Image Super-resolution Reconstruction based on Deep Learning and Sparse Representation

Qian LEI[1], Zhao-hui ZHANG[2,*] and Cun-ming HAO[3]

[1] SJZ JKSS Technology Co., Ltd., Shijiazhuang 050081, China

[2] School of mathematics and information science, Hebei Normal University, Shijiazhuang 050024, China

[3] Institute of Applied Mathematics, Hebei Academy of Sciences,

Shijiazhuang 050081, China

Keywords: Super-resolution, Deep learning, Denoising auto-encoders, Joint dictionary learning, Sparse Representation.

Abstract. This paper addresses the problem of super-resolution(SR) image reconstruction based on sparse representation and deep learning. we approached this problem from the dictionary learning. Firstly, in order to realize the correspondence between the sparse representation coefficients, we proposed the method of joint dictionary learning based on Sparse Denoising Auto-Encoders(NSDAE). Secondly, at the stage of reconstruction, in order to achieve high frequency compensation, we proposed the algorithm of iterative error back projection. Finally, experimental results show that the recovered high-resolution image is competitive in quality to images produced by other SR methods.

Introduction

Image super-resolution(SR) reconstruction [1] is the process of generating a high resolution image from one or more low-resolution images with software technology. Yang et al. [2] realized SR from the perspective of compressed sensing. The SR algorithm based on sparse representation is developing rapidly and it avoids the artificial selection for the number of image blocks. However, the process of dictionary learning is computationally large and slow; and most of the learned dictionaries do not take into account the consistency of sparse coefficients between high and low resolution image blocks. Therefore, how to learn a dictionary with better universality from the large training set becomes the difficulty in recent research.

The concept of Deep learning [3][4] comes from the artificial neural network, the model of deep learning can learn more effective features: Restricted Boltzmann Machines(RBM), Deep Belief Networks (DBN), Convolutional Neural Networks (CNN), and Auto-Encoders (AE). Auto-Encoders (AE) is a typical form of deep learning model: the nonlinear mapping from the visible layer to the hidden layer can be regarded as the nonlinear feature dimensionality reduction of the input; the nonlinear mapping from the hidden layer to the output layer is equivalent to the reconstruction of the input. Considering the non-negativity of the image itself and the denoising of SR, the paper proposes a NSDAE-based joint dictionary learning algorithm.

Image Super-resolution Reconstruction Based on Sparse Representation

The key idea of SR based on sparse representation [5] is as follows: suppose $\boldsymbol{x} \in \Re^{\mathrm{m}}$ is an image block extracted from a high-resolution image, $\boldsymbol{y} \in \Re^{\mathrm{n}}$ is the corresponding low-resolution image block extracted from the same position. $\boldsymbol{D}_h$ and $\boldsymbol{D}_l$ is the high-resolution and low-resolution dictionaries. For the input image block $\boldsymbol{y}$, we firstly solve the equation $\boldsymbol{y} \approx \boldsymbol{D}_l \boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \Re^{\boldsymbol{k}}, \|\boldsymbol{\alpha}\|_0 << k$ is the sparse coefficient vector. Then we can get the corresponding high-resolution image block $\boldsymbol{x}$ by solving the equation: $\boldsymbol{x} \approx \boldsymbol{D}_h \boldsymbol{\alpha}$.

Joint Dictionary Learning based on NSDAE

Auto-Encoders(AE)

AE [6] can be regarded as a neural network model that makes the target output equal to the input. Figure 1 shows the AE network architecture with a hidden layer. The AE model consists of a coding module and a decoding module. The coding module maps the input $\boldsymbol{x} \in [0,1]^d$ to the hidden layer as a representation vector $\boldsymbol{y} \in [0,1]^{d'}$:

$$\boldsymbol{y} = f_\theta(\boldsymbol{x}) = s(\boldsymbol{Wx} + \boldsymbol{b}), \tag{1}$$

where $s(x) = 1/(1+e^{-x})$, $\theta = \{\boldsymbol{W}, \boldsymbol{b}\}$ is the parameter, $\boldsymbol{W}$ is the weight matrix, $\boldsymbol{b}$ is the bias vector. Then, $\boldsymbol{y}$ is remapped at the decoding module to obtain a reconstruction of the input vector $\boldsymbol{x}$, i.e. the decoded output vector $\boldsymbol{z} \in [0,1]^d$:

$$\boldsymbol{z} = g_{\theta'}(\boldsymbol{y}) = s(\boldsymbol{W}'\boldsymbol{y} + \boldsymbol{b}'), \tag{2}$$

where $\theta' = \{\boldsymbol{W}', \boldsymbol{b}'\}$ is the parameter, according to the nature of AE weight tied [7], we can see that $\boldsymbol{W}' = \boldsymbol{W}^{\mathrm{T}}$.

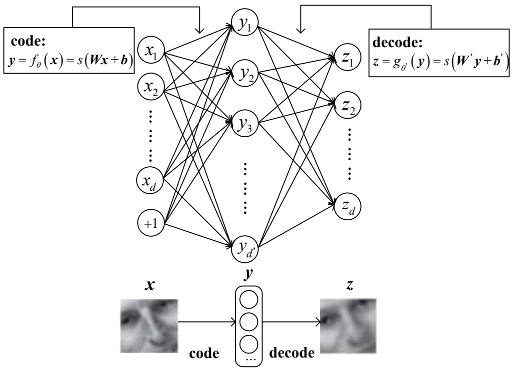

Figure 1. AE network architecture with a hidden layer

Denoising Auto-Encoders(DAE)

The DAE [8] is an improvement to the traditional AE: adding noise to the input. For SR, the training samples are achieved by doing degradation to the high-resolution images, so the reconstruction of high-resolution images is equivalent to denoising. So the improved model can learn more robust features and effectively suppress the noise. Figure 2 shows the DAE network architecture.

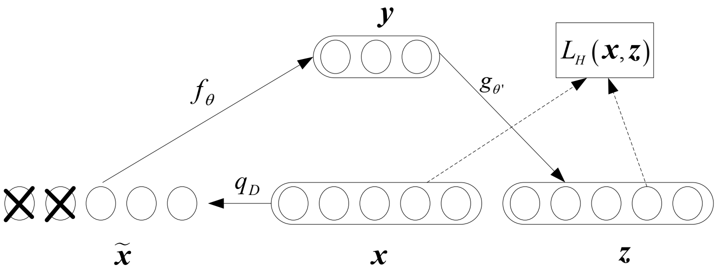

Figure 2. The DAE network architecture

The training of DAE is similar to that of traditional AE. Firstly, get the degraded input $\bar{\boldsymbol{x}}$ from the initial input $\boldsymbol{x}$, i.e. $\bar{\boldsymbol{x}} \sim q_D\left(\bar{\boldsymbol{x}} \mid \boldsymbol{x}\right)$, where q is a random mapping function. Then the degraded input $\bar{\boldsymbol{x}}$ is mapped to the hidden layer:

$$\boldsymbol{y} = f_\theta\left(\bar{\boldsymbol{x}}\right) = s\left(\boldsymbol{W}\,\bar{\boldsymbol{x}} + \boldsymbol{b}\right), \tag{3}$$

finally, decode module gives the reconstruction result:

$$\boldsymbol{z} = g_{\theta'}(\boldsymbol{y}) = s\left(\boldsymbol{W}'\boldsymbol{y} + \boldsymbol{b}'\right), \tag{4}$$

Non-negtive Sparse Denoising Auto-Encoders(NSDAE)

Non-negativity. Image itself has the characteristics of non-negativity, so the paper combines the non-negativity with AE. For the non-negativity of the input, map the samples to $[0,1]$ with normalization. For the non-negativity of the network weights, an asymmetric decay mechanism is introduced. Note: $\boldsymbol{x}^c \in \Re^D$ is the original clean input, $\boldsymbol{x} \in \Re^D$ is the degraded form; $\boldsymbol{W}^{inp}$ and $\boldsymbol{W}^{out}$ are the encoding and decoding matrices, and satisfy the relation: $\boldsymbol{W} = \boldsymbol{W}^{inp} = \left(\boldsymbol{W}^{out}\right)^{\mathrm{T}}$. $\boldsymbol{f}(\cdot)$ is a parameterized nonlinear activation function, and denoted as:

$$f_i\left(g_i \mid a_i, b_i\right) = \frac{1}{1 + e^{(-a_i g_i - b_i)}} \in [0,1], \tag{5}$$

where a_i is slope, b_i is bias, $\boldsymbol{g} = \boldsymbol{W}^{inp}\boldsymbol{x}$ is the input to hidden layer. The reconstructed result of the network can be expressed as follows:

$$\hat{\boldsymbol{x}} = \boldsymbol{W}^{out}\boldsymbol{f}\left(\boldsymbol{W}^{inp}\boldsymbol{x}\right) = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{f}\left(\boldsymbol{W}\boldsymbol{x}\right). \tag{6}$$

The network takes the reconstruction error as the target function:

$$E = \frac{1}{K}\sum_{i=1}^{K}\left\|\boldsymbol{x}_i^c - \hat{\boldsymbol{x}}_i\right\|^2, \tag{7}$$

where K is the number of training samples, $\boldsymbol{x}_i^c$ is the clean non-negative sample i.

Introduce the weight decay to target function Eq. (7) by adding the regular term $\lambda\|\boldsymbol{W}\|^2$, and update the network parameters based on the gradient descent method. The weight decay term is $d\left(w_{ij}\right) = -\lambda w_{ij}$, and the online error correction rule is:

$$\Delta w_{ij} = \eta\left(x_i - \hat{x}_i\right)h_j + d\left(\tilde{w}_{ij}\right). \tag{8}$$

Where w_{ij} is the weight between hidden layer neuron j to the output layer neuron i, h_j is the activation of neuron j and η is the adaptive learning rate. Eq. (8) is used to emphasize the non-negativity of the weights, and it is a non-symmetric, piecewise linear decay function:

$$d\left(\tilde{w}_{ij}\right) = \begin{cases} -\alpha \tilde{w}_{ij} & if\ \tilde{w}_{ij} < 0 \\ -\beta \tilde{w}_{ij} & else \end{cases}, \tag{9}$$

where $\tilde{w}_{ij} = w_{ij} + \Delta w_{ij}$ is the new computed weight after error correction according to Eq. (9), α and β are parameters which control the sign of weight.

Sparsity. In order to ensure sparse coding [9], we introduce the Intrinsic Plasticity(IP) mechanism proposed in 2005 by Triesch [10]. Assuming the input x satisfies the distribution: $f_x(x)$, and the corresponding output is:

$$h = f(x) = \frac{1}{\exp(-(ax+b))}. \tag{10}$$

where f is nonlinear activation function, the distribution of h is:

$$f_h(h) = f_x(x) / \left(\frac{\partial h}{\partial x}\right). \tag{11}$$

The core idea of IP is adjusting the parameters a and b to minimize the divergence distance between $f_h(h)$ and the expected exponential distribution $f_{\exp}(h) = \frac{1}{\mu}e^{-\frac{h}{\mu}}$, so as to optimize the information transmission of the neuron. Where μ is the average activation level of the output. The update rule for the parameters a and b is:

$$\begin{cases} a = a + \Delta a & \Delta a = \eta_{IP}\left(\frac{1}{a} + x - \left(2 + \frac{1}{\mu}\right)xh + \frac{1}{\mu}xh^2\right) \\ b = b + \Delta b & \Delta b = \eta_{IP}\left(1 - \left(2 + \frac{1}{\mu}\right)h + \frac{1}{\mu}h^2\right) \end{cases}, \tag{12}$$

where η_{IP} is the IP learning rate.

Joint Dictionary Learning

The NSDAE-based joint dictionary learning [11] network model is built as follows:

(1) The number of hidden nodes corresponds to the number of atoms in the dictionary;

(2) The input of the model is divided into two parts: the image sub-blocks extracted from the upsampled version of the low-resolution images, and the features extracted from the corresponding low-resolution image sub-blocks;

(3) The target output of the model is the original high-resolution image sub-blocks and the features extracted from the corresponding low-resolution image sub-blocks.

Suppose $\boldsymbol{D}_h \in K \times m$ is the weight matrix from the first part of the input to the hidden layer; $\boldsymbol{D}_l \in K \times n$ is the weight matrix from the second part of the input to the hidden layer; so the weight matrix $\boldsymbol{W} = (\boldsymbol{D}_h, \boldsymbol{D}_l)^{\mathrm{T}}$ can be regarded as the learned dictionary. The structure of the joint dictionary learning network is shown in Figure 3.

In Figure 3, suppose visible layer: $\boldsymbol{input} = (\boldsymbol{x}, \boldsymbol{y})^{\mathrm{T}}$, target output: $\boldsymbol{u}^p = (\boldsymbol{o}, \boldsymbol{y})^{\mathrm{T}}$, hidden layer output: $\boldsymbol{h} = 1/\left(1 + e^{-a.*(\boldsymbol{W} \cdot \boldsymbol{input}) - \boldsymbol{b}}\right)$, the actual output: $\boldsymbol{u}^o = \boldsymbol{Wh}$, error vector: $\boldsymbol{e} = \boldsymbol{u}^p - \boldsymbol{u}^o$. The entire joint dictionary learning process is summarized as Algorithm 1.

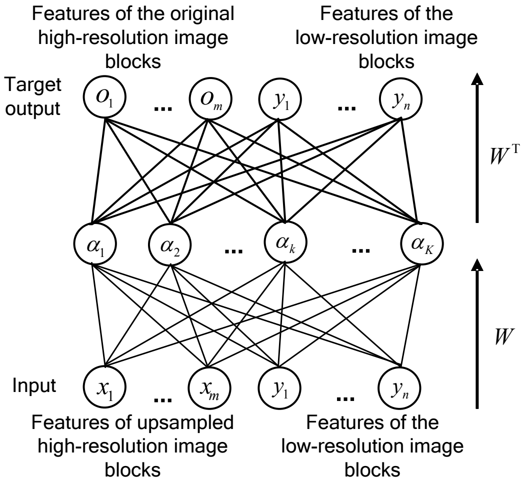
Features of the original high-resolution image blocks
Features of the low-resolution image blocks
Target output
o_1 ... o_m y_1 ... y_n
W^{T}
α_1 α_2 ... α_k ... α_K
W
Input
x_1 ... x_m y_1 ... y_n
Features of upsampled high-resolution image blocks
Features of the low-resolution image blocks

Figure 3. The structure of the joint dictionary learning network

Algorithm 1 (joint dictionary learning)

1: **Input:** training samples $\left\{(\boldsymbol{x}_t, \boldsymbol{y}_t)_{t=1}^{T}\right\}$, target output $\boldsymbol{u}^p = (\boldsymbol{o}, \boldsymbol{y})^{\mathrm{T}}$.

2: **Initialization:** learning rate η_0, η_{IP}, regularization factor η_g, decay factor α and β, average output value of activation function μ; parameters $\boldsymbol{W}$, a, and b.

3: **Network training:** target function:

$$[\boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b}] = \arg\min_{\boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b}} \frac{1}{T} \sum_{i=1}^{T} L\left(\boldsymbol{u}^{p(i)}, \boldsymbol{u}^{o(i)}\right) = \arg\min_{\boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b}} \frac{1}{T} \sum_{i=1}^{T} \left\| \boldsymbol{u}^{p(i)} - \boldsymbol{u}^{o(i)} \right\|_2^2$$

updata parameters according Eq. (8), Eq. (9), Eq. (10).

4: **Output:** dictionary $\boldsymbol{W} = (\boldsymbol{D}_h, \boldsymbol{D}_l)^{\mathrm{T}}$, slope and bias a and b.

Image Super-resolution Reconstruction based on Sparse Representation

(1) The reconstruction of high-resolution image sub-blocks.

For a sub-block $\boldsymbol{y}$ in a low-resolution image $\boldsymbol{Y}$, get the the sparse representation coefficient vector $\boldsymbol{q}$ by solving the follow sparse representation model:

$$\min_{\boldsymbol{q}} \|\boldsymbol{y} - \boldsymbol{D}_l \boldsymbol{q}\| + \lambda \|\boldsymbol{q}\|_1, \tag{13}$$

where $\|\boldsymbol{q}\|_1$ is 1-norm of vector $\boldsymbol{q}$. Based on the assumption of sparse description consistency, we reconstruct the high-resolution image sub-block as follow:

$$\boldsymbol{x} = \boldsymbol{D}_h \boldsymbol{q} . \tag{14}$$

(2) Generation of high-resolution image $\boldsymbol{X}_h$ by splicing reconstructed sub-blocks.

(3) Global high frequency compensation.

In step (2), the smoothing operation of the overlapped region of neighboring sub-blocks leads to the loss of detail information. The paper introduces the method of iterative error back projection based on the residual image to realize high frequency compensation. The process is summarized as Algorithm 2.

Algorithm 2 (iterative error back projection based on the residual image)

```
1: Input: original low-resolution image $\boldsymbol{Y}$ , initial high-resolution image $\boldsymbol{X}_h$ ;
2: Construct and normalize the gauss filter template size of $\boldsymbol{P} = \lfloor P_{ij} \rfloor_{k \times k}$ .
3: For $t = 1,2,...,T$
       down-sampling: $\boldsymbol{X}_{down} = downsample(\boldsymbol{X}_h)$ ;
       difference image: $\boldsymbol{C} = \boldsymbol{Y} - \boldsymbol{X}_{down}$ ;
       up-sampling: $\boldsymbol{D} = upsample(\boldsymbol{C})$ ;
       convolution: $\boldsymbol{X}_h = \boldsymbol{D} * \boldsymbol{P} + \boldsymbol{X}_h$ .
   End
4: Output: super-resolution image $\boldsymbol{X} = \boldsymbol{X}_h$ .
```

Experiment

Experimental Settings

For the joint dictionary learning, the training samples are all searched from the network, including landscapes, people, natural images and so on. In the experiment, the dataset size is 200000, and the sampling factor is uniformly set. The filters used to extract features from the low-resolution image are:

$$f_1 = [-1, 0, 1], \ f_2 = [-1, 0, 1]'$$

$$f_2 = [-1, 0, -2, 0, 1], \ f_2 = [-1, 0, -2, 0, 1]'$$

The size of dictionary is 512, 768, 1024, the size of the image sub-block is set to 5×5, 6×6, 7×7, 8×8; The width of the overlap between blocks is 4, 5, 6, 7, 8. In the stage of dictionary learning, the iteration number of the dataset is 100; the iteration number of back projection is 20. We will study the effects of dictionary dimension, size of image sub-block and overlap width between image sub-blocks on the quality of reconstructed image. Root Mean Square Error (RMSE), Peak Signal to Noise Ratio(PSNR), and Structural Similarity Measure (SSIM) as the index.

Experimental Results

Figure 4 shows the SR results of different dictionary dimensions. The size of the image sub-blocks is 5×5; Table 1 lists the quality evaluation results. Figure 5 shows the SR results of different sub-block size; Table 2 lists the quality evaluation results. Figure 6 shows the SR results of different overlap width; Table 3 lists the quality evaluation results. All the size of the dataset is set to 200000.

Experimental results verify the following conclusions. The higher the dictionary dimension, the higher the quality of reconstructed image, but also the longer of the time. The size of image sub-block should be moderate, too small or too big will lead to the quality degradation of the results. The overlap width between adjacent sub-blocks will increase the image quality, but the time will gradually increase. At last, the parameters selected are as follows: the size of dictionary is 1024, the size of sub-block is 6×6, the overlap width is 5 pixels.

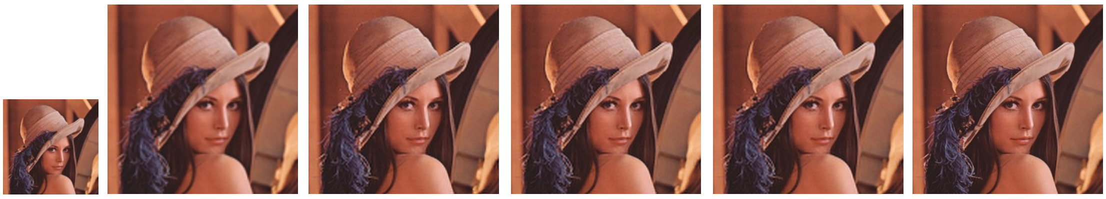

Figure 4. The Lena image magnified by a factor of size 2. Left to right: input, bicubic interpolation, size of the dictionary is 512, 768, 1024, and the original.

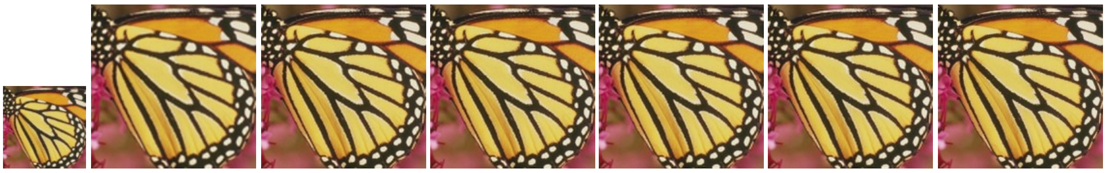

Figure 5. The Butterfly image magnified by a factor of size 2. Left to right: input, bicubic interpolation, size of the sub-block is 5×5, 6×6, 7×7, 8×8, and the original.

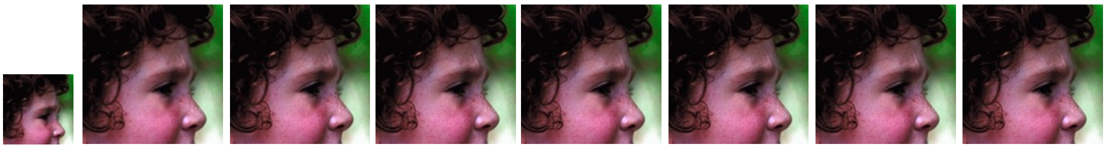

Figure 6. The Head image magnified by a factor of size 2. Left to right: input, bicubic interpolation, overlap width of 1, 2, 3, 4, 5, and the original.

Lena image is reconstructed based on the method proposed in this paper, and compared with other methods based on LLE-SR and Yang' method, Figure 7 shows the comparison results, it is clear that the method proposed in this paper is better. Finally, three natural images are selected for super-resolution reconstruction, and compared with the method of bicubic interpolation, Figure 8 shows the comparison results.

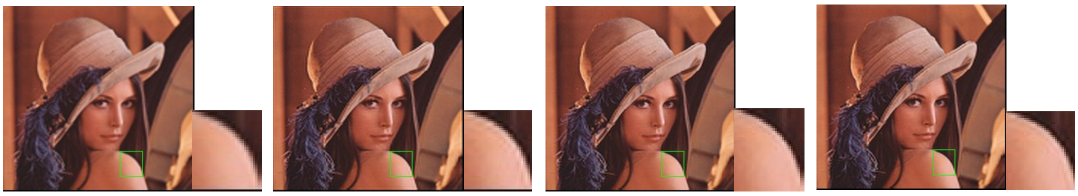

Figure 7. The Lena image magnified by a factor of size 2. Left to right: bicubic interpolation(RMSE=5.8453), LLE-SR method(RMSE=5.3862), Yang method(RMSE=4.8358), our method(RMSE=4.8079)

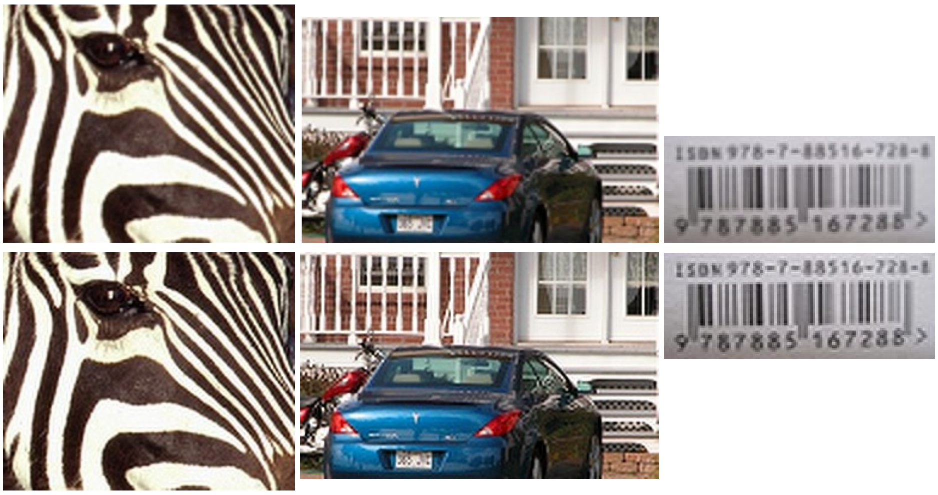

Figure 8. More super-resolution results on natural images. Top: bicubic interpolation. Bottom: our method, with magnification factor 2.

Table 1 The quality evaluation results of super-resolution reconstructed images with different size of dictionary

Size of dictionary	512	768	1024
RMSE	5.0032	4.9931	**4.9860**
PSNR	34.1459	34.1634	**34.1757**
SSIM	0.9909	0.9909	**0.9910**

Table 2 The quality evaluation results of super-resolution reconstructed images with different size of sub-blocks

Size of sub-blocks	RMSE	PSNR	SSIM
5×5	8.2273	29.8257	0.9886
6×6	**7.7981**	**30.2911**	**0.9897**
7×7	7.9294	30.1460	0.9894
8×8	8.0319	30.0345	0.9891

Table 3 The quality evaluation results of super-resolution reconstructed images with different overlap width

Overlap width	RMSE	PSNR	SSIM
1	5.1046	33.9715	0.9954
2	4.9692	34.2051	0.9956
3	4.7019	34.6853	0.9961
4	4.4384	35.1863	0.9965
5	**4.2738**	**35.5146**	**0.9967**

Sparseness Analysis of Dictionary Matrix and Reconstruction Coefficient

Due to the non-negativity of the model parameters, the nodes in the hidden layer are in a competitive state: some nodes can match the input image very well, and the others converge to an identical structure which the weight components tend to zero. Therefore, the learned weight matrix can be regarded as a component-based representation of non-negative input signals, and only a few of the base images are important. That is, the weight dictionary matrix is also sparse. In order to investigate the sparseness of dictionary matrix and reconstruction coefficient, we study the sparse metric function for a given vector:

$$s(\boldsymbol{v})=\left(\sqrt{n}-\left(\sum_{i=1}^{n}|v_i|\right)/\sqrt{\sum_{i=1}^{n}v_i^{\,2}}\right)\left(\sqrt{n}-1\right)^{-1}, \tag{15}$$

where $\boldsymbol{v}\in\Re^{(n)}$ with values between zero and one, the sparse vector is mapped to $s(\boldsymbol{v})>>0$. And the more sparse, the greater the value of $s(\boldsymbol{v})$.

Lena image is selected as test image, randomly select 200 image sub-blocks and calculate the sparseness of the coefficient vector from two angles, and the length of the vector is 1024:

(1) Calculate the sparse degree of the coefficient vector based on Eq. (15);

(2) Count the number of nonzero elements in the coefficient vector.

Figure 9 shows the results of the first method. Figure 10 shows the results of the second method, where the horizontal axis is the sequence number of sub-blocks.

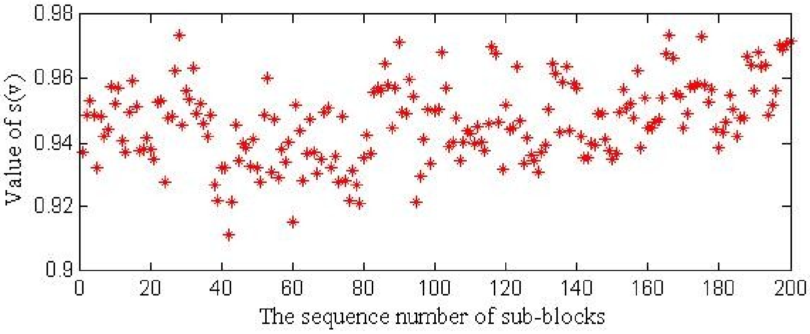

Figure 9. The sparseness of the coefficient vector based on Eq. (15)

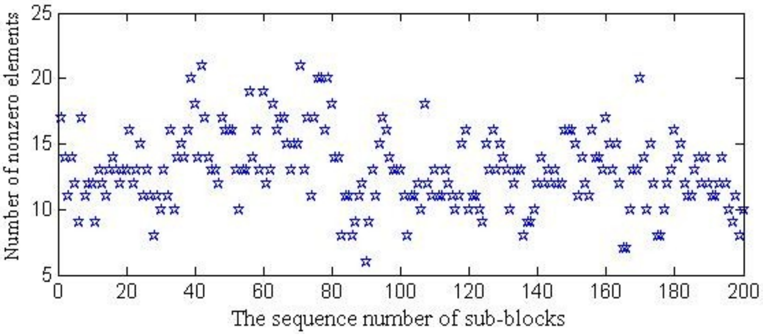

Figure 10. The sparseness of the coefficient vector based on the number of nonzero elements

Obviously, the value of $s(\boldsymbol{v})$ of each coefficient vector is greater than 0.90, and the number of non-zero elements in the coefficient vector is less than 26, and the proportion

$$\frac{number(non-zeros)}{number(total)}=\frac{25}{1024}<<0.025.$$

Conclusions

We firstly propose a joint dictionary learning method based on NSDAE model, which is inspired by the AE structure and the deblurring of the image super-resolution reconstruction. Then we reconstruct the high-resolution image by combining the sparse representation and the iterative error back projection method. The experiments show the effectiveness of the algorithm. Next we will do the following research: (1) Training dictionary with more deep learning models. (2) In order to meet various requirements we need increase the magnification factor of the image. (3) Study the super-resolution reconstruction algorithm for special image.

References

[1] Huan-Feng SHEN, Ping-Xiang LI, Liang-Pei ZHANG, Yi WANG. In Chinese: Overview on super resolution image reconstruction [J]. Optical technique, 2009, 35(2): 196–203.

[2] J C YANG, WRIGHT J, T HUANG, Y MA. Image super-resolution as sparse representation of raw image patches[J]. IEEE Conference of Computer Vision and Pattern Recognition (CVPR), 2008: 1–8.

[3] Zhi-Jun SUN, Lei XUE, Yang-Ming XU, Zheng WANG. In Chinese: Overview of deep learning[J]. Application Research of Computers, 2012, 29(8): 2805–2810.

[4] G E HINTON, P R SALAKHUTDINOV. Reducing the dimensionality of data with neural networks[J]. Science, 2006(313): 504–507.

[5] J C YANG, WRIGHT J, T HUANG, Y MA. Image super-resolution via sparse representation[C]. IEEE Transactions on Image Processing, 2010, 19(11): 2861–2873.

[6] ALAIN D, OLIVIER S. Gated Autoencoders with Tied Input Weights[C]. International Conference on Machine Learning, 2013: 154–162.

[7] VINCENT P, LAROCHELLE H, BENGIO Y, MANZAGOL P A. Extracting and composing robust features with denoising autoencoders[C]. International Conference on Machine Learning, 2008: 1096–1103.

[8] VINCENT P, LAROCHELLE H, BENGIO Y, MANZAGOL P A. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion[J]. Journal of Machine Learning Research, 2010(11):3371–3408.

[9] CHO K. Simple Sparsification Improves Sparse Denoising Autoencoders in Denoising Highly Noisy Images. Proceedings of the 30th International Conference on Machine Learning (ICML-13) , 2013: 432–440.

[10]TRIESCH J. A gradient rule for the plasticity of a neuron's intrinsic excitability[C]. Computer Science, 2005(3696): 65–70.

[11] J C YANG, Z W WANG, Z LIN, S COHEN, T HUANG. Coupled dictionary training for image super-resolution[C]. IEEE Transactions on Image Processing, 2012: 2359–2390.