

Design of Network Cascade Structure for Image Super-Resolution

Jianwei Zhang, Zhenxing Wang, Yuhui Zheng and Guoqing Zhang*

Nanjing University of Information Science and Technology, Nanjing, 210044, China

*Corresponding Author: Guoqing Zhang. Email: xiayang14551@163.com

Received: 06 March 2021; Accepted: 07 March 2021

Abstract: Image super resolution is an important field of computer research. The current mainstream image super-resolution technology is to use deep learning to mine the deeper features of the image, and then use it for image restoration. However, most of these models mentioned above only trained the images in a specific scale and do not consider the relationships between different scales of images. In order to utilize the information of images at different scales, we design a cascade network structure and cascaded super-resolution convolutional neural networks. This network contains three cascaded FSRCNNs. Due to each sub FSRCNN can process a specific scale image, our network can simultaneously exploit three scale images, and can also use the information of three different scales of images. Experiments on multiple datasets confirmed that the proposed network can achieve better performance for image SR.

Keywords: Super-resolution; cascade structure; convolutional neural network

1 Introduction

Single image super-resolution (SR) is an important issue in the field of computer vision research. Given a low resolution (LR) image, the purpose of single image SR is to recover a high resolution (HR) image from its corresponding LR image. Currently, single image SR based methods are generally divided into two categories: sparse coding-based methods [1–4] and deep learning-based methods [5–8].

Most sparse coding-based methods assume that each pair of patches from LR and HR images have similar coding coefficients in the patch space. Therefore, the HR image patches can be reconstructed as a linear combination of a few LR image patches. Yang et al. [1–2] learned a pair of dictionaries D_L and D_H for LR and HR image patches respectively. The final reconstructed HR image can be obtained through the learned dictionary D_H and coding coefficient \mathbf{a} , which encoded by the learned dictionary D_L [9–11]. At present, existing sparse coding-based methods mainly focus on how to learn better dictionaries [12–13]. For example, Zhao et al. [12] designed a transfer robust sparse coding based on graph for image representation and Ahmed et al. [13] proposed a new multiple dictionary learning strategy.

Deep learning-based methods aim to directly learn an end-to-end mapping from LR images to HR images [5–8]. Dong et al. [7] first applied convolutional neural network (CNN) for the single image SR, and proposed a super-resolution convolutional neural network (SRCNN) and achieved better restoration performance. However, SRCNN only learns the mapping from bicubic interpolated LR images (not the original LR images) to HR images, the calculation cost of the network will be increased quadratically with the size of HR image increase [8]. In addition, SRCNN adopts a 5×5 convolution kernel to learn nonlinear mapping which obviously limited the learning ability of network in such a setting. To solve the above problems, researchers [14–16] proposed lots of approaches such as the large super-resolution convolutional neural network (SRCNN-ex), fast super-resolution convolutional neural network (FSRCNN), very deep super-resolution network (VDSR), Laplacian pyramid super-resolution network (LaPSRN) and enhanced deep super-resolution network (EDSR) to improve the structure of SRCNN. In



addition, Christian et al. [17] exploited generative adversarial networks (GAN) [18] to handle out image super-resolution problem and proposed super-resolution generative adversarial network (SRGAN). GAN is different from convolutional neural network (CNN), and the main difference is that GAN aims to generate more realistic HR images that are more consistent with human eye, while the purpose of CNN is to faithfully restore the high-frequency information of images. Lots of improved networks based on GAN have been proposed in recent years [19–20]. These networks have solved the above problems to some extent and achieved better restoration performance.

However, most of these models mentioned above only trained the images in a specific scale and do not consider the relationships between different scales of images. For example, if we set the scale factor is 4 during the training process, FSRCNN only utilizes the image information, where the image scale is 4, and does not use the complementary information resided in different scales. If we can simultaneously use the information of different scales of images in the training process, the recovered image qualities will be further improved.

Based on the above concerns, we design a cascaded super-resolution convolutional neural networks (CSRCNN), which consists of three cascaded FSRCNNs. Due to each sub FSRCNN can process a specific scale image, our network can simultaneously exploit three scale images, and can also use the information of three different scales of images in the training process. For each FSRCNN, we set its scaling factor is 2, which can double enlarge the size of input image. Since the images with different scales are trained together, the learned network can exploit more image information resided in different scales. In addition, we use L1 loss function to make the reconstructed image in texture and edge clearer. Experiments on multiple datasets show that the proposed our proposed network can achieve better performance for image SR.

2 Related Work

2.1 Deep Learning for Image Super-Resolution

The purpose of image super-resolution is to reconstruct high resolution image from a given low resolution one. Dong et al. [7] first applied convolutional neural network in the field of image super-resolution and proposed a super-resolution convolutional neural network (SRCNN) which achieved impressive performance for image restoration. At present, deep learning technology is widely used in the field of image super-resolution. Various network structures have been developed, such as deep network with residual learning [14], Laplace pyramid structure [15], residual block [21], and residual dense network [22]. Besides supervised learning and unsupervised learning [23–26], reinforcement learning [27–29] are also introduced to solve the problem of image super-resolution in recent years. Specifically, the literature [30] has a systematic description of image super resolution.

2.2 Fast Super-Resolution Convolutional Neural Networks

Dong et al. [8] proposed a fast super-resolution convolutional neural network (FSRCNN), aiming to accelerate the speed of SRCNN. Compared with SRCNN, FSRCNN can directly learn the mapping from the original LR image to HR image by introducing the deconvolution layer at the end of the network. In addition, FSRCNN adds a shrinking by and an expanding layer at the beginning and the end of the mapping layer to enhance the presentation capability of non-linear mapping [31]. In the nonlinear layer, FSRCNN adopts a smaller filter (size = 3×3) to reduce the computational cost of the network. The whole network is designed as a compact hourglass-type CNN structure, which includes five parts: feature extraction, shrinking, non-linear mapping, expanding, deconvolution. The shrinking layer uses a filter of size 1×1 to reduce the LR feature dimension from d to s , where d is the feature dimension of LR image after feature extraction and s is the number of filters ($s \ll d$). The expanding layer uses d filters with a size of 1×1 to maintain consistency with the shrinking layer, which is the inverse operation of the shrinking layer. To avoid the “dead features” [32] caused by zero gradients in ReLU , the author uses PReLU

[33] as the activation function after each convolution layer. Thus, a complete FSRCNN can be represented as:

$$\begin{aligned} & Conv(5, d, 1) - PReLU - Conv(1, s, d) - PReLU - Conv(3, s, s) - PReLU - mConv(1, d, s) \\ & - PReLU - DeConv(9, 1, d). \end{aligned} \quad (1)$$

3 Cascaded Super-Resolution Convolutional Neural Network

3.1 Network Structure

We propose a cascaded super-resolution convolutional neural network (CSRCNN) framework in this section. Our proposed network consists of three cascaded FSRCNNs, where each FSRCNN can process a specific scale image. Fig. 1 shows the structure of our CSRCNN.

In each sub FSRCNN, we set its scaling factor to 2 for double enlarging the size of input image. Suppose that all images have the same width-to-height ratio. We denote W and H respectively as the width and height of the original HR image. For the first sub FSRCNN, the image shape of the input is $W/8 * H/8$ and the output will be as the input of next sub FSRCNN. Due to each sub FSRCNN can double enlarge the size of input image, the image shape of the input for the second sub FSRCNN is $W/4 * H/4$. Similarly, for the third sub FSRCNN, the image shape of the input is $W/2 * H/2$. During the training process, images of different scales are trained simultaneously. Here, we use I^k to represent the input image of each sub FSRCNN, where $k = 0, 1, 2$ is the ID of each sub FSRCNN and the scale index of the input image. We set r_k to represent the scale ratio of LR image to HR image, the size of input image I^k is described as $r_k W * r_k H$. The scale ratios are set as $r_0 = 1/8, r_1 = 1/4$ and $r_2 = 1/2$ respectively. $I^{k+1} = F_k(I^k)$ is the super-solved HR image obtained by sub FSRCNN-k. For each

cascaded FSRCNN, the output image is twice the size of input image. We use \tilde{I}^{k+1} represents the real HR image, which has the same size with the super-solved image $F_k(I^k)$. Our goal is to learn super-resolved functions that estimating the reconstructed HR images I^{k+1} and real HR images \tilde{I}^{k+1} .

3.2 Loss Function

Our network includes three cascaded FSRCNNs. For each subnetwork, the output of previous network will be the input of the next cascaded subnetwork. We use L1 loss functions as the sub-loss functions for all three subnetworks, and the sum of these sub-loss functions will eventually form the total loss function of the whole network. We use $L_1^{SR_0}$, $L_1^{SR_1}$ and $L_1^{SR_2}$ represent the loss functions of three subnetworks, respectively. The loss function of the whole network can be expressed as:

$$L_{total} = L_1^{SR_0} + L_1^{SR_1} + L_1^{SR_2}. \quad (2)$$

For each subnetwork, the loss function is defined as follows:

$$L_1^{SR_k} = \frac{1}{r_{k+1}^2 WH} \sum_{x=1}^{r_{k+1}W} \sum_{y=1}^{r_{k+1}H} \left\| I_{x,y} - F_k(I^k)_{x,y} \right\|_1, \quad k = 0, 1, 2. \quad (3)$$

In Eq. (2), we select L1 loss function to replace MSE loss function. The main reason is that L1 loss function can make the reconstructed image clearer in texture and edge, while MSE loss function will lose high-frequency information in the reconstruction process, such as texture and edge [34].

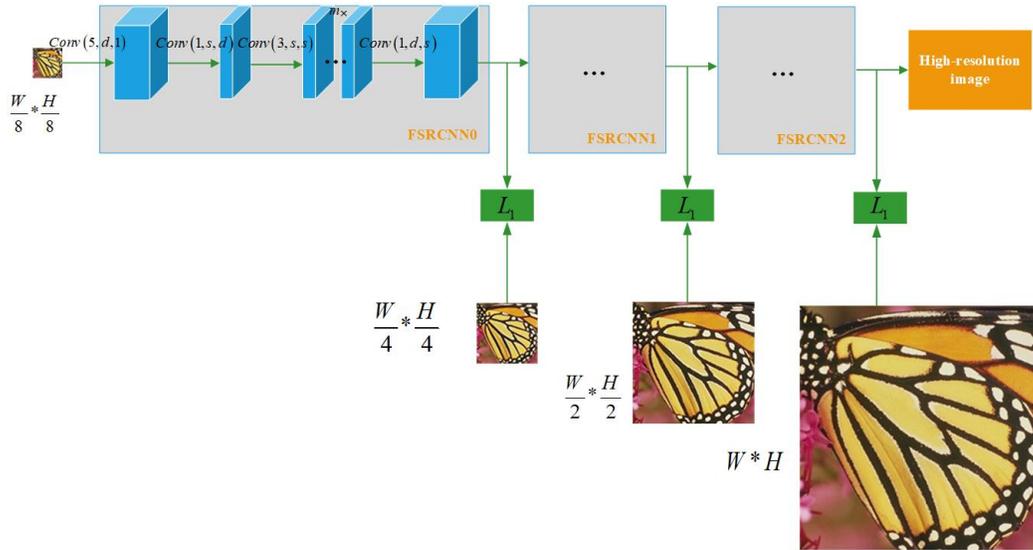


Figure 1: The network structure of our CSRCNN. The network is composed of three cascaded FSRCNNs, in which the upscaling factor of each FSRCNN is 2. For the first sub FSRCNN, the input is an $W/8 * H/8$ LR image and the output of previous layer will be the input of the next layer. Each sub FSRCNN can double enlarge the size of input image. We select L1 loss function as sub-loss functions of three subnetworks, and the sum of three sub-loss functions will eventually form the loss function of the whole network

3.3 Assessment Process

According to the scale ratios of LR images to HR images, we assign LR images to different stages of the cascaded networks during the evaluation process. When a LR image is input into FSRCNN-k, the input shape will be resized based on the corresponding network $r_k W * r_k H$. For example, for a given test image with size of $W/3 * H/3$, we will resize the shape of the image as $W/2 * H/2$ by bicubic interpolation. The resized image will be arranged into FSRCNN-2. Here, each sub-network can double enlarge the size of input image. Finally, all images are enlarged to the uniform HR images [35]. Fig. 2 shows the entire evaluation process of our network.

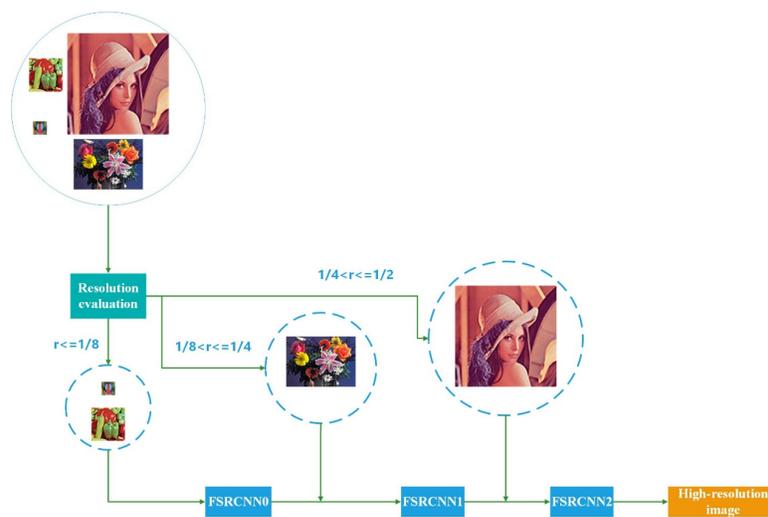


Figure 2: LR images are assigned to different stages of the cascaded FSRCNNs according to their resolution scale ratios. Through the CSRCNN, all LR images are enlarged to the uniform HR ones

3.4 The Difference between Our Model with FSRCNN

Our network contains three cascaded FSRCNNs, where each FSRCNN can process a specific scale image. For each sub FSRCNN, we use the L1 loss function, whereas original FSRCNN uses the MSE loss function. In addition, FSRCNN only considers a specific scale image, while our network can simultaneously train different scales of images due to adopting the cascaded structure. In the test phase, our network can generate multiple intermediate results in a prediction process, while FSRCNN only obtains a single scale result.

4 Experiments

4.1 Dataset

Training and testing dataset: Following SRCNN and FSRCNN, we combine 91 images dataset and the general-100 dataset to train our network. In particular, general-100 dataset [8] contains 100 bmp-format images with ranges from 710×704 (large) to 131×112 (small) in size. All of these images are favorable quality with clear edges and less smooth regions. Following the setting used in [8], we adopt the data augmentation method [36], includes scaling and rotating images. 1) Scaling: each image is down-sampling with the factor 0.9,0.8,0.7 and 0.6. 2) Rotation: each image is rotated with the degree of 90, 180 and 270. For testing dataset, we use Set5 [37], Set14 [38] and BSD200 [39] datasets. All testing images will be cropped according to the model structure which make the size of output image of the model as an integer.

Training samples: Cascaded structure are adopted in our network. To form LR/HR sub-image pairs, we first down-sample the original training images by different scale factors n , then crop LR images and real HR images obtained in sampling process.

4.2 Training Details

Training strategy: In our experiment, we combine 91 images dataset and the general-100 dataset [8] for training. The 91-image dataset is used to train a network from scratch. Then, we add General-100 dataset to the network for fine-tuning when the training is saturated. Using this strategy, the training converges much earlier compare with training with both two datasets.

The choice of learning rate: The learning rate plays a very import role for the performance of our network. For FSRCNN, the learning rates in the convolution layers and deconvolution layer are set to be 10^{-3} and 10^{-4} , the learning rates of all layers are reduced by half during fine-tuning. Obviously, the learning rates of convolution layer and deconvolutional layer are static in each iteration, while we adopt a dynamic strategy to update the learning rates for convolution layer and deconvolutional layer in our network. This strategy can make our network produce better learning rates in different iterations. We set $a_0 = 10^{-3}, 10^{-4}$ as the initial learning rates of the convolution layer and the deconvolution layer, and the total number of iterations in our network is set as n . When the number of iterations is m , the learning rates in convolution layer and the deconvolution layer are represented as:

$$a_m = a_0 * 0.1^{\rho\left(\frac{m}{0.8*n}\right)} \quad (4)$$

Here, $\rho\left(\frac{m}{0.8*n}\right)$ represents the integer part of the final result.

The choice of network initialization: Network initialization has a great influence on network training, a good initialization can largely reduce the training time of the network. In our network, we select *PReLU* [33] as the activation function and use MSRA initialization [33]. MSRA is a gaussian with a mean of zero and variance of $n/2$, which make each layer neuron input/output variance is consistent.

Parameter setting: Our proposed cascaded CSRCNN consists of three subnetworks. For each subnetwork, there are three sensitive variables governing the network's performance. The LR feature

dimension d , the number of shrinking filters s , and the mapping of the depth m . Following the same setting in FSRCNN, we set $d = 56, s = 12$ and $m = 4$ for each subnetwork.

4.3 Experiments for Different Loss Functions

In this section, we conduct experiments to evaluate the effects of different loss functions on our network on three datasets when the upscaling factor is 3. We compare MSE-Loss, Huber-Loss, Charbonnier-loss and L1-loss functions and the results are listed in Table 1. The delta of Huber-Loss is 0.9 and the delta of Charbonnier-Loss is 0.0001. From Table1, we can see that using L1-loss, our cascaded network can achieve better performance. Fig. 3 shows the convergence curves of these four loss functions on Set5 datasets with upscaling factor 3. From the Fig. 3, we first observed that using L1-Loss, our cascaded network gets the best average test PSNR values, followed by Charbonnier-Loss, and Huber-Loss get the worst average test PSNR values. In addition, after 160 iterations, the PSNR value curves of these four loss functions tend to be stable. Among them, the network has a small fluctuation when we use Charbonnier-Loss and MSE-Loss as our network loss functions. We also find that the initial value of PSNR is low when we use L1 loss as the loss function of the network, with the increase of the number of iterations, the PSNR value will gradually increases and eventually tends to be stable. In this section, we evaluate the performance of our network versus the number of cascaded FSRCNNs. Since the scaling factor of each FSRCNN is set to 2, if we cascaded more than three FSRCNNs, the scale of the input image to the network will too small, which makes the effective information contained in these images are very limited. Therefore, in our model, we only cascaded three FSRCNNs. These three networks are respectively denoted as Net-1, Net-2 and Net-3. Net-1 contains one FSRCNN with upscaling factor is 4, and Net-2 contains two cascaded FSRCNNs, where upscaling factor is 2. We use PSNR as the evaluation criteria and test on three datasets for $4\times$ SR. Table 2 lists the results our model versus different number of FSRCNNs and the number of FSRCNNs varies from 1 to 3. From Tab. 2, we can observe that Net-3 obtains the highest PSNR value on all testing datasets. Net-2 obtains higher results than Net-1. This indicates that cascading multiple basic models can achieve better performance than a single basic model. In addition, the increasing trend of PSNR is gradually decreasing with the increase of the number of cascaded FSRCNNs, one possible reason is that the extra information provided by LR images with small scale is limited.

4.4 Evaluation for the Number of Cascaded FSRCNNs

In this section, we evaluate the performance of our network versus the number of cascaded FSRCNNs. Since the scaling factor of each FSRCNN is set to 2, if we cascaded more than three FSRCNNs, the scale of the input image to the network will too small, which makes the effective information contained in these images are very limited. Therefore, in our model, we only cascaded three FSRCNNs. These three networks are respectively denoted as Net-1, Net-2 and Net-3. Net-1 contains one FSRCNN with upscaling factor is 4, and Net-2 contains two cascaded FSRCNNs, where upscaling factor is 2. We use PSNR as the evaluation criteria and test on three datasets for $4\times$ SR. Tab. 2 lists the results our model versus different number of FSRCNNs and the number of FSRCNNs varies from 1 to 3. From Tab. 2, we can observe that Net-3 obtains the highest PSNR value on all testing datasets. Net-2 obtains higher results than Net-1. This indicates that cascading multiple basic models can achieve better performance than a single basic model. In addition, the increasing trend of PSNR is gradually decreasing with the increase of the number of cascaded FSRCNNs, one possible reason is that the extra information provided by LR images with small scale is limited.

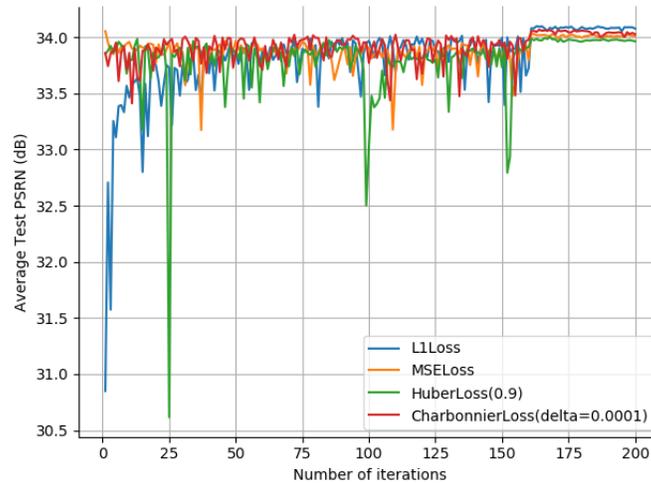


Figure 3: Experiments for different loss functions on Set5 datasets when the upscaling factor is 3

Table 1: The influence of different loss choices on CSRCNN with upscaling factor 3, where the test datasets are Set5, Set14 and BSD200

Test-dataset	Upscaling factor	MSE-Loss	Huber-Loss (delta = 0.9)	Charbonnier-Loss (delta = 0.0001)	L1-Loss
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Set-5	3	34.02/0.9218	33.97/0.9213	34.04/0.9219	34.10/0.9233
Set-14	3	29.98/0.8321	28.86/0.8319	30.02/0.8332	30.02/0.8346
BSD200	3	29.69/0.8282	28.77/0.8279	29.68/0.8285	29.78/0.8302

Table 2: The influence of the number of cascaded FSRCNNs with upscaling factor 4, where the test datasets are Set5, Set14 and BSD200

Test-dataset	Upscaling factor	Net-1	Net-2	Net-3
		PSNR	PSNR	PSNR
Set-5	4	30.55	30.82	31.01
Set-14	4	27.50	27.66	27.71
BSD200	4	26.92	27.42	27.68

4.5 Comparison with Popular SR Algorithms

In order to verify the reconstruction performance of our proposed CSRCNN. We compare our model with three popular SR algorithms: Bicubic, SRCNN [7], and FSRCNN [8]. We carry out extensive experiments on three datasets: Set5 [37], Set14 [38] and BSD200 dataset [39]. We adopt two image quality indicators to evaluate the SR images, i.e., PSNR and SSIM [40]. Tab. 3 reports quantitative comparisons for $\times 2$, $\times 3$ and $\times 4$. From Tab. 3, we can observe that among all the compared methods, our proposed CSRCNN achieves the best results in three evaluation datasets for $\times 2$, $\times 3$ and $\times 4$. Fig. 4 shows the convergence curves of CSRCNN on three evaluation datasets with upscaling factor 3. Among them, the PSNR value of the proposed CSRCNN achieve 0.51 db, 1.04 db and 0.46 db improvement over FSRCNN in Set5 datasets for $\times 2$, $\times 3$ and $\times 4$. We observe that the PSNR value of CSRCNN obtains 1.04db improvement over that of FSRCNN on factor 3. One possible reason is that we use bicubic interpolation to resize the size of the image before throwing the LR image into the network. Considering that our network is made up of three cascaded FSRCNNs, where each FSRCNN's upscaling factor is 2. We also made a quantitative comparison for $\times 8$. This comparison result is shown in Tab. 4. Obviously, our network achieves higher PSNR and SSIM values than FSRCNN due to the cascaded design.

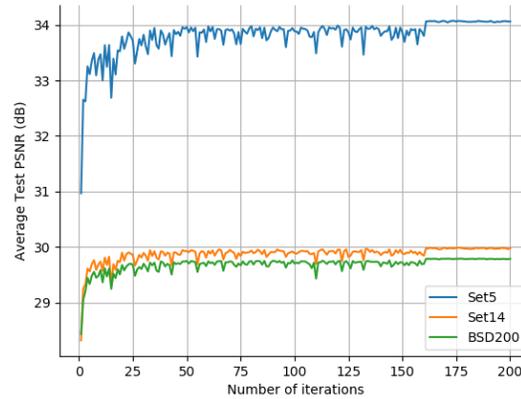


Figure 4: Visual comparisons on Set5, Set14 and BSD200 with upscaling factor 3

Table 3: Quantitative evaluation of popular SR algorithms average PSNR/SSIM for scale factors 2, 3 and 4

Test-dataset	Upscaling factor	Bicubic	SRCNN	FSRCNN	CSRCNN (our)
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Set-5	2	33.66/0.9299	36.33/0.9521	36.94/0.9558	37.45/0.9570
Set-14	2	30.23/0.8677	32.15/0.9039	32.54/0.9088	32.82/0.9112
BSD200	2	29.70/0.8625	31.34/0.9287	31.73/0.9074	32.92/0.9122
Set-5	3	30.39/0.8682	32.45/0.9033	33.06/0.9140	34.10/0.9233
Set-14	3	27.54/0.7736	29.01/0.8145	29.37/0.8242	30.02/0.8346
BSD200	3	27.26/0.7638	28.27/0.8038	28.55/0.8137	29.78/0.8302
Set-5	4	28.42/0.8104	30.15/0.8530	30.55/0.8657	31.01/0.8702
Set-14	4	26.00/0.7019	27.21/0.7413	27.50/0.7535	27.71/0.7589
BSD200	4	25.97/0.6949	26.72/0.7291	26.92/0.7398	27.68/0.7552

Table 4: Quantitative evaluation of popular SR algorithms average PSNR/SSIM for scale factor 8

Test-dataset	Upscaling factor	Bicubic	SRCNN	FSRCNN	CSRCNN (our)
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Set-5	8	24.39/0.657	25.33/0.689	25.41/0.682	25.74/0.715
Set-14	8	23.19/0.568	23.85/0.593	23.93/0.592	24.30/0.614
BSD200	8	23.67/0.547	24.13/0.565	24.21/0.567	24.50/0.581

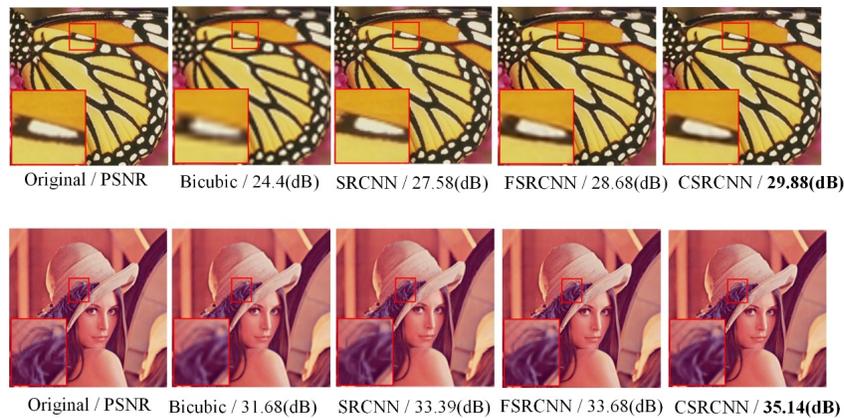


Figure 5: Visual comparisons on Set5 and Set14 datasets with upscaling factor 3. The first row is “butterfly” image from Set5 dataset, and the second row is “lenna” image from Set14 dataset

Fig. 5 shows the visual comparisons on Set5 and Set14 datasets with upscaling factor 3. As you can see, our method can accurately reconstruct the details of the image, such as texture and edge.

5 Conclusion

In this paper, we proposed a more efficient network for mining image information in different scales, named Cascaded super-resolution convolutional neural network (CSRCNN). We adopted a cascaded network structure to mine the information of image in different scales. Furthermore, we adopted L1 loss function as the loss function of each subnetwork in the training process which make the reconstructed image clearer in texture and edge. Finally, we also explored the effects of the number of cascaded FSRCNNs for the performance of our network. Extensive experiments show that the proposed method achieves satisfactory SR performance.

Acknowledgement: We gratefully acknowledge the support from NVIDIA Corporation for providing us the 2080Ti GPU that was used in this research. I am deeply thanking to my tutors and my parents for their continuous support and encouragement.

Funding Statement: This research is supported in part by the National Natural Science Foundation of China under Grant 61806099, in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20180790, in part by the Natural Science Research of Jiangsu Higher Education Institutions of China under Grant 8KJB520033, in part by Startup Foundation for Introducing Talent of Nanjing University of Information Science and Technology under Grant 2243141701077.

Conflicts of Interest: We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

- [1] F. Farhadifard, E. Abar and E. Abar, "Single image super resolution based on sparse representation via directionally structured dictionaries," in *Proc. Signal Proc. and Communications Applications Conf.*, Mersin, Turkey, pp. 1718–1721, 2014.
- [2] J. Yang, Z. Wang and Z. Lin, "Coupled dictionary training for image super-resolution," *IEEE Trans. on Image Proc.*, vol. 21, no. 2, pp. 3467–3478, 2012.
- [3] Z. L. Yuan and S. U. Cai, "Image super-resolution via sparse representation," *IEEE Trans. Image Proc.*, vol. 19, no. 4, pp. 2861–2873, 2016.
- [4] J. Ahmed, A. R. Memon and M. A. Shah, "Single image super-resolution by directionally structured coupled dictionary learning," *IEEE Trans. on Image and Video Proc.*, vol. 36, no. 1, pp. 1–12, 2016.
- [5] Z. Cui, H. Chang, S. Shan and B. Zhong, "Deep network cascade for image super-resolution," in *Proc. ICCV*, Santiago, Chile, pp. 49–64, 2014.
- [6] J. H. Choi and J. H. Kim, "Deep learning-based image super-resolution considering quantitative and perceptual quality," in *Proc. ECCV*, Munich, Germany, pp. 347–359, 2018.
- [7] C. Dong, C. C. Loy, K. He and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. ECCV*, Columbus, America, pp. 184–199, 2014.
- [8] C. Dong, C. C. Loy and X. Tang, "Accelerating the super resolution convolutional neural network," in *Proc. ECCV*, Amsterdam, Holland, pp. 391–407, 2016.
- [9] G. Zhang, H. Sun and Y. Zheng, "Optimal discriminative projection for sparse representation-based classification via bilevel optimization," *IEEE Trans. on Circuits and Syst. for Video Technology*, vol. 30, no. 2, pp. 3065–3077, 2019.
- [10] G. Zhang, Y. Zheng and G. Xia, "Domain adaptive collaborative representation based classification," *Multimedia Tools and Applications*, vol. 78, no. 1, pp. 175–196, 2019.
- [11] Y. Zheng, X. Wang and G. Zhang, "Multiple kernel coupled projections for domain adaptive dictionary learning," *Multimedia Tools and Applications*, vol. 21, no. 2, pp. 2292–2304, 2019.

- [12] P. Zhao, W. Wang and Y. Lu, "Transfer robust sparse coding based on graph and joint distribution adaption for image representation," *IEEE Trans. on Knowledge-Based Syst.*, vol. 147, no. 5, pp. 135–167, 2018.
- [13] J. Ahmed, R. Memon and M. Waqas, "Selective sparse coding based coupled dictionary learning algorithm for single image super-resolution," in *Proc. ECCV*, Munich, Germany, pp. 236–251, 2018.
- [14] J. Kim, J. K. Lee and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. CVPR*, Las Vegas, America, pp. 1646–1654, 2016.
- [15] W. S. Lai, J. B. Huang and N. Ahuja, "Fast and accurate image super-resolution with deep laplacian pyramid networks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 2599–2613, 2017.
- [16] B. Lim, S. Son and H. Kim, "Enhanced deep residual networks for single image super-resolution," in *Proc. CVPR*, Hawaii, America, pp. 136–144, 2017.
- [17] C. Ledig, L. Theis and F. Huszar, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. ICCV*, Venice, Italy, pp. 4681–4690, 2017.
- [18] X. Wang, K. Yu and S. Wu, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. CVPR*, Salt Lake City, America, pp. 2560–2572, 2018.
- [19] W. Zhang, Y. Liu and C. Dong, "RanksRGAN: Generative adversarial networks with ranker for image super-resolution," in *Proc. ICCV*, Seoul, Korea, pp. 3096–3105, 2019.
- [20] Y. Zhang, Y. Tian and Y. Kong, "Residual dense network for image super-resolution," in *Proc. CVPR*, Las Vegas, America, pp. 2472–2481, 2016.
- [21] J. H. Choi and J. H. Kim, "Deep learning-based image super-resolution considering quantitative and perceptual quality," in *Proc. CVPR*, Salt Lake City, America, pp. 347–359, 2018.
- [22] I. J. Goodfellow, J. P. Abadie and M. Mirza, "Generative adversarial nets," in *Proc. ICCV*, Barcelona, Spain, pp. 2672–2680, 2017.
- [23] Y. Peng, "Super-resolution reconstruction using multiconnection deep residual network combined an improved loss function for single-frame image," *Multimedia Tools and Applications*, vol. 78, no. 1, pp. 123–135, 2019.
- [24] R. Timofte, V. De and L. V. Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. ICCV*, Sydney, Australia, pp. 1920–1927, 2013.
- [25] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *Proc. ICCV*, Santiago, Chile, pp. 2794–2802, 2015.
- [26] K. D. Meyer and M. W. Spratling, "Multiplicative gain modulation arises through unsupervised learning in a predictive coding model of cortical function," *Neural Computation*, vol. 23, no. 6, pp. 1536–1567, 2011.
- [27] J. J. Choi and D. Laibson, "Reinforcement learning and savings behavior," *Finance*, vol. 64, no. 6, pp. 136–155, 2010.
- [28] R. Timofte, V. De, L. Smet and V. Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. CVPR*, Columbus, USA, pp. 111–126, 2014.
- [29] S. J. Yun, J. Choi, Y. Yoo and K. Yun, "Action decision networks for visual tracking with deep reinforcement learning," in *Proc. CVPR*, Hawaii, USA, pp. 2711–2720, 2017.
- [30] Z. Wang, J. Chen and S. C. H. Hoi, "Deep learning for image super-resolution: A survey," in *Proc. CVPR*, Long Beach, USA, pp. 2520–2538, 2019.
- [31] K. He, X. Zhang and S. Ren, "Deep residual learning for image recognition," in *Proc. CVPR*, Las Vegas, America, pp. 770–778, 2016.
- [32] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, Zurich, Switzerland, pp. 818–833, 2014.
- [33] K. He, X. Zhang, J. Sun and S. Ren, "Delving deep into rectifiers: Surpassing human-level performance on image-net classification," in *Proc. ICCV*, Santiago, Chile, pp. 1026–1034, 2015.
- [34] C. Yi and J. Huang, "Semismooth newton coordinate descent algorithm for elastic-net penalized huber loss regression and quantile regression," *Statistics*, vol. 26, no. 2, pp. 547–557, 2015.
- [35] J. Pang, W. Sun and S. Ren, "Cascade residual learning: A two-stage convolutional neural network for stereo matching," in *Proc. CVPR*, Hawaii, America, pp. 887–895, 2017.
- [36] Z. Wang, D. Liu, J. Yang, W. Han and T. Huang, "Deeply improved sparse coding for image super-resolution," in *Proc. ICCV*, Santiago, Chile, pp. 370–378, 2015.

- [37] M. Bevilacqua, A. Roumy, A. Guillelot, C. Morel and M. L. A, “Low-complexity single-image super-resolution based on nonnegative neighbor embedding,” in *Proc. BMVC*, Surrey, British, pp. 135.1–135.10, 2012.
- [38] R. Zeyde and M. Elad and M. Protter, “On single image scale-up using sparse-representations,” in *Proc. Curves and Surfaces–7th Int. Conf.*, Avignon, France, pp. 711–730, 2010.
- [39] 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 *Proc. ICCV*, Vancouver, British Columbia, Canada, pp. 416–423, 2001.
- [40] Z. Wang, A. C. Bovik and H. R. Sheikh, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. on Image Proc.*, vol. 13, no. 2, pp. 600–612, 2004.