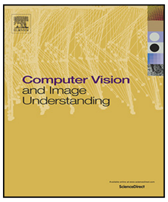
Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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



Residual network with detail perception loss for single image super-resolution

Zhijie Wen [a], Jiawei Guan [a], Tieyong Zeng [b], Ying Li [c,*]

[a] Department of Mathematics, School of Science, Shanghai University, Shanghai, 200444, China
[b] Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong, 999077, China
[c] School of Computer Engineering and Sciences, Shanghai University, Shanghai, 200444, China

ARTICLE INFO

MSC:
41A05
41A10
65D05
65D17

Keywords:
Detail perception loss
Multi-layer perceptron
Single image super-resolution

ABSTRACT

Recently, deep convolutional neural networks have demonstrated high-quality reconstruction for single image super-resolution. In this study, we present a network by using residual blocks with cascading simple blocks to improve the image resolution. Cascading simple blocks with a multi-layer perceptron are conducive to extract features and approximate a complex mapping with fewer parameters. Skip connections can help to alleviate the vanishing-gradient problem of deep networks. In addition, our network contains two pathways. One is to predict the high frequency information of the high resolution image and the other is to predict the low frequency information of the high resolution image. Then the information of two pathways is fused, and pixel-shuffle is used for upsampling. Moreover, to capture texture details of images, we introduce a novel loss function called detail perception loss, which is used to measure the difference of the wavelet coefficients from the reconstructed image and ground truth. By reducing detail perception loss, texture details of the reconstructed image are becoming more similar with texture details of ground truth. Extensive quantitative and qualitative experiments on four benchmark datasets show that our method achieves superior performance over typical single image super-resolution methods.

1. Introduction

Single image super-resolution (SISR) is a classical and ill-posed inverse problem, which aims to reconstruct a high resolution (HR) image from its corresponding low resolution (LR) image. SISR has many wide applications where more image details and textures are greatly desired, e.g. face hallucination (Huang et al., 2017), medical image (Peled and Yeshurun, 2001), satellite image (Thornton et al., 2006), security and surveillance (Zhang et al., 2010).

Recently, convolutional neural network has gained wide attention in computer vision community. It achieves state-of-the-art performance in many applications, e.g. image classification, image segmentation and object recognition. Meanwhile, many researchers also have introduced convolutional neural network into SISR problem. Dong et al. (2016a) first proposed a super-resolution convolutional neural network, which models a nonlinear mapping from LR images to HR images in an end-to-end manner. Their method does not require any specialized knowledge and shows good results. Since then, many methods based on deep learning have been developed.

Although existing methods show good results, there are some issues. First, most models optimize the network by a pixel-wise mean squared error loss (l_2 loss) to make the output pixel-wise closer to the HR image. However, l_2 loss tends to generate a blurry prediction, which lacks some textural details. The final outputs are overly-smooth and unlike human visual perception for natural images. Second, as an ill-posed inverse problem, there are not enough conditions to constrain the final solution in training phase, especially the intermediate layers. It results in the very big solution space. Third, to achieve better performance, many networks become deeper and wider, which leads to larger computational cost and memory consumption. In addition, from a scientific standpoint, this is baseless. The development of better model is reduced to trial-and-error.

As shown in Fig. 1, we upsample the low resolution image by bicubic interpolation. The residual error is the difference between ground truth and the bicubic interpolated image. It can be found that the original image loses many texture details, also called the high frequency information. The residual error is similar to the high frequency information of the image, as shown in Fig. 2, which is obtained by Haar wavelet transform of HR image. In a word, the high frequency information prediction is important in the SISR problem.

In this paper, we propose a residual network with cascading simple blocks, abbreviated CSBRN. Meanwhile, our network has two pathways. One of them is used to pass the low frequency information. The

* Corresponding author.
E-mail address: yinglotus@shu.edu.cn (Y. Li).

https://doi.org/10.1016/j.cviu.2020.103007
Received 26 December 2019; Received in revised form 8 April 2020; Accepted 31 May 2020
Available online 3 June 2020
1077-3142/© 2020 Elsevier Inc. All rights reserved.

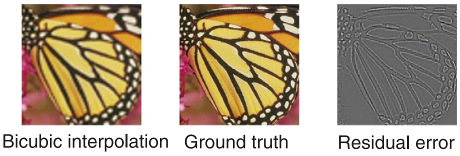


Fig. 1. The comparison of the bicubic interpolation image and ground truth.

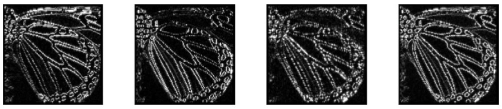


Fig. 2. From left to right: the horizontal information, the vertical information, the diagonal information and sum of them.

other one is used to predict the high frequency information of HR images. Moreover, to enable the network to capture texture details, we take the error of the wavelet coefficients of the reconstructed image and the wavelet coefficients of ground truth as the loss function. Main contributions of our work can be summarized as follows:

(1) Cascading simple blocks are introduced into residual block, which can approximate a more complex function with fewer parameters to predict details of images and help to interact cross channel information.

(2) A novel loss function is proposed, called detail perception loss, which is used to measure the difference of the wavelet coefficients from the reconstructed image and the wavelet coefficients of ground truth. It can be used in other SR networks to improve performance.

(3) The proposed network is divided into two pathways, which are used for low frequency information transfer and high frequency information prediction respectively.

(4) The proposed method achieves superior performance on benchmark datasets.

2. Related work

Recently, many SISR methods based on deep learning have been proposed. Different from traditional interpolation methods, they achieve very good results. Especially, convolutional neural network models a nonlinear mapping from LR image to HR image effectively. At present, the problem of single image super-resolution mainly focuses on the study of upsampling method, network architecture and loss function.

In the SISR problem, upsampling operation is an essential process. Dong et al. (2016a) first proposed a preprocessed method by interpolating the input image. However, it results in increasing computation cost because all convolution operations are computed on the high resolution space. And the interpolation operation does not introduce new information. Then, they (Dong et al., 2016b) adopted a deconvolutional layer at the end of the network in FSRCNN, which can reduce computation cost. However, the deconvolution operation is easy to introduce checkerboard artifacts (Odena et al., 2016). Different from the first two methods, Shi et al. (2016) adopted a novel upsampling method, called pixel shuffle, which is designed specifically for SR problem without extra parameters.

In addition to the study of upsampling operation, there are also a large amount of study about network architecture. Kim et al. (2016a) first introduced skip connection into VDSR, which is used to address vanishing gradient problem and helps with training the deeper and wider network. Furthermore, it can increase information flow in the network. Thereafter, skip connection is widely used in super-resolution convolutional neural networks, such as SRDenseNet (Tong et al., 2017), DRRN (Tai et al., 2017a), EDSR (Lim et al., 2017), WDSR (Yu et al., 2018), DBPN (Haris et al., 2018), IND (Hui et al., 2018), LFFN (Yang et al., 2018). However, as the network becomes deeper, the parameters of the model increase significantly. To address this drawback, Kim et al. (2016b) utilized a recursive layer for reducing the model parameters. Because more recursions are performed in the recursive layer, the mapping is highly nonlinear. Tai et al. (2017a) introduced recursive layers and residual blocks into the network called DRRN, which are used to control the model parameters and mitigate the difficulty of training deep networks.

Moreover, there are many other studies. Huang et al. (2017) proposed a wavelet-based approach, which transforms single image super resolution to wavelet coefficients prediction task in deep learning framework. Tian et al. (2019) proposed a loss function combining the statistic loss with semantic priors and quality assessment loss. Li et al. (2018) proposed a multi-scale residual network to fully exploit the image features, which introduces convolution kernels of different sizes to adaptively detect the image features in different scales. Bhowmik et al. (2018) constructed a dynamic convolutional network to learn the relation between the Gaussian and Laplacian pyramid for predicting the detail. Lai et al. (2017) proposed a convolutional network within a Laplacian pyramid framework, which progressively predicts high-frequency residual in a coarse-to-fine manner. Cheong and Park (2017) jointly used external and internal examples on deep CNN framework. Zhang and Lu (2011) proposed a Hopfield neural network to produce high-resolution images, which takes into consideration point spread function blurring and additive noise. Zhong et al. (2018) designed a new up-sampling module which uses IDWT to change the size of feature maps. The network jointly learns all sub-band coefficients depending on the edge feature property. Fang et al. (2020) proposed a soft-edge assisted network to reconstruct the high-quality SR image with the help of image soft-edge.

3. Methodology

3.1. Network architecture

As shown in Fig. 3, our network mainly consists of two pathways: information transfer pathway (or low frequency information pathway) and high frequency information pathway. We denote $I^{LR} \in \mathbb{R}^{h \times w \times 3}$ as the input LR image and $I^{HR} \in \mathbb{R}^{sh \times sw \times 3}$ as the output HR image, where h and w are the height and the width of the input image. s is the upsampling scale. The information transfer pathway passes low frequency information from low resolution images. The high frequency information pathway predicts high frequency information from low resolution images by a complex mapping. They do not change the size (h, w) of the input image. However, the number of features has increased. Finally, we add them together and upsample by pixel shuffle.

3.2. Residual block with Cascading simple blocks

The Low resolution image is fed into the high frequency information pathway. It consists of convolutional layers and residual blocks with cascading simple blocks. The first convolutional layer is used to increase the number of features. And these features will be fed into residual blocks with cascading simple blocks. The final convolutional layer is used to limit the number of features for pixel shuffle. It outputs $3s^2$ features. And the size of them is $h \times w$. Residual blocks with a complex mapping are used to predict high frequency information. This process can be written as:

$$F_1 = f_1(I^{LR}) \in \mathbb{R}^{h \times w \times N} \tag{1}$$

$$F_2 = f_2(F_1) \in \mathbb{R}^{h \times w \times N} \tag{2}$$

$$F_{HF} = f_3(F_2) \in \mathbb{R}^{h \times w \times 3s^2} \tag{3}$$

where N is the number of channels for residual blocks. f_1 and f_3 are the mapping of a convolutional layer. f_2 is the mapping of multiple residual blocks.

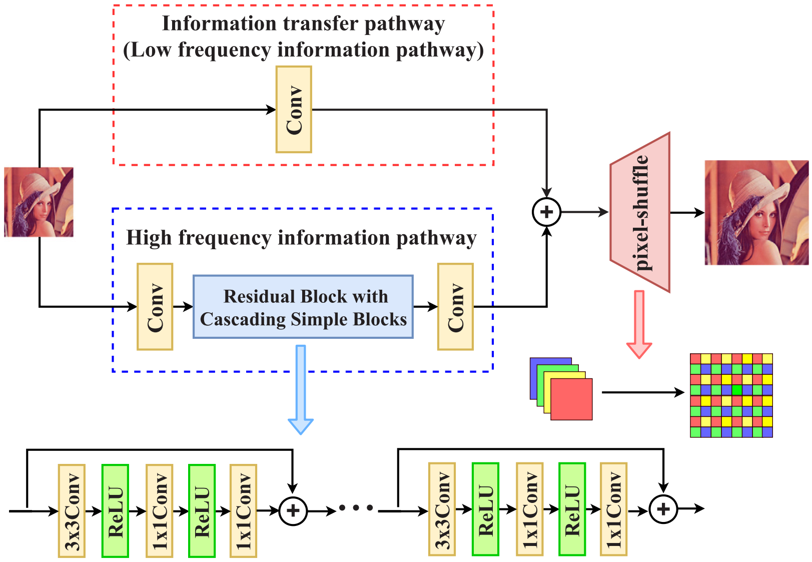

Information transfer pathway
(Low frequency information pathway)
Conv
High frequency information pathway
Conv
Residual Block with Cascading Simple Blocks
Conv
pixel-shuffle
3x3Conv
ReLU
1x1Conv
ReLU
1x1Conv
3x3Conv
ReLU
1x1Conv
ReLU
1x1Conv

Fig. 3. The architecture of the proposed network.

Recently, the residual block (Tai et al., 2017a; Lim et al., 2017; Yu et al., 2018; Ledig et al., 2017) shows excellent performance in image super-resolution. In this paper, we propose a residual block with cascading simple blocks, which is shown in Fig. 3. Reference to the idea of Network in Network (Lin et al., 2013), we introduce the micro neural network with a multi-layer perceptron, which is a potent function approximator. It can approximate a complex function with fewer parameters and help to interact cross channel information. It is equivalent to a convolutional layer with 1 × 1 convolution kernel. And 3 × 3 convolutional layers can enlarge the field of perception and fuse of surrounding information. Thus, in our residual blocks, we cascade 3 × 3 convolutional layers and 1 × 1 convolutional layers. If the larger receptive field is needed, 3 × 3 convolutional layers should be increased. If it is necessary to enhance the nonlinearity and reduce parameters as much as possible, 1 × 1 convolutional layers should be increased. We can adjust the number of 3 × 3 and 1 × 1 convolutional layers according to the task.

3.3. Information transfer pathway and pixel shuffle

In VDSR (Kim et al., 2016a), the network uses a skip connection, which adds the input and output image together. This network predicts the residual image between the LR and HR image. Similarly, our network uses an information transfer pathway. However, to add to high frequency information pathway, we need to use a convolutional layer to increase the number of feature channels. The Low resolution image is fed into the information transfer pathway. Then we use it to pass low frequency information. It takes $3 \times H \times W$ low resolution image as an input and outputs $3s^2 \times H \times W$ features. It is formulated as follow:

$$F_{LF} = f_4(I^{LR}) \in \mathbb{R}^{h \times w \times 3s^2} \tag{4}$$

Then, we add low frequency information and high frequency information. Finally, the features are upsampled by pixel shuffle (Shi et al., 2016). This process can be written as:

$$F = F_{LF} + F_{HF} \in \mathbb{R}^{h \times w \times 3s^2} \tag{5}$$

$$I^{HR} = PS(F) \tag{6}$$

where PS is the pixel shuffle operation.

3.4. Detail perception loss function

In this paper, we propose a novel loss function called detail perception loss. Unlike L_1 and L_2 loss functions, which focus on pixel-wise similarity, our loss function is concerned with texture details of real images at the same time. The network trained by L_1 and L_2 loss function sometimes has some differences and errors between the super-resolution image and ground truth in detail features. We hope that loss function designed by us can learn detail features and make up for the deficiency of L_1 loss function.

If the reconstructed image is similar to the real image, they should have similar detail features. We try to use different frequency bands of the image through wavelet transform to describe detail features. Features are extracted through hand-crafted filters. Through wavelet decomposition, we can get the horizontal information, the vertical information and the diagonal information of the image. Such information can be used as the detail feature of the image. Our detail perception loss is used to measure the difference in detail features. It is formulated as follow:

$$L_{detail}(\Theta) = \frac{1}{N} \sum_{i=1}^{N} \|T(y^{(i)}) - T(\hat{y}^{(i)})\|_1 \tag{7}$$

where $\hat{y}^{(i)}$ is the reconstructed image of the network. $T(\cdot)$ is wavelet transform function. In this paper, we choose the simplest wavelet, Haar wavelet, for it is enough to depict texture details of images. Furthermore, we can change the form of transformation according to the needs. $T(\cdot)$ can be any function transformation or feature extractor. Its purpose is to enable the model to learn specific features in the training process. As the prior knowledge, it can be combined with traditional image processing methods.

Although detail perception loss function can depict texture details of the image, the color information of the image cannot be obtained. Therefore, L_1 loss is still needed, which is used to keep the color information of the image. Therefore, our loss function is formulated as follow:

$$L_{total} = \alpha L_1 + (1-\alpha) L_{detail} + \beta \|\theta\|_2^2 \tag{8}$$

where α is the balance parameter, β is weight decay.

4. Experiments

In this section, we evaluate the performance of the proposed method on several typical datasets. First, datasets used for training and testing are described. Then, we introduce implementation and training details. Finally, the proposed method is compared with other SISR methods.

Table 1
Quantitative evaluation of typical SR algorithms: average PSNR and SSIM for scale factors ×2, ×3 and ×4. Red text indicates the best and blue text indicates the second best performance.

Models	Mag.	Set5		Set14		BSDS100		Urban100		Avg.	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	×2	33.66	0.9299	30.24	0.8688	29.56	0.8431	26.88	0.8403	28.47	0.8454
SRCNN (Dong et al., 2016a)		36.66	0.9542	32.42	0.9063	31.36	0.8879	29.50	0.8946	30.70	0.8936
VDSR (Kim et al., 2016a)		37.53	0.9587	33.03	0.9124	31.90	0.8960	30.76	0.9140	31.58	0.9067
LapSRN (Lai et al., 2017)		37.52	0.9591	32.99	0.9124	31.80	0.8952	30.41	0.9103	31.37	0.9047
DRCN (Kim et al., 2016b)		37.63	0.9588	33.04	0.9118	31.85	0.8942	30.75	0.9133	31.56	0.9055
DRFN (Yang et al., 2019)		37.71	0.9595	33.29	0.9142	32.02	0.8979	31.08	0.9179	31.80	0.9095
DRRN (Tai et al., 2017a)		37.74	0.9591	33.23	0.9136	32.05	0.8973	31.23	0.9188	31.88	0.9096
MemNet (Tai et al., 2017b)		37.78	0.9597	33.28	0.9142	32.08	0.8978	31.31	0.9195	31.94	0.9102
IDN (Hui et al., 2018)		37.83	0.9600	33.30	0.9148	32.08	0.8985	31.27	0.9196	31.92	0.9106
MSRN (Li et al., 2018)		38.08	0.9605	33.74	0.9170	32.23	**0.9013**	**32.22**	0.9326	**32.46**	0.9179
CSBRN		**38.11**	**0.9611**	**33.81**	**0.9188**	**32.28**	0.9008	32.12	**0.9357**	32.44	**0.9193**
CSBRN+		**38.17**	**0.9613**	**33.86**	**0.9199**	**32.31**	**0.9011**	**32.36**	**0.9379**	**32.57**	**0.9205**
Bicubic	×3	30.39	0.8682	27.55	0.7742	27.21	0.7385	24.46	0.7349	26.05	0.7421
SRCNN (Dong et al., 2016a)		32.75	0.9090	29.28	0.8209	28.41	0.7863	26.24	0.7989	27.57	0.7971
VDSR (Kim et al., 2016a)		33.66	0.9213	29.77	0.8314	28.82	0.7976	27.14	0.8279	28.22	0.8164
LapSRN (Lai et al., 2017)		33.81	0.9220	29.79	0.8325	28.82	0.7980	27.07	0.8275	28.20	0.8165
DRCN (Kim et al., 2016b)		33.82	0.9226	29.76	0.8311	28.80	0.7963	27.15	0.8276	28.22	0.8157
DRFN (Yang et al., 2019)		34.01	0.9234	30.06	0.8366	28.93	0.8010	27.43	0.8359	28.43	0.8220
DRRN (Tai et al., 2017a)		34.03	0.9244	29.96	0.8349	28.95	0.8004	27.53	0.8378	28.48	0.8225
MemNet (Tai et al., 2017b)		34.09	0.9248	30.00	0.8350	28.96	0.8001	27.56	0.8376	28.50	0.8223
IDN (Hui et al., 2018)		34.11	0.9253	29.99	0.8354	28.95	0.8013	27.42	0.8359	28.43	0.8221
MSRN (Li et al., 2018)		34.38	0.9262	30.34	0.8395	29.08	0.8041	**28.08**	**0.8554**	**28.82**	0.8326
CSBRN		**34.44**	**0.9287**	**30.43**	**0.8449**	**29.22**	**0.8087**	27.82	0.8536	28.78	**0.8343**
CSBRN+		**34.53**	**0.9292**	**30.50**	**0.8463**	**29.26**	**0.8093**	**28.01**	**0.8568**	**28.89**	**0.8361**
Bicubic	×4	28.42	0.8104	26.00	0.7027	25.96	0.6675	23.14	0.6577	24.73	0.6685
SRCNN (Dong et al., 2016a)		30.48	0.8628	27.49	0.7503	26.90	0.7101	24.52	0.7221	25.93	0.7216
VDSR (Kim et al., 2016a)		31.35	0.8838	28.01	0.7674	27.29	0.7251	25.18	0.7524	26.47	0.7439
LapSRN (Lai et al., 2017)		31.54	0.8852	28.09	0.7700	27.32	0.7275	25.21	0.7562	26.50	0.7469
DRCN (Kim et al., 2016b)		31.53	0.8854	28.02	0.7670	27.23	0.7233	25.14	0.7510	26.42	0.7424
DRFN (Yang et al., 2019)		31.55	0.8861	28.30	0.7737	27.39	0.7293	25.45	0.7629	26.66	0.7511
DRRN (Tai et al., 2017a)		31.68	0.8888	28.21	0.7721	27.38	0.7284	25.44	0.7638	26.65	0.7510
MemNet (Tai et al., 2017b)		31.74	0.8893	28.26	0.7723	27.40	0.7281	25.50	0.7630	26.69	0.7505
IDN (Hui et al., 2018)		31.82	0.8903	28.25	0.7730	27.41	0.7297	25.41	0.7632	26.65	0.7514
MSRN (Li et al., 2018)		32.07	0.8903	28.60	0.7751	27.52	0.7273	**26.04**	**0.7896**	**27.01**	**0.7625**
CSBRN		**32.20**	**0.8965**	**28.61**	**0.7832**	**27.64**	**0.7382**	25.38	0.7677	26.77	0.7582
CSBRN+		**32.30**	**0.8975**	**28.70**	**0.7848**	**27.67**	**0.7390**	**25.51**	**0.7705**	**26.86**	**0.7599**

4.1. Datasets for training and testing

We train all networks using images from DIV2K (Timofte et al., 2017), which is large enough and contains many high-resolution images. The DIV2K dataset consists of 800 training images, 100 validation images and 100 test images. We use 800 training images for training and 10 validation images for validation during training. The proposed method is evaluated on four widely used benchmark datasets: Set5 (Bevilacqua et al., 2012), Set14 (Zeyde et al., 2010), BSDS100 (Martin et al., 2001), Urban100 (Huang et al., 2015).

4.2. Implementation and training details

For preparing the input LR images, we downsample the original HR image with upscaling factor m ($m = 2, 3, 4$) by using the bicubic downsampling to generate the corresponding LR image. We crop 96 × 96 RGB patches from HR images. And according to the upscaling factor m, the corresponding LR image is divided into sub-images with size $\frac{96}{m} \times \frac{96}{m}$. The mini-batch size is set to 16. Adam optimizer (Kingma and Ba, 2014) is used with $\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The learning rate is initially set to 10^{-4} and halved at every 2×10^5 iterations. We conduct all experiments using Tensorflow. In this paper, we use two 3 × 3 convolutional layers and two 1 × 1 convolutional layers in the residual block.

4.3. Comparison with other SR methods

Performance comparison. We compare the result of the proposed method with the results of other SR methods, including Bicubic, SRCNN (Dong et al., 2016a), VDSR (Kim et al., 2016a), LapSRN (Lai et al., 2017), DRCN (Kim et al., 2016b), DRFN (Yang et al., 2019), DRRN (Tai et al., 2017a), MemNet (Tai et al., 2017b), IDN (Hui et al., 2018), MSRN (Li et al., 2018). Furthermore, we introduce the self-ensemble strategy to improve our method and denote the self-ensemble version as CSBRN+. In this paper, we cascade two 3 × 3 convolutional layers and two 1 × 1 convolutional layers. We set the number of blocks to 32 and the width of network to 128. Table 1 shows the average PSNR and SSIM on benchmark datasets. For comparison, we measure PSNR and

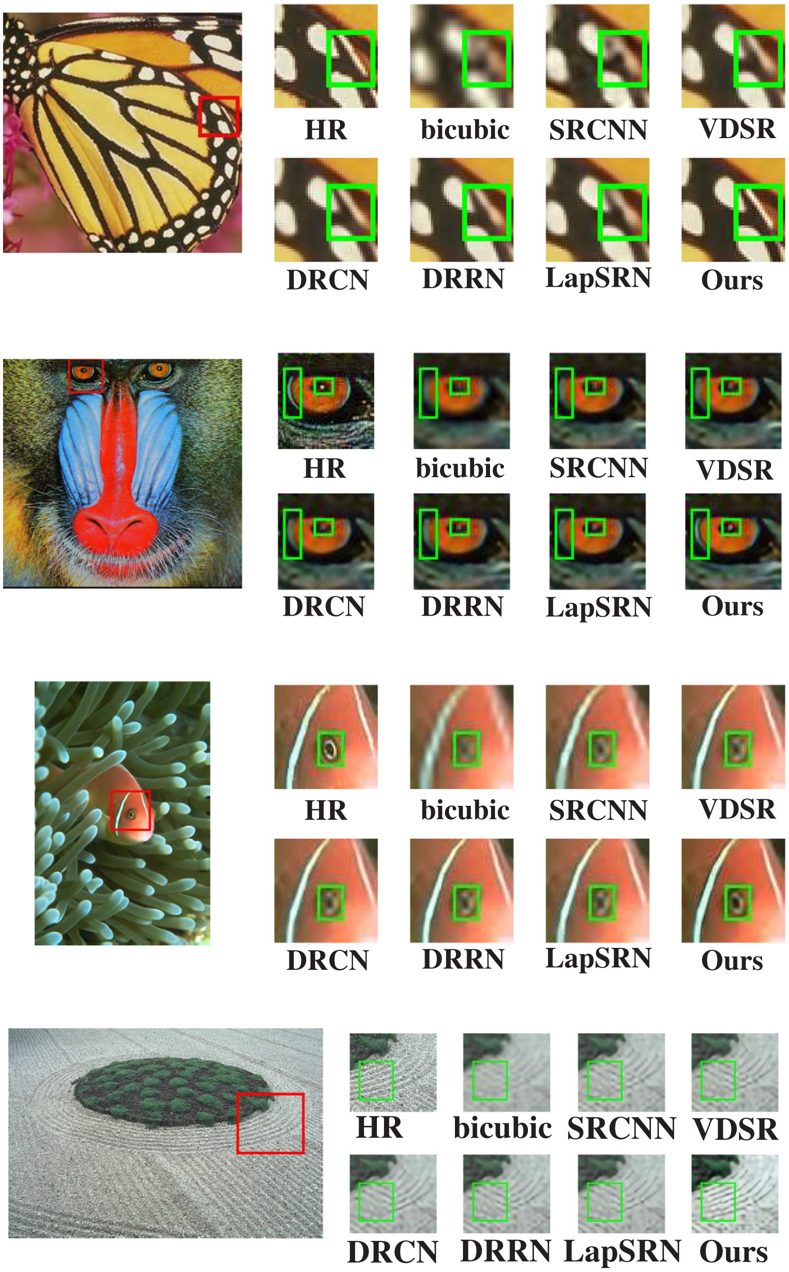


Fig. 4. Visual comparison for ×4 SR on different datasets. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

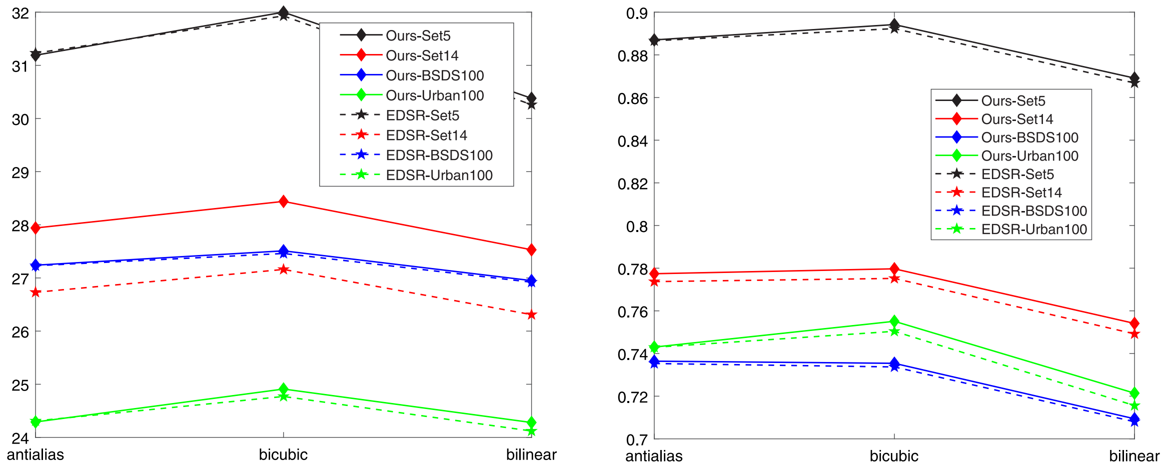
Table 2
The comparison of model parameters and performance with EDSR and RDN on BSDS100.

Algorithm	Feature extraction	Filters	Layers	Parameters	PSNR	SSIM
EDSR (Lim et al., 2017)	32 blocks	256	69	43M	32.32	0.9013
RDN (Zhang et al., 2018)	16 blocks	64	149	23.7M	32.34	0.9017
CSBRN	32 blocks	128	131	11.4M	32.28	0.9008

SSIM on the luminance channel and ignore the same amount of pixels as scale from the border. Compared with other methods, our method exhibits good performance.

Visual comparison. In Fig. 4, we show qualitative results and visual comparisons on different datasets. In order to compare details of the reconstructed image, we mark the magnified area with a red rectangle on the ground truth. And we mark the position of detail improvement with a green rectangle on the magnified area. It is obvious that our method has a better effect on texture detail restoration. On butterfly image, the wing texture of our result is clearer than that of other methods. On baboon image, our result has a white line like ground truth on the baboon's eye. On goldfish image, goldfish's eye of our result is closer to that of ground truth in shape. On the sand image, the texture of sand is interlaced. The texture of other results are disturbed by other wrong textures nearby. While our result generates texture details which subjectively closer to the ground truth.

Parameters comparison. The parameters of the proposed method are compared with those of other advanced methods. As shown in Table 2, the parameter of our algorithm is about a third of EDSR and



(a) PSNR (b) SSIM

Fig. 5. Stability of EDSR and our method for different datasets and downsampling methods.

one half of RDN. They have similar performance on BSDS100. Our algorithm effectively reduces the amount of parameters and computation, and prevents the performance degradation as much as possible.

Stability of the model for different datasets and downsampling methods. The low-resolution image of the training set is obtained by using the bicubic downsampling, but the actual low-resolution image is more complex. Thus, we use antialias, bicubic and bilinear downsampling methods to test the stability of the model. We compare our method with EDSR on the same training dataset and processing. As shown in Fig. 5, the stability of two methods is slightly better. But for PSNR of Set14, our method is better.

5. Discussion

This section discusses the model in three parts: parameter settings, ablation experiments and loss function analysis.

5.1. Parameter settings

Initializer analysis. Different weight initialization methods have different influence on the training process of the network. In this paper, we contrast the effect of Xavier initialization (Dong et al., 2016a) and Normal initialization in SISR problem. We record the MSE of the validation set every 500 iterations. As illustrated in Fig. 6(a), the network with the Xavier initialization converges better and faster.

Network depth. We train the proposed method with the different number of blocks, $B = 8, 16, 24, 32, 40$. As shown in Fig. 6(b), as the number of blocks increases, the performance of model becomes better. However, as the number of blocks increases, the speed of performance improvement slows down. And the model with 32 blocks and 40 blocks are almost identical in performance. The dataset information determines the upper bound of the model effect, and the representation ability of the model has a certain limit. In addition, the deeper networks have higher computational costs. In Table 3, we show the trade-off between performance and speed on Set5 and Set14. In order to strike a balance between performance and speed, we choose $B = 32$ for our method.

Network width. We train our network with different width, $W = 32, 64, 128, 256$. We show the performance of networks in Fig. 6(c). The model with 128 width and 256 width are almost identical in performance. Although the width of the network becomes larger, the performance does not always become better. This indicates that there is redundant in convolution kernels of the network. In addition, the wider network has higher computational costs. In Table 4, we show the trade-off between performance and speed on Set5 and Set14. In order to strike a balance between performance and speed, we choose $W = 128$ for our method.

Table 3
The trade-off between performance and speed on the number of block at each level of the proposed network.

Blocks	Set5			Set14		
	PSNR	SSIM	Time (s)	PSNR	SSIM	Time (s)
8	33.88	0.8912	1.250	30.07	0.7811	1.422
16	33.98	0.8922	1.382	30.09	0.7819	1.526
24	34.02	0.8934	1.489	30.16	0.7834	1.688
32	**34.10**	**0.8947**	1.644	30.15	0.7832	1.846
40	34.09	0.8941	1.827	**30.18**	**0.7836**	1.973

Table 4
Trade-off between performance and speed on the width at each level of the proposed network.

Width	Set5			Set14		
	PSNR	SSIM	Time (s)	PSNR	SSIM	Time (s)
32	33.88	0.8912	1.250	30.07	0.7811	1.422
64	34.10	0.8945	1.270	30.25	0.7854	1.427
128	**34.23**	0.8961	1.278	**30.36**	**0.7876**	1.565
256	**34.23**	**0.8964**	1.360	30.35	0.7871	1.820

Weight decay. By training the network with width $W = 128$, we find that the network will lead to slight overfitting. Therefore, the training should be regularized by weight decay. We train the network with different weight decay, $\beta = 0, 0.001, 0.0001, 0.00001$. As shown in Fig. 6(d), the network with weight decay set to 0.0001 has the best performance.

5.2. Ablation experiments

Ablation experiments of residual block. In order to prove that our residual block is more effective, we conducted ablation experiments on the residual block. Our residual blocks have two 3 × 3 convolutional layers and two 1 × 1 convolutional layers, but the usual residual blocks only have two 3 × 3 convolutional layers. We use two kinds of residual blocks to carry out ablation experiments. As shown in Fig. 7, the network with our residual block converges better and faster.

The output of information transfer pathway and high frequency information pathway. In order to prove that the output of two pathways respectively corresponds to the high-frequency information and the low-frequency information, we output the image of two pathways respectively. As shown in Fig. 8, the output of information transfer

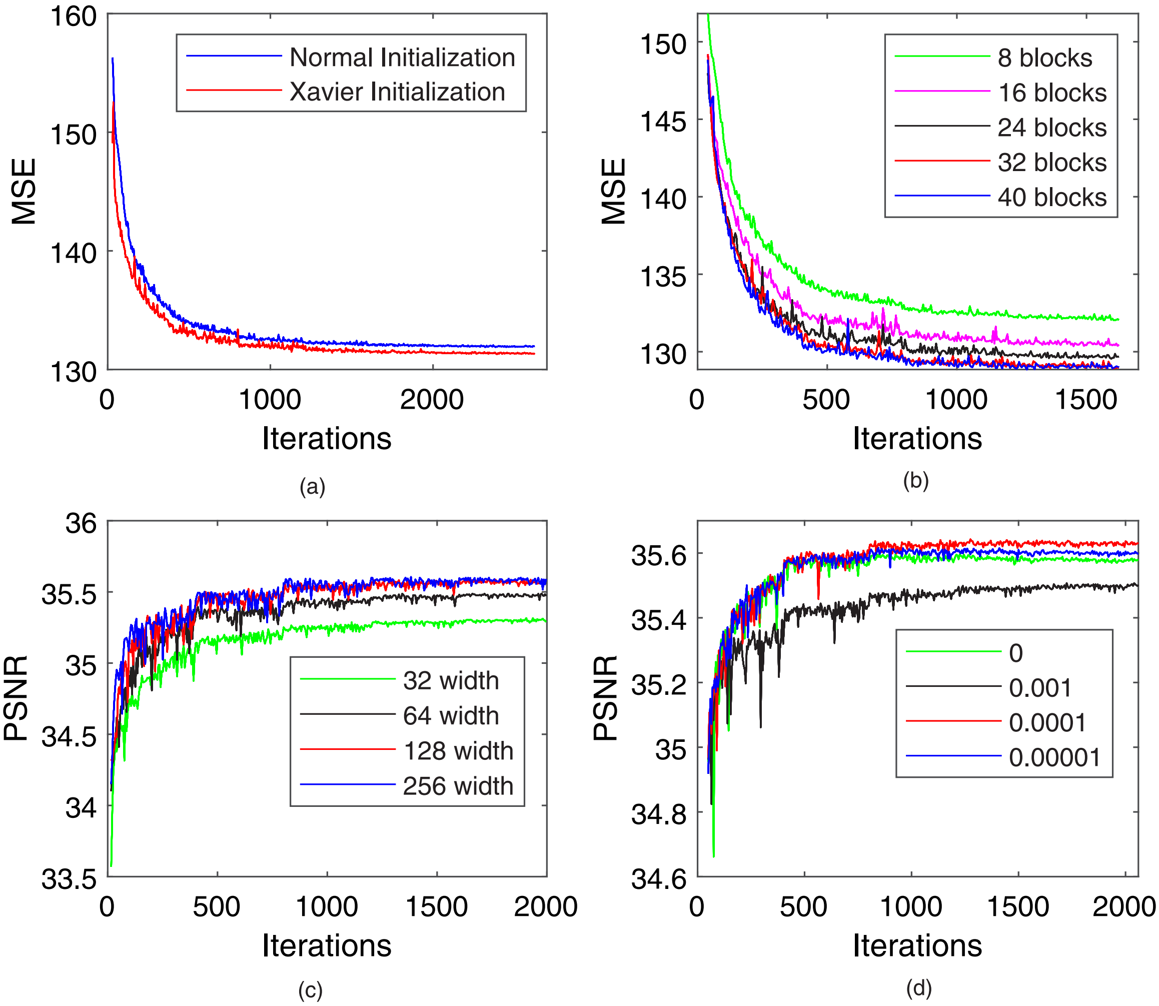


Fig. 6. Method analysis: (a) initializer analysis; (b) network depth; (c) network width; (d) weight decay.

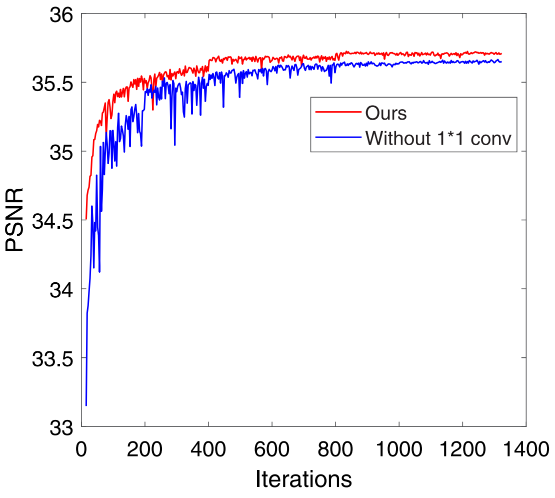


Fig. 7. Ablation experiment of residual block..

pathway is mainly low-frequency information. And the output of high frequency information pathway is mainly high-frequency information.

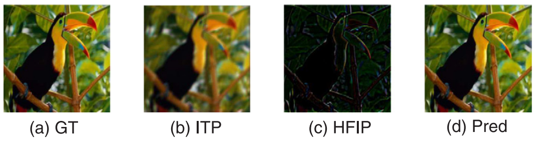
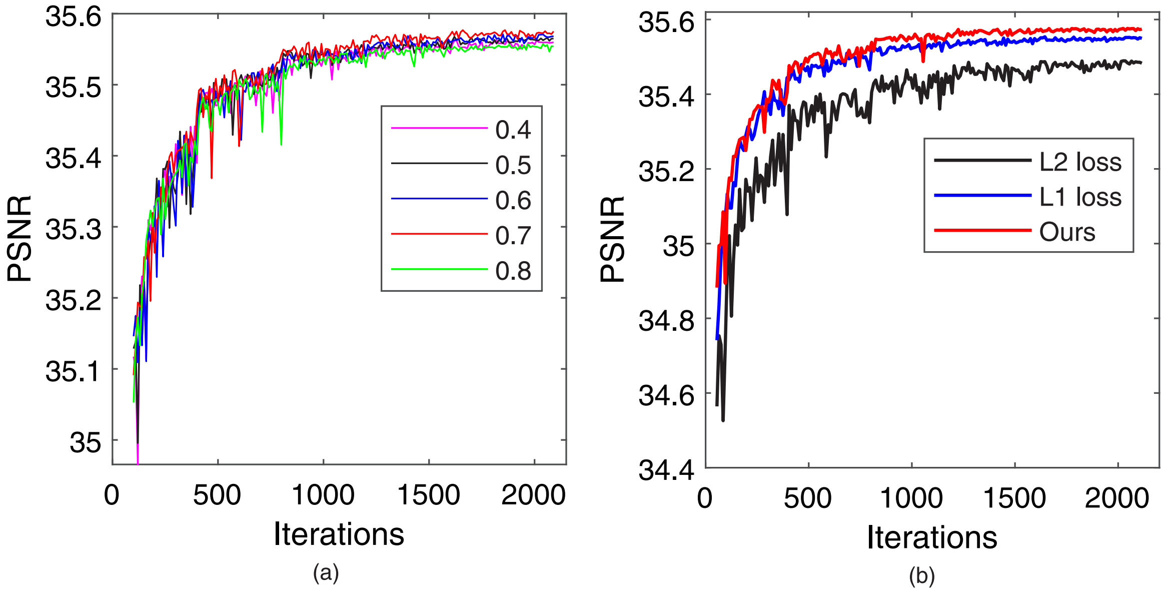


Fig. 8. The output of information transfer pathway and high frequency information pathway: (a) ground truth(GT); (b) information transfer pathway(ITP); (c) high frequency information pathway(HFIP); (d) prediction(Pred)..

5.3. *Loss function analysis*

The balance parameter of loss function. This experiment is to determine the balance parameter α of loss function. We train our network with different balance parameters, $\alpha = 0.4, 0.5, 0.6, 0.7, 0.8$, and same network parameters, $B = 8, W = 64$. As shown in Fig. 9(a), when the α is set to 0.7, the convergence of the network is the best.

Loss function. We train the network with different loss functions, L_1 loss function, L_2 loss function and the proposed loss function. As shown in Fig. 9(b), the convergence of the proposed loss function is the best. Moreover, we compare the visual effect of the network with different loss functions. As shown in Fig. 10, our loss function is helpful for detail restoration of low resolution images. On ppt image, the letter 'ow' of our result is clearer than other methods with different loss function. On penguin image, our result has a white line like ground truth on the body of the penguin. L_1 and L_2 loss functions directly learn the



(a) (b)

Fig. 9. Method analysis: (a) the balance parameter of loss function; (b) the different loss functions.

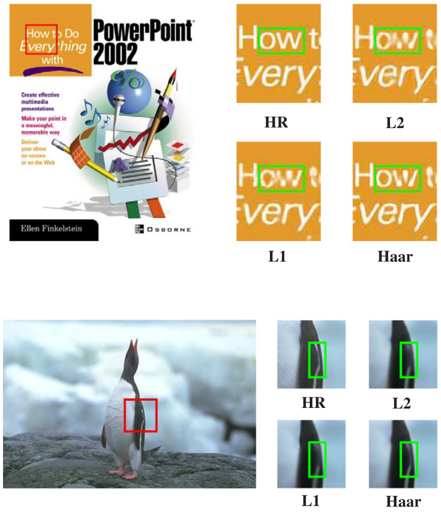


Fig. 10. Visual comparison of networks with different loss functions for ×4 SR.

pixel value. And the ability of image reconstruction is limited. Texture details of the reconstructed image are sometimes wrong. Through detail perception loss, our network can effectively learn texture features.

6. Conclusion

In this paper, we proposed a novel residual network with cascading simple blocks for SISR. We transform the pixel prediction of image super-resolution problem into the prediction of high-frequency information and the expansion of low-frequency information. The global skip connection with a convolutional layer passes low frequency information from LR images. The high frequency information pathway composed of residual blocks with cascading simple blocks predicts high frequency information from LR images by a complex mapping. The network uses many skip connections, which contribute to alleviating gradient disappearance and gradient explosion. In the residual block, we cascade 3×3 and 1×1 convolutional layer to increase the receptive field and nonlinearity, respectively. We can adjust the number of 3×3 and 1×1 convolutional layers according to our needs. Moreover, to capture more texture details, we adopt detail perception loss function, which is used to measure the difference of the reconstructed image and ground truth by Haar wavelet transform. The proposed loss function can be used in most image super-resolution networks to improve the model performance. The proposed method achieves a good result on four benchmark datasets.

CRediT authorship contribution statement

Zhijie Wen: Conceptualization, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Jiawei Guan:** Methodology, Software, Validation, Writing - original draft. **Tieyong Zeng:** Formal analysis, Investigation, Data curation. **Ying Li:** Resources, Visualization.

Declaration of competing interest

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

Acknowledgments

This research is supported by the National Natural Science Foundation of China (11701357, 11971296, 81830058, 11601315) and The Capacity Construction Project of Local Universities in Shanghai, China under Grant (18010500600).

References

Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L., 2012. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: Br. Machine Vis. Conf., pp. 1–10.

Bhowmik, A., Shit, S., Seelamantula, C., 2018. Training-free, single-image super-resolution using a dynamic convolutional network. IEEE Signal Process. Lett. 25, 85–89.

Cheong, J., Park, I., 2017. Deep CNN-based super-resolution using external and internal examples. IEEE Signal Process. Lett. 24, 1252–1256.

Dong, C., Loy, C., He, K., Tang, X., 2016a. Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. 30, 295–307.

Dong, C., Loy, C., Tang, X., 2016b. Accelerating the super-resolution convolutional neural network. In: Eur. Conf. on Comput. Vis., pp. 391–407.

Fang, F., Li, J., Zeng, T., 2020. Soft-edge assisted network for single image super-resolution. IEEE Trans. Image Process. 29, 4656–4668.

Haris, M., Shakhnarovich, G., Ukita, N., 2018. Deep back-projection networks for super-resolution. In: IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1664–1673.

Huang, H., He, R., Sun, Z., Tan, T., 2017. Wavelet-SRNet: a wavelet-based CNN for multi-scale face super resolution. In: The IEEE Int. Conf. Comput. Vis., pp. 1698–1706.

Huang, J.B., Singh, A., Ahuja, N., 2015. Single image super-resolution from transformed self-exemplars. In: IEEE Conf. Comput. Vis. Pattern Recognit., pp. 5197–5206.

Hui, Z., Wang, X., Gao, X., 2018. Fast and accurate single image super-resolution via information distillation network. In: IEEE Conf. Comput. Vis. Pattern Recognit., pp. 723–731.

Kim, J., Lee, J.K., Lee, K.M., 2016a. Accurate image super-resolution using very deep convolutional networks. In: IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1646–1654.

Kim, J., Lee, J.K., Lee, K.M., 2016b. Deeply-recursive convolutional network for image super-resolution. In: IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1637–1645.

Kingma, D.P., Ba, J., 2014. Adam: a method for stochastic optimization.

Lai, W., Huang, J., Ahuja, N., Yang, M., 2017. Deep Laplacian pyramid networks for fast and accurate super-resolution. In: IEEE Conf. Comput. Vision Pattern Recognit., pp. 624–632.

Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W., 2017. Photo-realistic single image super-resolution using a generative adversarial network. In: IEEE Conf. Comput. Vis. Pattern Recognit., pp. 4681–4690.

Li, J., Fang, F., Mei, K., Zhang, G., 2018. Multi-scale residual network for image super-resolution. In: Eur. Conf. on Comput. Vis., pp. 527–542.

Lim, B., Soen, S., Kim, H., Nah, S., Lee, K.M., 2017. Enhanced deep residual networks for single image super-resolution. In: IEEE Conf. Comput. Vis. Pattern Recognit., pp. 136–144.

Lin, M., Chen, Q., Yan, S., 2013. Network in network.

Martin, D., Fowlkes, C., Tal, D., Malik, J., 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: IEEE Conf. Comput. Vis. Pattern Recognit., pp. 416–423.

Odena, A., Dumoulin, V., Olah, C., 2016. Deconvolution and checkerboard artifacts. http://dx.doi.org/10.23915/distill.00003.

Peled, S., Yeshurun, Y., 2001. Superresolution in MRI: application to human white matter fiber tract visualization by diffusion tensor imaging. Magn. Reson. Med. 45, 29–35.

Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z., 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1874–1883.

Tai, Y., Yang, J., Liu, X., 2017. Image super-resolution via deep recursive residual network. In: IEEE Conf. Comput. Vis. Pattern Recognit., pp. 3147–3155.

Tai, Y., Yang, J., Liu, X., Xu, C., 2017. MemNet: a persistent memory network for image restoration. In: IEEE Conf. Comput. Vis. Pattern Recognit. pp. 4539–4547.

Thornton, M.W., Atkinson, P.M., a. Holland, D., 2006. Subpixel mapping of rural land cover objects from fine spatial resolution satellite sensor imagery using super-resolution pixel-swapping. Int. J. Remote Sens. 27, 473–491.

Tian, S., Zou, L., Yang, Y., Kong, C., Liu, Y., 2019. Learning image block statistics and quality assessment losses for perceptual image super-resolution. J. Electron. Imaging 28 (1), 013042, (1–15).

Timofte, R., Agustsson, E., Gool, L.V., Yang, M., Zhang, L., 2017. Ntire 2017 challenge on single image super-resolution: methods and results. In: IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, pp. 1110–1121.

Tong, T., Li, G., Liu, X., Gao, Q., 2017. Image super-resolution using dense skip connections. In: IEEE Int. Conf. Comput. Vis., pp. 4799–4807.

Yang, X., Mei, H., Zhang, J., Xu, K., Yin, B., Zhang, Q., Wei, X., 2019. DRFN: deep recurrent fusion network for single-image super-resolution with large factors. IEEE Trans. Multimedia 21, 328–337.

Yang, W., Wang, W., Zhang, X., Sun, S., Liao, Q., 2018. Lightweight feature fusion network for single image super-resolution. In: IEEE Conf. Comput. Vis. Pattern Recognit., pp. 723–731.

Yu, J., Fan, Y., Yang, J., Xu, N., Wang, Z., Wang, X., Huang, T., 2018. Wide activation for efficient and accurate image super-resolution.

Zeyde, R., Elad, M., Protter, M., 2010. On single image scale-up using sparse-representations. In: Int. Conf. on Curves and Surfaces, pp. 711–730.

Zhang, S., Lu, Y., 2011. Image resolution enhancement via image restoration using neural network. J. Electron. Imaging 20 (2), 023013, (1–10).

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y., 2018. Residual dense network for image super-resolution. In: IEEE Conf. Comput. Vis. Pattern Recognit., pp. 2472–2481.

Zhang, L., Zhang, H., Shen, H., Li, P., 2010. A super-resolution reconstruction algorithm for surveillance images. IEEE Signal Process. 90, 848–859.

Zhong, Z., Shen, T., Yang, Y., Lin, Z., Zhang, C., 2018. Joint sub-bands learning with clique structures for wavelet domain super-resolution. Neural Inf. Process. Syst.