Coarse-to-Fine CNN for Image Super-resolution

Chunwei Tian, Yong Xu*, *Senior Member, IEEE,* Wangmeng Zuo, *Senior Member, IEEE,* Bob Zhang, *Senior Member, IEEE,* Lunke Fei, *Member, IEEE,* and Chia-Wen Lin, *Fellow, IEEE*

***Abstract*—Deep convolutional neural networks (CNNs) have been popularly adopted in image super-resolution (SR). However, deep CNNs for SR often suffer from the instability of training, resulting in poor image SR performance. Gathering complementary contextual information can effectively overcome the problem. Along this line, we propose a coarse-to-fine SR CNN (CFSRCNN) to recover a high-resolution (HR) image from its low-resolution version. The proposed CFSRCNN consists of a stack of feature extraction blocks (FEBs), an enhancement block (EB), a construction block (CB) and, a feature refinement block (FRB) to learn a robust SR model. Specifically, the stack of FEBs learns the long- and short-path features, and then fuses the learned features by expending the effect of the shallower layers to the deeper layers to improve the representing power of learned features. A compression unit is then used in each FEB to distill important information of features so as to reduce the number of parameters. Subsequently, the EB utilizes residual learning to integrate the extracted features to prevent from losing edge information due to repeated distillation operations. After that, the CB applies the global and local LR features to obtain coarse features, followed by the FRB to refine the features to reconstruct a high-resolution image. Extensive experiments demonstrate the high efficiency and good performance of our CFSRCNN model on benchmark datasets compared with state-of-the-art SR models. The code of CFSRCNN is accessible on https://github.com/hellloxiaotian/CFSRCNN.**

***Index Terms*—Image super-resolution, convolutional neural network, cascaded structure, feature fusion, feature refinement**

I. Introduction

SINGLE image super-resolution (SISR) aims to reconstruct a high-resolution (HR) image from a given low-resolution (LR) image, which has been widely applied in many fields, such as visual analysis [1], medical images [2], and person identification [3, 4]. Since SISR is an ill-posed inverse problem, prior information is important to guarantee the quality of reconstructed SR image. For example, a set of patterns through Bayesian knowledge was learned for SISR in [5]. The method proposed in [6] utilizes sparse representation to predict the SR counterparts of LR patches. The random forest-based method proposed in [7] can directly map SR patches from LR patches to overcome the difficulty of training. In addition, non-local self-similarity (NLSS) [8], regression [9], dictionary learning [10], and gradient methods [11] were shown to be effective for SISR.

Although these SISR methods can achieve impressive performances, most of them still suffer from two drawbacks. First, they usually rely on complex optimization methods to recover HR images, which are time-consuming. Second, they need manually-tuned parameters to obtain a good performance of SISR, making them inflexible.

Deep learning techniques have found wide applications in low-level vision, such as image denoising [12], rain removal [13, 14], deblurring [15] and image SR [16]. Taking image SR as an example, the super-resolution CNN (SRCNN) proposed in [17] utilizes three convolutional layers to predict the HR image in a pixel-mapping manner, which, however, leads to slow convergence and large training cost. To break the bottleneck of the SRCNN, a very deep SR (VDSR) network [18] uses residual learning and small filter sizes to accelerate the speed of training while achieving good visual quality. Moreover, reducing the number of parameters is effective to overcome the difficulty in training a SR model. For example, the deeply-recursive convolutional network (DRCN) [19] and the deep recursive residual network (DRRN) [20] utilize recursive learning and residual learning techniques to improve training efficiency. Besides, using skip connections to fuse global and local features, such as the 30-layer convolutional residual encoder-decoder network (RED30) [21], is shown effective in enhancing the expressive ability of the SISR model. Moreover, the very deep persistent memory network (MemNet) in [22] applies recursive and gate units to mine useful features for some low-level tasks. However, very deep networks are not easy to train. Additionally, some of these methods perform bicubic interpolation to upscale a LR image to the same size as the HR image on SISR, which resulted in low efficiency for training [23]. Some existing methods only extract LR features for the SR task and magnify the obtained LR features in the final layer, which ignores the effect of HR features on SISR and can lead to the instability of training.

In this paper, we propose a coarse-to-fine super-resolution CNN (CFSRCNN) for SISR. It consists of a stack of feature extraction blocks (FEBs), an enhancement block (EB), a construction block (CB), and a feature refinement block

This work was supported in part by the National Nature Science Foundation of China Gant No. 61876051 and in part by the Shenzhen Key Laboratory of Visual Object Detection and Recognition under Grant No. ZDSYS20190902093015527. (Corresponding author: Yong Xu (Email: yongxu@ymail.com).)

Chunwei Tian and Yong Xu are with the Bio-Computing Research Center, Harbin Institute of Technology, Shenzhen, Shenzhen, 518055, Guangdong, China. They are also with the Shenzhen Key Laboratory of Visual Object Detection and Recognition, Shenzhen, 518055, Guangdong, China. And Yong Xu is also with the Peng Cheng Laboratory, Shenzhen, 518055, Guangdong, China. (Email: chunweitian@163.com; yongxu@ymail.com.)

Wangmeng Zuo is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, Heilongjiang, China. And he is also with the Peng Cheng Laboratory, Shenzhen, 518055, Guangdong, China. (Email: wmzuo@hit.edu.cn).

Bob Zhang is with the Department of Computer and Information Science, University of Macau, Macau, 999078, China (e-mail: bobzhang@umac.mo).

Lunke Fei is with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, 510006, Guangdong, China (e-mail: flksxm@126.com)

Chia-Wen Lin is with the Department of Electrical Engineering and the Institute of Communications Engineering, National Tsing Hua University, Hsinchu, Taiwan (Email: cwlin@ee.nthu.edu.tw)

(FRB) to train a robust SR model. The combination of the stacked FEBs, EB and CB can more effectively make use of hierarchical LR features extracted from a LR image with much fewer parameters to enhance LR features and infer better initial HR features. Specifically, the stack of FEBs learns the long- and short-path features, and then fuses these features by expending the effect of the shallower layers to the deeper layers to improve the representing power of the learned features. A compression unit is then used in each FEB to distill important information of features so as to reduce the number of parameters. Subsequently, the EB utilizes residual learning to integrate the learned features to prevent from losing edge information due to repeated distillation operations. After that, the CB applies global and local LR features to obtain initial HR features, followed by the FRB to refine the HR features to reconstruct the final SR image. The proposed CFSRCNN has the following contributions.

(1) We propose a cascaded network that combines LR and HR features to prevent possible training instability and performance degradation caused by upsampling operations.

(2) We propose a novel feature fusion scheme based on heterogeneous convolutions to well resolve the long-term dependency problem and prevent information loss so as to significantly improve the efficiency of SISR without sacrificing the visual quality of reconstructed SR images.

(3) The proposed network achieves both good performance and high computational efficiency for SISR.

The remainder of this paper is organized as follows. Section II provides related work. Section III presents the proposed method. Section IV shows extensive experimental results. Section V reports conclusion.

II. Related Work

A. Deep CNNs based on cascaded structures for SISR

In image SR, using LR information, especially with a large upscaling factor to recover a HR image is very challenging. To address this problem, deep CNNs based on cascaded structures have been proposed to minimize the error between prediction results and their ground-truths. They can be divided into two categories in general. The first category applies bicubic interpolation to upscale a given LR image to the same size as the HR image, and then uses the upscaled image to predict a SR image. Specifically, a simple, effective, robust, and fast (SERF) method [24] cascades several linear least squares functions to extract effective features and then compresses the model in a coarse-to-fine manner. The second category gradually uses upscaling in different stages to predict the SR image. Specifically, the deep network cascade (DNC) in [25] magnifies a LR image layer by layer and utilizes NLSS in each sub-network to extract HR texture features. The cascaded multi-scale cross (CMSC) network in [26] cascades different sub-networks to obtain SR features. Then, it uses a reconstruction stage to fuse the obtained features in a weighted way to reconstruct a SR image. To reduce computation, the cascading residual network (CARN) in [27] cascades residual networks with small filter sizes to train a fast, accurate, and lightweight model. All the methods demonstrated the effectiveness of cascading operations in mitigating the discrepancy between a predicted SR image and its ground-truth.

B. Deep CNNs based on modules for SISR

Due to their flexible end-to-end architectures, CNNs have been widely adopted in many fields, i.e., image processing [29, 30], video surveillance [31] and speech processing [32], and text recognition [33]. To facilitate more features, CNNs based on modules are developed for image SR. Specifically, these methods can be divided into two categories in general: high accuracy (also referred to as performance) and efficiency.

For improving the accuracy of SR, fusing multiple features has been found useful in enhancing the expressive ability of a SR model. For example, the multi-scale dense network (MSDN) in [34] employs a multi-scale dense block to fuse intermediate features of different layers for SISR. Further, different views of an image are used as the inputs of a SR network to improve accuracy. The residual channel attention network (RCAN) in [35] utilizes residual channel attention blocks to mine and integrate different features from the channels of a LR image. The deep networks in [36, 37] fuse color and depth information of a given LR image to enhance the expressive ability of learned features. Besides, to make better use of hierarchical features, the residual dense network (RDN) in [38] combines local and global features via residual dense blocks to recover HR image details. Using residual learning techniques to learn hierarchical features is shown beneficial to build a depth map for SISR [39]. To obtain more detailed information, the SR CliqueNet (SRCliqueNet) in [40] utilizes a clique block and an up-sampling module to extract textural details useful for SISR. The recursively dilated residual network in [16] increases the impact of local spatial information by using a recursion module, followed by a refinement module to learn more accurate LR features for recovering HR image details.

For improving efficiency, reducing the number of parameters is a common way. The information distillation network (IDN) in [41] applies a feature extraction block, a stacked information distillation block, and a reconstruction block to recover HR details. The stacked information distillation block exploits part convolutional filters of size 1×1 to compress the model for improving the speed of training. To reduce complexity and computational cost, the block state-based recursive network (BSRN) in [42], comprising an initial feature extractor, a recursive residual block and an upscaling part, increases the resolution of input images at the last stage to reduce the model complexity. That was also extended to a deeper or wider network, the convolutional anchored regression network (CARN) [43] employs regression blocks to convert a LR input image to an other domain, according to the regression and similarity so as to achieve a better trade-off between speed and accuracy in contrast to other SISR methods. The Laplacian pyramid distillation network (LapIDN) in [44] first applies a Laplacian pyramid module to upscale extracted features gradually, followed by a distillation network to achieve high SR performance while compressing the network complexity. Moreover, there are other effective SISR methods, such as the lightweight feature fusion network (LFFN) in [45] that aggregates different hierarchical features in an adaptively convex weighed manner to control the number of parameters through a spindle block and a softmax feature fusion module, and the adaptive weighted SR network (AWSRN) in [46] that utilizes adaptive weighted residual unit and local residual fusion units, and an adaptive weighted multi-scale module to reduce parameters, according to the contribution of obtained features at different scales.

To achieve both high performance and efficiency in SISR, we propose a novel cascaded structure consisting of modular CNN blocks to learn accurate features.

III. Proposed Method

As shown in Fig. 1, our proposed CFSRCNN is composed of a stack of Feature Extraction Blocks (FEBs), an Enhancement Block (EB), a Construction Block (CB) and a Feature Refinement Block (FRB). The combination of the stacked FEBs, EB and CB can make use of hierarchical LR features extracted from the LR image with fewer parameters to enhance obtained LR features and derive coarse SR features. Specifically, combining an FEU and a CU into an FEB obtains long- and short-path features. Also, fusing the obtained features via the two closest FEUs can enlarge the effects of shallow layers on deep layers to improve the representing power of the SR model. The CU can distill more useful information and reduce the number of parameters. The EB fuses the features of all FEUs to offer complementary features for the stacked FEBs and prevent from the loss of edge information caused by the repeated distillation operations. Gathering several extra stacked FEUs into the EB removes over-enhanced pixel points from the previous stage of the EB. After that, the CB utilizes the global and local LR features to obtain coarse SR features. Finally, the FRB utilizes HR features to more effectively learn HR features and reconstruct a HR image. We introduce these techniques in the later sections.

A. Network architecture

The proposed 46-layer CFSRCNN is composed of four parts, a stacked FEBs, an EB, a CB and an FRB. Let I_{LR} and I_{SR} denote the input LR image and its corresponding SR output image of CFSRCNN, respectively. We divide the four blocks into two kinds, according to the obtained feature types (i.e., LR and HR features): the combination of the stack of FEBs, EB and CB, and FRB. For the combination of the three functional blocks, a 40-layer network is used to extract LR features from a given LR image and derive the coarse SR features, as formulated below:

$$O_{CB} = f_{CB}(f_{EB}(f_{sFEBs}(I_{LR}))), \tag{1}$$

where f_{sFEBs}, f_{EB} and f_{CB} denote the functions of the stack of FEBs, EB and CB, respectively, O_{CB} is the output of the the combination of the stack of FEBs, EB and CB. Specifically, O_{CB} is used as the input of a 6-layer FRB, which utilizes HR features to reduce the discrepancy between the predicted SR and target HR images, as formulated below:

$$\begin{aligned} O_{FRB} &= f_{FRB}(O_{CB}) \\ &= f_{CFSRCNN}(I_{LR}), \end{aligned} \tag{2}$$

where f_{FRB} and O_{FRB} denote the function and output of FRB, respectively. $f_{CFSRCNN}$ is the function of the CFSRCNN. Also, $O_{FRB} = I_{SR}$. Finally, CFSRCNN is optimized by the loss function that will be explained in Section III.B.

B. Loss function

We use a set of training pairs $\{I^j_{LR}, I^j_{HR}\}^N_{j=1}$ to train the model, where N is the size of training set, and I^j_{LR} and I^j_{HR} denote the j-th LR and HR training images, respectively. We choose mean square error (MSE) [47] as the loss function to minimize the difference between the predicted SR and target HR images as follows:

$$l(\theta) = \frac{1}{2N}\sum_{j=1}^{N}\left\| f_{CFSRCNN}(I^j_{LR}) - I^j_{HR}\right\|^2_2, \tag{3}$$

where θ denotes the parameter set of the trained model.

C. The combination of stacked FEBs, EB, and CB

The 40-layer block consists of 33-layer stacked FEBs, 6-layer EB and 1-layer CB. Specifically, FEBs aim to improve the efficiency and performance of SR by combining a FEU with a CU, where CU is used to distill more useful information. EB offers complementary features by fusing hierarchical LR features to address the loss of edge information from the CUs. Besides, EB also utilizes several additional stacked FEUs to learn finer LR features to mitigate over-enhanced pixel points caused by the previous stage of EB. Subsequently, CB can apply the global and local LR features to obtain coarse SR features. The detailed information is shown in latter subsections.

1) Stacked FEBs: As illustrated in Fig. 2, the 33-layer stacked FEBs fuses obtained features from the pair of FEU and CU to enhance the performance and efficiency of the SR model. FEB of the stacked FEBs is composed of a pair of FEU and CU that respectively perform 3×3 and 1×1 heterogeneous convolutions, except for the last FEB that only has a FEU without CU. Each FEB (except the last one) first concatenates the output of the FEU from the previous FEB (aka the long-path features) and that of its own FEU (aka the short-path features) as the input of its own CU to enhance the representing power for SR. Subsequently, CU distills useful information from the enhanced LR features above, which can reduce the number of parameters and improve the training efficiency for a SR model. In practice, each FEU and CU comprise Conv+ReLU: a convolution filter followed by a rectified linear unit [48]. As mentioned above, the filter sizes of the FEU and CU are 3×3 and 1×1, respectively. The ReLU is used to non-linearly transform the fused features, which is then used as the input of the following CU. Further, the sizes of the first FEU and CU are $3 \times 3 \times 3 \times 64$ and $64 \times 1 \times 1 \times 64$, respectively, where 3 and 64 are the channel numbers of the input and output of the first FEU, respectively, and 64 is the channel number of the input and output of the first CU, respectively. The sizes of FEU and CU of the other FEBs are $64{\times}3{\times}3{\times}64$ and $128{\times}1{\times}1{\times}64$, respectively, where 64 represents the channel numbers of input and output except

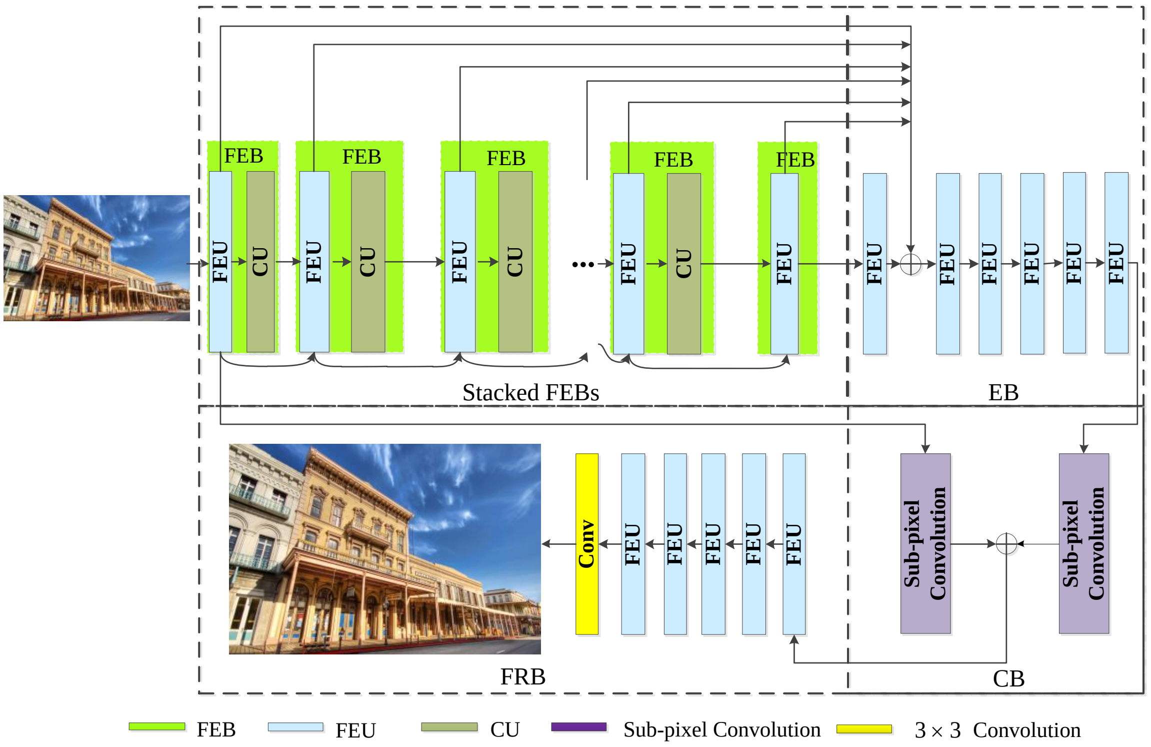


Fig. 1. Network architecture of CFSRCNN.

for the first FEU, and 128 and 64 are the channel numbers of input and output except the first CU, respectively. Due to the 1×1 convolution, CUs can better distill useful information as well as reduce the number of parameters [41], as will be shown in Section IV.D.

We further clarify the formulation of the stacked FEBs in details. Let O^i_{FEU} and O^i_{CU} denote the output of the FEU and CU of the i-th FEB, respectively. According to the previous descriptions, the function of the i-th FEB can be expressed as

$$O^i_{CU} = CU(Cat(FEU(O^{i-1}_{CU}), O^{i-1}_{FEU})), \tag{4}$$

where $i = 2, 3, , ..., 16$ and Cat represents a concatenation operation. Specifically, the first FEB can be represented as $O^1_{CU} = CU(FEU(I_{LR}))$. Because the last FEB (the 17th FEB) only has a FEU, it can be formulated as

$$O^{17}_{CU} = Cat(FEU(O^{16}_{CU}), O^{16}_{FEU}), \tag{5}$$

where $O_{sFEBs} = O^{17}_{CU}$ and O_{sFEBs} is the output of the stacked FEBs.

2) Enhancement block: It is known that as the depth of the network increases, the extracted features are more accurate. However, this will also lead to the information loss of shallow layers [49]. Also, although the CUs can reduce the number of parameters and improve the efficiency for a SR model, repeated distillation operations lead to the loss of edge information extracted from the shallow layers. Taking into account these two factors, we propose a two-step feature enhancement block (EB) as demonstrated in Fig. 2. The first step of the EB, involving one Conv+ReLU with a filter size of $128 \times 3 \times 3 \times 64$, gathers hierarchical features extracted by the FEUs of all FEBs through residual learning to offer features complementary to the stacked FEBs. The second step, involving five Conv+ReLU with the same filter size of $64 \times 3 \times 3 \times 64$, fine-tunes the LR features to mitigate possible over-enhancement caused by the first step. The first-step operation of EB can be formulated as follows:

$$O^1_{EB} = \sum_{i=1}^{17} (FEU(O_{sFEBs}) + O^i_{FEU}), \tag{6}$$

where O^1_{EB} denotes the first-step output of EB. The EB's second-step can be expressed as:

$$O^2_{EB} = FEU(FEU(FEU(FEU(FEU(O^1_{EB}))))), \tag{7}$$

where O^2_{EB} represents the final output of EB.

3) Construction block: It is known that a given LR image uses bicubic interpolation to obtain the same size as the HR image as the input of a SR network for training the model, which can result in high computational cost and memory consumption [23]. To address this problem, the up-sampling technique was proposed [50]. For example, a fast SR convolutional neural network (FSRCNN) [23] utilized deconvolution as the last layer to upscale the extracted LR features for SISR. FSRCNN directly utilized a LR image as input to extract LR features, then applied the deconvolution technique in the last layer to reconstruct the HR image. Although this approach can improve the efficiency of training, its performance in SISR is not satisfactory. In this paper, we use two steps in the construction block (CB) to overcome this problem. CB first performs a sub-pixel convolution with a filter size of $64 \times 3 \times 3 \times 64$ to obtain global- and local-features. Subsequently, CB fuses the obtained features by residual learning to derive coarse HR features. The operation of CB can be formulated as follows:

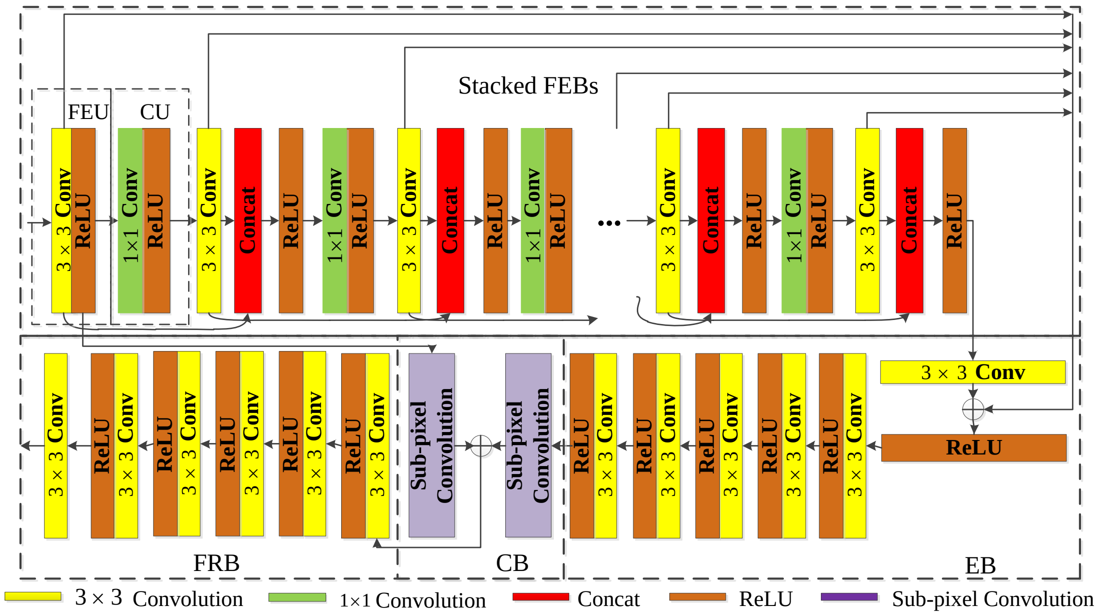


Fig. 2. Architecture of the CFSRCNN.

$$O_{CB} = S(O^1_{FEU}) + S(O^2_{EB}), \tag{8}$$

where S and $+$ stand for the functions of the sub-pixel convolution and residual learning, respectively. In addition, the residual learning operation is denoted as $\bigoplus$ in Figs. 1 and 2.

D. Feature refinement block

Feature refinement block (FRB) is used to reduce the discrepancy between the predicted SR and ground-truth HR images. It involves five cascaded FEUs followed by a convolutional filter, where each FEU consists of Conv+ReLU with the same filter size of $64 \times 3 \times 3 \times 64$, where 64 is the channel numbers of the input and output. This operation of cascaded FEUs is shown in Eq. (9).

$$O_{FRB1} = FEU(FEU(FEU(FEU(FEU(O_{CB}))))), \tag{9}$$

where O_{FRB1} is the output of five cascaded FEUs of FRB, which is then filtered by the final convolution with a filter size of $64 \times 3 \times 3 \times 3$ as follows:

$$I_{SR} = C(O_{FRB1}), \tag{10}$$

where C denotes the final convolution function.

IV. Experimental Results

A. Training dataset

Following the state-of-the-art SR methods in [27, 51, 52], we utilize the public DIV2K dataset [53] to train our model. The DIV2K dataset contains 800 training images, 100 validation images, and 100 test images at three different scales: ×2, ×3, and ×4. We merge the training and validation datasets of the DIV2K to expand the training dataset. Further, to improve the efficiency of model training, we divide each LR image into patches of size 77×77. Besides, we use random horizontal flipping and 90° rotation operations [27] to augment the training patches.

B. Testing datasets

We utilize five benchmark datasets, including Set5 [54], Set14 [6], BSD100 [55], Urban100 [56] and 720p, which are constructed at three different scales (×2, ×3, and ×4), to evaluate the performance of the trained SR model. Set5 and Set14 respectively contain 5 and 14 images from different scenes, and BSD100 (a.k.a. B100) and Urban100 (a.k.a. U100) both consist of 100 images. The 720p dataset is composed of three typical images from clean images in the PolyU dataset [57], which are cropped to 1280×720.

Note, existing methods, such as DnCNN [58] and RED30 utilize YCbCr channel (also named Y channel) to conduct experiments. Thus, we covert the RGB image predicted by CFSRCNN into Y channel to evaluate the performance for SISR.

C. Implementation details

In the training process, the initial parameters are batch size of 64, beta_1 of 0.9, beta_2 of 0.999 and epsilon of 1e-8. All steps of the training are 6e+5. The initial learning rate is 1e-4 and halved every 4e+5 steps. Also, the initial weights and biases are the same as [27].

The propose CFSRCNN is implemented on Pytorch 0.41 and Python 2.7 for training and inference, respectively. Besides, all experiments are conducted on Ubuntu 16.04 on a PC equipped with an Intel Core i7-7800 CPU, 16G RAM, and two GPUs of Nvidia GeForce GTX 1080Ti with Nvidia CUDA 9.0, and CuDNN 7.5.

D. Network analysis

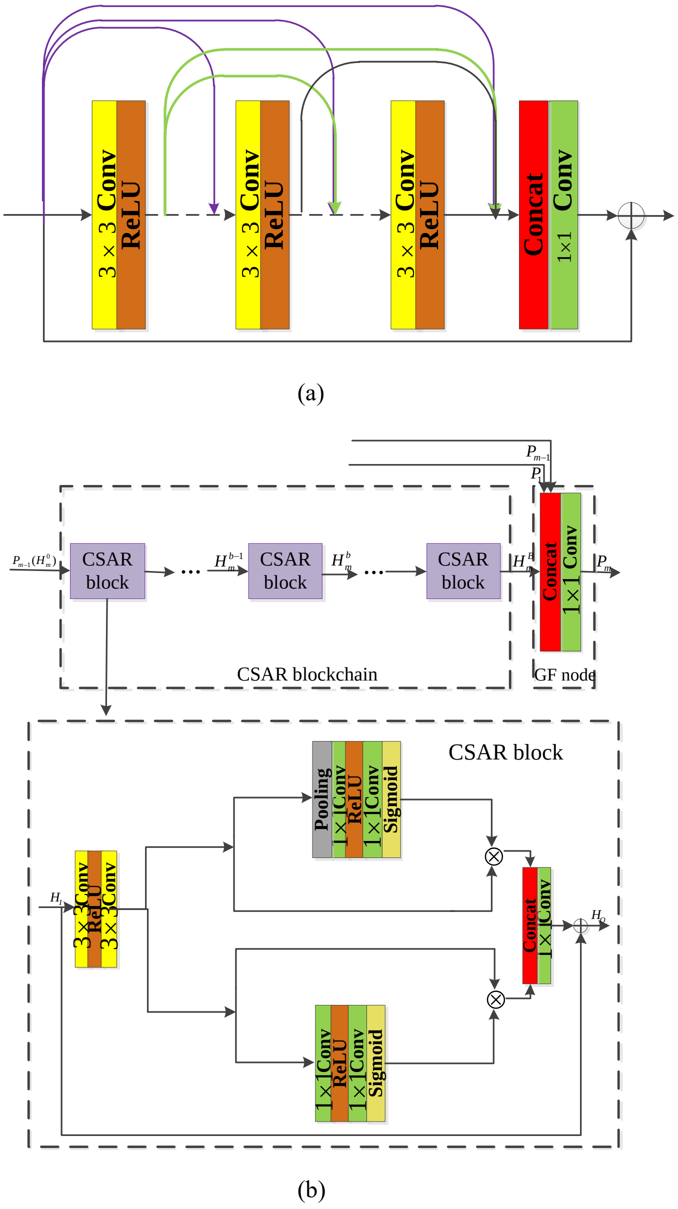
Extracting suitable features has been shown useful to accelerate the training process and improve performance in many image applications [59, 60]. To this end, feature extraction for a SR network usually involves HR features, LR features, and the combination of HR and LR features. For HR features, using bicubic interpolation to upscale a LR image as the input for training a SR model was popular [61]. This method, however, may lose some LR features, thereby resulting in performance degradation in SISR. Also, it leads to high computational cost as well. To address this issue, directly using LR features to train a SR model has been proposed. For example, FSRCNN [23] accelerates the SR task by utilizing only LR features prior to upsampling LR features in the last layer to obtain the HR image. However, as reported in [62], the upsampling operation may lead to a sudden shock to the model, which makes the training procedure unstable. To address the problem, an additional refinement process was found useful in [62] which refines the HR features obtained by the upsampling process by combining HR and LR information. Motivated by this, we also cascade a feature refinement block (FEB) to recover HR image details. Moreover, when the network depth goes deeper, the shallow-layer features would have weaker effect on deep-layer features. To address this problem, fusing hierarchical information has been proposed. Notably, the RDN in [38] fuses hierarchical non-linear features extracted from all convolutional layers via a residual dense block (see Fig. 3(a)) to enhance the memory ability of shallow layers. Besides, it employs global residual learning to learn global features complementary to local features obtained from the residual dense blocks, thereby achieving performance improvement in SISR. Moreover, the channel-wise and spatial feature modulation (CSFM) method proposed in [63] applies channel-wise and spatial attentions as block to extract hierarchical features and fuse them for enhancing the expressive ability of the SR model, as illustrated in Fig. 3(b). These methods achieve great performances for SISR. Similarly, our proposed CFSRCNN makes full use of hierarchical features to enhance LR features in the cascaded network for SISR. Nevertheless, as can be easily seen by comparing Figs. 1–2 and Fig. 3, CFSRCNN is different from the RDN and CSFM in the following aspects:

(1) We concatenate the features of two neighboring FEBs, instead of using solely the current layer (used in RDB and CSFM), as the input of all the following layers to propagate the effect of shallower-layer features to deeper layers. Besides, we use a pair of heterogeneous (3×3 and 1×1) convolutions with two layers, rather than stacking multiple (3×3) convolutions, and fuse them into a block (FEB) to reduce the network depth and complexity (i.e., the numbers of parameters and flops). The above two changes together largely reduce the number of parameters to only 5.5% of RDB and 9.3% of CSFM, and the run-time, without severely sacrificing the visual quality of reconstructed SR images, as shown in the next subsection.

(2) The Enhancement Block (EB) is not simply a concatenation of multiple FEUs. Instead, it adopts residual learning, rather than concatenating FEUs, to integrate the hierarchical LR features obtained from FEUs for enhancing the robustness of obtained LR features, which are complementary to the sFEBs. To prevent possible over-enhancement caused by previous operations, inspired by VDSR [18], we stack several convolutional layers to smooth out sharp features.

(3) By gathering the global and local features via the residual learning and sub-pixel convolution to obtain coarse HR features, rather than using solely local features, CB can effectively address the long-term dependency problem.

(4) Different from most existing learning-based SR methods which utilize LR features to train their models, we additionally make use of HR features to boost the SR performance via FRB that learns more accurate HR features by stacking multiple convolutional layers to reduce the discrepancy between the predicted SR image and its ground-truth. This also can enhance the stability of the training process.



(a)

(b)

Fig. 3. (a) The residual dense block (RDB) architecture proposed in [38]; (b) The FMM module in the CFSM [63].

CFSRCNN is composed of a number of stacked feature extraction blocks (sFEBs), an enhancement block (EB), a construction block (CB), and a feature refinement block (FRB). Such a combination efficiently makes full use of extracted hierarchical features to enhance the LR features, that are then used to infer initial SR features. The feature refinement block further employs HR features to learn more robust SR features to reduce the discrepancy between the predicted SR and its ground-truth. Thus, the combination of the sFEBs, EB, CB, and FRB can enhance the stability of SR model training. These key modules constituting CFSRCNN are elaborated below.

1) Stacked Feature Extraction Blocks (sFEBs): The sFEBs well collaborate with the EB and CB to extract more robust LR features for training a SR model, where each FEB includes an FEU and a CU. The design of FEB breaks two rules: less parameters and better performance for a SR model. For the first aspect, we use partial heterogeneous convolutions with $P = 2$ [64] to reduce computation and improve efficiency of training the SR network, where P denotes part. These heterogeneous convolutions consist of 16 standard convolutions with size 3×3 and 16 small convolutions with size 1×1. The convolutions of size 3×3 and 1×1 are integrated into FEU and CU, respectively. Since the convolution of size 1×1 can remove redundant information and distill important features [41], the proposed CU can effectively reduce the number of parameters and improve the efficiency of training CFSRCNN.

The higher diversity the network architecture is, the better performance the SR model achieves as revealed in [38]. We therefore append a standard convolutional layer of 3×3 to heterogeneous convolutions. Given a LR input image, we consolidate the sub-pixel layer to learn SR features after heterogeneous convolutions by using a standard convolution. To reconstruct the final SR image, we employ a standard convolution with size $64 \times 3 \times 3 \times 3$ as the final layer. The depth of heterogeneous convolutional network (HCN) is set to be 35. For fair comparison, we compare HCN with a standard convolutional network (SCN) of the same depth as that of HCN. Table I shows that HCN consumes significantly fewer computational cost and memory space than SCN for ×2 upscaling. Besides, HCN is also more computationally efficient than SCN in run-time complexity as shown in Table II. These results verify that our method effectively reduces the number of parameters and complexity. Additionally, Table III show that HCN achieves the same Peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) performance as that of SCN on U100 for ×2 upscaling.

Note, with the increase of network depth, shallow-layer features would make weaker effect on deep-layer ones, making a deep network suffer from SR performance degradation [49]. To address this problem, gathering hierarchical features can offer complementary contextual information from shallower layers to deeper layers [38]. Motivated by this, we propose a two-step enhancement mechanism in the sFEBs to enhance the expressive ability of the SR model. The first step utilizes the FEUs and CUs from two contiguous FEBs to extract long- and short-path features, respectively. The second step exploits a concatenation operation to fuse the extracted long- and short-path features to address the long-term dependency problem. Further, the effectiveness of the two-step enhancement mechanism for SR is illustrated in Table III, where the sFEBs with long and short-path achieve higher PSNR and SSIM than that of HCN without long and short-path on B100 and U100, showing the effectiveness of the combination of long- and short-path.

2) Enhancement Block (EB): Although a convolution of size 1×1 can distill useful features [41], repeated distillation operations may lead to information loss of the original images. To address this issue, a two-step enhancement mechanism has been adopted in the sFEBs. However, the two-step enhancement mechanism can fuse long- and short-path features by concatenating their half feature points to enhance the generalization ability of a SR model. The features extracted from the deep layers only inherit partial features from shallow layers, which cannot completely address the above-mentioned problem. To better handle this problem, we propose a two-phase EB. The first phase of EB (named EB1) applies the residual learning technique to integrate hierarchical features of all FEUs to offer complementary features for the sFEBs. The second phase uses several additional stacked FEUs to refine the learned LR features, which can mitigate over-enhanced pixels from EB1. The two phases not only provide extra information for SISR, but also enhance the diversity of the network architecture. These are useful to recover a latent HR image as verified in Table III. That is, the ‘combination of stacked FEBs and EB1’ outperforms ‘sFEBs’ in PSRN and SSIM on B100 and U100, showing the effectiveness of EB1 for SISR. ‘The combination of stacked FEBs and EB’ achieves higher PSNR and SSIM performances than ‘The combination of stacked FEBs and EB1’, implies that the second phase of EB improves SR performance.

3) Construction Block (CB): Using bicubic interpolation to upscale a given LR image as the input of a SR model leads to high computational cost and memory consumption [23]. Taking this into consideration, we use the sub-pixel technique to magnify the obtained LR features as SR features, which can improve the training efficiency by reducing the computational cost and memory consumption. However, local LR features benefit from multiple convolutions may ignore some useful information in the original LR images, making the extracted SR features anemic. It is known that global features are complementary with local features to promote the expressive ability for SISR [38]. Inspired by that, we utilize the residual learning technique to fuse global and local features for enhancing the robustness of the extracted SR features as follows. First, the outputs of the FEU from the first FEB and the EB are treated as global and local SR features, respectively. The global and local features are then upscaled by the sub-pixel technique as the global and local SR features, respectively. Second, residual learning is employed to fuse the obtained global and local SR features to derive the coarse SR features. This phase makes full use of global and local information from LR and SR features to improve the expressive ability of a SR model, where is tested by Table III. That is, the ‘The combination of sFEBs, EB and CB’ has better results of PSNR and SSIM on B100 and U100 than that of ‘The combination of sFEBs and EB’ in SISR.

TABLE I
COMPLEXITY COMPARISON OF TWO DIFFERENT NETWORKS.

Methods	Parameters	Flops
HCN	757k	5.18G
SCN	1257K	8.14G

TABLE II
RUN-TIME PERFORMANCE COMPARISON OF HCN AND SCN FOR $\times 2$ UPSCALING ON IMAGES OF SIZES 256×256, 512×512, AND 1024×1024.

Sizes	Methods	
	HCN	SCN
	$\times 2$	
256×256	0.009466	0.009536
512×512	0.011093	0.011369
1024×1024	0.019960	0.026624

TABLE III
AVERAGE PSNR AND SSIM PERFORMANCES OF VARIOUS SR METHODS FOR $\times 2$ UPSCALING ON TWO BENCHMARK DATASETS: B100 AND U100.

Methods	B100	U100
	PSNR/SSIM	PSNR/SSIM
HCN	14.64/0.4132	12.86/0.3788
SCN	14.64/0.4132	12.86/0.3788
sFEBs	31.83/0.8954	31.07/0.9169
The combination of sFEBs and EB1	32.03/0.8974	31.77/0.9243
The combination of sFEBs and EB	32.05/0.8976	31.80/0.9247
The combination of sFEBs, EB and CB	32.06/0.8981	31.91/0.9261
FRNet	14.64/0.4132	12.86/0.3788
CFSRCNN (Ours)	32.11/0.8988	32.03/0.9273

4) Feature Refinement Block (FRB): The combination of the sFEBs, EB and CB mainly extracts LR features to learn coarse SR features, which, however, cannot fully characterize the latent HR image. Also, the training of a SISR model is unstable. To tackle these problems, a six-layer FRB is proposed to refine SR features. That is, FRB can use HR features to refine SR features so as to reduce the discrepancy between the predicted SR and its ground-truth, which is complementary with the combination of the sFEBs, EB and CB. As a consequence, combining the sFEBs, EB, CB and FRB can enhance the stability of training the SR model. This fact is verified by comparing 'CFSRCNN' with 'The combination of sFEBs, EB and CB' about PSNR and SSIM on U100 and B100 in Table III. Also, we employ FRB in the LR space (called FRNet) to extract features with approximately the same number of parameters to validate the effectiveness of FRB as shown in Table III. However, solely using FRB in the LR space behalves like a VGG network with a very deep depth that usually leads to gradient vanishing/explosion, thereby significantly degrading performance. Since such validation method does not provide meaningful comparison, we do not include it in the comparison. Finally, we use a convolution of size $64 \times 3 \times 3 \times 3$ as the final layer to reconstruct a SR image.

E. Comparisons with state-of-the-arts

We conduct quantitative and qualitative analyses to evaluate the performance of CFSRCNN for SISR. Specifically, we evaluate quantitatively the average PSNR and SSIM performances and the run-time and model complexities of various SR methods on five benchmark datasets: Set5, Set14, B100, U100 and 720p. The compared methods include Bicubic, A+ [9], RFL [7], self-exemplars SR method (SelfEx) [56], cascade of sparse coding based network (CSCN) [67], RED30 [21], DnCNN [58], trainable nonlinear reaction diffusion (TNRD) [68], fast dilated SR method (FDSR) [69], SRCNN [17], FSRCNN [23], residue context sub-network (RCN) [61], VDSR [18], DRCN [19], context-wise network fusion (CNF) [51], Laplacian SR network (LapSRN) [70], DRRN [20], balanced two-stage residual networks (BTSRN) [71], MemNet [22], CARN-M [27], CARN [27], end-to-end deep and shallow network (EEDS)+ [72], two-stage convolutional network (TSCN) [73], deep recurrent fusion network (DRFN) [74], RDN [38], CSFM [63], and super-resolution feedback network (SRFBN) [75]. We also demonstrate a few reconstructed SR images for subjective visual comparisons.

The average PSNR and SSIM performances of various SR methods on the Set5, Set14, B100, U100 and 720p datasets are demonstrated in Tables IV–VIII, respectively. As shown in Table IV, our CFSRCNN with scaling factors of $\times 3$ and $\times 4$ outperforms the state-of-the-art SR methods, such as DRFN, TSCN, EEDS+ and CARN-M on Set5, and achieves a comparable performance with that of CARN-M for $\times 2$ upscaling. Specifically, compared with CARN-M, CFSRCNN achieves a notable gain of 0.14dB in PSNR for $\times 4$ upscaling. Moreover, as shown in Tables V–VIII, CFSRCNN achieves excellent performances for all the three scaling factors: $\times 2$, $\times 3$ and $\times 4$. For example, Table V shows that, CFSRCNN outperforms MemNet by 0.23dB, 0.27dB, and 0.31dB in PSNR on Set14 for $\times 2$, $\times 3$, and $\times 4$ upscaling, respectively. Similarly, Table VI also shows that CFSRCNN outperforms several popular SR methods, such as CARN-M, TSCN and DRFN. As illustrated in Table VII, CFSRCNN achieves a significant PSNR gain over CARN-M by 1.24dB on U100 for $\times 2$ upscaling. In Table VIII, CFSRCNN outperforms CARN for all the three scaling factors on 720p. Besides, these tables also show that CFSRCNN performs stably well.

Figs. 4–6 compare the subjective visual quality of CFSRCNN with that of four SR methods, including Bicubic, SRCNN, SelfEx, and CARN-M for $\times 2$ upscaling on Set14, $\times 3$ upscaling on B100 and $\times 4$ upscaling U100, respectively. To facilitate subjective comparison visually, we enlarge selected regions in the SR images, showing that the images super-resolved by CFSRCNN are clearer than those super-resolved by the other methods for $\times 2$, $\times 3$, and $\times 4$ upscaling.

We then compare the run-time complexity of CFSRCNN with that of six methods, including VDSR, DRRN, MemNet, RDN, SRFBN, and CARN-M, on HR images of sizes 256×256, 512×512 and 1024×1024 for $\times 2$ upscaling. Table IX shows that CFSRCNN achieves the fastest processing speed. Besides run-time, we also evaluate the number of parameters and flops [76] in Table X that reflect the SR model complexity (i.e., computational cost and memory consumption) for SR images of size 154×154. Table X shows that CFSRCNN consumes the third fewest number of flops while faithfully reconstructing high-quality SR images. Due to its shallower architecture and fewer concatenation operations, CFSRCNN does not outperform some deeper SR networks, such as RDN,

TABLE IV
COMPARISON OF AVERAGE PSNR/SSIM PERFORMANCES OF VARIOUS SR METHODS FOR ×2, ×3, AND ×4 UPSCALING ON SET5.

Dataset	Model	×2	×3	×4
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Set5	Bicubic	33.66/0.9299	30.39/0.8682	28.42/0.8104
	A+ [9]	36.54/0.9544	32.58/0.9088	30.28/0.8603
	RFL [7]	36.54/0.9537	32.43/0.9057	30.14/0.8548
	SelfEx [56]	36.49/0.9537	32.58/0.9093	30.31/0.8619
	CSCN [67]	36.93/0.9552	33.10/0.9144	30.86/0.8732
	RED30 [21]	37.66/0.9599	33.82/0.9230	31.51/0.8869
	DnCNN [58]	37.58/0.9590	33.75/0.9222	31.40/0.8845
	TNRD [68]	36.86/0.9556	33.18/0.9152	30.85/0.8732
	FDSR [69]	37.40/0.9513	33.68/0.9096	31.28/0.8658
	SRCNN [17]	36.66/0.9542	32.75/0.9090	30.48/0.8628
	FSRCNN [23]	37.00/0.9558	33.16/0.9140	30.71/0.8657
	RCN [61]	37.17/0.9583	33.45/0.9175	31.11/0.8736
	VDSR [18]	37.53/0.9587	33.66/0.9213	31.35/0.8838
	DRCN [19]	37.63/0.9588	33.82/0.9226	31.53/0.8854
	CNF [51]	37.66/0.9590	33.74/0.9226	31.55/0.8856
	LapSRN [70]	37.52/0.9590	-	31.54/0.8850
	IDN [41]	37.83/0.9600	34.11/0.9253	31.82/0.8903
	DRRN [20]	37.74/0.9591	34.03/0.9244	31.68/0.8888
	BTSRN [71]	37.75/-	34.03/-	31.85/-
	MemNet [22]	37.78/0.9597	34.09/0.9248	31.74/0.8893
	CARN-M [27]	37.53/0.9583	33.99/0.9255	31.92/0.8903
	CARN [27]	37.76/0.9590	34.29/0.9255	32.13/0.8937
	EEDS+ [72]	37.78/0.9609	33.81/0.9252	31.53/0.8869
	TSCN [73]	37.88/0.9602	34.18/0.9256	31.82/0.8907
	DRFN [74]	37.71/0.9595	34.01/0.9234	31.55/0.8861
	RDN [38]	38.24/0.9614	34.71/0.9296	32.47/0.8990
	CSFM [63]	38.26/0.9615	34.76/0.9301	32.61/0.9000
	SRFBN [75]	38.11/0.9609	34.70/0.9292	32.47/0.8983
	CFSRCNN (Ours)	37.79/0.9591	34.24/0.9256	32.06/0.8920

TABLE V
COMPARISON OF AVERAGE PSNR/SSIM PERFORMANCES OF VARIOUS SR METHODS FOR ×2, ×3, AND ×4 UPSCALING ON SET14.

Dataset	Model	×2	×3	×4
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Set14	Bicubic	30.24/0.8688	27.55/0.7742	26.00/0.7027
	A+ [9]	32.28/0.9056	29.13/0.8188	27.32/0.7491
	RFL [7]	32.26/0.9040	29.05/0.8164	27.24/0.7451
	SelfEx [56]	32.22/0.9034	29.16/0.8196	27.40/0.7518
	CSCN [67]	32.56/0.9074	29.41/0.8238	27.64/0.7578
	RED30 [21]	32.94/0.9144	29.61/0.8341	27.86/0.7718
	DnCNN [58]	33.03/0.9128	29.81/0.8321	28.04/0.7672
	TNRD [68]	32.51/0.9069	29.43/0.8232	27.66/0.7563
	FDSR [69]	33.00/0.9042	29.61/0.8179	27.86/0.7500
	SRCNN [17]	32.42/0.9063	29.28/0.8209	27.49/0.7503
	FSRCNN [23]	32.63/0.9088	29.43/0.8242	27.59/0.7535
	RCN [61]	32.77/0.9109	29.63/0.8269	27.79/0.7594
	VDSR [18]	33.03/0.9124	29.77/0.8314	28.01/0.7674
	DRCN [19]	33.04/0.9118	29.76/0.8311	28.02/0.7670
	CNF [51]	33.38/0.9136	29.90/0.8322	28.15/0.7680
	LapSRN [70]	33.08/0.9130	29.63/0.8269	28.19/0.7720
	IDN [41]	33.30/0.9148	29.99/0.8354	28.25/0.7730
	DRRN [20]	33.23/0.9136	29.96/0.8349	28.21/0.7720
	BTSRN [71]	33.20/-	29.90/-	28.20/-
	MemNet [22]	33.28/0.9142	30.00/0.8350	28.26/0.7723
	CARN-M [27]	33.26/0.9141	30.08/0.8367	28.42/0.7762
	CARN [27]	33.52/0.9166	30.29/0.8407	8.60/0.7806
	EEDS+ [72]	33.21/0.9151	29.85/0.8339	28.13/0.7698
	TSCN [73]	33.28/0.9147	29.99/0.8351	28.28/0.7734
	DRFN [74]	33.29/0.9142	30.06/0.8366	28.30/0.7737
	RDN [38]	34.01/0.9212	30.57/0.8468	28.81/0.7871
	CSFM [63]	34.07/0.9213	30.63/0.8477	28.87/0.7886
	SRFBN [75]	33.82/0.9196	30.51/0.8461	28.81/0.7868
	CFSRCNN (Ours)	33.51/0.9165	30.27/0.8410	28.57/0.7800

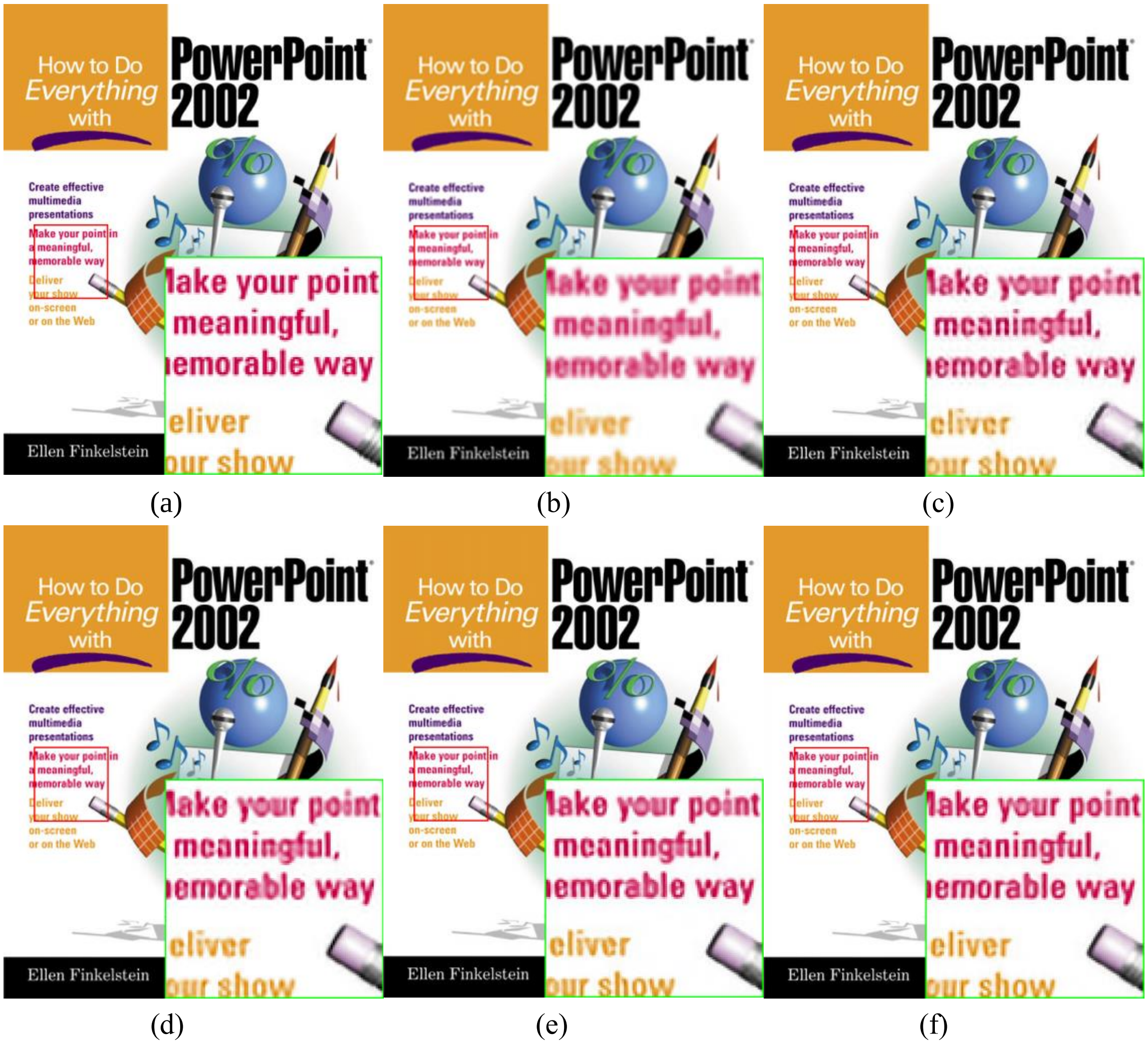
CSFM and SRFBM. However, CFSRCNN offers an excellent trade-off among visual quality, computational efficiency, and model complexity.

TABLE VI
COMPARISON OF AVERAGE PSNR/SSIM PERFORMANCES OF VARIOUS SR METHODS FOR ×2, ×3, AND ×4 UPSCALING ON B100.

Dataset	Model	×2	×3	×4
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
B100	Bicubic	29.56/0.8431	27.21/0.7385	25.96/0.6675
	A+ [9]	31.21/0.8863	28.29/0.7835	26.82/0.7087
	RFL [7]	31.16/0.8840	28.22/0.7806	26.75/0.7054
	SelfEx [56]	31.18/0.8855	28.29/0.7840	26.84/0.7106
	CSCN [67]	31.40/0.8884	28.50/0.7885	27.03/0.7161
	RED30 [21]	31.98/0.8974	28.92/0.7993	27.39/0.7286
	DnCNN [58]	31.90/0.8961	28.85/0.7981	27.29/0.7253
	TNRD [68]	31.40/0.8878	28.50/0.7881	27.00/0.7140
	FDSR [69]	31.87/0.8847	28.82/0.7797	27.31/0.7031
	SRCNN [17]	31.36/0.8879	28.41/0.7863	26.90/0.7101
	FSRCNN [23]	31.53/0.8920	28.53/0.7910	26.98/0.7150
	VDSR [18]	31.90/0.8960	28.82/0.7976	27.29/0.7251
	DRCN [19]	31.85/0.8942	28.80/0.7963	27.23/0.7233
	CNF [51]	31.91/0.8962	28.82/0.7980	27.32/0.7253
	LapSRN [70]	31.80/0.8950	-	27.32/0.7280
	IDN [41]	32.08/0.8985	28.95/0.8013	27.41/0.7297
	DRRN [20]	32.05/0.8973	28.95/0.8004	27.38/0.7284
	BTSRN [71]	32.05/-	28.97/-	27.47/-
	MemNet [22]	32.08/0.8978	28.96/0.8001	27.40/0.7281
	CARN-M [27]	31.92/0.8960	28.91/0.8000	27.44/0.7304
	CARN [27]	32.09/0.8978	29.06/0.8034	27.58/0.7349
	EEDS+ [72]	31.95/0.8963	28.88/0.8054	27.35/0.7263
	TSCN [73]	32.09/0.8985	28.95/0.8012	27.42/0.7301
	DRFN [74]	32.02/0.8979	28.93/0.8010	27.39/0.7293
	RDN [38]	32.34/0.9017	29.26/0.8093	27.72/0.7419
	CSFM [63]	32.37/0.9021	29.30/0.8105	27.76/0.7432
	SRFBN [75]	32.29/0.9010	29.24/0.8084	27.72/0.7409
	CFSRCNN (Ours)	32.11/0.8988	29.03/0.8035	27.53/0.7333

TABLE VII
COMPARISON OF AVERAGE PSNR/SSIM PERFORMANCES OF VARIOUS SR METHODS FOR ×2, ×3, AND ×4 UPSCALING ON U100.

Dataset	Model	×2	×3	×4
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
U100	Bicubic	26.88/0.8403	24.46/0.7349	23.14/0.6577
	A+ [9]	29.20/0.8938	26.03/0.7973	24.32/0.7183
	RFL [7]	29.11/0.8904	25.86/0.7900	24.19/0.7096
	SelfEx [56]	29.54/0.8967	26.44/0.8088	24.79/0.7374
	RED30 [21]	30.91/0.9159	27.31/0.8303	25.35/0.7587
	DnCNN [58]	30.74/0.9139	27.15/0.8276	25.20/0.7521
	TNRD [68]	29.70/0.8994	26.42/0.8076	24.61/0.7291
	FDSR [69]	30.91/0.9088	27.23/0.8190	25.27/0.7417
	SRCNN [17]	29.50/0.8946	26.24/0.7989	24.52/0.7221
	FSRCNN [23]	29.88/0.9020	26.43/0.8080	24.62/0.7280
	VDSR [18]	30.76/0.9140	27.14/0.8279	25.18/0.7524
	DRCN [19]	30.75/0.9133	27.15/0.8276	25.14/0.7510
	LapSRN [70]	30.41/0.9100	-	25.21/0.7560
	IDN [41]	31.27/0.9196	27.42/0.8359	25.41/0.7632
	DRRN [20]	31.23/0.9188	27.53/0.8378	25.44/0.7638
	BTSRN [71]	31.63/-	27.75/-	25.74-
	MemNet [22]	31.31/0.9195	27.56/0.8376	25.50/0.7630
	CARN-M [27]	30.83/0.9233	26.86/0.8263	25.63/0.7688
	CARN [27]	31.92/0.9256	28.06/0.8493	26.07/0.7837
	TSCN [73]	31.29/0.9198	27.46/0.8362	25.44/0.7644
	DRFN [74]	31.08/0.9179	27.43/0.8359	25.45/0.7629
	RDN [38]	32.89/0.9353	28.80/0.8653	26.61/0.8028
	CSFM [63]	33.12/0.9366	28.98/0.8681	26.78/0.8065
	SRFBN [75]	32.62/0.9328	28.73/0.8641	26.60/0.8015
	CFSRCNN (Ours)	32.07/0.9273	28.04/0.8496	26.03/0.7824


How to Do Everything with
PowerPoint 2002
Create effective multimedia presentations
Make your point in a meaningful, memorable way
Deliver your show on-screen or on the Web
Ellen Finkelstein

(a) (b) (c)

(d) (e) (f)

Fig. 4. Visual qualitative comparison of various SR methods for ×2 upscaling on Set14: (a) HR image (PSNR/SSIM), (b) Bicubic (26.85/0.9468), (c) SRCNN (30.24/0.9743), (d) SelfEx (31.49/0.9823), (e) CARN-M (33.63/0.9888) and (f) CFSRCNN (34.45/0.9901).

TABLE VIII
COMPARISON OF AVERAGE PSNR/SSIM PERFORMANCES OF VARIOUS SR METHODS FOR ×2, ×3, AND ×4 UPSCALING ON 720P.

Dataset	Model	×2	×3	×4
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
720p	CARN-M [27]	43.62/0.9791	39.87/0.9602	37.61/0.9389
	CARN [27]	44.57/0.9809	40.66/0.9633	38.03/0.9429
	CFSRCNN (Ours)	44.77/0.9811	40.93/0.9656	38.34/0.9482

TABLE IX
COMPARISON OF RUN-TIME (SECONDS) OF VARIOUS SR METHODS ON HR IMAGES OF SIZES 256 × 256, 512 × 512 AND 1024 × 1024 FOR ×2 UPSCALING.

Single Image Super-Resolution			
Size	256 × 256	512 × 512	1024 × 1024
VDSR [18]	0.0172	0.0575	0.2126
DRRN [20]	3.063	8.050	25.23
MemNet [22]	0.8774	3.605	14.69
RDN [38]	0.0553	0.2232	0.9124
SRFBN [75]	0.0761	0.2508	0.9787
CARN-M [27]	0.0159	0.0199	0.0320
CFSRCNN (Ours)	0.0153	0.0184	0.0298

TABLE X
COMPARISON OF MODEL COMPLEXITIES OF VARIOUS SR METHODS FOR ×2 UPSCALING.

Methods	Parameters	Flops
VDSR [18]	665K	15.82G
DnCNN [58]	556K	13.20G
DRCN [19]	1,774K	42.07G
MemNet [22]	677K	16.06G
CARN-M [27]	412K	2.50G
CARN [27]	1,592K	10.13G
CSFM [63]	12,841K	76.82G
RDN [38]	21,937K	130.75G
SRFBN [75]	3,631K	22.24G
CFSRCNN (Ours)	1,200K	11.08G

V. CONCLUSION

In this paper, we proposed a coarse-to-fine super-resolution CNN (CFSRCNN) for single-image super-resolution. CFSRCNN combines low-resolution and high-resolution features by cascading several types of modular blocks to prevent possible training instability and performance degradation caused by upsampling operations. We have also proposed a novel feature

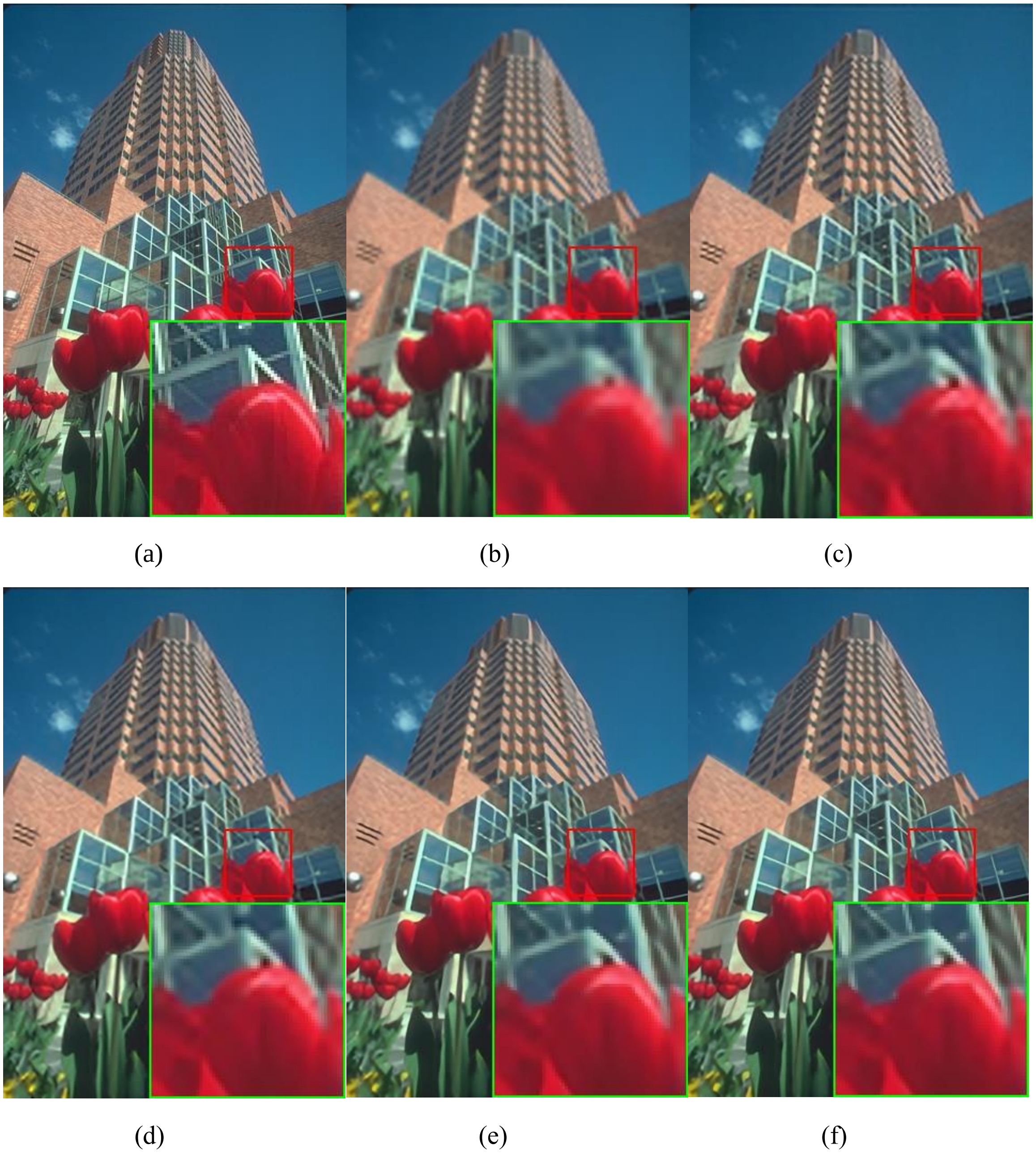


Fig. 5. Subjective visual quality comparison of various SR methods for ×3 upscaling on B100: (a) HR image (PSNR/SSIM), (b) Bicubic (25.52/0.7731), (c) SRCNN (26.58/0.8217), (d) SelfEx (27.32/0.8424), (e) CARN-M (27.90/0.8626) and (f) CFSRCNN (28.56/0.8732).

fusion scheme based on heterogeneous convolutions to address the long-term dependency problem as well as prevent information loss so as to significantly improve the computational efficiency of super-resolution without sacrificing the visual quality of reconstructed images. Comprehensive evaluations on four benchmark datasets demonstrate that CFSRCNN offers an excellent trade-off among visual quality, computational efficiency, and model complexity.

References

[1] S. Li, F. He, B. Du, L. Zhang, Y. Xu, and D. Tao, "Fast spatio-temporal residual network for video super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 10 522–10 531.

[2] Y. Huang, L. Shao, and A. F. Frangi, "Simultaneous super-resolution and cross-modality synthesis of 3d medical images using weakly-supervised joint convolutional sparse coding," in *Proc. IEEE Conf. Computer Vis. Pattern Recog.*, 2017, pp. 6070–6079.

[3] X. Luo, Y. Xu, and J. Yang, "Multi-resolution dictionary learning for face recognition," *Pattern Recog.*, vol. 93, pp. 283–292, 2019.

[4] X. Liang, D. Zhang, G. Lu, Z. Guo, and N. Luo, "A novel multicamera system for high-speed touchless palm recognition," *IEEE Trans. Syst., Man, Cybern. Syst.*, 2019.

[5] G. Polatkan, M. Zhou, L. Carin, D. Blei, and I. Daubechies, "A bayesian nonparametric approach to image super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 346–358, 2014.

[6] Y. Jchao, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, 2010.

[7] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proc. IEEE Conf. Computer Vis. Pattern Recog.*, 2015, pp. 3791–3799.

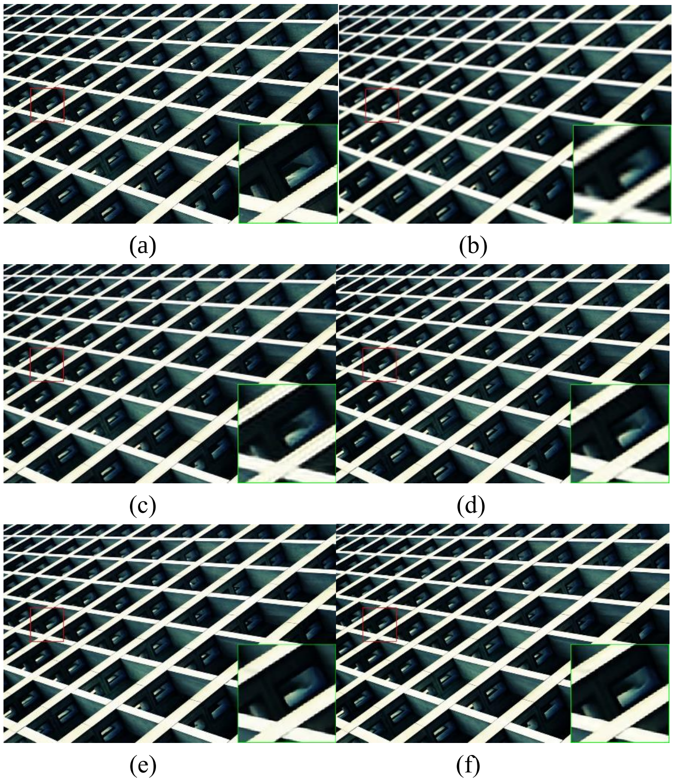


Fig. 6. Subjective visual quality comparison of various SR methods for ×4 upscaling on U100: (a) HR image (PSNR/SSIM), (b) Bicubic (22.10/0.7862), (c) SRCNN (26.08/0.8547), (d) SelfEx (28.02/0.9026), (e) CARN-M (31.80/0.9324) and (f) CFSRCNN (33.21/0.9377).

[8] K. Zhang, X. Gao, D. Tao, and X. Li, “Single image super-resolution with non-local means and steering kernel regression,” *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4544–4556, 2012.

[9] R. Timofte, V. De Smet, and L. Van Gool, “A+: Adjusted anchored neighborhood regression for fast super-resolution,” in *Asian Conf. Comput. Vis.*. Springer, 2014, pp. 111–126.

[10] X. Lu, Y. Yuan, and P. Yan, “Alternatively constrained dictionary learning for image superresolution,” *IEEE Trans. Cybern.*, vol. 44, no. 3, pp. 366–377, 2013.

[11] W. Zuo and Z. Lin, “A generalized accelerated proximal gradient approach for total-variation-based image restoration¡? query valign=”-12pt” copyedited file?” *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2748–2759, 2011.

[12] C. Tian, Y. Xu, and W. Zuo, “Image denoising using deep cnn with batch renormalization,” *Neural Net.*, vol. 121, pp. 461–473, 2020.

[13] S. Li, W. Ren, J. Zhang, J. Yu, and X. Guo, “Fast single image rain removal via a deep decomposition-composition network,” *arXiv preprint arXiv:1804.02688*, 2018.

[14] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, “Progressive image deraining networks: a better and simpler baseline,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3937–3946.

[15] J. Pan, W. Ren, Z. Hu, and M.-H. Yang, “Learning to deblur images with exemplars,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018.

[16] F. Li, H. Bai, and Y. Zhao, “Detail-preserving image super-resolution via recursively dilated residual network,” *Neurocomputing*, vol. 358, pp. 285–293, 2019.

[17] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, 2015.

[18] J. Kim, J. Kwon Lee, and K. Mu Lee, “Accurate image super-resolution using very deep convolutional networks,” in *Proc. IEEE Conf. Computer Vis. Pattern Recog.*, 2016, pp. 1646–1654.

[19] J. Kim, J. Kwon Lee, and K. Mu Lee, “Deeply-recursive convolutional network for image super-resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 1637–1645.

[20] Y. Tai, J. Yang, and X. Liu, “Image super-resolution via deep recursive residual network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 3147–3155.

[21] X. Mao, C. Shen, and Y.-B. Yang, “Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2802–2810.

[22] Y. Tai, J. Yang, X. Liu, and C. Xu, “Memnet: A persistent memory network for image restoration,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4539–4547.

[23] C. Dong, C. C. Loy, and X. Tang, “Accelerating the super-resolution convolutional neural network,” in *Proc. European Conf. Comput. Vis.*. Springer, 2016, pp. 391–407.

[24] Y. Hu, N. Wang, D. Tao, X. Gao, and X. Li, “Serf: a simple, effective, robust, and fast image super-resolver from cascaded linear regression,” *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4091–4102, 2016.

[25] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, “Deep network cascade for image super-resolution,” in *Proc. European Conf. Comput. Vis.*. Springer, 2014, pp. 49–64.

[26] Y. Hu, X. Gao, J. Li, Y. Huang, and H. Wang, “Single image super-resolution via cascaded multi-scale cross network,” *arXiv preprint arXiv:1802.08808*, 2018.

[27] N. Ahn, B. Kang, and K.-A. Sohn, “Fast, accurate, and lightweight super-resolution with cascading residual network,” in *Proc. European Conf. Comput. Vis.*, 2018, pp. 252–268.

[28] R. Timofte, R. Rothe, and L. Van Gool, “Seven ways to improve example-based single image super resolution,” in *Proc. IEEE Conf. Computer Vis. Pattern Recog.*, 2016, pp. 1865–1873.

[29] B. Du, L. Ru, C. Wu, and L. Zhang, “Unsupervised deep slow feature analysis for change detection in multi-temporal remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9976–9992, 2019.

[30] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, “Deep learning on image denoising: An overview,” *arXiv preprint arXiv:1912.13171*, 2019.

[31] Q. Liu, Z. He, X. Li, and Y. Zheng, “Ptb-tir: A thermal infrared pedestrian tracking benchmark,” *IEEE Trans. Multimedia*, 2019.

[32] S. Zhang, S. Zhang, T. Huang, and W. Gao, “Speech emotion recognition using deep convolutional neural network and discriminant temporal pyramid matching,” *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1576–1590, 2017.

[33] C. Duan, L. Cui, X. Chen, F. Wei, C. Zhu, and T. Zhao, “Attention-fused deep matching network for natural language inference.” in *IJCAI*, 2018, pp. 4033–4040.

[34] C.-Y. Chang and S.-Y. Chien, “Multi-scale dense network for single-image super-resolution,” in *IEEE Int. Conf. Acoustics, Speech Signal Process.*. IEEE, 2019, pp. 1742–1746.

[35] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, “Image super-resolution using very deep residual channel attention networks,” in *Proc. European Conf. Comput. Vis.*, 2018, pp. 286–301.

[36] L. Zhao, H. Bai, J. Liang, B. Zeng, A. Wang, and Y. Zhao, “Simultaneous color-depth super-resolution with conditional generative adversarial networks,” *Pattern Recog.*, vol. 88, pp. 356–369, 2019.

[37] M. Ni, J. Lei, R. Cong, K. Zheng, B. Peng, and X. Fan, “Color-guided depth map super resolution using convolutional neural network,” *IEEE Access*, vol. 5, pp. 26 666–26 672, 2017.

[38] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, “Residual dense network for image super-resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 2472–2481.

[39] C. Guo, C. Li, J. Guo, R. Cong, H. Fu, and P. Han, “Hierarchical features driven residual learning for depth map super-resolution,” *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2545–2557, 2018.

[40] Z. Zhong, T. Shen, Y. Yang, Z. Lin, and C. Zhang, “Joint sub-bands learning with clique structures for wavelet domain super-resolution,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 165–175.

[41] Z. Hui, X. Wang, and X. Gao, “Fast and accurate single image super-resolution via information distillation network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 723–731.

[42] J.-H. Choi, J.-H. Kim, M. Cheon, and J.-S. Lee, “Lightweight and efficient image super-resolution with block state-based recursive network,” *arXiv preprint arXiv:1811.12546*, 2018.

[43] Y. Li, E. Agustsson, S. Gu, R. Timofte, and L. Van Gool, “Carn: convolutional anchored regression network for fast and accurate single image super-resolution,” in *Proc. European Conf. Comput. Vis.*, 2018, pp. 0–0.

[44] M. Cheng, Z. Shu, J. Hu, Y. Zhang, and Z. Su, “Single image super-resolution via laplacian information distillation network,” in *Proc. Int. Conf. Digital Home*. IEEE, 2018, pp. 24–30.

[45] W. Yang, W. Wang, X. Zhang, S. Sun, and Q. Liao, “Lightweight feature fusion network for single image super-resolution,” *IEEE Signal Process. Lett.*, vol. 26, no. 4, pp. 538–542, 2019.

[46] C. Wang, Z. Li, and J. Shi, “Lightweight image super-resolution with adaptive weighted learning network,” *arXiv preprint arXiv:1904.02358*, 2019.

[47] C. Douillard, M. Jézéquel, C. Berrou, D. Electronique, A. Picart, P. Didier, and A. Glavieux, “Iterative correction of intersymbol interference: Turbo-equalization,” *European Trans. Telecom.*, vol. 6, no. 5, pp. 507–511, 1995.

[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[49] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[50] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 1874–1883.

[51] H. Ren, M. El-Khamy, and J. Lee, “Image super resolution based on fusing multiple convolution neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 54–61.

[52] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, “Multi-level wavelet-cnn for image restoration,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 773–782.

[53] E. Agustsson and R. Timofte, “Ntire 2017 challenge on single image super-resolution: Dataset and study,” in *Proc. IEEE Conf. Computer Vis. Pattern Recog. Workshops*, 2017, pp. 126–135.

[54] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, “Low-complexity single-image super-resolution based on nonnegative neighbor embedding,” 2012.

[55] D. Martin, C. Fowlkes, D. Tal, J. Malik *et al.*, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics.” ICCV Vancouver:, 2001.

[56] J.-B. Huang, A. Singh, and N. Ahuja, “Single image super-resolution from transformed self-exemplars,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 5197–5206.

[57] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang, “Real-world noisy image denoising: A new benchmark,” *arXiv preprint arXiv:1804.02603*, 2018.

[58] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017.

[59] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng, “Patch group based nonlocal self-similarity prior learning for image denoising,” in *Proc. IEEE int. conf. comput. vis.*, 2015, pp. 244–252.

[60] J. Xu, L. Zhang, D. Zhang, and X. Feng, “Multi-channel weighted nuclear norm minimization for real color image denoising,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1096–1104.

[61] Y. Shi, K. Wang, C. Chen, L. Xu, and L. Lin, “Structure-preserving image super-resolution via contextualized multitask learning,” *IEEE Trans. Multimedia*, vol. 19, no. 12, pp. 2804–2815, 2017.

[62] N. Ahn, B. Kang, and K.-A. Sohn, “Image super-resolution via progressive cascading residual network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, 2018, pp. 791–799.

[63] Y. Hu, J. Li, Y. Huang, and X. Gao, “Channel-wise and spatial feature modulation network for single image super-resolution,” *IEEE Trans. Circuits Syst. Video Technol.*, 2019.

[64] P. Singh, V. K. Verma, P. Rai, and V. P. Namboodiri, “Hetconv: Heterogeneous kernel-based convolutions for deep cnns,” *arXiv preprint arXiv:1903.04120*, 2019.

[65] A. Hore and D. Ziou, “Image quality metrics: Psnr vs. ssim,” in *2010 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 2366–2369.

[66] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli *et al.*, “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[67] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, “Deep networks for image super-resolution with sparse prior,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 370–378.

[68] Y. Chen and T. Pock, “Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, 2016.

[69] Z. Lu, Z. Yu, P. Ya-Li, L. Shi-Gang, W. Xiaojun, L. Gang, and R. Yuan, “Fast single image super-resolution via dilated residual networks,” *IEEE Access*, 2018.

[70] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, “Deep laplacian pyramid networks for fast and accurate super-resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 624–632.

[71] Y. Fan, H. Shi, J. Yu, D. Liu, W. Han, H. Yu, Z. Wang, X. Wang, and T. S. Huang, “Balanced two-stage residual networks for image super-resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, 2017, pp. 161–168.

[72] Y. Wang, L. Wang, H. Wang, and P. Li, “End-to-end image super-resolution via deep and shallow convolutional networks,” *IEEE Access*, vol. 7, pp. 31 959–31 970, 2019.

[73] Z. Hui, X. Wang, and X. Gao, “Two-stage convolutional network for image super-resolution,” in *Proc. Int. Conf. Pattern Recog.*. IEEE, 2018, pp. 2670–2675.

[74] X. Yang, H. Mei, J. Zhang, K. Xu, B. Yin, Q. Zhang, and X. Wei, “Drfn: Deep recurrent fusion network for single-image super-resolution with large factors,” *IEEE Trans. Multimedia*, vol. 21, no. 2, pp. 328–337, 2018.

[75] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, “Feedback network for image super-resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3867–3876.

[76] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu, “Attention-guided cnn for image denoising,” *Neural Net.*, 2020.

Chunwei Tian is currently pursuing the Ph.D degree in the School of Computer Science and Technology at Harbin Institute of Technology, Shenzhen. He received the M.S. degree and B.S. degree at Harbin University of Science and Technology, in 2017 and 2014, respectively. His research interests include image denoising, image super-resolution, pattern recognition and deep learning.

He has published over 20 papers in academic journals and conferences, including Neural Networks, Pattern Recognition Letters and ICASSP. He is a PC of the 18th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC 2020), a PC Assistant of IJCAI 2019, a reviewer of some journals and conferences, such as the IEEE Transactions on Industrial Informatics, the Computer Vision and Image Understanding, the Nerocomputing, the Visual Computer, the Journal of Modern Optics, the IEEE Access, the CAAI Transactions on Intelligence Technology, the International Journal of Biometrics (IJBM), the International Journal of Image and Graphics, the ACAIT2019, the 2019 International Conference on Artificial Intelligence and the Information Processing and Clod Computing (AIIPC 2019).

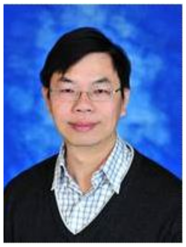
Yong Xu (Senior Member, IEEE) received his B.S. degree, M.S. degree in 1994 and 1997, respectively. He received the Ph.D. degree in Pattern Recognition and Intelligence system at NUST (China) in 2005. Now he works at Harbin Institute of Technology, Shenzhen. His current interests include pattern recognition, deep learning, biometrics, machine learning and video analysis. He has published over 70 papers in toptier academic journals and conferences. He has served as an Co-Editors-in-Chief of the International Journal of Image and Graphics, an Associate Editor of the CAAI Transactions on Intelligence Technology, an editor of the Pattern Recognition and Artificial Intelligence. More information please refer to http://www.yongxu.org/lunwen.html.

Wangmeng Zuo (Senior Member, IEEE) received the Ph.D. degree in computer application technology from the Harbin Institute of Technology, Harbin, China, in 2007. He is currently a Professor in the School of Computer Science and Technology, Harbin Institute of Technology. His current research interests include image enhancement and restoration, object detection, visual tracking, and image classification. He has published over 70 papers in toptier academic journals and conferences. He has served as a Tutorial Organizer in ECCV 2016, an Associate Editor of the IET Biometrics and Journal of Electronic Imaging, and the Guest Editor of Neurocomputing, Pattern Recognition, IEEE Transactions on Circuits and Systems for Video Technology, and IEEE Transactions on Neural Networks and Learning Systems.

Bob Zhang (Senior Member, IEEE) received the B.A. degree in computer science from York University, Toronto, ON, Canada, in 2006, the M.A.Sc. degree in information systems security from Concordia University, Montreal, QC, Canada, in 2007, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2011.

He is currently an Associate Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His current research interests include medical biometrics, pattern recognition, and image processing.

Lunke Fei (Member, IEEE) received the B.S. degree in computer science and technology and the M.S. degree in computer application technology from East China Jiaotong University, Nanchang, China, in 2004 and 2007, respectively, and the Ph.D. degree in computer science and technology from the Harbin Institute of Technology, Harbin, China, in 2016. He is currently an Associate Professor with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China. His current research interests include biometrics, pattern recognition, image processing, and machine learning.



Chia-Wen Lin (Fellow, IEEE) received the Ph.D degree in electrical engineering from National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 2000. He is currently Professor with the Department of Electrical Engineering and the Institute of Communications Engineering, NTHU. He is also Deputy Director of the AI Research Center of NTHU. His research interests include computer vision and image and video processing. He has served as a Distinguished Lecturer for IEEE Circuits and Systems Society from 2018 to 2019, a Steering Committee member for the IEEE TRANSACTIONS ON MULTIMEDIA from 2014 to 2015, and the Chair of the Multimedia Systems and Applications Technical Committee of the IEEE Circuits and Systems Society from 2013 to 2015. His articles received the Best Paper Award of IEEE VCIP 2015, Top 10% Paper Awards of IEEE MMSP 2013, and the Young Investigator Award of VCIP 2005. He has been serving as President of the Chinese Image Processing and Pattern Recognition Association, Taiwan. He served as a General Co-Chair of IEEE VCIP 2018, a Technical Program Co-Chair of IEEE ICME 2010, and a Technical Program Co-Chair of IEEE ICIP 2019. He has served as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, IEEE MULTIMEDIA, and the *Journal of Visual Communication and Image Representation*.