Knowledge-Based Systems 205 (2020) 106235

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Lightweight image super-resolution with enhanced CNN

Chunwei Tian [a,b], Ruibin Zhuge [a,b], Zhihao Wu [a,b], Yong Xu [a,b,c,*], Wangmeng Zuo [d,c], Chen Chen [e], Chia-Wen Lin [f]

[a] *Bio-Computing Research Center, Harbin Institute of Technology, Shenzhen, Shenzhen, 518055, Guangdong, China*
[b] *Shenzhen Key Laboratory of Visual Object Detection and Recognition, Shenzhen, 518055, Guangdong, China*
[c] *Peng Cheng Laboratory, Shenzhen, 518055, Guangdong, China*
[d] *School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, Heilongjiang, China*
[e] *Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, NC, 28223, USA*
[f] *Department of Electrical Engineering and the Institute of Communications Engineering, National Tsing Hua University, Hsinchu, Taiwan*

ARTICLE INFO

Article history:
Received 29 February 2020
Received in revised form 25 May 2020
Accepted 8 July 2020
Available online 15 July 2020

Keywords:
Image super-resolution
CNN
Lightweight enhanced network
Enhancement and compression
Information refinement

ABSTRACT

Deep convolutional neural networks (CNNs) with strong expressive ability have achieved impressive performances on single image super-resolution (SISR). However, their excessive amounts of convolutions and parameters usually consume high computational cost and more memory storage for training a SR model, which limits their applications to SR with resource-constrained devices in real world. To resolve these problems, we propose a lightweight enhanced SR CNN (LESRCNN) with three successive sub-blocks, an information extraction and enhancement block (IEEB), a reconstruction block (RB) and an information refinement block (IRB). Specifically, the IEEB extracts hierarchical low-resolution (LR) features and aggregates the obtained features step-by-step to increase the memory ability of the shallow layers on deep layers for SISR. To remove redundant information obtained, a heterogeneous architecture is adopted in the IEEB. After that, the RB converts low-frequency features into high-frequency features by fusing global and local features, which is complementary with the IEEB in tackling the long-term dependency problem. Finally, the IRB uses coarse high-frequency features from the RB to learn more accurate SR features and construct a SR image. The proposed LESRCNN can obtain a high-quality image by a model for different scales. Extensive experiments demonstrate that the proposed LESRCNN outperforms state-of-the-arts on SISR in terms of qualitative and quantitative evaluation. The code of LESRCNN is accessible on https://github.com/hellloxiaotian/LESRCNN.

1. Introduction

Single image super-resolution (SISR) aims at recovering a high-resolution (HR) image from a low-resolution (LR) observation. Since multiple HR images can be downsampled to the same LR image, SISR is an ill-posed problem [1]. For solving this issue, prior knowledge methods were developed by constraining the solution space [2,3]. For instance, the sparse-coding method in [4] combines LR patches and dictionary learning to obtain super-resolution (SR) patches, and then applies weighted averaging to produce the high-resolution (HR) image [5]. The non-local means with steering kernel regression method in [6] jointly utilizes non-local and local priors to extract complementary information for enhancing SR performance. To accelerate the training of a SR model, it was proposed in [7] to generalize the original accelerated proximal gradient to take the place of the Lipschitz constant to improve the convergence process. In addition, there have been some good tools, such as random forest [8] and regression [9] proposed for SISR. However, these methods rely on external example information to improve the performance of SISR, which leads to two drawbacks significantly limiting their applications. First, most of these methods resort to complex optimization methods to enhance the qualities of recovered HR images and other low-level tasks at the expense of efficiency [10,11]. Second, they usually require manually tuning parameters to boost the SR performance.

To address the above problems, various convolutional neural networks (CNNs) with flexible end-to-end network architectures and effective training strategies were proposed [12–14], having brought prosperous development in image restoration tasks, especially image super-resolution [15]. Dong et al. [16] proposed a pioneering three-layer SRCNN to obtain the SR image in a pixel mapping manner. Although the shallow SRCNN was simpler and more effective than traditional SR techniques, it was hard to make a tradeoff among depth, effectiveness and performance. Since then, the designs of deeper networks which pursue superior SR

[*] Corresponding author at: Bio-Computing Research Center, Harbin Institute of Technology, Shenzhen, Shenzhen, 518055, Guangdong, China.
E-mail address: yongxu@ymail.com (Y. Xu).

https://doi.org/10.1016/j.knosys.2020.106235
0950-7051/© 2020 Elsevier B.V. All rights reserved.

performance have become popular. For example, the cascade of sparse coding based on networks (CSCN) in [17] utilize a sparse coding technique to guide a deep network for accelerating the training speed and compressing the SR model. A very deep SR network (VDSR) was proposed in [18] to enlarge dramatically the depth of the network by stacking multiple layers to enhance SR performance. To prevent vanishing or exploding gradients, skip connections and recursive operations in deep networks were proposed [19]. The deep recursive residual network (DRRN) in [20] uses recursive learning to control the number of parameters. Besides, global and local residual learning (RL) techniques are incorporated into DRRN to facilitate the training for SISR. A very deep persistent memory network (MemNet) was proposed in [21] that applies multiple recursive units and gate units to extract and fuse multi-level features to enhance the visual qualities of reconstructed HR images. The 60-layer residual encoder–decoder network (RED30) in [22] employs a symmetric network architecture via skip connections to effectively extract details of a HR image. Although most of the above-mentioned methods can boost the visual qualities of SR results, the inputs of these networks are first upsampled to the same resolution as the output sizes for training a SR model, thereby increasing the computational cost and memory consumption significantly [23].

To better trade SR performance for resource consumption, the fast SR CNN (FSRCNN) in [23] utilizes sub-pixel convolution as the final layer to upscale the resolution of obtained feature map, thereby speeding up the SR process while degrading visual quality. To address this issue, novel network architecture based on image characteristics and multi-level feature integration have been attracting increasing attention. For example, the enhanced deep SR network (EDSR) [24] exploits improved ResNet architecture in [25] and multi-scale techniques to gain performance improvement in image SR. The residual dense network (RDN) in [26] based on EDSR utilizes global and local features to enhance the diversity of the network architecture for improving the SR performance. The multi-level wavelet CNN (MWCNN) in [27] incorporates signal processing technique into the U-Net to promote the performance and computational efficiency for restoration tasks.

Most of these algorithms above resort to increasing the depth of a SR network to enhance the expressive power of the network for performance improvement. However, this usually results in more parameters and excessive memory consumption, which is usually not affordable for resource-constrained mobile devices in practical applications.

In this paper, we propose a lightweight enhanced super-resolution CNN (LESRCNN) by cascading three sub-blocks, an information extraction and enhancement block (IEEB), a reconstruction block (RB), and an information refinement block (IRB). The IEEB uses hierarchical LR features and residual learning techniques to enhance the memory ability of shallow layers for improving the SR performance. By incorporating the heterogeneous architecture proposed in [28] into the IEEB, the amounts of parameters and memory consumption for the IEEB are significantly reduced, so as the training time. Then, the RB fuses the extracted global and local features to transform low-frequency features (i.e., the LR features) into high-frequency features (i.e., the HR features) via residual learning and sub-pixel convolution methods. This also leads to an auxiliary effect of preventing the long-term dependency problem with the IEEB. Finally, the IRB uses the coarse high-frequency features from the RB to learn more accurate SR features and construct a SR image.

The contributions of the proposed LESRCNN are summarized as follows.

(1) LESRCNN significantly reduces the number of parameters for achieving excellent performance on SISR by cascading an information extraction and enhancement block, a reconstruction block and an information refinement block. As a result, the low computational cost and memory consumption make LESRCNN particularly suitable for resource-constrained mobile devices for real-world applications.

(2) The information extraction and enhancement block extracts hierarchical LR features and fuses them to enhance the memory ability of shallow layers for improving the SISR performance. Besides, we also propose a heterogeneous architecture in the information extraction and enhancement block for compressing the network, which significantly reduces the computational cost and memory consumption. Moreover, since LR patches are used to train the SR model, the training process can be significantly accelerated.

(3) The reconstruction block combines global and local features by residual learning and sub-pixel convolution techniques to convert low-frequency features into high-frequency features, which is complementary with the information extraction and enhancement block in preventing the long-term dependency problem.

(4) The information refinement block applies coarse high-frequency features extracted by the reconstruction block to learn more accurate high-frequency features to effectively enhance the fidelity of the predicted SR image with respect to its HR ground-truth. The proposed method can deal with different scales via a model for SISR.

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 illustrates the proposed method. Section 4 provides extensive experimental results and Section 5 concludes the paper.

2. Related work

2.1. Deep CNNs based cascaded structures for SISR

With the rapid development of big data and graphic processing unit (GPU), deep CNNs have widely applied in SISR. The SR techniques based on deep CNNs mainly consist of three kinds: using high-frequency features for training a SR model, using low-frequency features for training a SR model and combination of high- and low-frequency features for training a SR model. The first method, such as DRRN [20], MemNet [21] and RED [22] upsamples the LR image the same as the given HR image as the input of deep SR network, which causes higher computational cost and more memory consumption. The second method, i.e., FSRCNN [23] only used sub-pixel convolution technique as the final layer in the SR network to amplify the extracted low-frequency features, which ignored the detailed information from high-frequency features. Although this method is superior to training speed, their SR performance is unsatisfactory. The third method simultaneously uses high- and low-frequency features to recover the high-quality image, which is very popular in SISR. Specifically, deep CNNs based cascaded structures can better express the third method above. Deep CNNs based cascaded structures can be divided into two categories from measuring SR effects: performance and efficiency.

In improving the SISR performance, cascading multistage networks can improve the resolution step by step [29]. A coarse-to-fine CNN [30] uses heterogeneous convolutions in a stack of feature extraction blocks to extract low-frequency features, then, applies feature refinement block to learn more accurate high-frequency features for image-resolution. A cascading dense network (CDN) [31] can extract hierarchical features from each convolutional layer, then, densely connect these obtained features to eliminate vanishing gradient and enhance the SR performance. Enlarging the width of network can urge more robust features in SISR. Thus, cascading two sub-networks to enlarge the width was a good choice to improve the expressive ability of the SISR model [32].

In improving the efficiency of training a SR model, compressing deep networks is very effective. A cascading residual network (CARN) [33] used multiple cascading connections to gather recursive blocks for guaranteeing performance of the SR model. Also, convolutions of size 1 × 1 were fused into the CARN to reduce the number of parameters and cut down the training time. Channels were divided into groups to simultaneously learn new features, which can improve the efficiency of training on SISR. The group convolutions and weight-typing were fed into a residual network [34] to obtain extreme efficiency for dealing with a LR image. Inspired by the facts above, we design a deep network based on cascaded structure to extract accurate LR and HR features for improving the training stability in terms of the SISR task.

2.2. Deep CNNs based blocks for SISR

Plug-and-play architectures enlarge the flexibilities of deep CNNs on different computer vision tasks, such as video [35,36], text-to-image synthesis [37], image denoising [38], image deraining [39], low-light image enhancement [40], image dehazing [41] and image super-resolution [42]. Specifically, deep CNNs based blocks can better cooperate with each component to facilitate more useful information, which is popular in real applications. This method can be divided into two groups: pursuing better performance and taking lower computational cost.

For the first aspect, features fusion methods can achieve superior performance against other methods. Gathering the same types of different feature levels of each cascaded sub-network can capture more contexts to achieve the aim of recovered high-quality image [43]. To address long-term dependency question, different types of different feature levels are merged to improve the discriminative ability for SISR. For example, Hu et al. [44] combined channel-wise and spatial features into the cascaded networks to promote the representation capacity in SISR. To deal with difficult training of the deep network, multi-scale technique jointed the dilated convolutional neural network to make obtained features better interact for obtaining more accurate SR image [45]. Additionally, according to the image nature, enhancing the effects of important features is very effective in low-level vision tasks. Zhang et al. [46] presented a channel attention mechanism into the residual network to adjust the influences of useful channels, which obtained better accuracy and visual improvements for SISR.

For the second aspect, compressing the network has become mainstream technology. [47] uses the knowledge distillation to transfer a general model to more personalized models, which can improve the training efficiency. In addition, the idea of distillation is first proposed by [48]. [49] used the combination of correlated embedding loss and knowledge distillation to resolve image recognition task. Using small convolutions is also popular to compress model. A novel information distillation network (IDN) [50] utilized group convolutions and small convolutions of 1 × 1 to remove the non-essential parts of deep network, which was useful to reduce the computational cost and complexity for training a SR model. Reducing the dimension of the data can also improve the speed in handling image restoration tasks. For example, Lai et al. [51] utilized Laplacian Pyramid network with progressive upsampling to reduce the number of parameters and obtain better performance in image super-resolution. Using signal processing idea or machine learning methods to guide the design of deep network can facilitate more features. Inspired by the fractal structure, Zhang et al. [52] applied adaptive importance learning to guide the CNN for SISR. Additionally, according to importance of candidate features, designing an efficient CNN was very effective. Hui et al. [53] exploited group convolutions of size 3 × 3 and convolutions of 1 × 1 to enhance and distill the obtained features, respectively, then, they applied an attention module to enhance the importance of key features, which obtained competitive result and fast execution ability.

According to previous researches, we can see that these methods applied different mechanisms to deal with SISR task. However, designs of their network architectures broke the rules of improving the performance or reducing computational resource. Motivated by that, we utilize deep CNNs based blocks to make a tradeoff between performance and computational cost for SISR, which is suitable to real applications.

3. Proposed method

Our proposed LESRCNN is described in Figs. 1 and 2. LESRCNN is implemented by cascading an information extraction and enhancement block (IEEB), a reconstruction block (RB) and an information refinement block (IRB). The IEEB extracts hierarchical low-frequency features and aggregates these features step-by-step by residual learning to increase the memory ability of shallow layers on deep layers. This can enhance the SR performance without significantly increasing computational cost. Also, to remove redundant information obtained, we propose a heterogeneous architecture in the IEEB. After that, the RB transforms the extracted low-frequency features into high-frequency features by fusing the global and local features. This complements of the information extraction and enhancement block can address the long-term dependency problem. Finally, the IRB refines the coarse high-frequency features to derive more accurate SR features and obtain a SR image. These blocks are explained in details in later subsections.

3.1. Network architecture

The proposed 23-layer LESRCNN consists of three blocks, an IEEB, a RB and an IRB. The 17-layer IEEB extracts and enhances the low-frequency features, and then refines the extracted low-frequency features to reduce computation. Then, the 1-layer RB converts these low-frequency features to high-frequency features. Finally, the 5-layer IRB refines the coarse high-frequency features extracted by the RB to derive more accurate SR features, which is useful to enhance the fidelity between the predicted SR and its HR ground-truth. To better explain these modules. We define the following terms. Let I_{LR} and I_{SR} represent the input LR image and the recovered SR image, respectively, f_{IEEB}, f_{RB} and f_{IRB} denote the functions of the IEEB, the RB and the IRB, respectively. The SR process with LESRCNN can be formulated as follows:

$$\begin{aligned} O_{SR} =& f_{IRB}(f_{RB}(f_{IEEB}(I_{LR}))), \\ =& f_{LESRCNN}(I_{LR}) \end{aligned} \tag{1}$$

where $f_{LESRCNN}$ denotes the function of LESRCNN. The SR performance of LESRCNN relies on an appropriately defined loss function, as will be elaborated in Section 3.2.

3.2. Loss function

We use mean squared error (MSE) [54] as the metric to measure the discrepancy between a reconstructed SR image with its HR ground-truth. We use a set of training image pairs $\{I_{LR}^i, I_{HR}^i\}_{i=1}^T$ to calculate the MSE, where T is the total number of training images. I_{LR}^i and I_{HR}^i are the ith LR and HR images, respectively. To

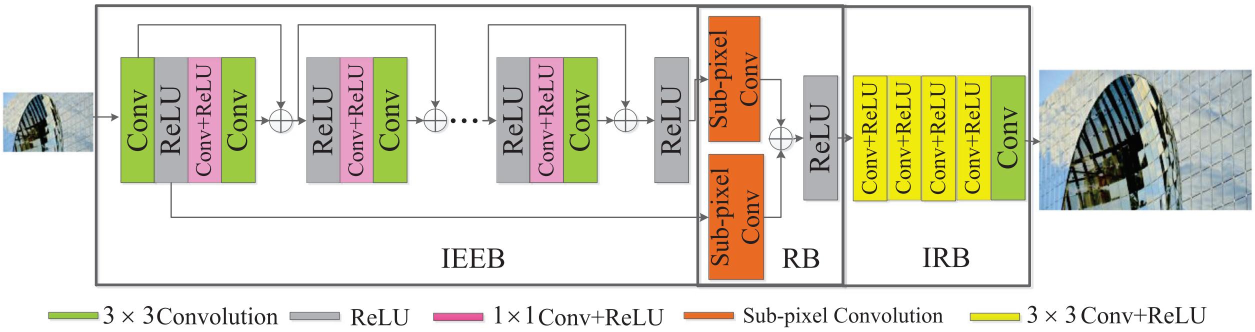

Fig. 1. Network architecture of the proposed LESRCNN.

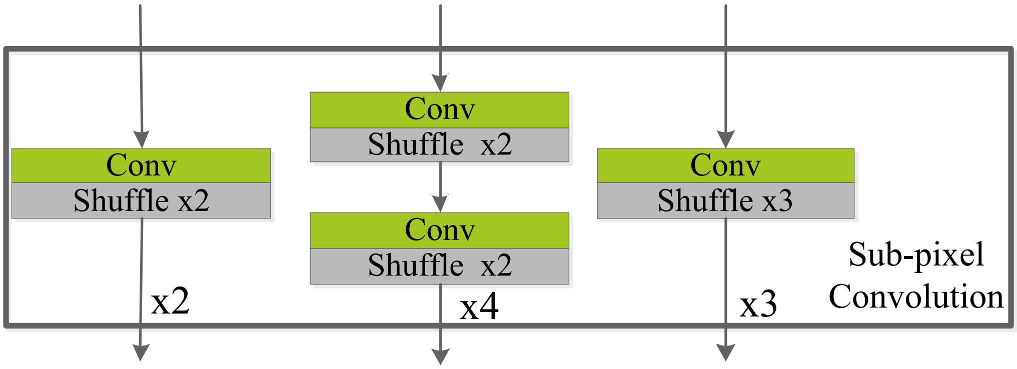

Fig. 2. Components of the sub-pixel convolution.

train LESRCNN, we aim at minimizing the following loss function:

$$l(p)=\frac{1}{2T}\sum_{i=1}^{T}\left\|f_{LESRCNN}(I_{LR}^{i})-I_{HR}^{i}\right\|^{2}, \tag{2}$$

where p is the parameter set of the SR model.

3.3. Information extraction and enhancement block (IEEB)

Typically, the deeper the depth of a network, the poorer the memory ability of the shallow layers. To handle this issue, we propose a 17-layer information extraction and enhancement block (IEEB) to achieve both good performance and high efficiency. IEEB extracts hierarchical LR features, and then uses residual learning to fuse the hierarchical features to preserve the effects of shallow-layer features on deep layers. Meanwhile, IEEB utilizes a heterogeneous architecture to distill the obtained features so as to reduce the number of parameters, computational cost, and memory consumption. The 17-layer IEEB involves two types of convolution: 3 × 3 Conv+ReLU and 1 × 1 Conv+ReLU, where Conv+ReLU represents a convolution followed by a ReLU activation function [55]. Specifically, the odd layers of 1, 3, 5, …, 17 are 3 × 3 Conv+ReLU. Note, the size of the first layer is $3\times3\times3\times64$, where 3, 3 × 3 and 64 denote the channel number of the input, filter and channel number of output, respectively. The sizes of other odd layers are 64 × 3 × 3 × 64, where 64, 3 × 3 and 64 are the channel of the input, filter and channel of output, respectively. To preserve the information of shallow layers, we fuse the hierarchical information via the residual learning method. That is, the current odd layer has effect on itself and the following odd layers. For example, the first-layer features work for layers 1, 3, 5, …, 17. Moreover, to reduce the computational cost and memory consumption, we set the convolution used in the even layers (i.e., layers 2, 4, 6, …, 16) as 1 × 1 Conv+ReLU. Let C_3 and C_1 denote the convolutional functions with sizes of 3 × 3 and 1 × 1, respectively. R the function of the ReLU. O_c^i the output of the convolution of the ith layer, and O_i the output of the ith layer, where $i=1,2,\ldots,17$. Specifically, $O_c^1=C_3(I_{SR})$ and $O_1=R(O_c^1)$. The outputs of the subsequent convolutional layers are as follows.

$$O_c^i=\begin{cases} C_3(O_{i-1}) & i \text{ is odd} \\ C_1(O_{i-1}) & i \text{ is even} \end{cases}, \tag{3}$$

where $i\in(2,17)$ is the layer index. According to the explanations above and Fig. 1, the output of each layer is formulated as

$$O_j=\begin{cases} R(O_c^j+\sum_{j=1}^{j-2}O_c^j) & j \text{ is odd} \\ R(O_c^j) & j \text{ is even} \end{cases}, \tag{4}$$

where odd $j=3,5,7,\ldots,17$. If j is even, its value falls in $2,4,6,\ldots,16$. Further, the output of IEEB is presented as $O_{17}=R(O_c^{17}+\sum_{i=1}^{15}O_c^j)$, where '+' denotes the residual learning technique and ⊕ is used to represent '+' in Fig. 1. Additionally, the output of the IEEB, O_{17} acts the RB.

3.4. Reconstruction block (RB)

Many existing methods use bicubic interpolation to upscale the input LR image to the target size as the input of a SR model, which, however, consumes significantly more computation and memory [23]. To address this issue, we incorporate the 1-layer sub-pixel convolution proposed in [56] into the reconstruction block (RB) to convert low-frequency features to high-frequency ones, where the convolution filter size is 3 × 3 and its channels of input and output are both 64. The sub-pixel convolution used in LESRCNN is divided into two kinds: a model for three scales and a model for one scale. When a SR model (i.e., LESRCNN-S) is trained for three scales, the sub-pixel convolution is composed of three components: Conv+Shuffle ×2, two Conv+Shuffle ×2, Conv+Shuffle ×3, where Conv+Shuffle ×2 denotes convolution of 3 × 3 connects Shuffle ×2 and Conv+Shuffle ×3 is convolution of 3 × 3 connects Shuffle ×3. The Conv+Shuffle ×2 and two Conv+Shuffle ×2 are used for ×2 and ×4, respectively. The Conv+Shuffle ×3 is applied for ×3. When a certain SR model (i.e., LESRCNN) is trained for single scale, the sub-pixel convolution technique only has a component from Conv+Shuffle ×2, two Conv+Shuffle ×2 and Conv+Shuffle ×3, as shown in Fig. 2.

In addition, to further address long-term dependency problem, we integrate global and local features to enhance the memory ability of shallow-layer features in deep layers. The function of RB is summarized in the following two steps. The first step uses the sub-pixel convolution to upsample the outputs of the 1st and 16th layers as global and local features, respectively. The second step utilizes residual learning to fuse the global and local features for enhancing the SR performance. After that, we apply the ReLU to convert the result of the second step into non-linearity. The process of RB can be formulated as

$$O_{RB} = R(S(O_1) + S(O_{17})), \tag{5}$$

where $S(\cdot)$ denotes the sub-pixel convolution, O_1 and O_{17} represent the global and local features, respectively, and O_{RB} represents the output of the RB.

3.5. Information refinement block (IRB)

As explained previously, combining the high- and low-frequency features to recover high-quality image is effective for SISR. However, the proposed IEEB only uses the LR image to extract low-frequency features. RB is then employed to convert the obtained low-frequency features to coarse high-frequency features, which may lack detailed information of high-frequency features. To address this problem, we propose an information refinement block (IRB) to learn more accuracy SR features and reconstruct the final SR image accordingly. The 5-layer IRB consists of 4-layer Conv+ReLU and 1-layer Conv. The Conv+ReLU layer contains a convolutional followed by a ReLU, where filter size of convolutional layer is 3 × 3 and the channel numbers of input and output are both 64. The filter size of the final convolutional layer is 3 × 3, and the channel numbers of input and output are 64 and 3, respectively. The function of IRB is formulated as.

$$O_{SR} = C_3(R(C_3(R(C_3(R(C_3(R(C_3(O_{RB}))))))))) \tag{6}$$

4. Experiments

4.1. Training dataset

By following [33], the public DIV2K dataset [57] is used as training dataset for a SR model in this paper. The DIV2K dataset comprises 800 training images, 100 validation images and 100 test images under different scales of ×2, ×3 and ×4, where they are saved in the format of '.png'. Specifically, enlarging the dataset is useful to improve the performance in image applications [55]. Motivated by that, we merge the training and validation datasets to form novel training dataset. Additionally, to reduce the training cost, each LR image is cropped as patches with size 64 × 64, which can improve the efficiency [58] of training. Further, random horizontal flips and 90° rotation operations are applied to augment obtained training patches.

4.2. Test datasets

For the test phase, four benchmark datasets [59], such as Set5 [60], Set14 [5], BSD100 [61] and Urban100 [62] with different scales of ×2, ×3 and ×4 are chosen, which are saved in the format of '.png'. The Set5 and Set14 have five and fourteen color images with different background, respectively. The 100 color images with different scenes are captured in the BSD100 (also treated as B100) and Urban100 (also named U100), respectively.

It is known that the SR methods (i.e., RED [22]) employed Y channel of YCbCr space to design experiments. Following the rule, the predicted RGB image from the LESRCNN is transformed into the Y channel to test the performance of SISR in this paper.

4.3. Implementation details

During the training, we set the initial parameters as follows. Batch size and epsilon are 64 and 1e-8, respectively. Beta_1 and beta_2 are 0.9 and 0.999, respectively. The training process has 6e+5 steps. The initial learning rate is set to 1e-4 and halved every 4e+5 steps. Additionally, other initial parameters are the same as [33]. The trained model is updated by Adam optimizer [63].

The LESRCNN is implemented by Pytorch of 0.41 and Python of 2.7. The related codes run on Ubuntu of 16.04 from a PC, which consists of a CPU of Inter Core i7-7800, a RAM of 16G and two GPUs of Nvidia GeForce GTX 1080Ti in this paper. The two GPUs can be accelerated by Nvidia CUDA of 9.0 and CuDNN of 7.5.

4.4. Network analysis

The proposed LESRCNN takes lower computational cost to obtain better performance in SISR. Its implementations by three blocks: an information extraction and enhancement block, a reconstruction block and an information refinement block. The information extraction and enhancement block, IEEB makes full use of hierarchical low-frequency features to enlarge the memory ability of shallow layers on deep layers. Meanwhile, a heterogeneous architecture is fused into the information extraction and enhancement block to distill obtained information, which is beneficial to reduce the computational cost and memory consumption. The reconstruction block converts the obtained low-frequency features from the IEEB into high-frequency features via the sub-pixel convolution technique. Then, it uses the RL method to fuse global and local features for better addressing long-term dependency problem, which can support the IEEB. Finally, the information refinement block is utilized to learn more accurate high-frequency features and construct a SR image. These techniques cooperate well to outperform state-of-the-art SR methods, such as the RED for SISR. Further, the design principles of the mentioned key techniques are shown in details as follows.

(1) Information extraction and enhancement block: In real applications, performance and computational cost are very important to mobile devices [64]. Thus, the design of the IEEB breaks the rules of lower computational cost and less memory consumption, and higher performance for SR task. For reducing the training cost, convolution of smaller filter size (e.g. 1 × 1) is a good choice to compress the network [50]. However, choosing the locations of convolutions of 1 × 1 are difficult. Due to the lower computational cost and excellent performance, a heterogeneous architecture [28] is chosen to address this problem. Specifically, the heterogeneous architecture comprises heterogeneous convolutions. Here heterogeneous convolutions with $P = 2$ [28] are used in the IEEB, where P represents part. The heterogeneous convolutions with $P = 2$ denote that each standard convolution of 3 × 3 and each convolution of 1 × 1 are connected. More information of heterogeneous convolutions refers to [28]. To convert

Table 1
Average PSNR and SSIM of different methods under scale of ×4 on Set5.

Scale	Methods	Set5 PSNR/SSIM
×4	SN	31.64/0.8864
	HN	31.62/0.8852
	IEEB	31.73/0.8877
	IEEB+RB	31.76/0.8881
	LESRCNN	31.88/0.8903

Table 2
Running time of two methods for predicting a SR image of sizes 256 × 256, 512 × 512 and 1024 × 1024.

Sizes	Methods	
	SN	HN
	×4	
256 × 256	0.00669	0.00651
512 × 512	0.00879	0.00869
1024 × 1024	0.01672	0.01651

Table 3
Complexity of two comparative methods.

Methods	Parameters	Flops
SN	630K	3.06G
HN	368K	1.38G

Table 4
PSNR and SSIM of different techniques with scale factors of ×2, ×3 and ×4 on Set5.

Dataset	Model	×2 PSNR/SSIM	×3 PSNR/SSIM	×4 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
	JOR [66]	36.58/0.9543	32.55/0.9067	30.19/0.8563
	RFL [8]	36.54/0.9537	32.43/0.9057	30.14/0.8548
	SelfEx [62]	36.49/0.9537	32.58/0.9093	30.31/0.8619
	CSCN [17]	36.93/0.9552	33.10/0.9144	30.86/0.8732
	RED [22]	37.56/0.9595	33.70/0.9222	31.33/0.8847
	DnCNN [67]	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 [16]	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 [70]	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 [71]	37.66/0.9590	33.74/0.9226	31.55/0.8856
	LapSRN [72]	37.52/0.9590	–	31.54/0.8850
	MemNet [21]	37.78/0.9597	34.09/0.9248	31.74/0.8893
	CARN-M [33]	37.53/0.9583	33.99/0.9236	31.92/0.8903
	WaveResNet [73]	37.57/0.9586	33.86/0.9228	31.52/0.8864
	CPCA [74]	34.99/0.9469	31.09/0.8975	28.67/0.8434
	NDRCN [75]	37.73/0.9596	33.90/0.9235	31.50/0.8859
	LESRCNN (Ours)	37.65/0.9586	33.93/0.9231	31.88/0.8903
	LESRCNN-S (Ours)	37.57/0.9582	34.05/0.9238	31.88/0.8907

Table 5
PSNR and SSIM of different techniques with scale factors of ×2, ×3 and ×4 on Set14.

Dataset	Model	×2 PSNR/SSIM	×3 PSNR/SSIM	×4 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
	JOR [66]	32.38/0.9063	29.19/0.8204	27.27/0.7479
	RFL [8]	32.26/0.9040	29.05/0.8164	27.24/0.7451
	SelfEx [62]	32.22/0.9034	29.16/0.8196	27.40/0.7518
	CSCN [17]	32.56/0.9074	29.41/0.8238	27.64/0.7578
	RED [22]	32.81/0.9135	29.50/0.8334	27.72/0.7698
	DnCNN [67]	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 [16]	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 [70]	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 [71]	33.38/0.9136	29.90/0.8322	28.15/0.7680
	LapSRN [72]	33.08/0.9130	29.63/0.8269	28.19/0.7720
	MemNet [21]	33.28/0.9142	30.00/0.8350	28.26/0.7723
	CARN-M [33]	33.26/0.9141	30.08/0.8367	28.42/0.7762
	WaveResNet [73]	33.09/0.9129	29.88/0.8331	28.11/0.7699
	CPCA [74]	31.04/0.8951	27.89/0.8038	26.10/0.7296
	NDRCN [75]	33.20/0.9141	29.88/0.8333	28.10/0.7697
	LESRCNN (Ours)	33.32/0.9148	30.12/0.8380	28.44/0.7772
	LESRCNN-S (Ours)	33.30/0.9145	30.16/0.8384	28.43/0.7776

obtained features into non-linearity, the activation function of ReLU is set behind each convolution in this paper. To verify the effects of the heterogeneous convolutions for SISR, we conduct the experiments by heterogeneous convolutional network (HN) and standard convolutional network (SN) in terms of performance, running time and complexity as shown in Tables 1–3. Specifically, the 17-layer HN comprises sixteen heterogeneous convolutions (eight convolutions of 3 × 3 and eight convolutions of 1 × 1) and a standard convolution of 3 × 3, where each convolution connects with a ReLU. It is known that enlarging the diversity of network is useful to promote the performance in image processing tasks [26]. Motivated by the fact, the seventeenth layer of the HN is added. Additionally, the aim of SR task is to obtain the HR image. Thus, we use a sub-pixel technique behind the HN to convert LR features into SR features. And a single convolution of 3 × 3 as the final layer of deep network is used to construct a predicted SR image. The SN has the same depth and components as the HN. However, it is noted that the sizes of the convolutions in the SN are 3 × 3.

In terms of SR performance, the SN is slightly higher than the HN on Set5 under scale of ×4 for both of peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [65] as illustrated in Table 1. However, the SN is higher than the HN in running time of a given LR image as shown in Table 2. Also, the SR has higher cost than that of the HR as presented in Table 3. According to the analysis above, we can see that the HN is more competitive than the SN in performance, running time and training cost for SISR. Thus, the HN fused into the IEEB for real applications, such as mobile device is reasonable.

It is known that as the growth of depth, memory ability of shallow layers gets weaker, which results in the consequence that the performance of image applications gets poorer [21,76]. For resolving this problem, multi-level feature fusion idea is applied in this IEEB. That is, we make full use of hierarchical information without increasing the computational cost to enhance the effects of shallow layers on deep layers in SISR. The detailed information of enhancement operation is given in Section 3.3. Additionally, the enhancement operation also increases the diversity of the network architecture, which is useful to promote the SR performance. These illustrations are proved in Table 1, where the 'IEEB' has higher PSNR and SSIM than that of the 'HN'. That shows that the enhancement operation is very effective for SISR. And the design of the IEEB is rational and effective in recovering a HR image.

(2) Reconstruction block: It is indisputable fact that employing bicubic interpolation to upsample the original LR image can bring greater training cost [23]. For tackling the problem, the sub-pixel convolution was proposed as the final layer of deep SR network [23]. However, using only low-frequency features to train the SR model may result in unstable training [29]. In terms of this issue, we set the sub-pixel into the reconstruction block, RB as the middle part of the LESRCNN to covert low-frequency features into high-frequency features, then, the output of the

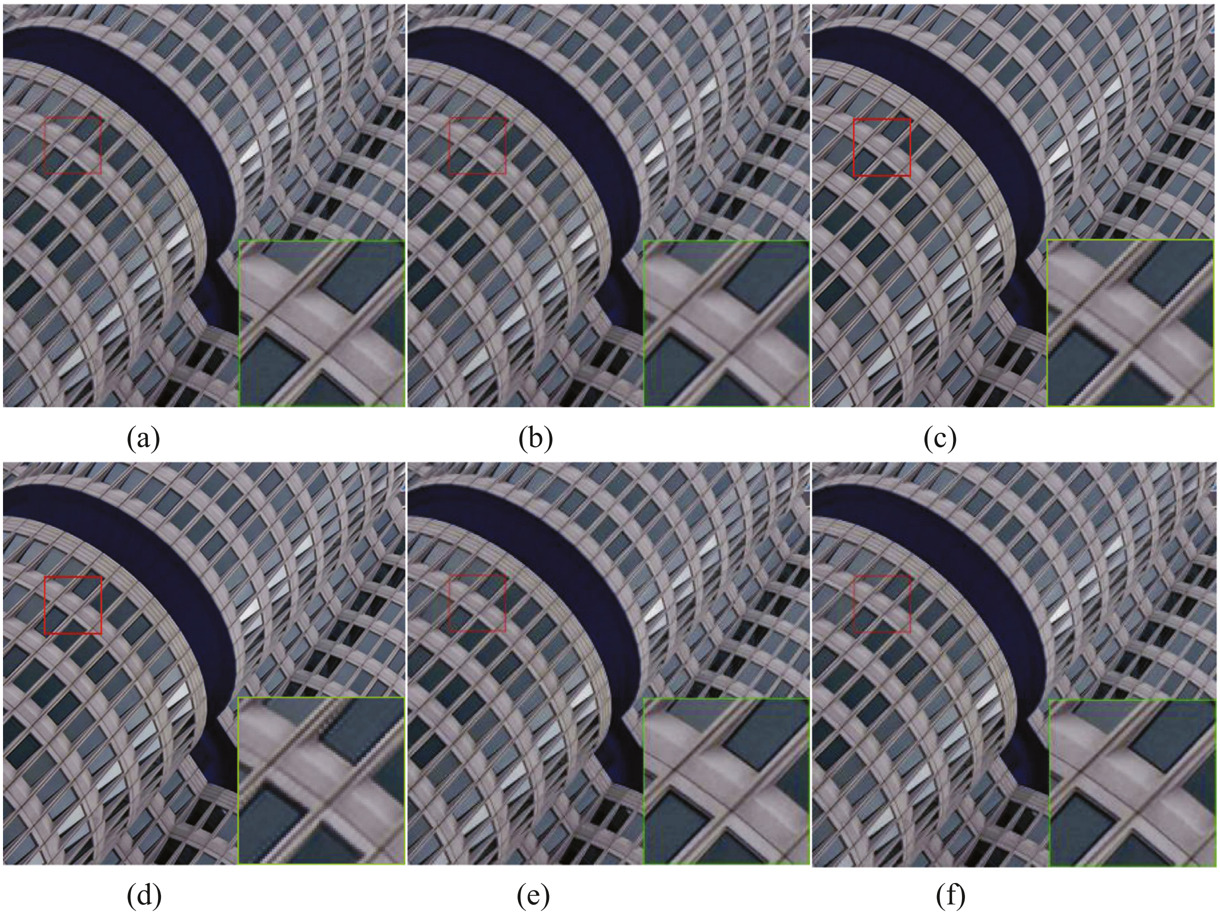

Fig. 3. Visual comparisons of different methods on an image from the U100 for ×2 scale: (a) A HR image (PSNR/SSIM), (b) Bicubic (31.88/0.9187), (c) SelfEx (32.92/0.9413), (d) SRCNN (31.88/0.9259), (e) CARN-M (37.59/0.9738) and (f) LESRCNN (37.95/0.9746).

Table 6
PSNR and SSIM of different techniques with scale factors of ×2, ×3 and ×4 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
	JOR [66]	31.22/0.8867	28.27/0.7837	26.79/0.7083
	RFL [8]	31.16/0.8840	28.22/0.7806	26.75/0.7054
	SelfEx [62]	31.18/0.8855	28.29/0.7840	26.84/0.7106
	CSCN [17]	31.40/0.8884	28.50/0.7885	27.03/0.7161
	RED [22]	31.96/0.8972	28.88/0.7993	27.35/0.7276
	DnCNN [67]	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 [16]	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 [71]	31.91/0.8962	28.82/0.7980	27.32/0.7253
	LapSRN [72]	31.80/0.8950	–	27.32/0.7280
	MemNet [21]	32.08/0.8978	28.96/0.8001	27.40/0.7281
	CARN-M [33]	31.92/0.8960	28.91/0.8000	27.44/0.7304
	WaveResNet [73]	32.15/0.8995	28.86/0.7987	27.32/0.7266
	NDRCN [75]	32.00/0.8975	28.86/0.7991	27.30/0.7263
	LESRCNN (Ours)	31.95/0.8964	28.91/0.8005	27.45/0.7313
	LESRCNN-S (Ours)	31.95/0.8965	28.94/0.8012	27.47/0.7321

Table 7
PSNR and SSIM of different techniques with scale factors of ×2, ×3 and ×4 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
	JOR [66]	29.25/0.8951	25.97/0.7972	24.29/0.7181
	RFL [8]	29.11/0.8904	25.86/0.7900	24.19/0.7096
	SelfEx [62]	29.54/0.8967	26.44/0.8088	24.79/0.7374
	DnCNN [67]	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 [16]	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 [72]	30.41/0.9100	–	25.21/0.7560
	MemNet [21]	31.31/0.9195	27.56/0.8376	25.50/0.7630
	CARN-M [33]	31.23/0.9193	27.55/0.8385	25.62/0.7694
	WaveResNet [73]	30.96/0.9169	27.28/0.8334	25.36/0.7614
	CPCA [74]	28.17/0.8990	25.61/0.8123	23.62/0.7257
	NDRCN [75]	31.06/0.9175	27.23/0.8312	25.16/0.7546
	LESRCNN (Ours)	31.45/0.9206	27.70/0.8415	25.77/0.7732
	LESRCNN-S (Ours)	31.45/0.9207	27.76/0.8424	25.78/0.7739

RB acts the IRB, where the IRB can further learn more robust high-frequency features. Although enhancement operation in the IEEB can make the information of shallow layers transmit the deep layers, up-sampling operation may loss some information of original LR image. To deal with this problem, we propose a two-step mechanism in the RB. The first step uses the sub-pixel convolution technique to upsample the outputs of the IEEB and the first layer of the IEEB as the local and global features, respectively. The second step utilizes the RL method to gather

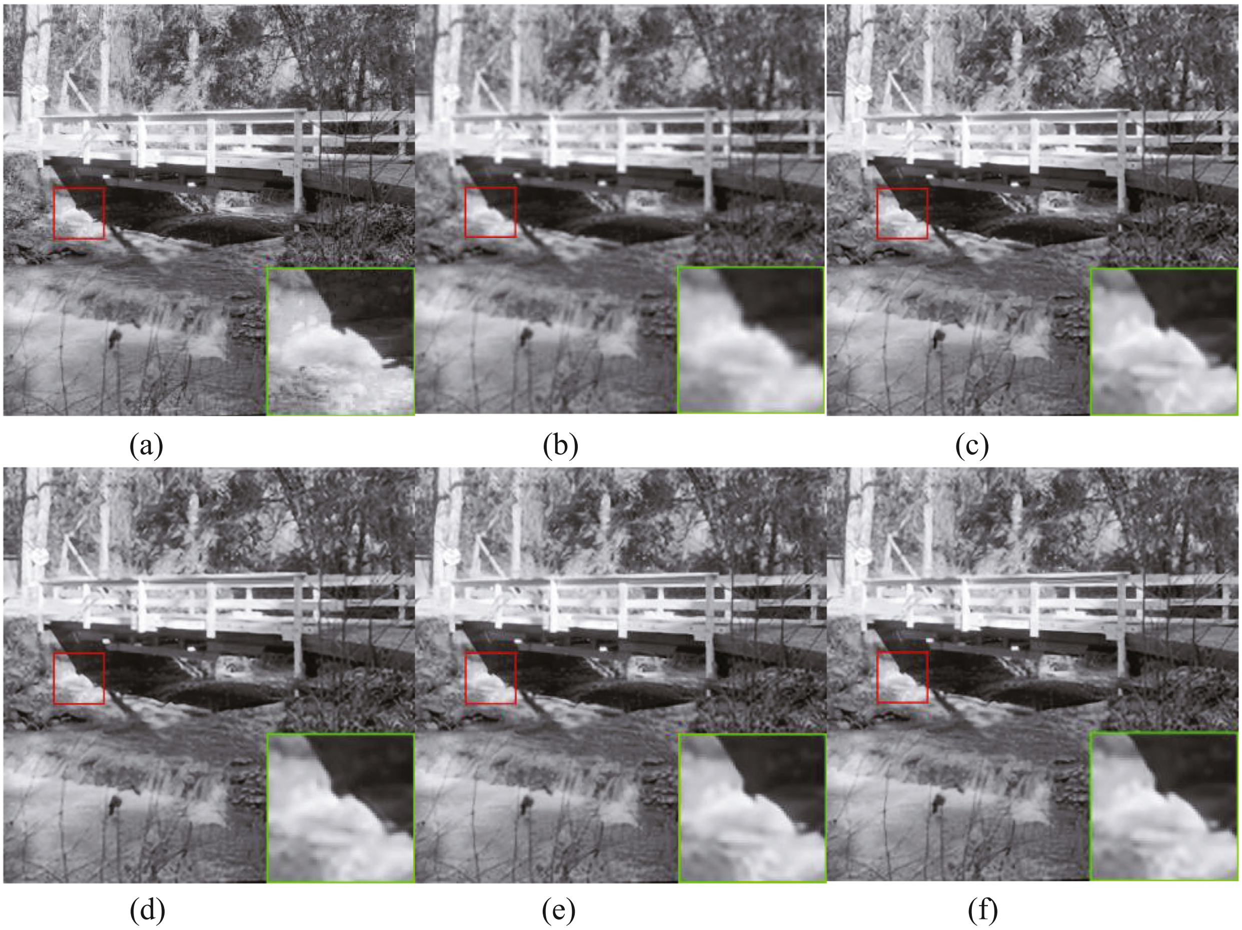

Fig. 4. Visual results of different methods on an image from the Set14 for ×3 scale: (a) A HR image (PSNR/SSIM), (b) Bicubic (25.65/0.6658), (c) SelfEx (26.03/0.7086), (d) SRCNN (25.92/0.7033), (e) CARN-M (26.34/0.7856) and (f) LESRCNN (26.82/0.7948).

Table 8
Running time of four networks for recovering the images of sizes 256 × 256, 512 × 512 and 1024 × 1024.

Single image super-resolution				
Size	VDSR [18]	MemNet [21]	CARN-M [33]	LESRCNN (Ours)
256 × 256	0.0172	0.8774	0.0159	0.0102
512 × 512	0.0575	3.605	0.0199	0.0129
1024 × 1024	0.2126	14.69	0.0320	0.0222

Table 9
Complexity of five networks for SISR.

Methods	Parameters	Flops
VDSR [18]	665K	10.90G
DnCNN [67]	556K	9.18G
DRCN [19]	1774K	29.07G
MemNet [21]	677K	11.09G
LESRCNN (Ours)	516K	3.08G

local and global features. The two-step mechanism above can further improve the expressive ability for a SISR model. Also, that is complementary to the IEEB in addressing the long-term dependency problem. The effectiveness of the two-step RB is tested by Table 1, where 'IEEB+RB' obtains better results of PSNR and SSIM than that of the 'IEEB' on the Set5.

(3) Information refinement block: According to Section 2.1, the combination of using low- and high-frequency features can achieve notable improvement over single technique. However, the IEEB emphasizes the effects of low-frequency features and the RB has power ability of converting low-frequency features into coarse high-frequency features, which ignores the influences of high-frequency features. Motivated by that, an information refinement block, IRB is developed. The IRB with four Conv+ReLU can learn more accurate high-frequency features via coarse high-frequency features from the RB, which can reduce the difference between the predicted SR image and the target HR image. Additionally, it can reconstruct a SR image by a single convolution. The IRB has great improvement on performance for the LESRCNN as shown in Table 1, where the 'LESRCNN' outperforms 'IEEB+RB' in both of PSNR and SSIM.

4.5. Comparisons with state-of-the-arts

To better evaluate the results of the LESRCNN in SISR, both of quantitative and qualitative analysis are chosen. The quantitative analysis depends on both of PSNR and SSIM, running time of a LR image and complexities of the popular methods, i.e., Bicubic, A+ [9], jointly optimized regressors (JOR) [66], RFL [8], self-exemplars super-resolution (SelfEx) [62], CSCN [17], RED [22], a denoising convolutional neural network (DnCNN) [67], trainable nonlinear reaction diffusion (TNRD) [68], fast dilated residual SR convolutional network (FDSR) [69], SRCNN [16], FSRCNN [23], residue context sub-network (RCN) [70], VDSR [18], deeply-recursive convolutional network (DRCN) [19], context-wise network fusion (CNF) [71], Laplacian SR network (LapSRN) [72], MemNet [21], CARN-M [33], wavelet domain residual network (WaveResNet) [73], convolutional principal component

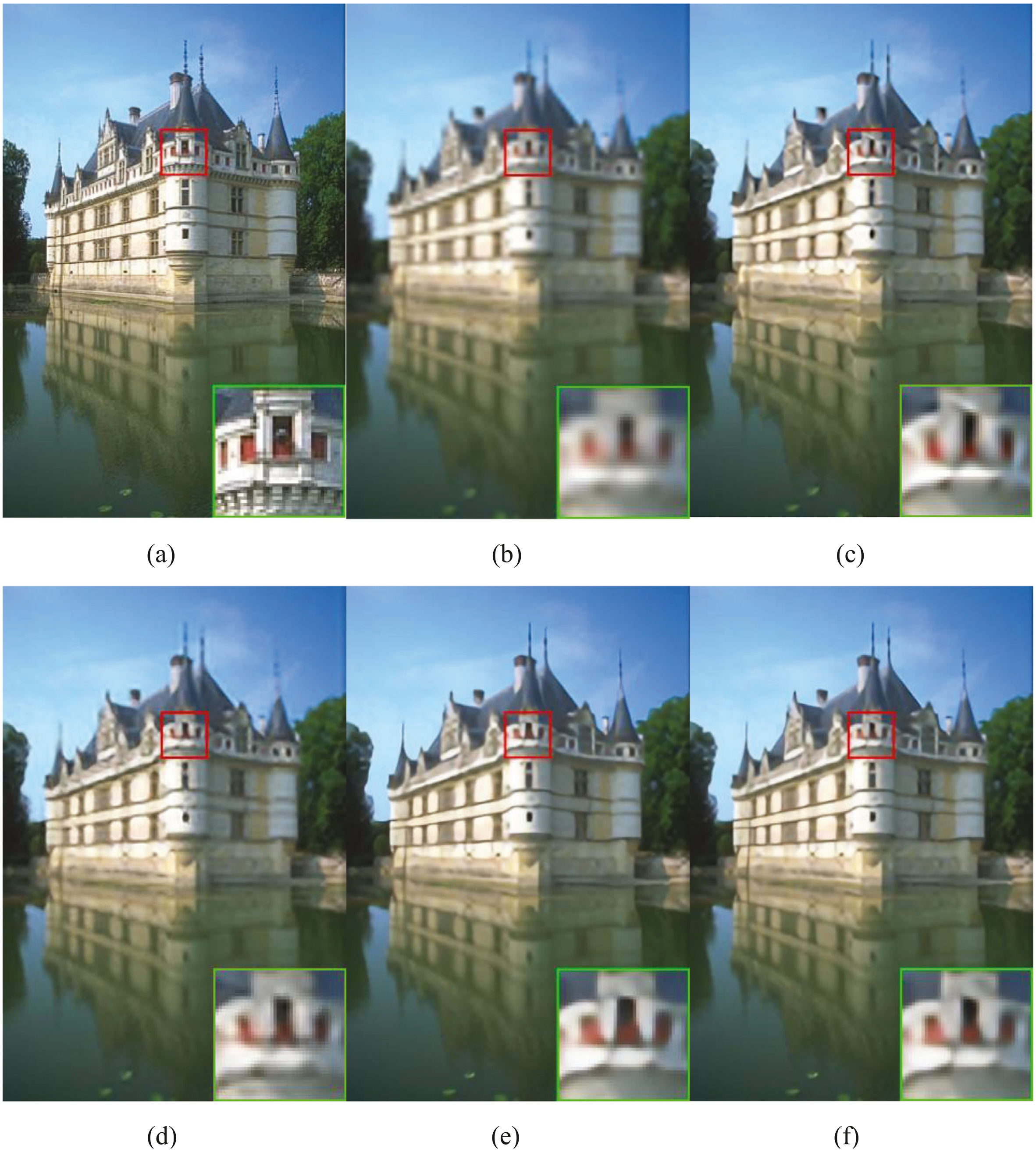

Fig. 5. Visual effects of different methods on an image from the B100 for ×4 scale: (a) A HR image (PSNR/SSIM), (b) Bicubic (25.26/0.7539), (c) SelfEx (25.83/0.7852), (d) SRCNN (25.78/0.7767), (e) CARN-M (26.39/0.8046) and (f) LESRCNN (26.46/0.8061).

analysis (CPCA) [74], new architecture of deep recursive convolution networks for SR (NDRCN) [75], LESRCNN and LESRCNN-S on four benchmark datasets (e.g. Set5, Set14, B100 and U100) for SISR. The qualitative analysis is explained by visual figures. Further, we introduce the quantitative and qualitative evaluation as follows.

Both of PSNR and SSIM are expressed through Tables 4–7. From Table 4, we can see that the proposed LESRCNN and LESRCNN-S can obtain superior performance against state-of-the-art SR methods with scale factors of ×3 and ×4 on Set5 for SISR, respectively, where the LESRCNN denotes a SR model for a certain scale and the LESRCNN-S is a SR model for three scales (i.e., ×2, ×3 and ×4). The LESRCNN-S is very suitable to real applications. Also, the LESRCNN achieves the similar result to the CNF in PSRN for ×2 on the Set5. The LESRCNN-S is 0.19 dB higher than the WaveResNet for ×3 in Table 4. The LESRCNN obtains the best performance with three different scale factors, such as ×2, ×3 and ×4 on the Set14 in SISR as illustrated in Table 5. For example, the LESRCNN obtains the improvements in PSNR of 0.04 dB and SSIM of 0.0006 than that of the popular methods, i.e., MemNet for scale factor of ×2 on the Set14.

The LESRCNN is suitable to large-scale datasets, such as B100 and U100. According to Tables 6 and 7, we can see that the proposed LESRCNN has obvious superiority than that of other popular methods, such as the CARN-M. For example, the LESRCNN has obtained notable gain both of PSRN of 0.22 dB and SSIM of 0.0013 in contrast to the CARN-M for scale factor of ×2 on the U100 as shown in Table 7. Additionally, the LESRCNN-S is also very competitive to the popular SR methods on B100 and U100. For instance, the LESRCNN-S can achieve the improvements in

PSNR of 0.15 dB and SSIM of 0.0045 than that of the CARN-M for ×4 on U100. According to the results, we find that the LESRCNN and LESRCNN-S perform well for SR task.

For running time, we choose four methods to test the running time of an image with different sizes (i.e., 256 × 256, 512 × 512 and 1024 × 1024). As explained in Table 8, it is known that the LESRCNN has faster execution to deal with the images of three different sizes than that of VDSR, MemNet and CARN-M.

In terms of the complexity, five methods are utilized to test the number of parameters and flops. From Table 9, we can see that the LESRCNN uses less parameters and flops than that of the state-of-the-art SR techniques, such as MemNet, which indicates the LESRCNN has lower computational cost and less memory consumption for training phase. In a summary, the proposed LESRCNN is superior to other state-of-the-art SR methods, such as MemNet and CARN-M in quantitative analysis.

For qualitative analysis, we use six visual figures from the given HR image, Bicubic, SelfEx, SRCNN, CARN-M and LESRCNN to test the effects of the predicted SR image for ×2, ×3 and ×4, respectively. As shown in Figs. 3–5, we can see that the magnified area of the predicted SR image from the LESRCNN is clearer than other methods, such as CARN-M for three different scales. That shows that our proposed LESRCNN is competitive in qualitative evaluation. According to the quantitative and qualitative analysis above, the LESRCNN is more effective for SISR.

5. Conclusion

In this paper, we propose a lightweight enhanced super-resolution CNN (LESRCNN) by cascading an IEEB, a RB and an IRB. The IEEB can extract and aggregate hierarchical low-frequency features, addressing the long-term dependency problem. Also, a heterogeneous architecture is fused into the IEEB to reduce the number of parameters and complexity for training a SR model. The RB can convert low-frequency features into high-frequency features by fusing global and local features, which is complementary with the IEEB in enhancing the memory ability of shallow layers on deep layers in SISR. The IRB uses coarse high-frequency features from the RB to learn more accurate SR features and construct a SR image. The LESRCNN obtains a high-resolution image via a model and multiple models for different scales. Extensive experiments illustrate that the proposed LESRCNN outperforms state-of-the-arts on SISR in terms of qualitative and quantitative evaluation.

CRediT authorship contribution statement

Chunwei Tian: Wrote this manuscript, Offered key ideas, Conducted some experiments. **Ruibin Zhuge:** Conducted some comparative experiments for this manuscript. **Zhihao Wu:** Provided part visual super-resolution images. **Yong Xu:** Revised this manuscript, Provided valuable comments for this manuscript. **Wangmeng Zuo:** Provided the partial experimental analysis. **Chen Chen:** Revised the experiment part of this manuscript. **Chia-Wen Lin:** Revised this manuscript, Offered valuable comments.

Declaration of competing interest

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

Acknowledgments

C. Tian, R. Zhu, Z. Wu and Y. Xu are supported in part by the National Nature Science Foundation of China Grant No. 61876051, and in part by the Shenzhen Key Laboratory of Visual Object Detection and Recognition under Grant No. ZDSYS20190902093015527 in this work.

References

[1] K. Zhang, W. Zuo, L. Zhang, Deep plug-and-play super-resolution for arbitrary blur kernels, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1671–1681.
[2] J. Xu, L. Zhang, D. Zhang, X. Feng, Multi-channel weighted nuclear norm minimization for real color image denoising, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1096–1104.
[3] J. Zhao, H. Hu, F. Cao, Image super-resolution via adaptive sparse representation, Knowl.-Based Syst. 124 (2017) 23–33.
[4] Z. Zha, X. Yuan, B. Wen, J. Zhou, J. Zhang, C. Zhu, A benchmark for sparse coding: When group sparsity meets rank minimization, IEEE Trans. Image Process. 29 (2020) 5094–5109.
[5] J. Yang, J. Wright, T.S. Huang, Y. Ma, Image super-resolution via sparse representation, IEEE Trans. Image Process. 19 (11) (2010) 2861–2873.
[6] K. Zhang, X. Gao, D. Tao, X. Li, Single image super-resolution with non-local means and steering kernel regression, IEEE Trans. Image Process. 21 (11) (2012) 4544–4556.
[7] W. Zuo, Z. Lin, A generalized accelerated proximal gradient approach for total-variation-based image restoration, IEEE Trans. Image Process. 20 (2011) 2748–2759.
[8] S. Schulter, C. Leistner, H. Bischof, Fast and accurate image upscaling with super-resolution forests, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3791–3799.
[9] R. Timofte, V. De Smet, L. Van Gool, A+: Adjusted anchored neighborhood regression for fast super-resolution, in: Asian Conference on Computer Vision, Springer, 2014, pp. 111–126.
[10] D. Ren, W. Zuo, D. Zhang, L. Zhang, M.-H. Yang, Simultaneous fidelity and regularization learning for image restoration, IEEE Trans. Pattern Anal. Mach. Intell. (2019).
[11] X. Li, B. Du, C. Xu, Y. Zhang, L. Zhang, D. Tao, Robust learning with imperfect privileged information, Artificial Intelligence 282 (2020) 103246.
[12] C. Tian, Y. Xu, W. Zuo, Image denoising using deep CNN with batch renormalization, Neural Netw. 121 (2020) 461–473.
[13] D. Yuan, N. Fan, Z. He, Learning target-focusing convolutional regression model for visual object tracking, Knowl.-Based Syst. (2020) 105526.
[14] C. Tian, Y. Xu, W. Zuo, B. Du, C.-W. Lin, D. Zhang, Designing and training of a dual CNN for image denoising, 2020, arXiv preprint arXiv:2007.03951.
[15] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, Q. Liao, Deep learning for single image super-resolution: A brief review, IEEE Trans. Multimed. (2019).
[16] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2015) 295–307.
[17] Z. Wang, D. Liu, J. Yang, W. Han, T. Huang, Deep networks for image super-resolution with sparse prior, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 370–378.
[18] J. Kim, J. Kwon Lee, K. Mu Lee, Accurate image super-resolution using very deep convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1646–1654.
[19] J. Kim, J. Kwon Lee, K. Mu Lee, Deeply-recursive convolutional network for image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1637–1645.
[20] Y. Tai, J. Yang, X. Liu, Image super-resolution via deep recursive residual network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3147–3155.
[21] Y. Tai, J. Yang, X. Liu, C. Xu, Memnet: A persistent memory network for image restoration, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4539–4547.
[22] X. Mao, C. Shen, Y.-B. Yang, Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections, in: Advances in Neural Information Processing Systems, 2016, pp. 2802–2810.
[23] C. Dong, C.C. Loy, X. Tang, Accelerating the super-resolution convolutional neural network, in: European Conference on Computer Vision, Springer, 2016, pp. 391–407.
[24] B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, Enhanced deep residual networks for single image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 136–144.
[25] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[26] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2472–2481.
[27] P. Liu, H. Zhang, K. Zhang, L. Lin, W. Zuo, Multi-level wavelet-CNN for image restoration, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 773–782.

[28] P. Singh, V.K. Verma, P. Rai, V.P. Namboodiri, Hetconv: Heterogeneous kernel-based convolutions for deep cnns, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4835–4844.

[29] N. Ahn, B. Kang, K.-A. Sohn, Image super-resolution via progressive cascading residual network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 791–799.

[30] C. Tian, Y. Xu, W. Zuo, B. Zhang, L. Fei, C.-W. Lin, Coarse-to-fine CNN for image super-resolution, IEEE Trans. Multimed. (2020).

[31] W. Wei, G. Feng, Q. Zhang, D. Cui, M. Zhang, F. Chen, Accurate single image super-resolution using cascading dense connections, Electron. Lett. (2019).

[32] D. Chowdhury, D. Androutsos, Single image super-resolution via cascaded parallel multisize receptive field, in: 2019 IEEE International Conference on Image Processing (ICIP), IEEE, 2019, pp. 2861–2865.

[33] N. Ahn, B. Kang, K.-A. Sohn, Fast, accurate, and lightweight super-resolution with cascading residual network, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 252–268.

[34] N. Ahn, B. Kang, K.-A. Sohn, Photo-realistic image super-resolution with fast and lightweight cascading residual network, 2019, arXiv preprint arXiv:1903.02240.

[35] D. Yuan, X. Li, Z. He, Q. Liu, S. Lu, Visual object tracking with adaptive structural convolutional network, Knowl.-Based Syst. (2020) 105554.

[36] S. Li, F. He, B. Du, L. Zhang, Y. Xu, D. Tao, Fast spatio-temporal residual network for video super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 10522–10531.

[37] J. Liang, W. Pei, F. Lu, CPGAN: Full-spectrum content-parsing generative adversarial networks for text-to-image synthesis, 2019, arXiv preprint arXiv:1912.08562.

[38] C. Tian, Y. Xu, L. Fei, J. Wang, J. Wen, N. Luo, Enhanced CNN for image denoising, CAAI Trans. Intell. Technol. 4 (1) (2019) 17–23.

[39] D. Ren, W. Shang, P. Zhu, Q. Hu, D. Meng, W. Zuo, Single image deraining using bilateral recurrent network, IEEE Trans. Image Process. (2020).

[40] W. Ren, S. Liu, L. Ma, Q. Xu, X. Xu, X. Cao, J. Du, M.-H. Yang, Low-light image enhancement via a deep hybrid network, IEEE Trans. Image Process. 28 (9) (2019) 4364–4375.

[41] W. Ren, J. Pan, H. Zhang, X. Cao, M.-H. Yang, Single image dehazing via multi-scale convolutional neural networks with holistic edges, Int. J. Comput. Vis. 128 (1) (2020) 240–259.

[42] K. Zhang, W. Zuo, L. Zhang, Learning a single convolutional super-resolution network for multiple degradations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3262–3271.

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

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

[45] W. Shi, F. Jiang, D. Zhao, Single image super-resolution with dilated convolution based multi-scale information learning inception module, in: 2017 IEEE International Conference on Image Processing (ICIP), IEEE, 2017, pp. 977–981.

[46] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, Image super-resolution using very deep residual channel attention networks, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 286–301.

[47] Y. Wu, L. Jiang, Y. Yang, Revisiting embodiedqa: A simple baseline and beyond, IEEE Trans. Image Process. 29 (2020) 3984–3992.

[48] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, 2015, arXiv preprint arXiv:1503.02531.

[49] X. Wu, R. He, Y. Hu, Z. Sun, Learning an evolutionary embedding via massive knowledge distillation, Int. J. Comput. Vis. (2020) 1–18.

[50] Z. Hui, X. Wang, X. Gao, Fast and accurate single image super-resolution via information distillation network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 723–731.

[51] W.-S. Lai, J.-B. Huang, N. Ahuja, M.-H. Yang, Fast and accurate image super-resolution with deep laplacian pyramid networks, IEEE Trans. Pattern Anal. Mach. Intell. (2018).

[52] L. Zhang, P. Wang, C. Shen, L. Liu, W. Wei, Y. Zhang, A. Van Den Hengel, Adaptive importance learning for improving lightweight image super-resolution network, Int. J. Comput. Vis. 128 (2) (2020) 479–499.

[53] Z. Hui, X. Gao, Y. Yang, X. Wang, Lightweight image super-resolution with information multi-distillation network, in: Proceedings of the 27th ACM International Conference on Multimedia, ACM, 2019, pp. 2024–2032.

[54] C. Douillard, M. Jézéquel, C. Berrou, D. Electronique, A. Picart, P. Didier, A. Glavieux, Iterative correction of intersymbol interference: Turbo-equalization, Eur. Trans. Telecommun. 6 (5) (1995) 507–511.

[55] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[56] W. Shi, J. Caballero, F. Huszár, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, Z. Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1874–1883.

[57] E. Agustsson, R. Timofte, Ntire 2017 challenge on single image super-resolution: Dataset and study, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 126–135.

[58] J. Xu, L. Zhang, W. Zuo, D. Zhang, X. Feng, Patch group based nonlocal self-similarity prior learning for image denoising, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 244–252.

[59] C.-Y. Yang, C. Ma, M.-H. Yang, Single-image super-resolution: A benchmark, in: European Conference on Computer Vision, Springer, 2014, pp. 372–386.

[60] M. Bevilacqua, A. Roumy, C. Guillemot, M.L. Alberi-Morel, Low-Complexity Single-Image Super-Resolution Based on Nonnegative Neighbor Embedding, BMVA press, 2012.

[61] 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.

[62] J.-B. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5197–5206.

[63] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[64] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, H. Liu, Attention-guided CNN for image denoising, Neural Netw. (2020).

[65] A. Hore, D. Ziou, Image quality metrics: PSNR vs. SSIM, in: 2010 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 2366–2369.

[66] D. Dai, R. Timofte, L. Van Gool, Jointly optimized regressors for image super-resolution, in: Computer Graphics Forum, vol. 34, (2) Wiley Online Library, 2015, pp. 95–104.

[67] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising, IEEE Trans. Image Process. 26 (7) (2017) 3142–3155.

[68] Y. Chen, T. Pock, Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2016) 1256–1272.

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

[70] Y. Shi, K. Wang, C. Chen, L. Xu, L. Lin, Structure-preserving image super-resolution via contextualized multitask learning, IEEE Trans. Multimed. 19 (12) (2017) 2804–2815.

[71] H. Ren, M. El-Khamy, J. Lee, Image super resolution based on fusing multiple convolution neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 54–61.

[72] W.-S. Lai, J.-B. Huang, N. Ahuja, M.-H. Yang, Deep laplacian pyramid networks for fast and accurate super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 624–632.

[73] W. Bae, J. Yoo, J. Chul Ye, Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 145–153.

[74] J. Xu, M. Li, J. Fan, X. Zhao, Z. Chang, Self-learning super-resolution using convolutional principal component analysis and random matching, IEEE Trans. Multimed. 21 (5) (2018) 1108–1121.

[75] F. Cao, B. Chen, New architecture of deep recursive convolution networks for super-resolution, Knowl.-Based Syst. 178 (2019) 98–110.

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