BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 70(1), 2022, Article number: e139616
DOI: 10.24425/bpasts.2021.139616

Deep networks for image super-resolution using hierarchical features

Xin YANG*, Yifan ZHANG, and Dake ZHOU

College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, 210016 Nanjing, Jiangsu, China

Abstract. To better extract feature maps from low-resolution (LR) images and recover high-frequency information in the high-resolution (HR) images in image super-resolution (SR), we propose in this paper a new SR algorithm based on a deep convolutional neural network (CNN). The network structure is composed of the feature extraction part and the reconstruction part. The extraction network extracts the feature maps of LR images and uses the sub-pixel convolutional neural network as the up-sampling operator. Skip connection, densely connected neural networks and feature map fusion are used to extract information from hierarchical feature maps at the end of the network, which can effectively reduce the dimension of the feature maps. In the reconstruction network, we add a 3×3 convolution layer based on the original sub-pixel convolution layer, which can allow the reconstruction network to have better nonlinear mapping ability. The experiments show that the algorithm results in a significant improvement in PSNR, SSIM, and human visual effects as compared with some state-of-the-art algorithms based on deep learning.

Key words: super-resolution, convolutional neural network, sub-pixel convolutional neural network, densely connected neural networks.

1. INTRODUCTION

Single image super-resolution (SISR) is the process of obtaining an HR image from a single LR image, which has been widely used in the field of security surveillance, medical images, satellite images, etc. [1–3]. Because there are multiple solutions to map from LR images to HR images, SR is an ill-posed problem. Especially when the up-sampling scale is large, it is difficult to recover high-frequency details in the reconstructed image. For the high-frequency information of reconstructed images, it is imperative to obtain the low-frequency information in the broad scope of the LR images.

In recent years, with the development of deep learning and the abundance of image datasets, deep learning based SR models have achieved excellent accuracy results and attracted wide attention of scholars. Notably, the deep convolution neural network (CNN) model completely outperforms the shallow CNN model. The deeper network model has a wider receptive field and can recover high-frequency information of the HR image by using low-frequency information in a wider spatial range of an LR image, which makes the edges in the HR image sharper. The receptive field is usually increased by the convolution layers with the convolution kernel larger than 1×1 or a pooling layer. Since the pooling layer will lose pixel information, the convolution layer is usually used to deepen the network to obtain a large receptive field for the SR task.

Dong *et al.* [4] propose a fully CNN-based method called SRCNN to learn the mapping between LR image blocks and HR image blocks. The SRCNN consists of three convolutional layers with filter sizes of 9×9, 1×1 and 5×5, respectively. The first convolution layer maps LR image blocks to high-dimensional vectors. Then, using another convolution layer, the LR high-dimensional vectors are mapped to HR high-dimensional vectors. Finally, the final HR image is reconstructed using a high-dimensional vector of HR image blocks. These three steps are based on different principles but can be done with the same CNN. Nevertheless, only one CNN is used to obtain the mapping from the LR image to the feature map, and the ability of feature extraction is limited. In addition, because the entire network has only three convolutional layers, nonlinear mapping capability is limited.

Kim *et al.* present the VDSR [5] on the basis of SRCNN. All filters of convolutional layers are set to 3×3. The receptive field of the network is proportional to the depth. The receptive field is $(2D+1) \times (2D+1)$ when the depth is D. Hari *et al.* found that the SR method based on a forward propagation network can not completely solve the problem of interdependence between LR and HR images at a large scale. Therefore, a deep back-projection network (DBPN) [6] is designed, which can achieve better performance in large-scale reconstruction. Because the depth of VDSR is too deep and the parameters of the network are excessive, it is difficult to train. Kim *et al.* [7] came up with a deeply-recursive convolutional network (DRCN) to recurse the single convolution layer of the mapping part. The output feature map of each recursion is connected to the reconstruction network through the skip connection to obtain the intermediate reconstruction result. Finally, The HR image can be obtained by weighted summation of all intermediate reconstruction results. The DRCN uses recursive techniques in the inference network to avoid introducing too many parameters in the model. Although the pre-operation of SRCNN, VDSR and DRCN only uses bicubic interpolation (BI), it still increases computational complexity and introduces artificial noise.

*e-mail: yangxin@nuaa.edu.cn

Manuscript submitted 2020-09-21, revised 2021-07-26, initially accepted for publication 2021-10-10, published in February 2022.

Tai *et al.* [8] proposed a memory block structure and built the MEMNET. With the increase of network depth, the feature maps of different convolutional layers have different receptive fields. Therefore, the feature maps in deep networks are hierarchical. Objects in different images have different scales, and the hierarchical feature map also provides more information for reconstruction. Based on the above theory, Zhang *et al.* [9] presented the RDN model, which introduces dense connections to make full use of feature maps at different levels.

Yamanaka *et al.* [10] constructed the DCSCN network, in which the whole network model is divided into two parts: the feature extraction network and the reconstruction network. All convolutional layers of the feature extraction network are connected to the output of the feature extraction network by means of skip connection. All feature maps are concatenated and output to the reconstruction network. Chen *et al.* [11] propose a content-guided deep residual network for single image SR. The network built a guided residual block through a convolution network. To train a deeper SR reconstruction network, Lim *et al.* [12] constructed the EDSR, which removes the BN layer in the residual network and introduces the residual scaling factor to make the training of the model more stable and to achieve remarkable results. Yang *et al.* [13] constructed a novel image SR network of multiple attention mechanism (MAMSR), which includes a channel attention mechanism and a spatial attention mechanism. Chen *et al.* [14] presented an extended layer named enhanced cycle residual block (CRB), and then developed a lightweight network with CRB as the feature inference layer. It improves feature expression ability by alternating and multiplexing convolutional layers without increasing parameters. Although the deeper convolution neural network can bring about a larger receptive field, this will also increase the amount of calculation required, make the training difficult and reduce efficiency. In addition, the feature information obtained by each channel plays a different and important role in detail recovery during the SR process. Yang *et al.* [15] developed a multi-branch attention SR model, which can boast excellent performance.

In this paper, to better extract feature maps from LR images and recover high-frequency information in the HR images, we propose a new SR algorithm based on CNN using hierarchical features.

2. DCSCN ALGORITHM

The model of DCSCN [10] can be divided into two parts, which are the feature extraction network and the reconstruction network. It is shown in Fig. 1. The feature extraction network of DCSCN is a directed acyclic structure. The output of the previous CNN is used as the input of the next CNN. The feature map of each convolution layer is directly connected to the output of the feature extraction network by means of skip connection. The output feature map of the feature extraction network can be expressed as $[F_1, F_2, \ldots, F_i]$.

In DCSCN, 7 CNNs with filter size of 3×3 are used in the feature extraction network. The number of filters in each CNN is reduced from 96 to 32. The detailed DCSCN network model is shown in Table 1.

Table 1
Number of filters in DCSCN model

Net	Feature extraction network							Reconstruction network			
	1	2	3	4	5	6	7	A1	B1	B2	L
DCSCN	96	76	65	55	47	39	32	64	32	32	4
cDCSCN	32	26	22	18	14	11	8	24	8	8	4

DCSCN constructs a parallel CNN structure similar to Network in Network. The output feature maps of two paths in parallel are concatenated, which is expressed as $[F_A, F_B]$. The dimension of $[F_A, F_B]$ is reduced to the dimension of HR by a 1×1 CNN. Because the input and output of the network are highly similar, DCSCN introduces a residual network to connect the input LR image to the output of the reconstructed network through the BI operation. Then the output of the network is $r = \hat{y} - x$.

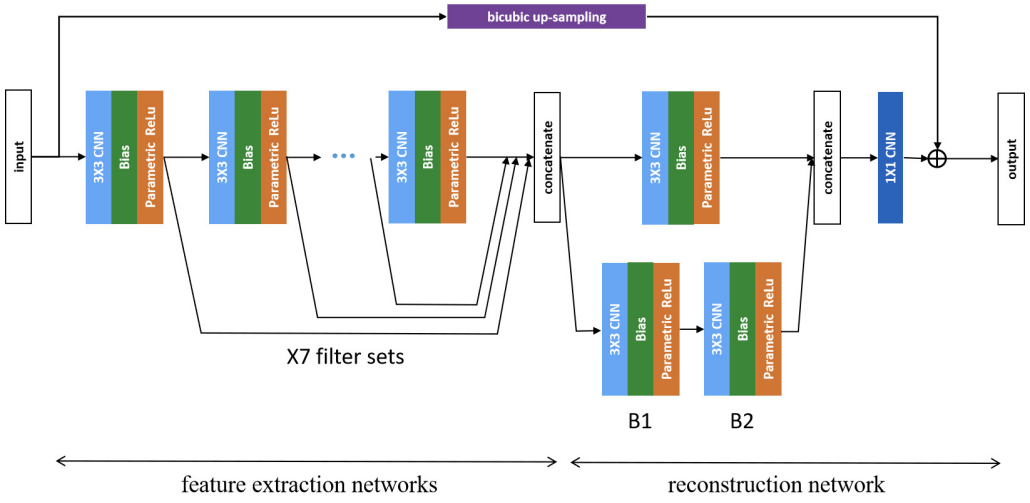

Fig. 1. Structure of DCSCN network

The output of DCSCN is $\hat{y} = r - x$. We adopt the mean squared error between the predicted output and the ground truth as the loss function.

3. THE PROPOSED METHOD

In this section, we describe the ideas for designing our network structure and the details for training the network.

3.1. Model structure

The proposed network of this paper is divided into two parts: the feature extraction network and the reconstruction network. The feature extraction network is divided into the feature extraction part and the feature fusion part. The feature extraction part adopts i identical convolutional layers. The feature fusion part is composed of a 1×1 CNN and a 3×3 CNN. The reconstruction network consists of an up-sample operator and a convolution layer (Fig. 2).

The output of CNN in the feature extraction network can be represented as:

$$F_i = \sigma\left(W_i[F_1, F_2, \ldots, F_{i-1}] + B_i\right), \tag{1}$$

where F_i is the output feature map of the i-th layer, W_i is the filter of the i-th layer, B_i is the bias, σ denotes activation function PReLU, and $[F_1, F_2, \ldots, F_{i-1}]$ is the input of layer i, which is the concatenation of the outputs of the previous layer.

The depth of the network in the SR task has deepened in the past few years. The structure of feature maps of different layers has different levels with different receptive fields, which is called hierarchical. SRCNN and VDSR only use the feature maps of the last layer of the network, which do not make good use of the information of the feature map of the middle convolution layer. The hierarchical feature maps can provide more clues for the reconstruction of the HR image, which is advantageous for obtaining better reconstruct accuracy. All feature maps of layers in our feature extraction part are concatenated as $[F_1, F_2, \ldots, F_i]$, i.e. output to the feature fusion part.

Because the dimension of the feature map $[F_1, F_2, \ldots, F_i]$ is too excessive, computational complexity is significantly increased. To solve this problem, we introduce the 1×1 CNN to adaptively control the information saved in F_{DF}, which can be given as:

$$F_{DF} = \sigma\left(W * [F_0, F_1, \ldots, F_i] + b\right), \tag{2}$$

where W is the G CNN filters with size $1 \times 1 \times (G \times i)$, and σ is the activation function of PReLU.

The convolution layer of 1×1 CNN integrates information from different channels of the feature map. Then the 3×3 convolution is used to extract feature maps from fused hierarchical feature maps. The output of the entire feature extraction network is expressed as F_{out}.

After the feature maps are extracted in the feature extraction network with LR spatial size, the sub-pixel convolutional neural network is adopted as the up-sample operator, which is expressed as $I^{UP} = F^L(F_{\text{out}}) = PS(W_L * f^{L-1}(F_{\text{out}}) + b_L)$. Finally, a 3×3 CNN is used to get the final HR image $I^{SR} = \sigma(W * I^{UP} + b)$.

3.2. Feature extraction network

Feature extraction is crucial for SR tasks. How to extract the low-level local features and preserve the useful texture information has been the focus of SR research.

The feature extraction operation usually uses the CNN cascade to obtain the feature maps of different receptive fields. The feature maps obtained by the different layers have different receptive fields and contain the image information with a different scale. Many traditional SR methods, such as VDSR, SRCNN and DRCN, are one-way structures which achieve good reconstruction accuracy. However, they do not make full use of the information contained in all the feature maps of the feature extraction network in the model.

MemNet [10] proposes the short-term memory to describe the phenomenon showing that a CNN layer is only affected by the previous connected layer directly in a single-path network. Some other methods, such as RED [16] and SRResNET [17], use a skip connection to connect the feature map of a CNN to a specific convolutional layer. MemNet [10] introduces restricted long-term memory to describe the phenomenon that one CNN could be affected by one particular previous CNN. Based on the above, MEMNET constructs the structure of the

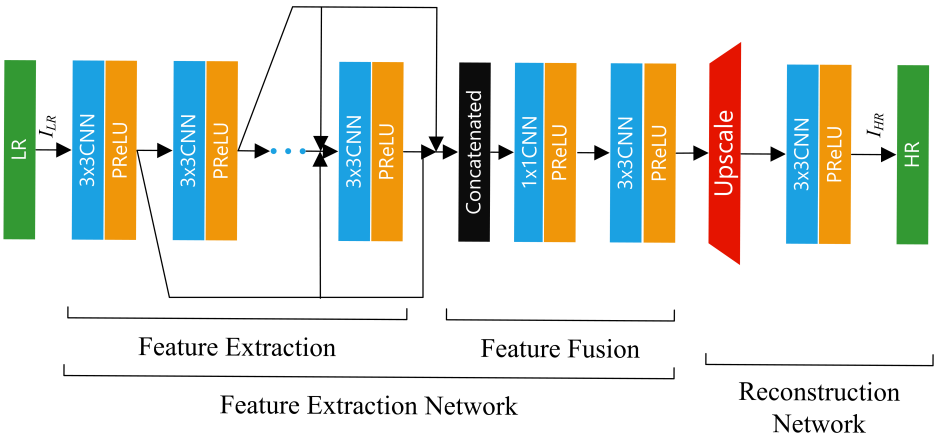

Fig. 2. Proposed network structure

memory block and densely connects it, which is described as long-term memory. The DCSCN adopts a similar structure to connect the feature maps of each CNN in the feature extraction network to the end of the feature extraction network and to concatenate them. Then the weights are adaptively learned to adjust the retention information of hierarchical feature maps. Such a structure enables the feature maps on different levels to be used in the reconstructive process. However, the CNN in the feature extraction network is still only affected by only one specific previous CNN. At the stage of feature extraction, the information contained in the hierarchical feature maps cannot be fully utilized to enhance the reconstructive effect.

The feature extraction network in our network adopts the idea of densely connected networks [18], which connects the feature map of one CNN in the feature extraction part to each subsequent CNN. It is beneficial to the forward propagation of the feature map and the backpropagation of the gradients. Meanwhile, the CNN in the feature extraction part can also utilize the information of the hierarchical feature maps with the hierarchical receptive filed. The dense connection enables the low-level feature map to become the input of the CNNs at later positions in the network. As a result, the input of the CNN itself contains the information of the low-level feature map. According to the above analysis, it is not necessary to decrease the number of the filter in the light of the gradual increase of model depth. The number of filters in the feature extraction part of our method is fixed to G.

Because all feature maps in the feature extraction part are concatenated, the dimension of feature maps is too large. If $[F_1, F_2, \ldots, F_i]$ is directly output to the reconstruction network, the calculation complexity is too heavy and real-time performance is poor. It is difficult to apply convergence in training. Considering that the computation of 1×1 CNN is only one-ninth of 3×3 CNN, we use the 1×1 convolution at the end of the feature extraction network to reduce the size of the feature maps from $G \times i$ to G. With just a small amount of information lost, it greatly reduces computational complexity and fuses the information in feature maps across channels. In addition, the 1×1 convolution can enhance the mapping ability of the entire network.

3.3. Reconstruction network

The up-sample operator in the classical model is mainly divided into the following three categories: 1. The LR image is directly input into the network following BI operation. The input image already has the same spatial size of the HR image. 2. The deconvolution layer is used as the up-sample operator [19]. Theoretically, using the specific deconvolution layer after training will get better results than using a prefixed up-sample operator. 3. We can use a large number of CNN in an LR image to get a large number of feature maps, and then the sub-pixel convolutional layer is used to change the spatial size and dimension of the feature maps.

In our method, we chose the sub-pixel convolutional neural network to be the reconstruction network.

Since 1×1 CNN has reduced the dimension of feature maps in the feature extraction network, the parallel structure is not indispensable. We chose a sub-pixel convolutional layer [20] to reconstruct i^{up}. In ESPCN [20], the output of the sub-pixel convolutional layer is directly used as the reconstruction result. In our method, a 3×3 convolutional layer is added. The dimension of i^{up} is C, which hardly increases computational complexity. The reconstructed network has better nonlinear mapping ability so that it can provide better reconstruction results.

3.4. Training

The deep learning network can be trained to get the map from LR to HR. The expected output of the network is as close as possible to the ground truth. In SR methods based on deep learning, both loss function and model structure have a significant influence on the SR task. In the training phase, the quality of the network structure and loss function will directly affect whether the model can converge to the expected mapping. Also, the computational complexity of the model will affect the real-time performance of the reconstruction.

A single RTX2080Ti graphics card (11GB memory) was used in training. Ubuntu 18.4 system, Pytorch 1.1.0, CUDA 10.0, and cuDNN 7.5.0 are all exploited as deep learning frameworks. The training dataset is defined as $\{x^{(i)}, y^{(i)}\}_{i=1}^{N}$, where N represents the number of training images. The process of training is to find optimal parameters θ so that the function $\hat{y} = f(x^i\theta)$ and y^i of the model are as close as possible. The definition of the loss function has a critical impact on the training model and the optimal solution of the model parameters. Intuitively, MSE can be an excellent loss function for obtaining high PSNR. Our method chooses MSE as the primary item of the loss function, which is given as:

$$l(\theta) = \frac{1}{2N} \sum_{i=1}^{N} \left\| y^{(i)} - \hat{y}^{(i)} \right\|^2. \tag{3}$$

To avoid over-fitting, the L2 norm of the CNN filters is further added to the loss function as a regularization term. The final loss function is given as:

$$L(\theta) = \sum_{i=1}^{N} \frac{1}{N} \left\| y^{(i)} - \hat{y}^{(i)} \right\|^2 + \beta \|\theta\|^2, \tag{4}$$

where β is constant.

4. EXPERIMENTS

BSD100 [21] is a data set provided by the University of California, Berkeley. It is mainly used in image segmentation and contour detection. Because of the complex content and rich scenes of images in the data sets, BSD100 is also often used in the test set of image SR reconstruction. To verify the validity of the model, we take SET5 [22], SET14 [23], BSD100, Urban100 [24] and DIV2K [25] dataset as the test image. We chose BI, Aplus [26], SelfExSR [27], SRCNN [4], LapSRN [28], DRCN [7], DRRN [29], VDSR [5], MemNet [10], TSCN [30], EDSR [12] and DBPN [6] as the comparison algorithm, and performed detailed comparative experiments with PSNR, SSIM and human visual effect.

4.1. Preprocessing

The recent DIV2K dataset [25] published by Timofte is an HR (2k resolution) image dataset for image restoration. The DIV2K contains 800 training images, 100 validation images, and 100 test images. In this paper, DIV2K is selected as the training dataset of the network.

Since the SR task is not sensitive to the direction of the training image, the training images are horizontally and vertically reversed and rotated 90 degrees to augment the dataset. Therefore, the training dataset in our method is preprocessed to contain 3200 images.

4.2. Implementation details

In our model, the filter size of all convolutional layers except for the 1×1 convolutional layer is set to 3×3. The depth i of the feature extraction part is set to 8. The number of convolution filters G is set to 64. The CNN with filter size 3×3 in the model is padded with 0 to keep the spatial size of the feature map unchanged. The up-sample operator uses subpixel convolutional neural network [20] to obtain the HR image from the fused feature map.

In the initialization of CNN, the bias term and the PReLU are set to zero. ADAM algorithm is used as the optimizer and the initial learning rate is set to 0.002. During the training process, the learning rate is divided by two while the loss is not descended in five continuous epochs. When the learning rate is lower than 0.00002, the entire training process is stopped.

A single RTX2080Ti graphics card (11GB memory) is used for the training. Ubuntu 18.4 system, Pytorch 1.1.0, CUDA 10.0 and cuDNN 7.5.0 are all exploited as deep learning frameworks.

4.3. Model convergence

Our deep SR model performs well in convergence, as shown in Fig. 3. In our model, the mean square error is chosen as the loss function. The PSNR increases rapidly in 10 epochs after training. Then, it fluctuates considerably between 10 and 40 epochs. We believe that this is due to the retention and abandonment of different levels of feature maps caused by the 1×1 convolution layer in the weight updating process. Overall, with the increase of epoch, PSNR continues to improve and remains stable until 50 epochs. This reflects the fast convergence performance.

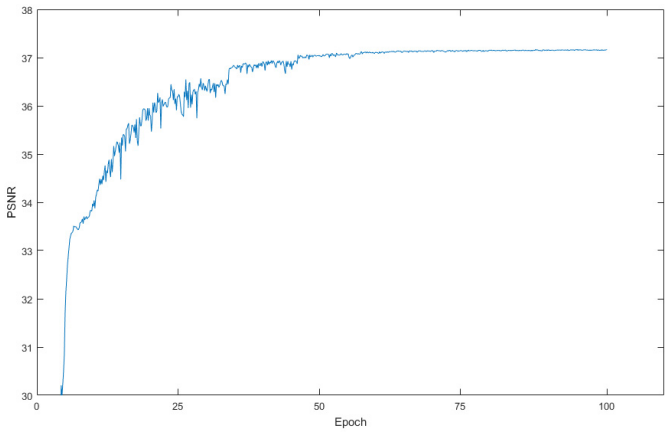

Fig. 3. Convergence of our method

4.4. Feature extraction and fusion

Given an input image, using our pre-training model, we can use different thresholds to get visual results and compare them with DCSCN. The experimental results are shown in Fig. 4.

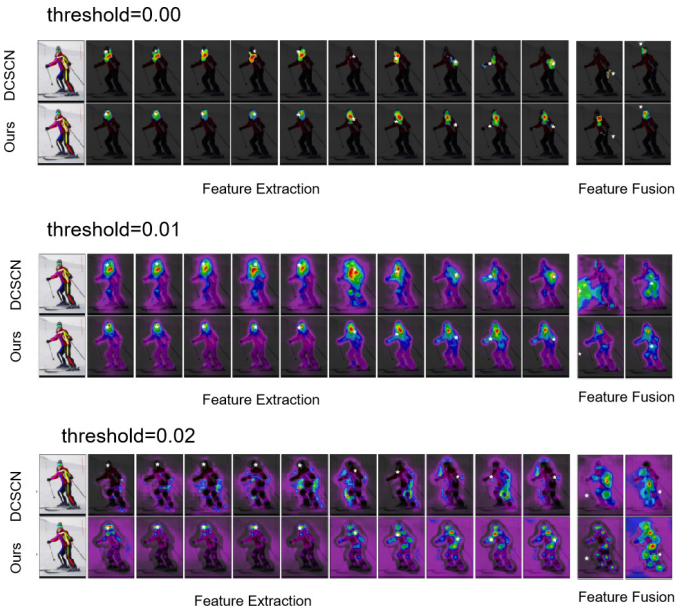

Fig. 4. Image sample in feature fusion and feature extraction

It can be seen that the feature extraction of the DCSCN algorithm exhibits a high level of randomness, resulting in the lack of some details of the characters identified in the final feature fusion. Although our model adopts i convolution layers with the same number as DCSCN in the feature extraction part, it can add the spatial correlation of predicted features through training, which can provide more details in the final feature fusion.

4.5. Comparison of objective criteria

The peak signal-to-noise ratio (PSNR) is an essential metric for quantitatively evaluating the accuracy of an SISR task, which is adopted as one of the indicators of reconstruction accuracy. The PSNR is defined as:

$$\mathrm{PSNR} = 10\log_{10}\frac{MN}{\left\|f-\hat{f}\right\|^2},$$

where M, N is the image spatial size, f is the ground truth, and $\hat{f}$ is the output reconstructive image provided by our algorithm. In the experiment, the BI, SRCNN, DRCN, VDSR, DCSCN and our algorithm are used to reconstruct the image on the same test set.

For the visual effect observed by the human eye, the luminance component is more critical. Therefore, in our experiments, the RGB image is converted to Ycbcr in the first place. Only the luminance component is reconstructed by the proposed algorithm. The blue-difference and the red-difference chroma components adopt BI for reconstruction. Finally, the reconstructed luminance component, blue-difference and red-difference chroma components are converted back to RGB space to obtain the reconstructed HR image.

Table 2 shows PSNR and SSIM results of the comparison experiments on Set5, set14, B100, urban100 and div2k data sets when the magnification scale is ×2, ×3, and ×4. Red and blue indicate the best and second-best performance, respectively. It can be seen from the table that our algorithm has achieved the best performance in most cases, especially when the scale is ×4. In a few cases, PSNR and SSIM of our model is slightly lower than that of EDSR.

Figure 5 shows the performance comparison between our algorithm and some SR models of different sizes. Those are implemented in the magnification factor of ×4 on Manga 109 [31]. As can be seen from the Figure, when compared with the existing algorithms EDSR, RDN and DBPN, our model has obvious advantages in PSNR, and the parameters of our model are not too many. Therefore, the results show that the size of our model is small but the effect provided is excellent.

Table 2

Comparison of the reconstruction effects of our method and several state-of-the-art SISR methods on Set5, Set14, B100, Urban100 and DIV2K. Red and blue indicate the best and second-best performance

Scale	Method	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM	DIV2k validation PSNR/SSIM
×2	BI	33.66 / 0.9299	30.24 / 0.8688	29.56 / 0.8431	26.88 / 0.8403	31.01 / 0.9393
	Aplus [26]	36.54 / 0.9544	32.28 / 0.9056	31.21 / 0.8863	29.20 / 0.8938	32.89 / 0.9570
	SelfExSR [27]	36.50 / 0.9536	32.22 / 0.9034	31.17 / 0.8853	29.52 / 0.8965	– / –
	SRCNN [4]	36.66 / 0.9542	32.45 / 0.9067	31.36 / 0.8879	29.51 / 0.8946	33.05 / 0.9581
	LapSRN [28]	37.44 / 0.9581	32.96 / 0.9117	31.78 / 0.8941	30.39 / 0.9093	– / –
	DRCN [7]	37.63 / 0.9588	33.04 / 0.9118	31.85 / 0.8942	30.75 / 0.9133	– / –
	DRRN [29]	37.74 / 0.9591	33.23 / 0.9136	32.05 / 0.8973	31.23 / 0.9188	– / –
	VDSR [5]	37.53 / 0.9587	33.03 / 0.9124	31.90 / 0.8960	30.76 / 0.9140	33.66 / 0.9625
	MemNet [10]	37.78 / 0.9597	33.28 / 0.9142	32.08 / 0.8978	31.31 / 0.9195	– / –
	TSCN [30]	37.88 / 0.9602	33.28/ 0.9147	32.09 / 0.8985	31.29 / 0.9198	– / –
	DBPN [6]	38.09 / 0.9600	33.85 / 0.9190	32.27 / 0.9000	32.55 / 0.9324	– / –
	EDSR [12]	38.11 / 0.9602	33.92 / 0.9195	32.32 / 0.9013	32.93 / 0.9351	– / –
	Ours	38.24/0.9611	33.99 / 0.9201	32.34 / 0.9014	32.94 / 0.9349	35.42 / 0.9718
×3	BI	30.39 / 0.8682	27.55 / 0.7742	27.21 / 0.7385	24.46 / 0.7349	28.22 / 0.8906
	Aplus [26]	32.58 / 0.9088	29.13 / 0.8188	28.29 / 0.7835	26.03 / 0.7973	29.50 / 0.9116
	SelfExSR [27]	32.64 / 0.9097	29.15 / 0.8196	28.29 / 0.7840	26.46 / 0.8090	– / –
	SRCNN [4]	32.75 / 0.9090	29.29 / 0.8215	28.41 / 0.7863	26.24 / 0.7991	29.64 / 0.9138
	DRCN [7]	33.82 / 0.9226	29.76 / 0.8311	28.80 / 0.7963	27.15/ 0.8276	– / –
	DRRN [29]	34.03 / 0.9244	29.96 / 0.8349	28.95 / 0.8004	27.53 / 0.8378	– / –
	VDSR [5]	33.66 / 0.9213	29.77 / 0.8314	28.82 / 0.7976	27.14 / 0.8279	30.09 / 0.9208
	MemNet [10]	34.09 / 0.9248	30.00 / 0.8350	28.96 / 0.8001	27.56 / 0.8376	– / –
	TSCN [30]	34.18 / 0.9256	29.99 / 0.8351	28.95 / 0.8012	27.46 / 0.8362	– / –
	EDSR [12]	34.65 / 0.9282	30.52 / 0.8462	29.25 / 0.8093	28.80 / 0.8653	– / –
	Ours	34.73/0.9292	30.51 / 0.8469	29.29 / 0.8104	28.86 / 0.8657	32.07 / 0.9212
×4	BI	28.42 / 0.8104	26.00 / 0.7027	25.96 / 0.6675	23.14 / 0.6577	26.66 / 0.8521
	Aplus [26]	30.28 / 0.8603	27.32/ 0.7491	26.82 / 0.7087	24.32 / 0.7183	27.70 / 0.8736
	SelfExSR [27]	30.30 / 0.8620	27.38/ 0.7516	26.84 / 0.7106	24.80 / 0.7377	– / –
	SRCNN [4]	30.48 / 0.8628	27.50 / 0.7513	26.90 / 0.7103	24.52 / 0.7226	27.78 / 0.8753
	LapSRN [28]	31.52/ 0.8854	28.08 / 0.7687	27.31 / 0.7255	25.21 / 0.7545	– / –
	DRCN [7]	31.53 / 0.8854	28.03 / 0.7673	27.24 / 0.7233	25.14 / 0.7511	– / –
	DRRN [29]	31.68 / 0.8888	28.21 / 0.7721	27.38 / 0.7284	25.44 / 0.7638	– / –
	VDSR [5]	31.35/ 0.8838	28.02 / 0.7678	27.29 / 0.7252	25.18/ 0.7525	28.17 / 0.8841
	MemNet [10]	31.74 / 0.8893	28.26 / 0.7723	27.40 / 0.7281	25.50 / 0.7630	– / –
	TSCN [30]	31.82/ 0.8907	28.28 / 0.7734	27.42 / 0.7301	25.44 / 0.7644	– / –
	DBPN [6]	32.47 / 0.8980	28.82 / 0.7860	27.72 / 0.7400	26.38 / 0.7946	– / –
	EDSR [12]	32.46 / 0.8968	28.80 / 0.7876	27.71 / 0.7420	26.64 / 0.803	– / –
	Ours	32.52 / 0.8975	28.85 / 0.7880	27.79 / 0.7431	26.58 / 0.8044	31.52 / 0.8925

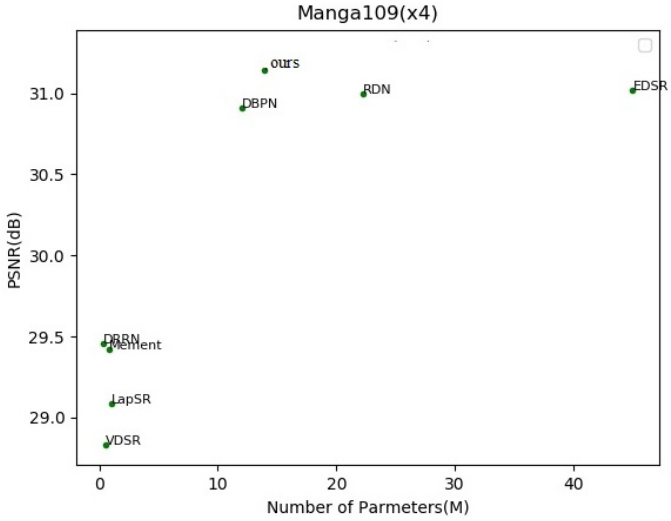

Fig. 5. Performance of different size models in PSNR with magnification factor of $\times 4$ on manga 109

4.6. Comparison of visual effect

Figures 6 and 7 present the comparison experiments of images named "baby" and "butterfly" in Set5 with $\times 2$ magnification factor. The first column is the original image. From the second column to the fifth column, there are the reconstructive images made by Bicubic, SRCNN, DRCN, VDSR, TSCN and our algorithm, respectively. The first row is the images in HR spatial size. The second row is the images of the zoomed local area. By enlarging some parts of the image, more texture details can be observed.

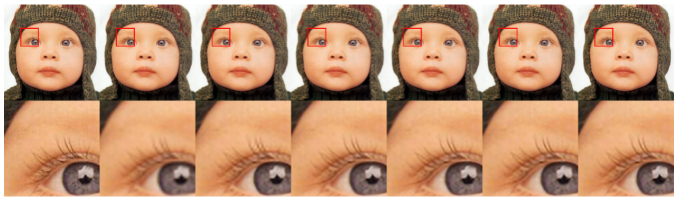

Fig. 6. SR results of "baby" with magnification factor of $\times 4$

Fig. 7. SR results of "butterfly" with magnification factor of $\times 4$

From the "baby" image, we can find that our algorithm provides competitive accuracy as compared with some state-of-the-art algorithms such as SRCNN, DRCN, VDSR and TSCN. Bicubic is a less accurate method. It is almost impossible to recognize the eyelash texture in the visual observation from the image reconstructed by Bicubic. From the "butterfly" image, the black and white textures of the butterfly's wing shown by our method are clear and sharp. There is no essential difference from the ground truth.

Figures 8 and 9 present the comparison experiments of "Lenna" and "comic" images in Set 14 with the magnification factor of $\times 4$. Through the figures, in the complex texture areas such as hair and so on, our algorithm provides a good reconstruction effect, which proves that the idea of the dense connection connecting low-level feature maps to the end of the feature extraction part to preserve detailed information is effective.

Fig. 8. SR results of "Lenna" with the magnification factor of $\times 4$

Fig. 9. SR results of "comic" with the magnification factor of $\times 4$

4.7. Analysis and discussion

From the detailed experiments, we can see that the effect of our model is excellent, which is mainly due to the following three points.

- The output of each convolution layer in the feature extraction network is connected to each subsequent convolution layer, which makes full use of the feature map of different levels in the hierarchical feature extraction part.
- Feature map fusion: At the end of the feature extraction network, a 1×1 CNN is used to integrate other channel information. Since the input of the 1×1 CNN is the concatenation of all the feature maps of all convolution layers in the feature extraction network, the most important task of the cross-channel is to integrate the feature maps of different levels with a different receptive field. Then, we use a 3×3 CNN to extract the feature map from the fused feature map.
- Sub-pixel convolution is used as the up-sampling operator to reconstruct the HR image.

5. CONCLUSION

In this paper, we constructed an improved single image SR algorithm based on deep learning. Firstly, our feature extraction network can make full use of the feature map of different levels in the hierarchical feature extraction part. Then, the model can integrate the feature maps of different levels with a different

receptive field. Finally, in the reconstruction network, the sub-pixel convolutional neural network is adopted to reduce computational complexity, which provides better reconstructive accuracy than some state-of-the-art methods. Detailed experiments show that our model outperforms DRCN, VDSR, TSCN and other classical networks in both PSNR and SSIM. In addition, our model performs better on human visual perception.

Compared with the DCSCN model, our model is more miniaturized and efficient, but it is less generalized for some scenes. In addition, our model can only be reconstructed at a specific magnification, which limits its application.

Our future work will focus on real-world image degradation models and image degradation models for special scenes. In addition, we will design a reconstruction model that can achieve any multiple, so as to further expand the application scope of SR technology.

ACKNOWLEDGEMENTS

This research was supported by the National Natural Science Foundation of China (61573182), and by the Fundamental Research Fund for Central Universities (NS2020025).

REFERENCES

[1] T. Karras, T. Aila, and S. Laine, "Progressive growing of gans for improved quality, stability, and variation", arXiv preprint arXiv:1710.10196, 2017.

[2] W. Shi, J. Caballero, and C. Ledig, "Cardiac image super-resolution with global correspondence using multi-atlas patchmatch", in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2013, pp. 9–16.

[3] X. Yang, D. Liu, and D. Zhou, "Super-resolution reconstruction of face images based on pre-amplification non-negative restricted neighborhood embedding", *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 66, no. 6, pp. 899–905, 2018.

[4] C. Dong, C.C. Loy, and K. He, "Learning a deep convolutional network for image super-resolution", in *European conference on computer vision (ECCV)*, 2014, pp. 184–199.

[5] J. Kim, J.K. Lee, and K.M. Lee, "Accurate image super-resolution using very deep convolutional networks", in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 1646–1654.

[6] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for single image super-resolution", arXiv preprint arXiv: 1904.05677, 2019.

[7] J. Kim, J.K. Lee, and K.M. Lee, "Deeply-recursive convolutional network for image super-resolution", in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 1637–1645.

[8] Y. Tai, J. Yang, and X. Liu, "Memnet: A persistent memory network for image restoration", in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2017, pp. 4539–4547.

[9] Y. Zhang, Y. Tian, and Y. Kong, "Residual dense network for image super-resolution", in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018, pp. 2472–2481.

[10] J. Yamanaka, S. Kuwashima, and T. Kurita, "Fast and accurate image super resolution by deep CNN with skip connection and network in network", in *International Conference on Neural Information Processing (ICONIP)*, 2017, pp. 217–225.

[11] L. Chen, Q. Kou, and D. Cheng, "Content-guided deep residual network for single image super-resolution", *Optik*, vol. 202, pp. 163678, 2020.

[12] B. Lim, S. Son, and H. Kim, "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.

[13] X. Yang, X. Li, and Z. Li, "Image super-resolution based on deep neural network of multiple attention mechanism", *J. Vis. Commun. Image Represent.*, vol. 75, p. 103019, 2021.

[14] L. Chen, L. Guo, and D. Cheng, "A lightweight network with bidirectional constraints for single image super-resolution", *Optik*, vol. 239, p. 166818, 2021.

[15] X. Yang, Y. Guo, and Z. Li, "Image super-resolution network based on a multi-branch attention mechanism", *Signal Image Video Process.*, vol. 15, pp. 1–9, 2021.

[16] X.J. Mao, C. Shen, and Y.B. Yang, "Image restoration using convolutional auto-encoders with symmetric skip connections", arXiv preprint arXiv:1606.08921, 2016.

[17] C. Ledig, L. Theis, and F. Huszar, "Photo-realistic single image super-resolution using a generative adversarial network", in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017, pp. 4681–4690.

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K.Q. Weinberger, "Densely connected convolutional networks", in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017, pp. 4700–4708.

[19] W. Shi, J. Caballero, and L. Theis, "Is the deconvolution layer the same as a convolutional layer?", arXiv preprint arXiv: 1609.07009, 2016.

[20] W. Shi, J. Caballero, and F. Huszár, "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 (CVPR)*, 2016, pp. 1874–1883.

[21] D. Martin, C. Fowlkes, and D. Tal, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics", in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001,* 2001, pp. 416–423.

[22] M. Bevilacqua, A. Roumy, and C. Guillemot, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding", in *Proceedings of the 23rd British Machine Vision Conference (BMVC)*, 2012, pp. 135.1–135.10.

[23] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations", in *International Conference on Curves and Surfaces*, 2010, pp. 711–730.

[24] J.B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars", in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2015, pp. 5197–5206.

[25] R. Timofte, E. Agustsson, and L.V. Gool, "Ntire 2017 challenge on single image super-resolution: Methods and results", in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 114–125.

[26] R. Timofte, V.D. Smet, and L.V. Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution", in *Asian conference on computer vision (ACCV)*, 2014, pp. 111–126.

[27] J.B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars", in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2015, pp. 5197–5206.

[28] W.S. Lai, J.B. Huang, and N. Ahuja, "Deep laplacian pyramid networks for fast and accurate super-resolution", in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017, pp. 624–632.

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

[30] Z. Hui, X. Wang, and X. Gao, "Two-stage convolutional network for image super-resolution", in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 2670–2675.

[31] Y. Matsui *et al.*, "Sketch-based manga retrieval using manga109 dataset", *Multimed Tools Appl*, vol. 76, no. 20, pp. 21811–21838, 2017.