Single Image Super-Resolution Based on Deep Learning and Gradient Transformation

Jingxu Chen, Xiaohai He*, Honggang Chen, Qizhi Teng, Linbo Qing
College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China
*Email: nic5602@scu.edu.cn

Abstract—**In this paper, an effective single image super-resolution method based on deep learning and gradient transformation is proposed. Firstly, the low-resolution image is upscaled by convolutional neural network. Then we calculate the gradients of the upscaled image, and transform them into desired gradients by using gradient transformation network. The transformed gradients are utilized as a constraint to establish the reconstruction energy function. Finally, we optimize this energy function to estimate the high-resolution image. Experimental results show that our proposed algorithm can produce sharp high-resolution images with few ringing or jaggy artifacts, and our results have high values of the objective assessment parameters.**

Keywords—super-resolution; convolutional neural network; gradient transformation; reconstruction energy function

I. Introduction

Single image super-resolution is one of the research focuses in digital image processing field. It aims at reconstructing a high-resolution (HR) image from a low-resolution (LR) image. Super-resolution is an ill-posed problem because just a few information is provided, and there are three kinds of methods to solve it: interpolation methods, reconstruction-based methods, and learning-based methods. In this paper, we focus on learning-based methods.

Learning-based methods usually utilize the common priors and the spatial similarities between LR images and HR images to build a corresponding mapping, and then apply this mapping to predict the HR images from these LR images. Freeman [1] firstly put forward the concept of learning-based super-resolution and proposed the example-based method. This method establishes a mapping between LR image blocks and HR image blocks by using the Markov network. Yang [2] presented the sparse coding method. This method first uses sample image database to train an over-complete dictionary with the assumption that the LR blocks and the corresponding HR blocks have the same sparse coefficients in the over-complete dictionary. In the process of reconstruction, overlapping blocks are first cropped from the LR image and encoded in the over-complete dictionary by sparse representation method to get the sparse coefficients. Then the corresponding HR blocks are reconstructed by the over-complete dictionary with the sparse coefficients, and these HR blocks are crowded to produce the final HR image. The Anchored Neighborhood Regression (ANR) method, introduced by Timofte [3], combines the sparse coding and neighbor embedding. This method generates a mapping function in advance, and then the super-resolution is simplified as the product of LR image and the mapping matrix. It greatly improves the speed of the super-resolution while the reconstructed images still have good restoration quality. Timofte [4] developed the Adjusted Anchored Neighborhood Regression method (A+) to obtain better performance of super-resolution.

In recent years, deep learning is more and more popular. For example, convolutional neural network (CNN) is used for denoising of natural image by Jain [5] and removing dirt and rain of single image by Eigen [6]. Quijas [7] trained a set of neural networks to remove the blocking artifacts of the compressed images. Osendorfer [8] presented a super-resolution method that uses CNN to approximate the sparse coding. Cui [9] proposed a super-resolution method that utilizes deep network cascade to upscale LR images layer by layer.

More recently, Dong [10] designed a novel method that uses CNN for super-resolution (CNN-SR). It trains a mapping between LR images and HR images, and this mapping is set as a deep CNN with three layers, including the layer of patch extraction and representation, the layer of non-linear mapping, and the layer of reconstruction. Although the structure of CNN-SR is very simple, it reaches state-of-the-art results compared with the existing super-resolution methods. Furthermore, because of the simple structure, CNN-SR is efficient and suitable for practical online usage. Dong also explained that the conventional sparse coding super-resolution methods can also be seen as a CNN, but CNN-SR optimizes all the operations in the training phase while sparse coding super-resolution methods do not. Consequently, the results of CNN-SR are better than the conventional sparse coding super-resolution methods.

The network of CNN-SR is only built with common architectures, so the reconstructed images of it generally suffer from some ringing and jaggy artifacts. One of the methods to suppress artifacts is enforcing prior to constrain the estimation. Sun [11] proposed a generic image prior—gradient profile prior. Based on this novel prior, a super-resolution method is developed with the constraint of gradient fields. This method can effectively remove artifacts of the reconstructed HR image.

978-1-5090-1345-6/16/$31.00 ©2016 IEEE

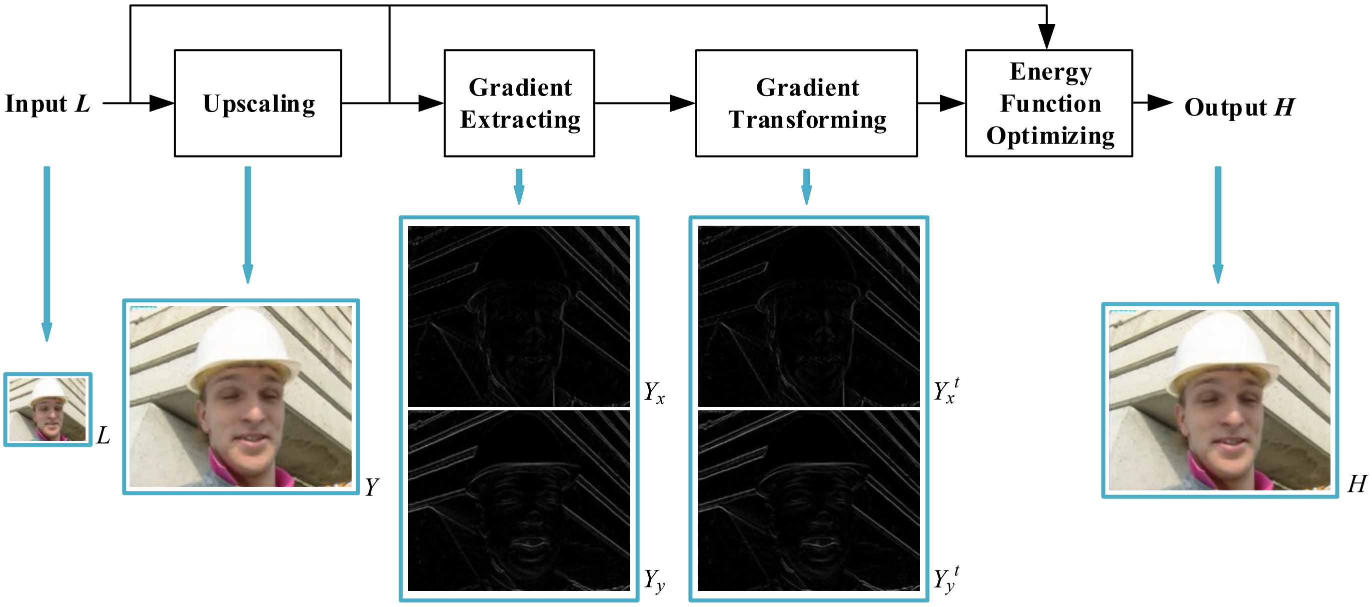


Fig. 1. Schematic diagram of the proposed method. We first upscale the LR image L. Next, we calculate the horizontal and vertical directions gradients Y_x and Y_y of the upscaled image Y, and transform them by using gradient transformation network. Finally, the HR image H is reconstructed from L, Y and the transformed gradients Y_x^t and Y_y^t by optimizing the energy function.

However, it cannot restore details and fine structures. To estimate the HR image with more fine structures but less artifacts, we propose a single image super-resolution method based on deep learning and gradient transformation in this paper. Firstly, convolutional neural networks in gradient field are trained to transform the gradients of upscaled image into the desired gradients, which are clearer and sharper. Then, the transformed gradients are utilized to construct a regularization term to constrain the HR image under the MAP-based reconstruction framework. Experimental results show that the reconstructed HR images of our method achieve high restoration quality with few ringing or jaggy artifacts.

II. Super-Resolution Based on Deep Leaning and Gradient Transformation

In this section, we will introduce the proposed single image super-resolution method. Fig. 1 shows the schematic diagram of our approach.

For a LR image L, we first use CNN-SR to upscale it and denote the upscaled image as Y. Then we calculate horizontal and vertical directions gradients Y_x and Y_y of Y by gradient operator, and each of the gradients is transformed into the desired gradient by using gradient transformation network individually. We denote the transformed gradients as Y_x^t and Y_y^t. Finally, the corresponding HR image H is reconstructed from L, Y, Y_x^t and Y_y^t by optimizing the energy function that is composed of reconstruction constraint and gradient constraint. We will describe the details of these steps respectively in the following subsections.

A. *Convolutional Neural Network for Gradient Transformation*

To transform the gradients of Y into the desired gradients, which are closer to the real gradients of ground truth X, the convolutional neural network is utilized for gradient transformation. We name the proposed network model as gradient transformation network. Our network structure consists of three convolutional layers. We use the horizontal direction gradient Y_x as an example to describe these layers' details, and the transformation of vertical direction gradient Y_y is the same.

For the input gradient Y_x, we use the first convolutional layer L_1 to extract the feature of each gradient patches. L_1 is composed of m_1 filters, and the spatial size of each filter is $s_1 \times s_1$. The extracted feature representation f is comprised of m_1 feature maps. The second convolutional layer L_2 is designed for mapping the extracted feature representation f into the transformed feature representation f_t that has m_2 dimensions. L_2 includes m_2 filters of size $m_1 \times s_2 \times s_2$. The transformed feature representation f_t will be used for producing the transformed gradient. The last convolutional layer L_3 is applied on f_t to produce the final output of our network. L_3 corresponds to one filter with spatial size $m_2 \times s_3 \times s_{3.}$ The output of L_3 is the desired transformed gradient Y_x^t that will be utilized for reconstructing the final HR image.

Furthermore, the Rectified linear units (ReLU), introduced by Nair [12], makes the convergence of training much faster, so we apply it on the filter responses in the training phase. All above layers form a convolutional neural network.

Mean squared error (MSE) is utilized as the loss function to train our network:

$$\min_{\Theta} \sum_{n} \| N(G_l^n;\Theta) - G_h^n \|_2^2 \qquad (1)$$

where G_l^n and G_h^n are the n-th training pairs. $N(G_l^n;\Theta)$ stands for the gradient block transformed by gradient transformation network with parameter Θ. We use the standard back-propagation algorithm [13] to minimize this loss function.

B. *Reconstruction of High-Resolution Image*

After we get the transformed gradients Y_x^t and Y_y^t, the corresponding HR image H can be reconstructed by minimizing the following energy function:

$$E(H\,|\,L,\nabla Y^t)=E_1(H\,|\,L)+\theta E_2(\nabla H\,|\,\nabla Y^t) \qquad (2)$$

where ∇Y^t denotes the transformed gradients Y_x^t and Y_y^t which are obtained in the previous subsection. ∇H stands for the gradients of output HR image H.

The first term $E_1(H\,|\,L)$ is the constraint of image field, and it demands that the down-sampled image of output H should be similar to the input L:

$$E_1(H\,|\,L)=\|H\downarrow -L\|_2^2 \qquad (3)$$

where $\downarrow$ is the down-sampling operation.

The second term $E_2(\nabla H\,|\,\nabla Y^t)$ is the constraint of gradient field, and it keeps the consistency between the gradients of output HR image H and the transformed gradients:

$$E_2(\nabla H\,|\,\nabla Y^t)=\|\nabla H-\nabla Y^t\|_2^2 \qquad (4)$$

The minimum of the energy function can be obtained by gradient descent algorithm:

$$H^{i+1}=H^{i}-\mu((H^{i}\downarrow -L)\uparrow -\theta\cdot(\nabla^2 H-\nabla^2 Y^t)) \qquad (5)$$

where H^i is the output of the i-th iteration. μ is the iteration step size. θ is the weight factor between two constraints. The parameter settings will be introduced in the following section.

III. Experiments

A. *Experiments I: Gradient Transformation Network*

The gradient transformation network is trained by Caffe package [14]. We adopt the BSDS500 database [15] as the training set. Particularly, we use its test-set and training-set for training and its val-set for testing. For each image in our training set, we first down-sample it and use CNN-SR to upscale it with upscaling factor 3. Then we calculate gradients from the upscaled image and divide them into 36 × 36 blocks G_l. Given a 36 × 36 block as input, gradient transformation network produces a 20 × 20 block as output to avoid border effects. So the corresponding real gradients are calculated from ground truth image and divided into 20 × 20 blocks G_h. G_l and G_h compose the training pairs $\{G_l, G_h\}$. The settings of convolutional natural network are selected by experience, i.e., $s_1 = 9$, $s_2 = 5$, $s_3 = 5$, $m_1 = 64$, and $m_2 = 32$. In order to get better results, the number of iterations is set as 10,000,000. We have to note that the gradient transformation networks for horizontal and vertical directions are trained separately.

Test images are listed in Fig. 2, and parts of transformed results of our network in the measures of PSNR and SSIM (the structure similarity index [16]) are illustrated in Table I. Fig. 3 shows one of the transformed gradients. As we can see, whether the horizontal direction or the vertical direction, the transformed gradients are closer to the real ones of ground truth image and have higher PSNR and SSIM scores compared with the untransformed gradients.

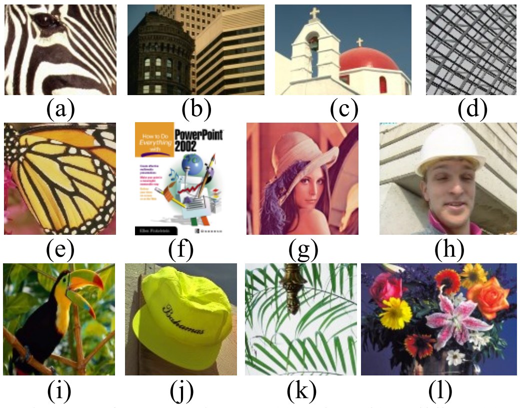


Fig. 2. Test images of our experiments. (a) Zebra. (b) Building. (c) House. (d) Rebar. (e) Butterfly. (f) Ppt3. (g) Lena. (h) Foreman. (i) Bird. (j) Hat. (k) Leaves. (l) Flowers.

TABLE I. The Transformation Results Of Gradients

Images	**Untransformed**	**Transformed**
PSNR (dB) / SSIM of Vertical Direction Gradients		
Zebra	30.12 / 0.8900	**30.76 / 0.8979**
Building	32.03 / 0.7743	**32.43 / 0.7832**
House	37.03 / 0.9224	**37.42 / 0.9249**
Rebar	27.49 / 0.7842	**28.19 / 0.8207**
PSNR (dB) / SSIM of Horizontal Direction Gradients		
Zebra	31.93 / 0.8818	**32.26 / 0.8889**
Building	29.94 / 0.7640	**30.50 / 0.7820**
House	36.28 / 0.9268	**36.80 / 0.9312**
Rebar	27.05 / 0.7823	**27.75 / 0.8161**

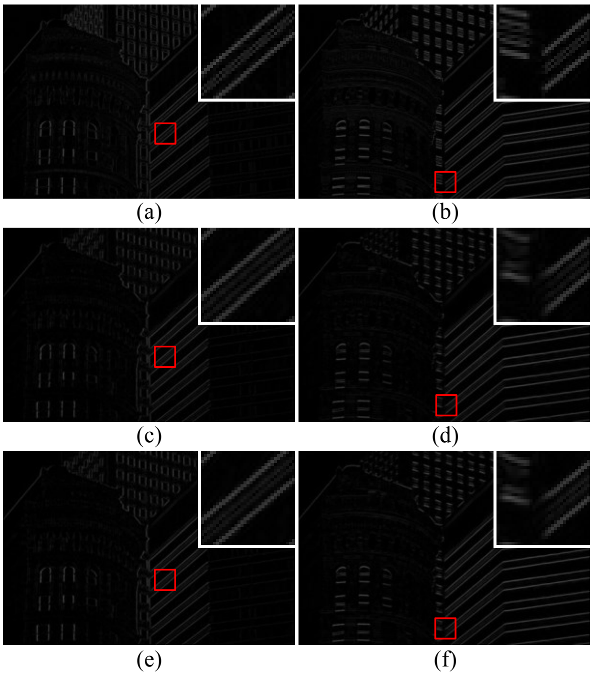


Fig. 3. Gradient transformation of "Building". (a) Original horizontal direction gradient. (b) Original vertical direction gradient. (c) Untransformed horizontal direction gradient (PSNR= 32.03 dB; SSIM= 0.7743). (d) Untransformed vertical direction gradient (PSNR= 29.94 dB; SSIM= 0.7640). (e) Transformed horizontal direction gradient (PSNR= 32.43 dB; SSIM= 0.7832). (f) Transformed vertical direction gradient (PSNR= 30.50 dB; SSIM= 0.7820).

B. *Experiments II: Super-resolution Using the Transformed Gradients*

To reconstruct a better HR image, we use the image that is upscaled by CNN-SR as the initial image. The upscaling factor is 3. In experiments, the iteration step size μ and the weighting factor θ in Eq. (5) are set to 1.5 and 0.05 separately. We set the number of iterations to 100 to produce the best reconstruction results.

TABLE II. THE SUPER-RESOLUTION RESULTS BY DIFFERENT METHODS

Images	**Bicubic**	**SC[2]**	**A+[4]**	**CNN-SR [10]**	**Proposed**
PSNR (dB)					
Zebra	22.67	24.45	27.83	27.70	**28.51**
Building	24.04	25.28	26.03	26.59	**27.20**
House	29.08	30.70	32.32	32.95	**33.64**
Rebar	21.21	22.35	23.80	24.42	**25.30**
Butterfly	24.05	25.63	27.29	28.01	**28.88**
Ppt3	23.65	24.92	26.03	26.98	**27.20**
Lena	31.64	32.61	33.49	33.64	**33.75**
Foreman	31.54	33.10	34.76	34.58	**35.19**
Bird	32.62	34.23	35.64	35.58	**35.80**
Hat	29.20	30.38	31.22	31.10	**31.45**
Leaves	23.53	25.11	26.48	26.92	**27.60**
Flowers	27.17	28.19	28.99	29.20	**29.41**
Average	26.70	28.08	29.49	29.81	**30.33**
SSIM					
Zebra	0.8648	0.8920	**0.9448**	0.9393	0.9435
Building	0.7085	0.7606	0.7909	0.8025	**0.8207**
House	0.9104	0.9187	0.9453	0.9462	**0.9513**
Rebar	0.7957	0.8548	0.8908	0.8947	**0.9133**
Butterfly	0.8543	0.8895	0.9282	0.9286	**0.9400**
Ppt3	0.8831	0.8988	0.9384	0.9437	**0.9485**
Lena	0.8783	0.8820	0.9032	0.9044	**0.9056**
Foreman	0.9216	0.9238	0.9503	0.9483	**0.9520**
Bird	0.9398	0.9513	0.9666	0.9658	**0.9673**
Hat	0.8499	0.8666	0.8914	0.8868	**0.8924**
Leaves	0.8437	0.8950	0.9282	0.9325	**0.9412**
Flowers	0.8326	0.8588	0.8798	0.8818	**0.8854**
Average	0.8569	0.8827	0.9132	0.9146	**0.9218**
IFC					
Zebra	3.89	4.15	6.35	5.72	**6.76**
Building	3.41	3.43	4.75	4.67	**5.39**
House	2.58	2.00	3.73	3.52	**4.11**
Rebar	3.93	4.23	5.77	5.51	**6.68**
Butterfly	3.43	3.65	5.31	5.10	**6.09**
Ppt3	2.79	2.63	4.12	3.90	**4.69**
Lena	3.76	2.85	4.77	4.59	**5.02**
Foreman	3.57	2.89	4.91	4.57	**5.03**
Bird	3.73	3.53	5.15	4.74	**5.19**
Hat	2.98	2.43	**4.14**	3.78	4.04
Leaves	3.98	4.23	5.99	5.47	**6.60**
Flowers	3.31	3.24	4.60	4.37	**4.91**
Average	3.45	3.27-	4.97	4.66	**5.38**

Table II illustrates super-resolution results of our proposed method and the compared methods in the measures of PSNR, SSIM, and IFC [17]. The compared methods are as follows: Bicubic (bicubic interpolation), SC (sparse coding method of yang [2]), A+ (Adjusted Anchored Neighborhood Regression method [4]) and CNN-SR (super-resolution convolutional neural network method [10]). Figs. 4, 5, and 6 show parts of the reconstructed images by different methods. As can be seen in the experimental results, the Bicubic has the most blurred results. The results of SC are clearer than Bicubic, but still blurred. A+ produces reconstructed images with few jaggy artifacts but they are too smooth. The results of CNN-SR are clear, while there are some ringing and jaggy artifacts along edges. Our proposed method can produce clear HR images with few ringing or jaggy artifacts and they have sharp edges. Moreover, our proposed method achieves the highest scores in evaluation measures. Particularly, the average scores of our method are 0.52 dB and 0.0072 higher than the second best method, CNN-SR, in the measures of PSNR and SSIM separately. The average score of our method is 0.41 higher than the second best method, A+, in the measure of IFC.

IV. CONCLUSION

In this paper, we propose an effective algorithm for single image super-resolution based on deep learning and gradient transformation. The observed gradients calculated from the upscaled images are transformed by gradient transformation network to obtain the desired gradients that are closer to real ones. Then we utilize the transformed gradients as a constraint to reconstruct the HR image. Experimental results show that the reconstructed images of our algorithm have fine structures with few ringing or jaggy artifacts, and the objective assessment parameters of our method are higher than the compared methods.

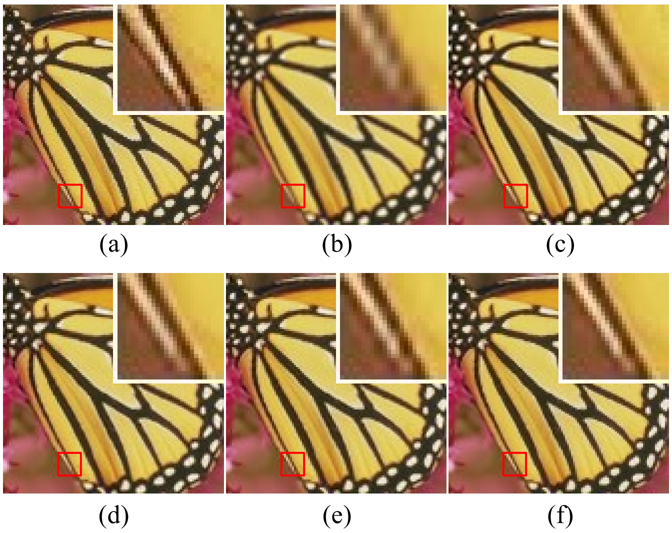
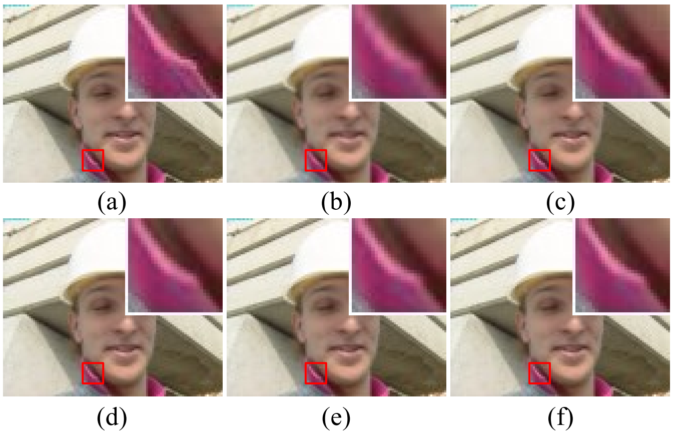


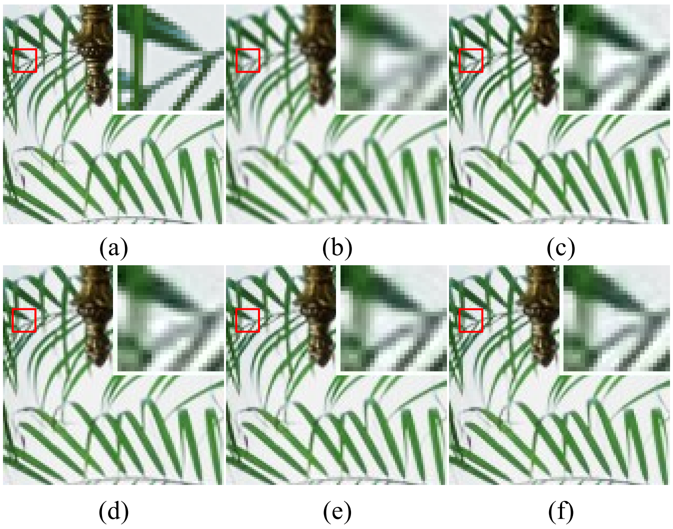
Fig. 4. Results of "Butterfly" by different methods. (a) Original. (b) Bicubic (PSNR= 24.05 dB; SSIM= 0.8543; IFC= 3.43). (c) SC [2] (PSNR= 25.63 dB; SSIM= 0.8895; IFC= 3.65). (d) A+ [4] (PSNR= 27.29 dB; SSIM= 0.9282; IFC= 5.31). (e) CNN-SR [10] (PSNR= 28.01 dB; SSIM= 0.9286; IFC= 5.10). (f) Proposed (PSNR= 28.88 dB; SSIM= 0.9400; IFC= 6.09).



(a) (b) (c)

(d) (e) (f)

Fig. 5. Results of “Foreman” by different methods. (a) Original. (b) Bicubic (PSNR= 31.54 dB; SSIM= 0.9216; IFC= 3.57). (c) SC [2] (PSNR= 33.10 dB; SSIM= 0.9238; IFC= 2.89). (d) A+ [4] (PSNR= 34.76 dB; SSIM= 0.9503; IFC= 4.91). (e) CNN-SR [10] (PSNR= 34.58 dB; SSIM= 0.9483; IFC= 4.57). (f) Proposed (PSNR= 35.19 dB; SSIM= 0.9520; IFC= 5.03).



(a) (b) (c)

(d) (e) (f)

Fig. 6. Results of “Leaves” by different methods. (a) Original. (b) Bicubic (PSNR= 23.53 dB; SSIM= 0.8543; IFC= 3.98). (c) SC [2] (PSNR= 25.11 dB; SSIM= 0.8950; IFC= 4.23). (d) A+ [4] (PSNR= 26.48 dB; SSIM= 0.9282; IFC= 5.99). (e) CNN-SR [10] (PSNR= 26.92 dB; SSIM= 0.9325; IFC= 5.47). (f) Proposed (PSNR= 27.60 dB; SSIM= 0.9412; IFC= 6.60).

Acknowledgment

This research is supported by the National Natural Science Foundation of China (No.61471248), the Applied Basic Research Project of Sichuan Province (No.2015JY0189) and the Innovation Project on the Education for Graduate Students of Sichuan Province (No.2014-Edu.-034).

References

[1] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution." IEEE Computer Graphics and Applications, vol. 22, no. 2, pp. 56-65, 2002.

[2] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation." IEEE Transactions on Image Processing, vol. 19, no. 11, pp. 2861-2873, 2010.

[3] R. Timofte, V. D. Smet, and L. V. Gool, "Anchored Neighborhood Regression for Fast Example-Based Super-Resolution." IEEE International Conference on Computer Vision, pp. 1920-1927, 2013.

[4] R. Timofte, V. D. Smet, and L. V. Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution." Computer Vision--ACCV 2014. Springer International Publishing, pp. 111-126, 2014.

[5] V. Jain, and H. S. Seung, "Natural Image Denoising with Convolutional Networks, " Advances in Neural Information Processing Systems, pp. 769-776, 2008.

[6] D. Eigen, D. Krishnan, and R. Fergus, "Restoring an Image Taken through a Window Covered with Dirt or Rain." IEEE International Conference on Computer Vision, pp. 633-640, 2013.

[7] J. Quijas, and O. Fuentes, "Removing JPEG blocking artifacts using machine learning." IEEE Southwest Symposium on Image Analysis and Interpretation, pp. 77-80, 2014.

[8] C. Osendorfer, H. Soyer, and P. V. D. Smagt, "Image Super-Resolution with Fast Approximate Convolutional Sparse Coding." Neural Information Processing. Springer International Publishing, pp. 250-257, 2014.

[9] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, "Deep network cascade for image super-resolution." Computer Vision–ECCV 2014. Springer International Publishing, pp. 49-64, 2014.

[10] C. Dong, C. C. Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 2, pp. 295-307, 2016.

[11] J. Sun, J. Sun, Z. Xu, and H. Shum, "Gradient profile prior and its applications in image super-resolution and enhancement." IEEE Transactions on Image Processing, vol. 20, no. 6, pp. 1529-1542, 2011.

[12] V. Nair, and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines." Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807-814, 2010.

[13] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition." Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] Y. Jia, E. Shelhamer, et al, "Caffe: Convolutional architecture for fast feature embedding." Proceedings of the ACM International Conference on Multimedia. ACM, pp. 675-678, 2014.

[15] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 5, pp. 898-916, 2011.

[16] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity. " IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600 – 612, 2004.

[17] H. R. Sheikh, A. C. Bovik, and G. D. Veciana, "An information fidelity criterion for image quality assessment using natural scene statistics." IEEE Transactions on Image Processing, vol. 14, no. 12, pp. 2117-2128, 2005.