps via a convolutional layer. A set of cascaded subnetworks is designed to reconstruct HR features from extracted shallow features in a coarse-to-fine manner. The reconstructed HR features are then fed into the reconstruction network to generate HR image via the convolution operations. In this section, we describe the proposed model in detail, from the multi-scale cross module to the residual-feature learning subnetwork and finally the overall cascaded network.

3.1. The multi-scale cross module

Due to the identity mappings to skip residual branches in ResNet [17], a deep residual network is easy to train. Similarly, to further reduce the training difficulty and improve information flow, a deep merge-and-run neural network, built by stacking a sequence of merge-and-run (MR) blocks, is proposed in Zhao et al. [42] for the standard image classification task. Depicted in Fig. 3(a), the MR block assembles the residual branches in parallel with a merge-and-run mapping: averaging the inputs of these residual branches and adding the average to the output of each residual branch as the input of subsequent residual branch respectively.

Inspired by MR block and considering that convolutional layers with different receptive fields can capture different contextual information which is very important for image SR, we propose a multi-scale cross (MSC) module to incorporate different information under different sizes of receptive fields. As shown in Fig. 3(b), an MSC module similarly integrates two residual branches in parallel with a merge-and-run mapping but operates different kernel sizes of convolutions on two branches, where different convolutions are differentiated by different color rectangles in Fig. 3(b). Each residual branch in MSC module contains two convolutional layers and each convolutional layer is followed by batch normalization (BN) [49] and LeakyReLU [50] layers. Moreover, the "+" operators in gray circles in Fig. 3 are between BN and LeakyReLU layers and denote the addition operation, while the "+" operators in white circles denote the average operation. Thus, with this design, these branches can provide complementary information at multiple scales which is further combined by the merge-and-run mapping. By denoting $H^{b1}(\cdot)$ and $H^{b2}(\cdot)$ as the transition functions of two residual branches respectively, the proposed module can be represented in matrix form as below.

$$\begin{bmatrix}\mathbf{x}_o^{b1}\\ \mathbf{x}_o^{b2}\end{bmatrix} = G\left(\begin{bmatrix}\mathbf{x}_i^{b1}\\ \mathbf{x}_i^{b2}\end{bmatrix}\right) = \begin{bmatrix}H^{b1}\left(\mathbf{x}_i^{b1}\right)\\ H^{b2}\left(\mathbf{x}_i^{b2}\right)\end{bmatrix} + \frac{1}{2}\begin{bmatrix}\mathbf{I} & \mathbf{I}\\ \mathbf{I} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{x}_i^{b1}\\ \mathbf{x}_i^{b2}\end{bmatrix}, \tag{1}$$

where $\mathbf{x}_i^{b1}$ and $\mathbf{x}_i^{b2}$ ($\mathbf{x}_o^{b1}$ and $\mathbf{x}_o^{b2}$) are the inputs (outputs) of two residual branches of the module and $\mathbf{I}$ is identity matrix. $\mathbf{P} = \frac{1}{2}\begin{bmatrix}\mathbf{I} & \mathbf{I}\\ \mathbf{I} & \mathbf{I}\end{bmatrix}$ is the transformation matrix of merge-and-run mapping. With this multi-scale information cross-fusion structure, the proposed module can exploit a wide variety of contextual information to infer missing high-frequency components. On the other hand, as analyzed in Zhao et al. [42], the transformation matrix $\mathbf{P}$ in merge-and-run mapping is idempotent, which is similarly possessed by the proposed MSC module. The property of idempotent not only helps rich information flow across different modules, but also encourages gradient back-propagation during training. Experimental analysis is described in Section 4.3.1.

3.2. The residual-feature learning subnetwork

Aiming to infer HR features, the subnetwork with the residual-feature learning (RFL) is built. As illustrated in Fig. 2(b), the building subnetwork contains a convolutional layer and a sequence of multi-scale cross (MSC) modules. At the end of the last MSC module, two outputs and the average of two inputs from two branches are fused by addition, which can be formulated as follows.

$$\mathbf{x}_o = G^{last}\left(\begin{bmatrix}\mathbf{x}_i^{b1}\\ \mathbf{x}_i^{b2}\end{bmatrix}\right) = H^{b1}\left(\mathbf{x}_i^{b1}\right) + H^{b2}\left(\mathbf{x}_i^{b2}\right) + \frac{1}{2}\left(\mathbf{x}_i^{b1} + \mathbf{x}_i^{b2}\right), \tag{2}$$

where $\mathbf{x}_o$ denotes the output from the last MSC module and the other notations have the same meanings as those in Eq. (1). The first convolutional layer in the subnetwork is utilized to transform the input as a specified number of feature maps. Then, with the parallel and intersecting structure, the stacked multi-scale cross

(MSC) modules can be used to process these features via different processing procedures. Meanwhile, multiple merge-and-run mappings in the subnetwork are exploited to integrate different information coming from two branches as well as to create quick paths for sending the information of the intermediate branches to the later modules.

Since the features of HR image are highly correlated with the features of corresponding LR image, we introduce residual-feature learning into our subnetwork by adding an identity branch from the input to the end of the subnetwork (blue curve in Fig. 2(b)). Therefore, instead of inferring HR features directly, the subnetwork is built to estimate the residual features between input and target features. We denote $\mathbf{D}_{q-1}$ and $\mathbf{D}_q$ as the input and the output of the qth subnetwork, M as the number of MSC modules for stacking, and $f_q(\cdot)$ as the convolution operation of the first convolutional layer. Thus, the output of the qth subnetwork is

$$\begin{aligned}\mathbf{D}_q = \xi^q\left(\mathbf{D}_{q-1}\right) &= G_q^{last}\left(G_q^{(M-1)}\left(\begin{bmatrix} f_q\left(\mathbf{D}_{q-1}\right) \\ f_q\left(\mathbf{D}_{q-1}\right)\end{bmatrix}\right)\right) + \mathbf{D}_{q-1} \\ &= G_q^{last}\left(G_q^{M-1}\left(\cdots G_q^1\left(\begin{bmatrix} f_q\left(\mathbf{D}_{q-1}\right) \\ f_q\left(\mathbf{D}_{q-1}\right)\end{bmatrix}\right)\cdots\right)\right) + \mathbf{D}_{q-1},\end{aligned} \tag{3}$$

where $\left\{G_q^m(\cdot)\right\}_{m=1}^{M-1}$ are functions for the MSC modules in the qth subnetwork (depicted in Eq. (1)) and $G_q^{last}(\cdot)$ denotes the function of the last MSC module (described in Eq. (2)). Since M-fold operations of $G(\cdot)$ and $G^{last}(\cdot)$ are performed as well as the residual-feature learning is applied, the subnetwork is able to capture different representations for inferring HR features.

3.3. The overall model with cascaded structure

In order to reconstruct HR features and reduce the difficulty of direct super-resolving the details, we build a cascade of subnetworks to reconstruct HR features from shallow features extracted by the feature extraction network. All subnetworks share the same structure and setting as mentioned in Section 3.2 and are jointly trained with simultaneous supervision. It is expected that each stage of subnetwork brings the predictions closer to the ground truth HR features and the cascade progressively minimizes the prediction errors. Then, all the reconstructed HR features from the subnetworks are fed into the corresponding reconstruction layers to obtain HR image. The full model is illustrated in Fig. 2 and termed as multi-scale information cross-fusion network (MSICF).

Let $\mathbf{x}$ and $\hat{\mathbf{y}}$ denote the input and output of the MSICF network. And, a convolutional layer followed by BN and LeakyReLU layers is utilized as a feature extraction network to extract shallow features from LR input image. The feature extraction network is formulated as below.

$$\mathbf{D}_0 = F(\mathbf{x}), \tag{4}$$

where $F(\cdot)$ denotes the operation of the feature extraction network and $\mathbf{D}_0$ is the extracted features which are then fed into the first stage of subnetwork. Supposing S subnetworks are cascaded to progressively infer HR features, following the notations in Section 3.2, the process of inference is represented as follows.

$$\mathbf{D}_S = \xi^S\left(\xi^{S-1}\left(\cdots\left(\xi^1(\mathbf{D}_0)\right)\cdots\right)\right), \tag{5}$$

where $\xi^q(\cdot)(q = 1, 2, \cdots, S)$ represents the function for the qth subnetwork, as depicted in Eq. (3). In order to make the output from each cascaded subnetwork closer to the ground truth HR features, we supervise all predictions from cascaded subnetworks via cascaded-supervision. The output of each subnetwork is fed into the reconstruction network where each convolutional layer takes the output of the corresponding stage as its input to reconstruct the corresponding residual image, and then S intermediate predictions from the cascaded stages are generated, as illustrated in Fig. 2(a). Similar to [10–13], we also adopt global residual learning in our network via adding an identity branch from the input to the reconstruction network. Thus, the qth$(q = 1, 2, \cdots, S)$ intermediate prediction is

$$\hat{\mathbf{y}}_q = R_q(\mathbf{D}_q) + \mathbf{x}, \tag{6}$$

where $\mathbf{D}_q$ is the output features of the qth stage and $R_q(\cdot)$ denotes the function for the corresponding reconstruction layer. And then, all of S intermediate predictions are assembled to generate the final output via weighted average.

$$\hat{\mathbf{y}} = \eta \cdot \sum_{q=1}^{S} w_q \cdot \hat{\mathbf{y}}_q, \tag{7}$$

where $\eta = \frac{1}{\sum_{q=1}^{S} w_q}$ and w_q denotes the weight for the prediction from the qth subnetwork. All of the weights in the above equation are learned during training and the final output $\hat{\mathbf{y}}$ is also supervised.

Given a training dataset $\left\{\mathbf{x}^{(k)}, \mathbf{y}^{(k)}\right\}_{k=1}^{K}$, where K is the number of training patches and $\{\mathbf{x}^{(k)}, \mathbf{y}^{(k)}\}$ are the kth LR and HR patch pairs, the loss function of our model with cascaded-supervision can be formulated as

$$\begin{aligned}L(\Theta) = \alpha \cdot \sum_{k=1}^{K} \frac{1}{2K}\left\|\mathbf{y}^{(k)} - \eta \cdot \sum_{q=1}^{S} w_q \cdot \hat{\mathbf{y}}_q^{(k)}\right\|^2 \\ +(1-\alpha) \cdot \sum_{q=1}^{S}\sum_{k=1}^{K} \frac{1}{2SK}\left\|\mathbf{y}^{(k)} - \hat{\mathbf{y}}_q^{(k)}\right\|^2,\end{aligned} \tag{8}$$

where Θ denotes the parameter set. The α balances the losses on the final output and on the intermediate outputs, and is empirically set as $\frac{2}{S+2}$.

When the depth of a network is defined as the longest path from the input to the output, the depth of our overall model can be calculated by

$$Depth = (M \cdot 2 + 1) \cdot S + 2, \tag{9}$$

where S and M denote the number of cascaded subnetworks and the number of MSC modules in each subnetwork. The 2 multiplied by M represents two convolutional layers contained in a branch of MSC module, and the 1 in the parentheses indicates the first convolutional layer in each subnetwork. Besides, the 2 at the end of equation represents two convolutional layers in feature extraction network and in reconstruction network.

The ensemble of intermediate predictions and multi-supervision are also used in DRCN [11] and MemNet [13]. However, there are two main differences between our MSICF and the two methods. The first difference is the design of the cascaded module. DRCN utilizes a convolutional layer as the cascaded module and recursively stacks the same convolutional layer in a chain structure. In MemNet, the memory block, containing recursive convolutional layers and gate unit, is constructed as the cascaded module. In both models, the most layers hardly receive the information from the preceding layers or blocks. In our model, the cascaded subnetwork is built by stacking multiple MSC modules with a parallel and intersecting structure and adding skip-connection for residual-feature learning. Since the transformation matrix in MSC module is an idemponent matrix (analyzed in Section 3.1), the information of each convolutional layer can easily flow into the subsequent or preceding layers during forward propagation or gradient back-propagation. Besides, the intersecting structure enables the multi-scale complementary information to be fused and the skip-connection can further improve information

flow across the network, the effectiveness of which in improving performance is demonstrated in Section 4. The second difference is that in DRCN and MemNet, the reconstruction network is shared by all cascaded stages, while in MSICF, the reconstruction layers are different for the cascaded subnetworks. And, as illustrated in Section 4, using different reconstruction layers can further boost SR performance, which is different from the description in Kim et al. [11].

4. Experiments and analysis

4.1. Benchmarks

We conduct comparative studies on widely used datasets, Set5 [51], Set14 [52], BSD100 [53] and Urban100 [54], which contain 5, 14, 100 and 100 images respectively. We use a training set of 291 images to train our model, which consists of 91 images from Yang et al. [4] and 200 images from the training set of BSD300 dataset [53]. In addition, data augmentation is used, which includes flipping, downscaling with the scales of 0.7 and 0.5, and rotating by 90°, 180° and 270°.

We use the peak signal-to-noise ratio (PSNR), the structural similarity (SSIM) index [55] and information fidelity criterion (IFC) [56] as metrics for evaluation. As commonly done in single image SR, all the criteria are calculated on the luminance channel of image after pixels near image boundary are removed.

4.2. Implementation details

Given an HR image, the input LR image is generated by first downsampling and then upsampling with bicubic interpolation to a certain scale and has the same size as the HR image. By following [10], we only train a single model for all different scales, including $2\times$, $3\times$ and $4\times$. The LR and HR pairs of all different scales are combined into one training dataset.

We split training images into 41×41 sub-images with no overlap and set the mini-batch size to 32 for stochastic gradient descent (SGD) solver. And, we set momentum parameter to 0.9 and the weight decay to 10^{-4}. The learning rate is initialized to 0.1 and then decreased by a factor of 10 for every 10 epochs. To deal with the vanishing and exploding gradients problem, we clip individual gradients to a certain range $[-\tau, \tau]$, where τ is set to 0.4. We find that our convergent procedure is fast and our model converges after about 50 epochs.

In our MSICF network, each convolutional layer has 64 filters and is followed by BN layer and LeakyReLU layer with a negative slope of 0.2. The kernel sizes of the convolutional layers are set to 3×3 except the convolutional layers in MSC modules, of which kernel sizes are determined according to the experimental analysis in Section 4.3. For weight initialization in all convolutional layers, we applied the same way as that in He et al. [57]. And, we apply PyTorch on an NVIDIA Titan X Pascal GPU for model training and testing.

4.3. Model analysis

In this section, the designs and the contributions of different components of our model are analyzed via the experiments, including MSC module, cascaded structure, residual-feature learning, cascaded-supervision and different reconstruction layers utilization. For all experiments in this section, all models are trained on 91 images [4] for faster training.

4.3.1. Multi-scale cross module

To design multi-scale cross (MSC) module for fusing complementary information as well as to demonstrate the superiority of designed MSC module, we build four modules for comparison. Two merge-and-run blocks (MR) with different filter sizes are shown in Fig. 4(a): (I) MR_L3R3, where four convolutional layers in two assembled residual branches have 64 filters of size 3×3, (II) MR_L5R5, in which each convolutional layer has 64 filters with the size of 5×5. In addition, Fig. 4(a) also depicts two MSC modules which are utilized to fuse information under different sizes of receptive fields: (III) MSC_L3R5, in which one residual branch contains two stacked convolutional layers with 64 filters of size 3×3, and another residual branch uses the same number of filters but with the size of 5×5 for the convolutional layers, (IV) MSC_L3R7, which has the similar structure with the MSC_L3R5, but stacks two convolutional layers with the kernel size of 7×7 in one residual branch. We stack these different modules to construct the corresponding plain networks with

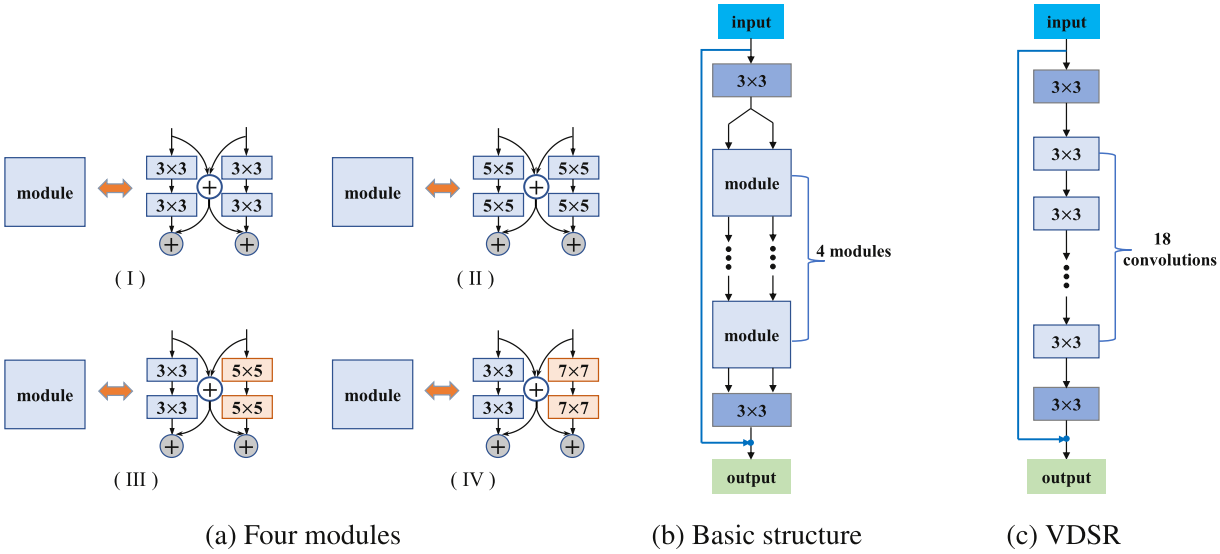


Fig. 4. Four network structures with different modules and the structure of VDSR. (a) Four built modules for comparison, including MR_L3R3, MR_L5R5, MSC_L3R5 and MSC_L3R7, which are depicted in (I), (II), (III) and (IV) respectively. (b) The built basic network structure with four modules, where all modules are replaced by one of the modules in (a) respectively to generate four corresponding networks for comparison. (c) The network structure of VDSR [10], which stacks 20 convolutional layers for image SR.

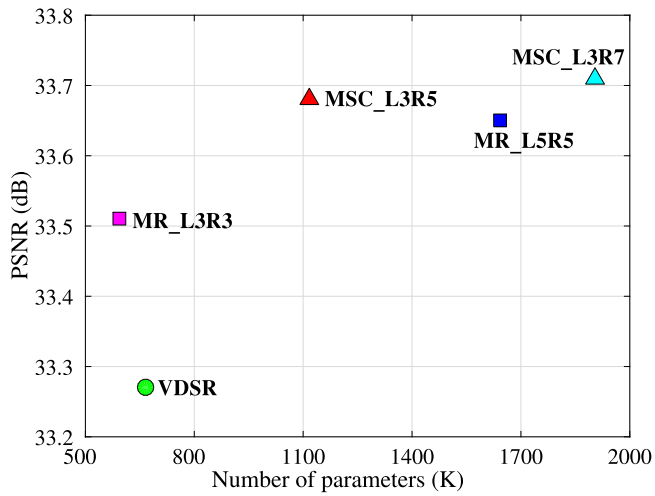


Fig. 5. PSNR performances of the networks in Fig. 4 for a scale factor of 3 × on *Set5* dataset. The circle denotes the model of VDSR, while the square and the triangle denote the models with MR blocks and with MSC modules respectively.

global residual learning (GRL) (Fig. 4(b)) for SR performance comparison. As illustrated in Fig. 4(b), the built basic network is composed of a convolutional layer, four stacked modules and another convolutional layer for reconstructing the residual image. By applying four modules (Fig. 4(a)) into the basic structure in Fig. 4(b) respectively, we obtain four networks which are named according to their containing modules (i.e., MR_L3R3, MR_L5R5, MSC_L3R5 and MSC_L3R7). In addition, we compare these four networks with VDSR [10] (illustrated in Fig. 4(c)) which has 20 stacked convolutional layers. We evaluate these networks on Set5 dataset and then illustrate the PSNRs of these models for a scale factor of 3 × in Fig. 5. One can see that, by introducing the interactions between branches, MR_L3R3 network with fewer parameters and layers achieves higher PSNR than VDSR, which manifests the effectiveness of the merge-and-run mapping for SR. On the other hand, our proposed MSC_L3R5 performs better than MR_L5R5 but contains fewer parameters, which suggests that the gains of our MSC module over MR block (MR_L3R3 and MR_L5R5) are not only from the larger receptive field but also from multi-scale complementary information fusion and richer representation. Among these models, MSC_L3R7 achieves the best performance. Considering both the performance and the number of parameters, we adopt MSC_L3R5 as our MSC module for the next experiments.

Table 1
Study on the effect of cascaded structure. Average PSNRs for a scale factor of 2 × are reported. **Font bold** indicates the best performance.

Model	Dataset		
	Set5	Set14	Urban100
PMSC	37.569	33.094	30.846
MSICF_NS	**37.609**	**33.131**	**30.899**

4.3.2. Cascaded structure

To validate the effectiveness of the cascaded structure, we use fifteen MSC modules to build a plain structure and a cascaded structure for comparison, which are shown in Fig. 6. For the plain structure, denoted as PMSC, we stack fifteen MSC modules to reconstruct HR features and also apply the residual-feature learning (RFL) as well as the global residual learning (GRL) in this structure. For cascaded structure, we utilize five MSC modules in each subnetwork and adopt three subnetworks for three stages of cascade. For fair comparison, the cascaded-supervision and the aggregation of predictions are excluded from the cascaded structure, which is denoted as MSICF_NS. Both the plain structure and the cascaded structure use two convolutional layers for feature extraction and image reconstruction respectively. Table 1 presents the SR results on three benchmark datasets, including Set5, Set14 and Urban100. From the results, it is seen that the cascaded structure (MSICF_NS) with the same number of parameters achieves moderate performance improvements over the plain structure. In addition, the cascaded structure enables the intermediate predictions to be conveniently supervised and assembled for further boosting SR performance. Therefore, the image SR can benefit from the cascaded structure.

4.3.3. Residual-feature learning, cascaded-supervision and different reconstruction layers utilization

To study contributions of residual-feature learning (RFL), cascaded-supervision and different reconstruction layers utilization to SR performance improvement, we rebuild three models for comparison besides our final model (MSICF), which are termed as MSICF_NRS, MSICF_NS and MSICF_SR respectively. For MSICF_NRS, the identity branch between the beginning and the end of each

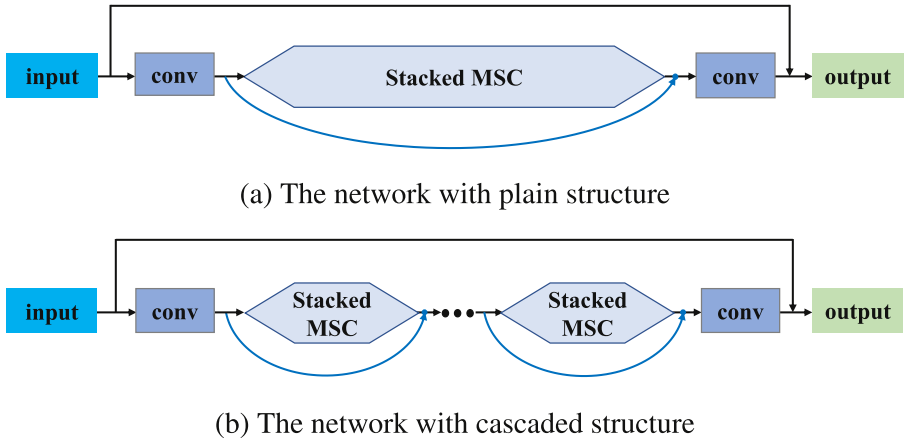
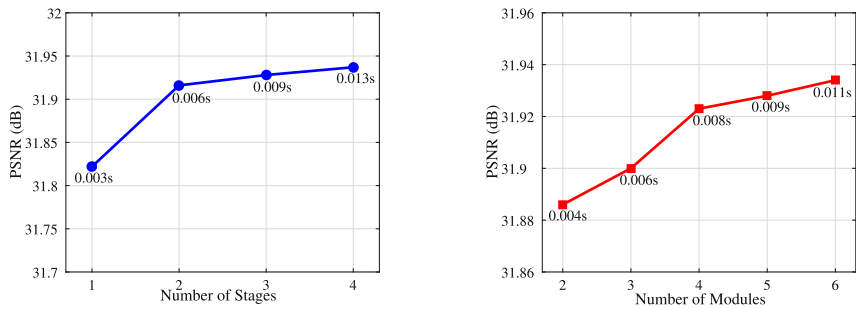


Fig. 6. The plain structure and the cascaded structure for the networks with 15 stacked MSC modules. (a) The network with plain structure (denoted as PMSC), which consists of two convolutional layers for feature representation and image reconstruction respectively and 15 MSC modules stacked in chain mode for inferring HR features. (b) The network with cascaded structure (denoted as MSICF_NS), where three subnetworks with five stacked MSC modules are cascaded for inferring HR features in a coarse-to-fine manner.



(a) (b)

Fig. 7. (a) PSNR performance versus the number of cascaded stages (S). (b) PSNR performance versus the number of modules (M) in each stage. The numbers marked on the side of the curves represent the execution time in seconds. The tests are conducted for a scale factor of 2 × on the dataset of *BSD100*.

Table 2
Study on the effects of residual-feature learning, cascaded-supervision and different reconstruction layers utilization. Average PSNRs/SSIMs on *Set5* dataset are reported. **Font bold** indicates the best performance.

Scale	MSICF_NRS	MSICF_NS	MSICF_SR	MSICF
	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM	PSNR / SSIM
2 ×	37.59 / 0.9592	37.61 / 0.9593	37.62 / 0.9594	**37.65** / **0.9596**
3 ×	33.94 / 0.9238	33.96 / 0.9239	33.98 / 0.9242	**34.03** / **0.9247**
4 ×	31.55 / 0.8870	31.58 / 0.8874	31.61 / 0.8879	**31.66** / **0.8888**

cascaded subnetwork (blue curve in Fig. 2(b)) is removed from MSICF, and the multiple supervisions are also excluded. Thus, by directly feeding the output from the last cascaded stage into the reconstruction network, the final prediction is obtained. Based on MSICF_NRS, we recover the RFL in each subnetwork to obtain MSICF_NS where the cascaded-supervision and assembling predictions are still not applied. The difference between MSICF_SR and MSICF is that all subnetworks in MSICF_SR share the same reconstruction layer to obtain intermediate predictions while the subnetworks in MSICF own their respective reconstruction layers. The four networks have the same number of cascaded subnetworks ($S = 3$) and the same number of MSC modules ($M = 5$) in each subnetwork. Table 2 shows the SR performances of four models in terms of PSNR and SSIM on Set5 dataset for three scale factors of 2 × , 3 × and 4 × . We can see that both residual-feature learning and cascaded-supervision make contributions to improving SR performance. Further, MSICF achieves better performance than MSICF_SR with a slight increase in the number of parameters, which manifests that applying different reconstruction layers for different cascaded stages can further boost SR performance.

Table 3
Quantitative evaluations of state-of-the-art SR methods. The average PSNRs/SSIMs/IFCs for scale factors of 2 × , 3 × and 4 × are reported. **Font bold** indicates the best performance and underline indicates the second-best performance.

Scale	Method	Set5			Set14			BSD100			Urban100		
		PSNR	SSIM	IFC	PSNR	SSIM	IFC	PSNR	SSIM	IFC	PSNR	SSIM	IFC
2 ×	Bicubic	33.68	0.9304	6.2821	30.24	0.8691	6.3035	29.56	0.8435	5.9201	26.88	0.8405	6.6493
	SRCNN [9]	36.66	0.9542	8.0356	32.45	0.9067	7.7844	31.36	0.8879	7.1260	29.51	0.8946	7.9958
	FSRCNN [15]	36.98	0.9556	7.8118	32.62	0.9087	7.5656	31.50	0.8904	6.8622	29.85	0.9009	8.0255
	VDSR [10]	37.53	0.9587	8.5801	33.05	0.9127	8.1594	31.90	0.8960	7.4942	30.77	0.9141	8.6288
	DRCN [11]	37.63	0.9588	8.7828	33.06	0.9121	8.3700	31.85	0.8942	7.5766	30.76	0.9133	8.9590
	LapSRN [39]	37.52	0.9591	9.0133	32.99	0.9124	8.5085	31.80	0.8949	7.7192	30.41	0.9101	8.9227
	DRRN [12]	37.74	0.9591	8.6698	33.23	0.9136	8.3178	32.05	0.8973	7.5128	31.23	0.9188	8.9200
	MemNet [13]	37.78	0.9597	8.8503	33.28	0.9142	8.4682	32.08	0.8978	7.6650	31.31	0.9195	9.1221
	IDN [20]	37.83	0.9600	9.2519	33.30	0.9148	8.8378	32.08	0.8985	7.9304	31.27	0.9196	9.5867
	MSICF (ours)	**37.89**	**0.9605**	**9.3922**	**33.41**	**0.9153**	**8.8932**	**32.15**	**0.8992**	**8.1380**	**31.47**	**0.9220**	**9.6666**
3 ×	Bicubic	30.40	0.8686	3.6160	27.54	0.7741	3.5346	27.21	0.7389	3.2339	24.46	0.7349	3.7759
	SRCNN [9]	32.75	0.9090	4.6582	29.29	0.8215	4.3377	28.41	0.7863	3.8435	26.24	0.7991	4.5790
	FSRCNN [15]	33.16	0.9140	4.9640	29.42	0.8242	4.5494	28.52	0.7893	4.0302	26.41	0.8064	4.8419
	VDSR [10]	33.66	0.9213	5.2029	29.78	0.8318	4.6906	28.83	0.7976	4.1514	27.14	0.8279	5.1594
	DRCN [11]	33.82	0.9226	5.3363	29.77	0.8314	4.7816	28.80	0.7963	4.1839	27.15	0.8277	5.3145
	LapSRN [39]	33.82	0.9227	5.1946	29.79	0.8320	4.6635	28.82	0.7973	4.0581	27.07	0.8271	5.1649
	DRRN [12]	34.03	0.9244	5.3962	29.96	0.8349	4.8773	28.95	0.8004	4.2350	27.53	0.8377	5.4496
	MemNet [13]	34.09	0.9248	5.5027	30.00	0.8350	4.9584	28.96	0.8001	4.3002	27.56	0.8376	5.5727
	IDN [20]	34.11	0.9253	5.6194	29.99	0.8354	5.0617	28.95	0.8013	4.3976	27.42	0.8359	5.6816
	MSICF (ours)	**34.24**	**0.9266**	**5.6612**	**30.09**	**0.8371**	**5.1019**	**29.01**	**0.8024**	**4.4427**	**27.69**	**0.8411**	**5.7819**
4 ×	Bicubic	28.43	0.8109	2.3425	26.00	0.7023	2.2593	25.96	0.6678	2.0210	23.14	0.6574	2.4459
	SRCNN [9]	30.48	0.8628	2.9910	27.50	0.7513	2.7506	26.90	0.7103	2.3961	24.52	0.7226	2.9632
	FSRCNN [15]	30.70	0.8657	2.9862	27.59	0.7535	2.7068	26.96	0.7128	2.3297	24.60	0.7258	2.8945
	VDSR [10]	31.35	0.8838	3.5419	28.02	0.7678	3.1065	27.29	0.7252	2.6785	25.18	0.7525	3.4623
	DRCN [11]	31.53	0.8854	3.5426	28.03	0.7673	3.0978	27.24	0.7233	2.6328	25.14	0.7511	3.4654
	LapSRN [39]	31.54	0.8866	3.5596	28.09	0.7694	3.1462	27.32	0.7264	2.6771	25.21	0.7553	3.5305
	DRRN [12]	31.68	0.8888	3.7022	28.21	0.7720	3.2518	27.38	0.7284	2.7459	25.44	0.7638	3.6741
	MemNet [13]	31.74	0.8893	3.7860	28.26	0.7723	3.3079	27.40	0.7281	2.7784	25.50	0.7630	3.7860
	IDN [20]	31.82	0.8903	3.8208	28.25	0.7730	3.3458	27.41	0.7297	2.8282	25.41	0.7632	3.7774
	MSICF (ours)	**31.91**	**0.8923**	**3.8758**	**28.35**	**0.7751**	**3.4056**	**27.46**	**0.7308**	**2.8767**	**25.64**	**0.7692**	**3.9437**

Fig. 8. Visual evaluation for a scale factor of 3 × on the "zebra" image from *Set14*. The proposed MSICF more faithfully reconstructs zebra stripes while others generate blurry results with severe distortions.

*4.3.4. The number of stages (*S*) and the number of modules (*M*)*

The capacity of MSICF model is mainly determined by two parameters: the number of subnetworks (S) and the number of MSC modules (M) in each subnetwork. In this subsection, we test the effects of the two parameters on image SR. Firstly, we fix the parameter of M to 5 and change the parameter of S from 1 to 4. Fig. 7(a) shows the curve of the PSNR performance versus the parameter S on the dataset of BSD100 for a scale factor of 2 × , and the corresponding average execution time in seconds is marked on the side of the curve. We can see that the performance is improved gradu-

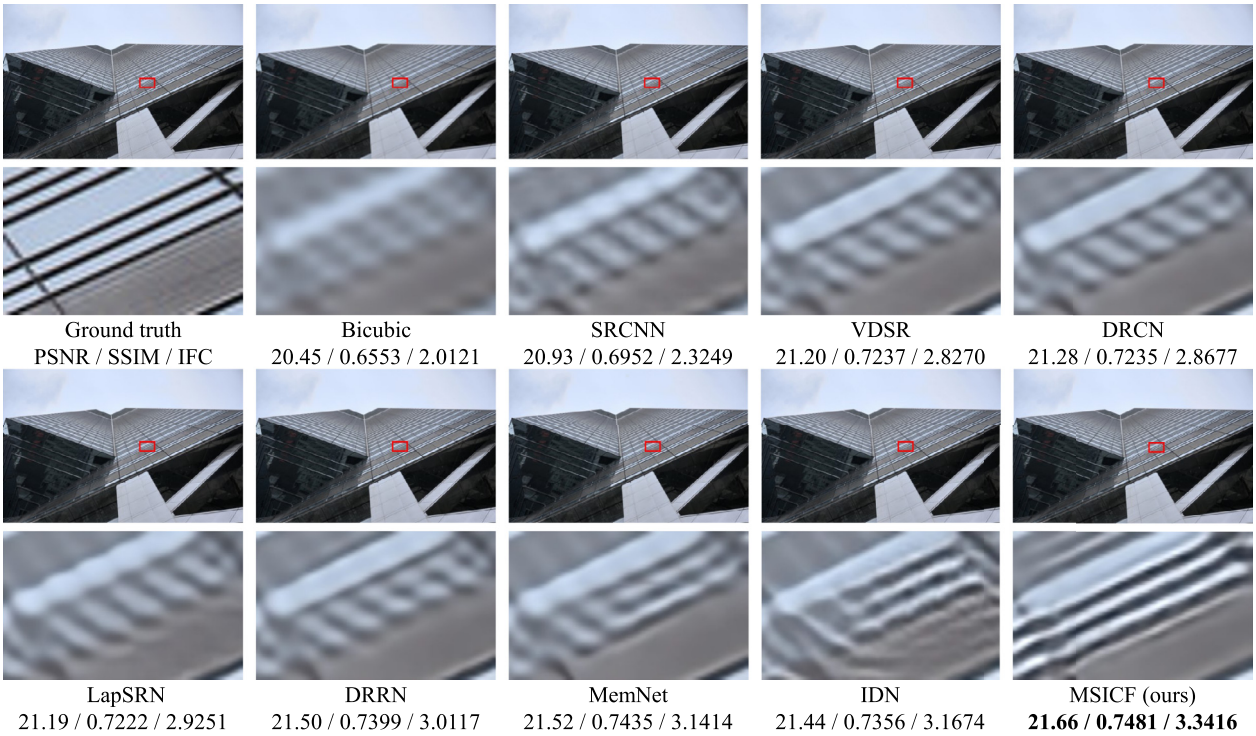


Fig. 9. Visual evaluation for a scale factor of 4 × on the "img059" image from *Urban100*. Only MSICF reconstructs the lines on the building which are visually closer to the ground truth while other methods produce blurry results with wrong directions.

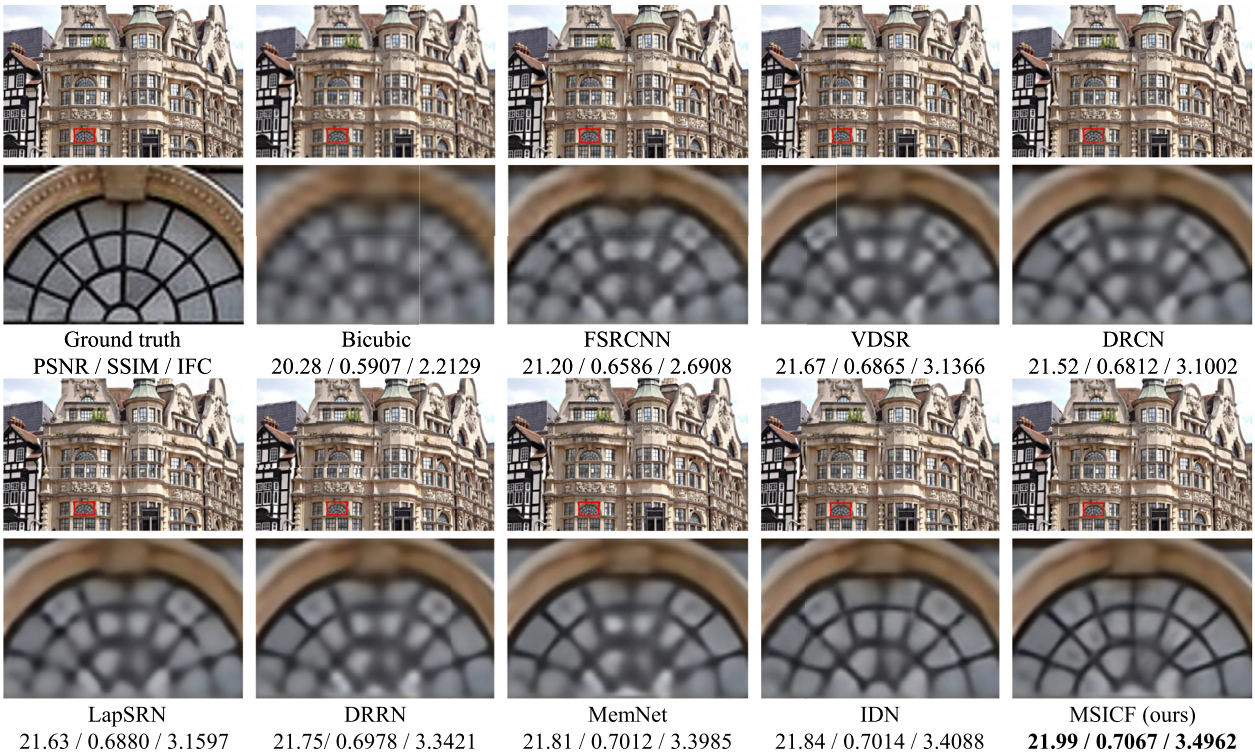


Fig. 10. Visual evaluation for a scale factor of 4 × on the "img053" image from *Urban100*. The contours of windows in the result of MSICF are sharper than those in other results.

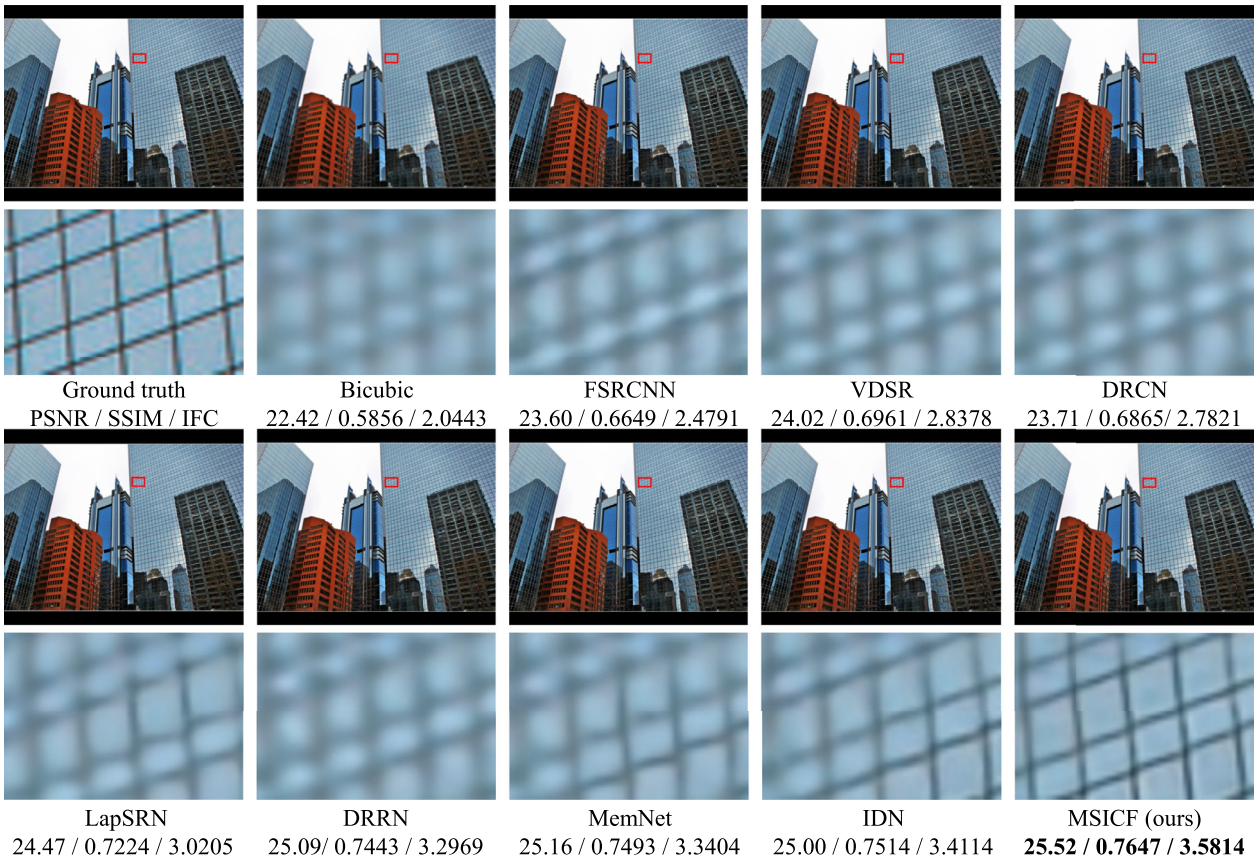


Fig. 11. Visual evaluation for a scale factor of 4 × on the "img099" image from *Urban100*. The proposed MSICF reconstructs the shaper grids which are very blurry in other results.

ally with the increase in the number of stages but at the expense of increased computational cost.

Next, the parameter S is fix to 3 and the parameter M in each stage is increased from 2 to 6. The curve of the PSNR performance versus the parameter M is illustrated in Fig. 7(b). Since the larger M results in a deeper network, the curve in Fig. 7(b) illustrates that the deeper network still achieves the better performance but with more execution time. To strike a balance between performance and speed, we choose $M = 5$ and $S = 3$ for our MSICF model, the depth of which is 35 according to Eq. (9).

4.4. Comparisons with the state-of-the-arts

To illustrate the effectiveness of the proposed MSICF model, several single image SR methods, including SRCNN [9], FSRCNN [15], VDSR [10], DRCN [11], LapSRN [39], DRRN [12], MemNet [13] and IDN [20], are compared in terms of quantitative evaluation, visual quality and execution time.

The quantitative evaluations on four benchmark datasets for three scale factors ($2\times$, $3\times$, $4\times$) are summarized in Table 3. Since the trained model for a scale factor of $3\times$ is not provided by LapSRN [39], we generate the corresponding results via downscaling its $4\times$ upscaling results as the way in Lai et al. [39]. It can be seen that our MSICF model significantly outperforms the exiting methods on all datasets for all upscaling factors, in terms of PSNR, SSIM and IFC. In particular, on the very challenging dataset Urban100, the proposed MSICF outperforms the state-of-the-art method (MemNet [13]) by the PSNR gains of 0.16 dB, 0.13 dB and 0.14 dB on scale factors of $2\times$, $3\times$ and $4\times$ respectively. In addition, objective image quality assessment values in terms of SSIM and IFC further validate the superiority of the proposed method.

The visual comparisons of different methods are shown in Figs. 8–11. Our proposed MSICF more clearly reconstructs the texture pattern, the grid pattern and the lines. It is observed that the severe distortions and the artifacts are contained in the results generated by the prior methods, such as the marked zebra stripes in Fig. 8 and the lines on the building in Fig. 9. In contrast, our method suppresses the distortions and artifacts via the multi-scale information cross-fusion, the residual-feature learning and the cascaded feature-reconstruction. In addition, in Figs. 10 and 11, only our method is able to reconstruct finer edges and clearer grids while other methods generate very blurry results. The above visual comparisons also demonstrate the superiority of our MSICF network especially on super-resolving the images with fine structures.

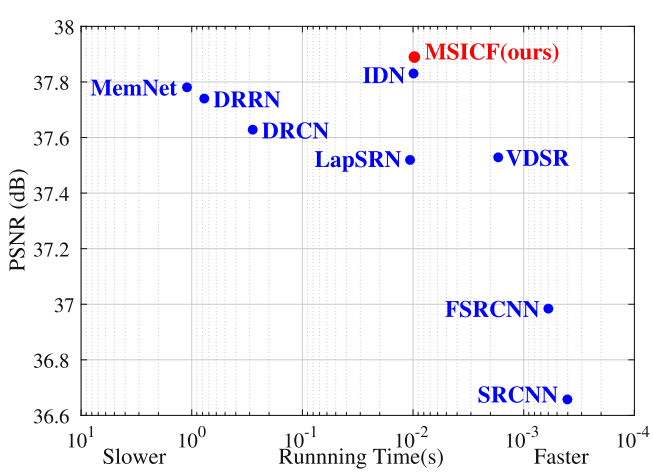


Fig. 12. PSNR performance versus runtime (evaluated in seconds). The results are evaluated on *Set5* dataset for a scale factor of $2\times$. The proposed MSICF achieves the best performance with relatively less execution time.

We also adopt the public source codes of compared methods to measure the execution time. Since the testing codes of SRCNN [9] and FSRCNN [15] are implemented on the CPU, we then rebuild both models as well as VDSR [10] and DRCN [11] in PyTorch with the same network parameters for evaluating the runtime on GPU. Fig. 12 shows the PSNR performance versus execution time in the testing phase on Set5 dataset for a scale factor of $2\times$. We can see that our proposed MSICF outperforms all mentioned methods with relatively less execution time.

5. Conclusion

In this paper, we propose a multi-scale information cross-fusion network (MSICF) for modeling the super-resolution reconstruction process, where a sequence of subnetworks is cascaded to gradually refine high resolution features with cascaded-supervision. In each cascaded subnetwork, multiple multi-scale cross (MSC) modules are stacked not only to fuse complementary information under different receptive fields but also to improve information flow across the layers. Besides, to make full use of relative information between multi-level features, residual-feature learning is introduced to the cascaded subnetworks for further boosting reconstruction performance. Comprehensive evaluations on benchmark datasets demonstrate that our MSICF network outperforms state-of-the-art super-resolution methods in terms of quantitative and qualitative evaluations with relatively less execution time.

Since the subnetworks in MSICF at all stages have the same structure and the same aim, it is possible for MSICF model to share the network parameters across the cascaded stages. In future work, we will explore a suitable strategy to share the parameters across as well as within the cascaded stages, and thus to control the number of model parameters without a decrease in performance. On the other hand, we will extend our MSICF model to other image restoration as well as heterogeneous image transformation and recognition fields [58,59].

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.

CRediT authorship contribution statement

Yanting Hu: Conceptualization, Methodology, Software. **Xinbo Gao:** Supervision, Writing - review & editing. **Jie Li:** Validation. **Yuanfei Huang:** Validation, Writing - review & editing. **Hanzi Wang:** Supervision, Writing - review & editing.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62036007, 61772402, 62061047, 61671339 and U1605252, in part by the National Key Research and Development Program of China under Grants 2018AAA0102702, 2018AAA0103202 and 2016QY01W0200.

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] H. Chang, D.-Y. Yeung, Y. Xiong, Super-resolution through neighbor embedding, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2004, pp. 275–282.
[3] X. Chen, C. Qi, Nonlinear neighbor embedding for single image super-resolution via kernel mapping, Signal Process. 94 (2014) 6–22.
[4] J. Yang, J. Wright, T.S. Huang, Y. Ma, Image super-resolution via sparse representation, IEEE Trans. Image Process. 19 (11) (2010) 2861–2873.

[5] R. Timofte, V.D. Smet, L.V. Gool, A+: adjusted anchored neighborhood regression for fast super-resolution, in: Proc. 12th Asian Conf. Comput. Vis. (ACCV), 2014, pp. 111–126.
[6] H. Wang, X. Gao, K. Zhang, J. Li, Fast single image super-resolution using sparse Gaussian process regression, Signal Process. 134 (2017) 52–62.
[7] S. Schulter, C. Leistner, H. Bischof, Fast and accurate image upscaling with super-resolution forests, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 3791–3799.
[8] W. Yang, X. Zhang, Y. Tian, W. Wang, J. Xue, Q. Liao, Deep learning for single image super-resolution: a brief review, IEEE Trans. Multimed. 21 (12) (2019) 3106–3121.
[9] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 295–307.
[10] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 1646–1654.
[11] J. Kim, J.K. Lee, K.M. Lee, Deeply-recursive convolutional network for image super-resolution, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 1637–1645.
[12] Y. Tai, J. Yang, X. Liu, Image super-resolution via deep recursive residual network, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 3147–3155.
[13] Y. Tai, J. Yang, X. Liu, C. Xu, MemNet: a persistent memory network for image restoration, in: Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2017, pp. 4549–4557.
[14] T. Plötz, S. Roth, Neural nearest neighbors networks, in: Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2018, pp. 1–12.
[15] C. Dong, C.C. Loy, X. Tang, Accelerating the super-resolution convolutional neural network, in: Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 391–407.
[16] C. Ledig, L. Theis, F. Huszar, J. Caballero, A.P. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, Photo-realistic single image super-resolution using a generative adversarial network, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 105–114.
[17] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 630–645.
[18] B. Lim, S. Son, H. Kim, S. Nah, K.M. Lee, Enhanced deep residual networks for single image super-resolution, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), 2017, pp. 136–144.
[19] R. Timofte, E. Agustsson, L.V. Gool, M.-H. Yang, L. Zhang, B. Lim, et al., NTIRE 2017 challenge on single image super-resolution: methods and results, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), 2017, pp. 1110–1121.
[20] Z. Hui, X. Wang, X. Gao, Fast and accurate single image super-resolution via information distillation network, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 723–731.
[21] Z. Hui, X. Gao, Y. Yang, X. Wang, Lightweight image super-resolution with information multi-distillation network, in: Proc. ACM Int. Conf. Multimedia, 2019, pp. 2024–2032.
[22] X. He, Z. Mo, P. Wang, Y. Liu, M. Yang, J. Cheng, ODE-inspired network design for single image super-resolution, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 1732–1741.
[23] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, Image super-resolution using very deep residual channel attention networks, in: Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 1–16.
[24] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, Residual non-local attention networks for image restoration, in: Int. Conf. Learn. Rep. (ICLR), 2019, pp. 1–13.
[25] W. Yang, W. Wang, X. Zhang, S. Sun, Q. Liao, Lightweight feature fusion network for single image super-resolution, IEEE Signal Process. Lett. 26 (4) (2019) 538–542.
[26] K. Chang, M. Li, P.L.K. Ding, B. Li, Accurate single image super-resolution using multi-path wide-activated residual network, Signal Process. 172 (2020), doi:10.1016/j.sigpro.2020.107567.
[27] G. Huang, Z. Liu, K.Q. Weinberger, L. van der Maaten, Densely connected convolutional networks, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 4700–4708.
[28] T. Tong, G. Li, X. Liu, Q. Gao, Image super-resolution using dense skip connections, in: Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2017, pp. 4799–4807.
[29] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image super-resolution, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 2472–2481.
[30] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, Q. Liao, LCSCNet: linear compressing-based skip-connecting network for image super-resolution, IEEE Trans. Image Process. 29 (2020) 1450–1463.
[31] M. Haris, G. Shakhnarovich, N. Ukita, Deep back-projection networks for super-resolution, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 1664–1673.
[32] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, W. Wu, Feedback network for image super-resolution, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 3862–3871.
[33] Z. Tang, L. Luo, H. Peng, S. Li, A joint residual network with paired ReLUs activation for image super-resolution, Neurocomputing 273 (2018) 37–46.
[34] J. Yamanaka, S. Kuwashima, T. Kurita, Fast and accurate image super resolution by deep CNN with skip connection and network in network, in: Int. Conf. Neural Inf. Process. (ICONIP), 2017, pp. 217–225.
[35] H. Ren, M. El-Khamy, J. Lee, Image super resolution based on fusing multiple convolution neural networks, in: IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), 2017, pp. 54–61.
[36] W. Wang, R. Guo, Y. Tian, W. Yang, CFSNet: toward a controllable feature space for image restoration, in: Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2019, pp. 4139–4148.
[37] Z. Wang, D. Liu, J. Yang, W. Han, T.S. Huang, Deep networks for image super-resolution with sparse prior, in: Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2015, pp. 370–378.
[38] Z. Cui, H. Chang, S. Shan, B. Zhong, X. Chen, Deep network cascade for image super-resolution, in: Proc. Eur. Conf. Comput. Vis. (ECCV), 2014, pp. 49–64.
[39] W.-S. Lai, J.-B. Huang, N. Ahuja, M.-H. Yang, Deep Laplacian pyramid networks for fast and accurate super-resolution, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 624–632.
[40] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, C. Schroers, A fully progressive approach to single-image super-resolution, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), 2018, pp. 977–986.
[41] S. Zagoruyko, N. Komodakis, DiracNets: training very deep neural networks without skip-connections, arXiv:1706.00388 (2017).
[42] L. Zhao, J. Wang, X. Li, Z. Tu, W. Zen, Deep convolutional neural networks with merge-and-run mappings, arXiv:1611.07718 (2017).
[43] R.K. Srivastava, K. Greff, J. Schmidhuber, Training very deep networks, in: Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2015, pp. 2377–2385.
[44] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.
[45] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 1–9.
[46] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 2818–2826.
[47] Y. Hu, N. Wang, D. Tao, X. Gao, X. Li, SERF: a simple, effective, robust, and fast image super-resolver from cascaded linear regression, IEEE Trans. Image Process. 25 (9) (2016) 4091–4102.
[48] R. Timofte, R. Rothe, L.V. Gool, Seven ways to improve example-based single image super resolution, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 1865–1873.
[49] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Int. Conf. Mach. Learn. (ICML), 2015, pp. 448–456.
[50] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Int. Conf. Mach. Learn. (ICML), 2013.
[51] M. Bevilacqua, A. Roumy, C. Guillemot, M.-L.A. Morel, Low-complexity single-image super-resolution based on nonnegative neighbor embedding, in: Proc. 23rd British Mach. Vis. Conf. (BMVC), 2012, pp. 135.1–135.10.
[52] R. Zeyde, M. Elad, M. Protter, On single image scale-up using sparse-representations, in: Proc. 7th Int. Conf. Curves Surfaces, 2010, pp. 711–730.
[53] P. Arbeláez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 898–916.
[54] J.-B. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 5197–5206.
[55] 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.
[56] H.R. Sheikh, A.C. Bovik, G. de Veciana, An information fidelity criterion for image quality assessment using natural scene statistics, IEEE Trans. Image Process. 14 (12) (2005) 2117–2128.
[57] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, in: in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2015, pp. 1026–1034.
[58] M. Zhang, J. Li, N. Wang, X. Gao, Compositional model-based sketch generator in facial entertainment, IEEE Trans. Cybern. 48 (3) (2018) 904–915.
[59] C. Peng, X. Gao, N. Wang, J. Li, Face recognition from multiple stylistic sketches: scenarios, datasets, and evaluation, Pattern Recognit. 84 (2018) 262–272.