Neurocomputing 172 (2016) 38–47

Contents lists available at ScienceDirect

Neurocomputing

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

Example-based super-resolution via social images

Yi Tang [a], Hong Chen [b], Zhanwen Liu [c], Biqin Song [b], Qi Wang [d,*]

[a] Department of Mathematics and Computer Science, Yunnan Minzu University, Kunming, Yunnan 650500, PR China
[b] College of Science, Huazhong Agricultural University, Wuhan 430070, PR China
[c] School of Information Engineering, Chang'an University, Xi'an, Shaanxi 710064, PR China
[d] School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, PR China

ARTICLE INFO

Article history:
Received 1 November 2013
Received in revised form
8 July 2014
Accepted 21 December 2014
Available online 15 May 2015

Keywords:
Social images
Single-image super-resolution
Matrix-based operator
Matrix-value operator learning

ABSTRACT

A novel image patch based example-based super-resolution algorithm is proposed for benefitting from social image data. The proposed algorithm is designed based on matrix-value operator learning techniques where the image patches are understood as the matrices and the single-image super-resolution is treated as a problem of learning a matrix-value operator. Taking advantage of the matrix trick, the proposed algorithm is so fast that it could be trained on social image data. To our knowledge, the proposed algorithm is the fastest single-image super-resolution algorithm when both training and test time are considered. Experimental results have shown the efficiency and the competitive performance of the proposed algorithm to most of state-of-the-art single-image super-resolution algorithms.

1. Introduction

Social images from online social platforms, for example, Flickr or YouTube, provide not only a new opportunity but also a challenge for various image processing tasks such as example-based super-resolution [1], visual query suggestion [2,3], saliency detection [4,5], ranking [6–8], and image classification [9–11] because a large amount of training images are available. The big size of training set means more information, however, it is also a huge burden during using all of these training samples. In this paper, we focus on the problem of example-based super-resolution with the help of a large amount of social image data, especially the problem of efficiently super-resolving with the help of a big size of training set.

1.1. Brief review of example-based super-resolution

Example-based super-resolution [1], also named as single-image super-resolution, is a problem of enhancing the resolution of some low-resolution images with the help of a set of training image pairs. Each of training image pairs consists of a low-resolution image and its corresponding high-resolution image. By learning on these training image pairs, the priori defining the relation between a low-resolution image and its high-resolution counterpart could be found. When a low-resolution image is observed, the learned priori could be applied on it for generating high-resolution estimation. The process of example-based super-resolution is summarized in Fig. 1.

Traditional example-based super-resolution algorithms could be generally divided into two categories according to the different ways of obtaining priori from training set. The first one belongs to implicit priori based algorithms where the priori is directly represented by the given training set. Most *K*-nearest neighbor based algorithms, such as Chang et al. [12], Tang et al. [13] and Gao et al. [14], belong to this category. It is clear that the implicit priori makes the learning process be omitted, but the *K*-nearest neighbor searching makes the price of recovering high-resolution estimation more expensive. The second one is explicit priori based algorithms. Dictionary [15–17] and regression function are two popular methods to represent the learned priori. Dictionary based algorithms such as Yang et al. [18], Lu et al. [19] focus on representing the priori between low- and high-resolution training images with the low- and high-resolution dictionary pair. Similarly, the priori on the relation between low- and high-resolution images is represented by a regression function which is learned by supervised or semi-supervised learning methods [20], such as, Ni and Nquyen [21], Kim and Kwon [22], and Tang et al. [23]. Generally, the training time of these explicit priori based algorithms is extremely long when the size of training set is big. Therefore, both of these traditional example-based super-resolution algorithms are not suitable for applying on a big size training set.

1.2. Motivation and our contributions

Two basic motivations make us focusing on the problem of super-resolution with the help of social images. Firstly, social images cover many image categories which could serve as a training set for super-resolving almost all kind of natural images. Secondly, huge training set

* Corresponding author.
E-mail address: crabwq@nwpu.edu.cn (Q. Wang).

http://dx.doi.org/10.1016/j.neucom.2014.12.102
0925-2312/

is changing the framework of example-based image processing. For example, Burger et al. [24] have shown that simple image denoising algorithm is competitive when big training set is available. Therefore, we discuss the problem of efficiently super-resolving natural images based on social images in this paper.

To benefit from a huge social image set, the computational complexity of an example-based super-resolution must be sufficiently low. However, the computational complexity of the traditional example-based super-resolution algorithms is too high to apply on a huge training set. To relieve the computational burden, a novel matrix-value operator based super-resolution algorithm is proposed based on the work [25]. Comparing with the work [25], more theoretical analysis and experiments are reported in this paper. Two main contributions of the novel algorithm can be summarized as follows:

- *Low computational complexity*. The computational complexity of the proposed super-resolution algorithm is only $O(N)$ where N is the number of training samples. The linear computational complexity makes the novel algorithm suitable for dealing with large training set.
- *Novel model for example-based super-resolution*. The proposed algorithm is designed based on representing images as matrices. And then, a novel matrix-value operator based learning model is introduced into example-based super-resolution. The novel learning model enables the computational and memory complexities of the proposed algorithm heavily reduced.

The rest of this paper is organized as follows. Main algorithm is introduced in Section 2. Some comments and theoretical analysis on the main algorithm are reported in Section 3. Experimental results are shown in Section 4.

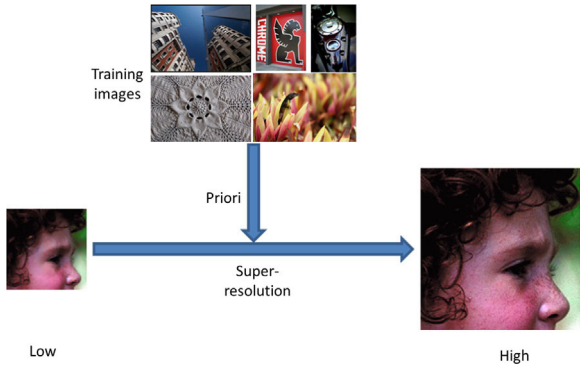

Fig. 1. The process of example-based super-resolution.

2. Main algorithm

Our algorithm is designed based on the idea of representing an image patch as a matrix. And then, the matrix-value operators are used to explicitly represent the relation between low- and high-resolution image patches. Taking advantage of operator learning techniques, our matrix-based super-resolution algorithm is fast enough for applying on a huge training set. The flowchart of our algorithm is summarized in Fig. 2.

Denote the matrix space as $R^{d\times d}$, where $d>0$ is an integer. Let low-resolution image patch space X and high-resolution image patch space Y be the subsets of $R^{d\times d}$ where d means the size of an image patch. Denote the training set

$$S_n=\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}\subseteq X\times Y, \tag{1}$$

where (x_i,y_i) is a pair of low- and high-resolution image patches, and n is the number of training pairs. The matrix-value operator $\mathcal{A}: X\mapsto Y$ is used to represent the super-resolution priori containing the training set S_n.

Least square operator regression model is used to learn the optimal matrix-value operator $\hat{\mathcal{A}}$ from the training set S_n:

$$\hat{\mathcal{A}}=\underset{\mathcal{A}}{\operatorname{argmin}}\sum_{i=1}^{n}\|y_i-\mathcal{A}x_i\|_F^2, \tag{2}$$

where $\|\cdot\|_F$ is Frobenius norm. By restricting the matrix-value operator $\mathcal{A}$ be Hilbert–Schmidt operator, the least square operator regression model (2) could be thought in Hilbert–Schmidt operator space, that is,

$$\begin{aligned}\hat{\mathcal{A}}&=\underset{\mathcal{A}}{\operatorname{argmin}}\sum_{i=1}^{n}\|y_i-\mathcal{A}x_i\|_F^2\\&=\underset{\mathcal{A}}{\operatorname{argmin}}\sum_{i=1}^{n}\langle y_i,y_i\rangle_F-2\langle y_i,\mathcal{A}x_i\rangle_F+\langle\mathcal{A}x_i,\mathcal{A}x_i\rangle_F\\&=\underset{\mathcal{A}}{\operatorname{argmin}}\sum_{i=1}^{n}\|y_iy_i^T\|_{HS}-2\langle y_ix_i^T,\mathcal{A}\rangle_{HS}+\langle x_ix_i^T,\mathcal{A}^*\mathcal{A}\rangle_{HS},\end{aligned} \tag{3}$$

where x^T is the transpose of the matrix x, $\langle\cdot,\cdot\rangle_F$ is the Frobenius inner on the matrix space, $\|\cdot\|_{HS}$ is the Hilbert–Schmidt norm on Hilbert-Schmidt operator space, the $\langle\cdot,\cdot\rangle_{HS}$ is the Hilbert–Schmidt inner on Hilbert–Schmidt operator space, and $\mathcal{A}^*$ is the adjoint operator of $\mathcal{A}$.

Denoting $F(\mathcal{A})=\sum_{i=1}^{n}\|y_iy_i^T\|_{HS}-2\langle y_ix_i^T,\mathcal{A}\rangle_{HS}+\langle x_ix_i^T,\mathcal{A}^*\mathcal{A}\rangle_{HS}$, the optimal operator $\hat{\mathcal{A}}$ satisfies the necessary conditions of the minimum, that is,

$$\frac{\partial}{\partial\mathcal{A}}F(\hat{\mathcal{A}})=0. \tag{4}$$

Because $F(\mathcal{A})=\sum_{i=1}^{n}\|y_iy_i^T\|_{HS}-2\langle y_ix_i^T,\mathcal{A}\rangle_{HS}+\langle x_ix_i^T,\mathcal{A}^*\mathcal{A}\rangle_{HS}$, there exists

$$\frac{\partial}{\partial\mathcal{A}}\left(\sum_{i=1}^{n}\|y_iy_i^T\|_{HS}-2\langle y_ix_i^T,\mathcal{A}\rangle_{HS}+\langle x_ix_i^T,\mathcal{A}^*\mathcal{A}\rangle_{HS}\right)=0$$

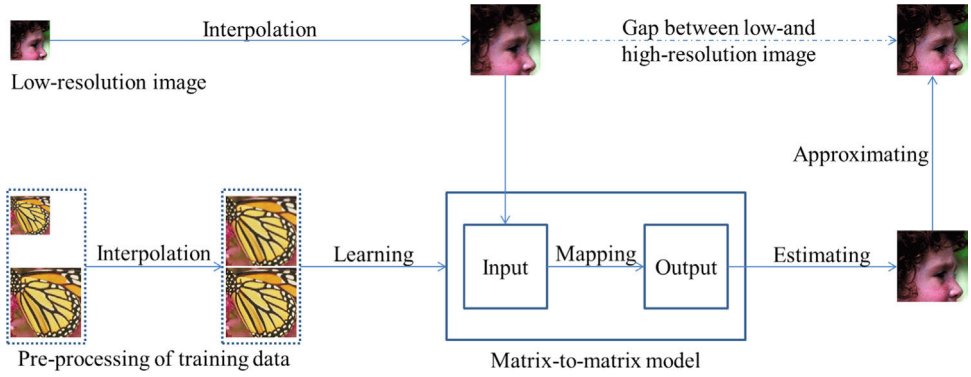

Fig. 2. The flowchart of the matrix-based super-resolution algorithm.

$$\Leftrightarrow \sum_{i=1}^{n} \mathcal{A}x_i x_i^T = \sum_{i=1}^{n} y_i x_i^T$$
$$\Leftrightarrow \mathcal{A}\left(\sum_{i=1}^{n} x_i x_i^T\right) = \sum_{i=1}^{n} y_i x_i^T$$
$$\Leftrightarrow \mathcal{A} = \left(\sum_{i=1}^{n} y_i x_i^T\right)\left(\sum_{i=1}^{n} x_i x_i^T\right)^{-1}, \quad (5)$$

where x^{-1} means the inverse matrix of x. Following (5), a close-form formula is there for representing the optimal matrix-value operator:

$$\hat{\mathcal{A}} = \sum_{i}^{n} y_i x_i^T \left(\sum_{i=1}^{n} x_i x_i^T\right)^{-1}. \quad (6)$$

An efficient algorithm can be designed based on (6). All details about the matrix-value regression algorithm for super-resolution are summarized in Algorithm 1.

Algorithm 1. Matrix-value operator regression for super-resolution.

Require:
Training set $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, the zooming parameter k, the size of image patch $d \times d$, and a test image T.

Matrix-value regression

1: Estimating the optimal matrix operator

$$\hat{\mathcal{A}} = \left(\sum_{i}^{n} y_i x_i^T\right)\left(\sum_{i=1}^{n} x_i x_i^T\right)^{-1};$$

Super-resolving

2: Interpolating the test image T with zooming parameter k;
3: Segmenting the interpolation image of the test image into a set of image patches with the size of $d \times d$

$$T \mapsto T = \{t_1, t_2, \ldots, t_l\};$$

4: **for** $s = 1, 2, \ldots, l$ **do**
5: Super-resolving the s-th test image patch

$$\tau_s = \hat{\mathcal{A}} t_s = \left(\sum_{i}^{n} y_i x_i^T\right)\left(\sum_{i=1}^{n} x_i x_i^T\right)^{-1} t_s;$$

6: **end for**
Ensure
7: Merge all $\tau_s, s = 1, 2, \ldots, l$, to obtain a super-resolution image $\tilde{T}$;
8: **return** $\tilde{T}$;

3. Comments on the main algorithm

3.1. Complexity of Algorithm 1

Because of Eq. (6), the optimal matrix-value operator $\hat{\mathcal{A}}$ could be obtained and stored by Algorithm 1 with tiny cost.

First of all, the memory complexity of the optimal matrix-value operator $\hat{\mathcal{A}}$ is just dependent on the size of the matrix $\hat{\mathcal{A}} = (\sum_i^n y_i x_i^T)(\sum_{i=1}^n x_i x_i^T)^{-1}$. It is clear that the size of $\hat{\mathcal{A}}$ is $d \times d$ because $S_n = \{(x_i, y_i)\} \subset R^{d \times d}$. In other word, d^2 parameters should be saved for recording the optimal matrix-value operator $\hat{\mathcal{A}}$.

Comparing with traditional example-based super-resolution algorithms, the memory complexity of Algorithm 1 is superior than them. For K-nearest based searching based algorithms, such as [1,26–28,14], all training samples should be recorded for recovering high-resolution estimations. It is clearly a heavy burden when a huge training set is considered. For dictionary based super-resolution algorithms [18,19,29], a pair of low- and high-resolution dictionaries should be recorded. Though the size of the pair of low- and high-resolution dictionaries is very smaller than the whole training set, more than

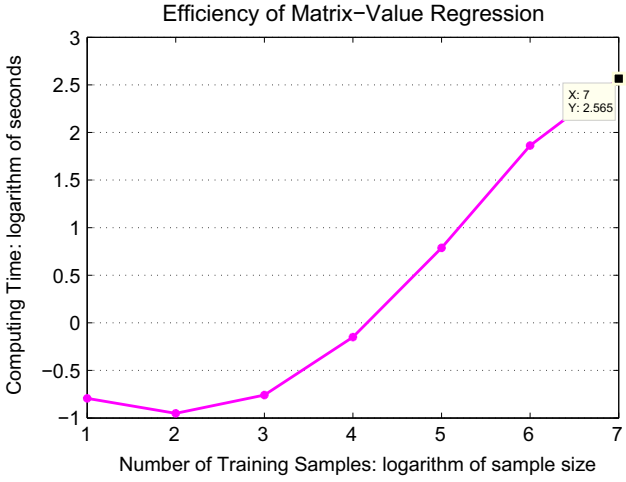

Fig. 4. Efficiency of matrix-value regression.

Fig. 3. Samples of training (a) and test (b) images.

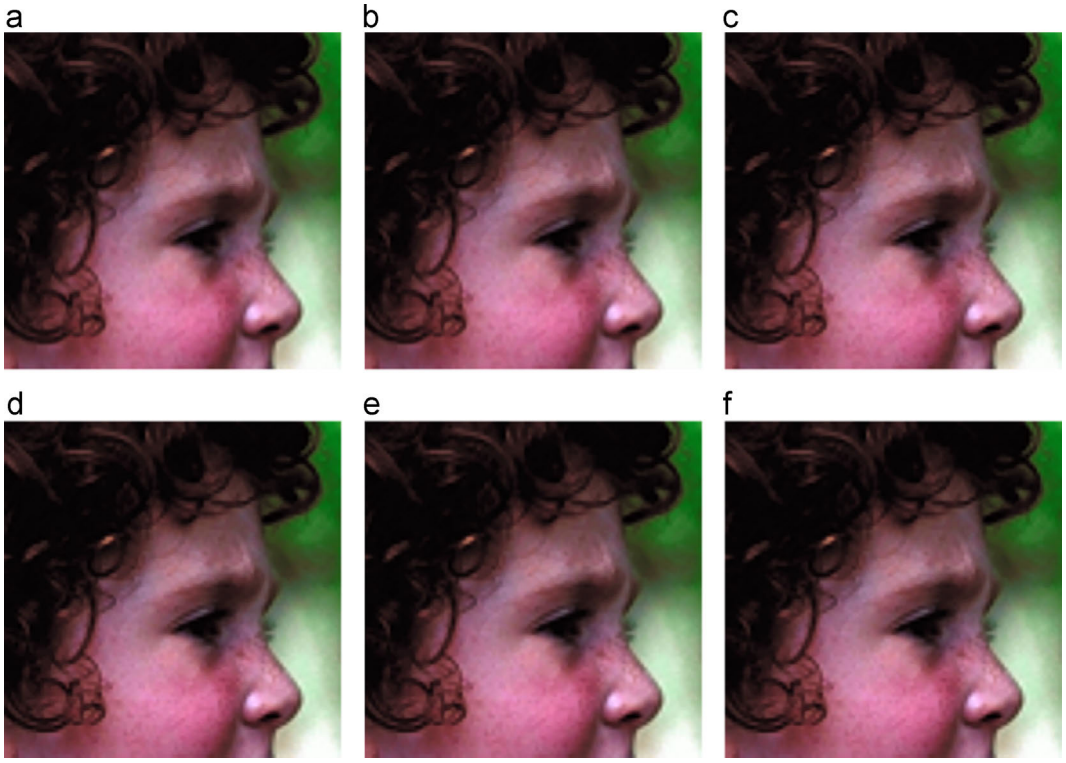

Fig. 5. Girl: performance of matrix-value regression where the sizes of training sets shift from 10 to 10^7 with a logarithm step equaling one. (a) #1. (b) #2. (c) #3. (d) #4. (e) #5. (f) #6.

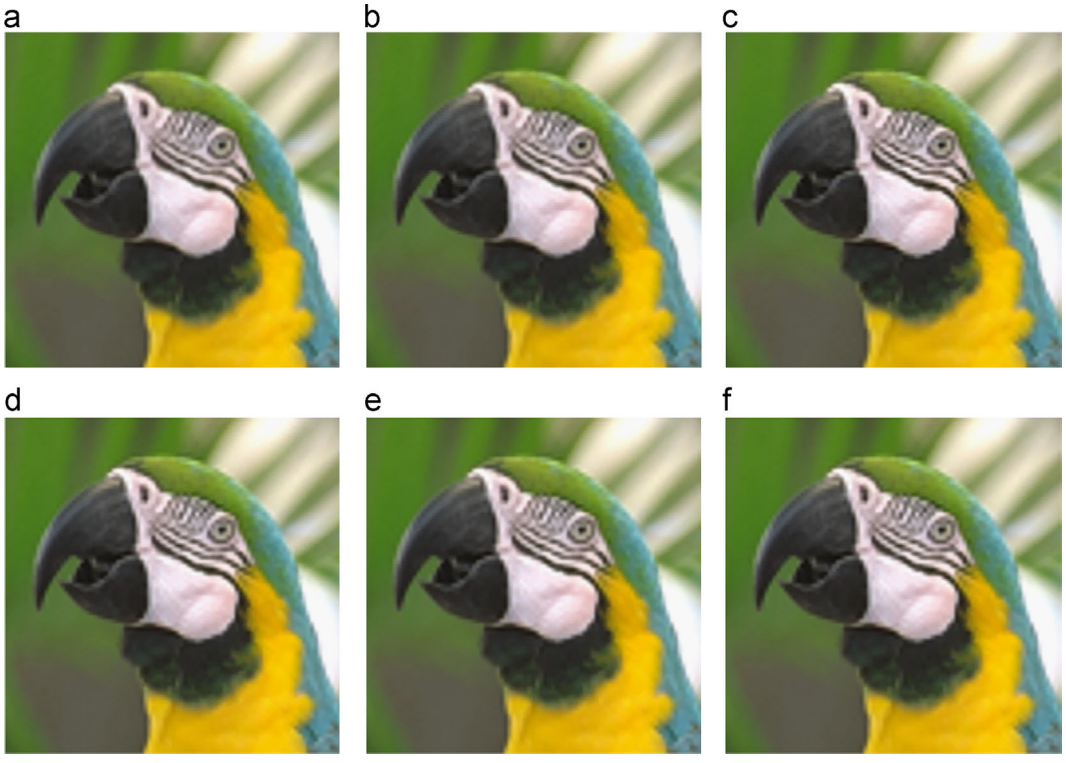

Fig. 6. Parrot: performance of matrix-value regression where the sizes of training sets shift from 10 to 10^7 with a logarithm step equaling one. (a) #1. (b) #2. (c) #3. (d) #4. (e) #5. (f) #6.

1000 vectors are generally needed to represent the dictionary pair. For each vector of the dictionary pair, tens of parameters are needed to represent the vector. Therefore, the memory complexity of the dictionary based algorithms is higher than that of our algorithm. For the regression based super-resolution algorithms, the super-resolution priori is explicitly represented by a regression function. Generally, the regression function is represented by a matrix whose number of rows is the number of all parameters used to represent a low-resolution image patch, and whose number of columns is the number of all parameters used to represent a high-resolution image patch because all training image patches are vectorized. For example, the regression function is represented by $d^2 \times d^2$ matrix within the setting that

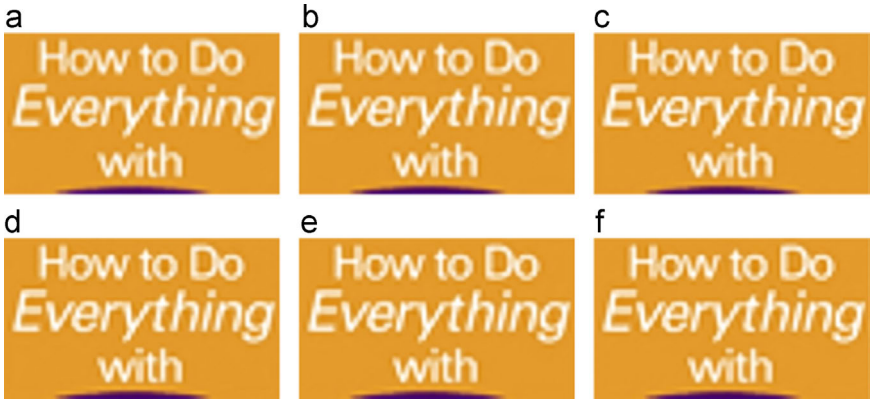

Fig. 7. Words: performance of matrix-value regression with where the sizes of training sets shift from 10 to 10^7 with a logarithm step equaling one. (a) #1. (b) #2. (c) #3. (d) #4. (e) #5. (f) #6.

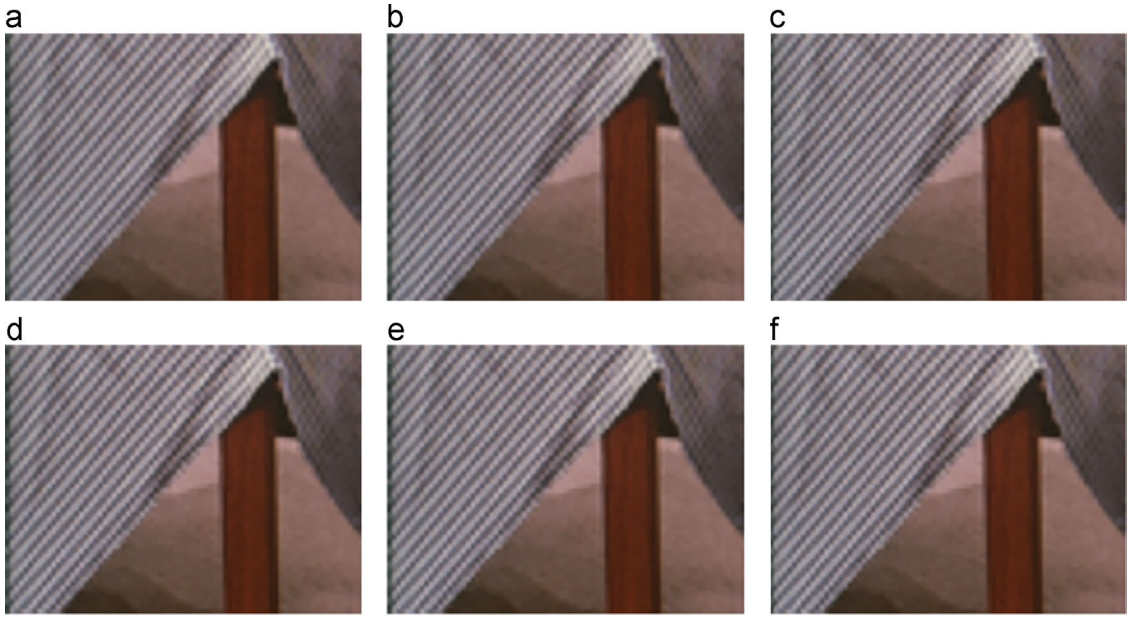

Fig. 8. Tablecloth: performance of matrix-value regression where the sizes of training sets shift from 10 to 10^7 with a logarithm step equaling one. (a) #1. (b) #2. (c) #3. (d) #4. (e) #5. (f) #6.

$(x, y) \in R^{d\times d}$. Then d^4 parameters are needed to record a regression function. It is clear that the memory complexity of regression based algorithms is also higher than ours.

Secondly, the computational complexity is also heavily reduced by using matrix-value operator learning compared with traditional example-based super-resolution algorithms. According to Eq. (6), the optimal matrix-value operator $\hat{A}$ is calculated by $\sum_{i=1}^{n} y_i x_i^T$, $\sum_{i=1}^{n} x_i x_i^T$ and the inverse matrix of $\sum_{i=1}^{n} x_i x_i^T$. It is clear that $2n$ matrix additions are needed to calculate $\sum_{i=1}^{n} y_i x_i^T$ and $\sum_{i=1}^{n} x_i x_i^T$, and d^2 operations are needed to generate the inverse matrix of $\sum_{i=1}^{n} x_i x_i^T$. Noticing the number of training samples n is much larger than the size of image patch d, the computational complexity of training our algorithm is $O(n)$. Therefore, our matrix-value operator based super-resolution algorithm could be smoothly trained when social image set is used as a training set. Meanwhile, the computational complexity of testing is independent of the number of training samples n, but dependent on the size of image patch d because the high-resolution image patch is estimated by $\hat{A}x$. When a test low-resolution image is divided into l test image patches, the computational complexity of estimating high-resolution image is $O(ld^2)$. Thus, the computational complexity of training and testing our algorithm is $O(n)+O(ld^2)$.

For the K-nearest neighbors searching based algorithms, the main computational burden is estimating high-resolution images because no training process is needed. Given l test image patches, the computational complexity of testing K-nearest neighbors searching based algorithms is larger than $O(\ln d^2)$ because the computational complexity of a naive K-nearest neighbors searching algorithm is $O(nd^2)$. It is clear that the total computational complexity of K-nearest neighbors searching based algorithms is much higher than ours because $O(\ln d^2)$ is always larger than $O(n)+O(ld^2)$.

For the dictionary based algorithms, the main computational burden is learning a dictionary. Because the computational complexity of learning a dictionary is changing with the dictionary learning algorithms, a very efficient dictionary learning algorithm K-SVD [15] is used as an example. According to the results reported in [30,31], the computational complexity of K-SVD is larger than $O(n(2d^2p+K^2p+3Kp+K^3))$ where the size of dictionary matrix is $d^2 \times p$, and K is the sparsity of estimations. Generally, there exists $p \gg d^2$ because of the redundancy of the learned dictionary. It should be noted that the training computational complexity of K-SVD is much larger than the total computational complexity of our algorithm. To our best knowledge, K-SVD is one of most efficient dictionary learning algorithms.

Thus, it could be inferred that the computational complexity of dictionary based algorithms is higher than ours.

Similarly, the computational complexity of our algorithm is much lower than the traditional regression based algorithms. For most least square regression based algorithms, it is needed to calculate an inverse matrix of a large matrix when a huge size of training set is given. For example, the inverse matrix of a larger kernel matrix with the size of $n \times n$ is needed for most of kernel based algorithms [21,22]. Generally, the computational complexity of calculating an inverse matrix of $n \times n$ is $O(n^3)$. Compared with the linear complexity of our algorithm, the computational complexity of kernel regression based algorithms is much higher.

Table 1
PSNRs/SSIMs of matrix-value regression algorithm.

Test image	Girl	Parrot	Words	Tablecloth
#1	32.7710/0.7698	28.0385/0.8711	26.9178/0.8523	26.2253/0.7703
#2	32.8669/0.7743	28.2307/0.8738	27.0930/0.8544	26.4411/0.7805
#3	32.9662/0.7786	28.3503/0.8777	27.1085/0.8549	26.5593/0.7868
#4	32.9508/0.7784	28.3209/0.8778	27.0441/0.8541	26.5552/0.7869
#5	32.9679/0.7790	28.3348/0.8781	27.0701/0.8538	26.5533/0.7864
#6	32.9937/0.7801	28.3704/0.8786	27.1090/0.8543	26.6039/0.7884

3.2. Merits of matrix-value operators

The advantage of low computational and memory complexity is connected with the trick of representing the image patches as matrices according to Eq. (6).

Firstly, a novel matrix-value similarity between two matrices could be found in Eq. (6). For any test image patch x, the high-resolution estimation $\hat{y}$ could be generated by the optimal matrix-value operator $\hat{\mathcal{A}}$, which is,

$$\begin{aligned}\hat{y} &= \hat{\mathcal{A}}x \\ &= \left(\sum_{i=1}^{n} y_i x_i^T\right)\left(\sum_{i=1}^{n} x_i x_i^T\right)^{-1} x \\ &= \sum_{i=1}^{n} y_i \left(x_i^T \left(\sum_{i=1}^{n} x_i x_i^T\right)^{-1} x\right)\end{aligned} \tag{7}$$

In the last line of Eq. (7), the matrix $x_i^T(\sum_{i=1}^{n} x_i x_i^T)^{-1}x$ could be roughly thought as a similarity measure between matrices x_i and x. Denote

$$S(x_i, x) = x_i^T \left(\sum_{i=1}^{n} x_i x_i^T\right)^{-1} x. \tag{8}$$

Table 2
Comparison of different algorithms.

Algorithm	Test image: girl			
	Statistics		Computing time	
	PSNR	SSIM	Training	Recovering
Chang's algorithm	32.3786	0.7453	1601.4 s	
Tang's algorithm	32.4827	0.7572	1047.0 s	
Yang's algorithm	33.1048	0.7899	≥ 12 h	28.2 s
Algorithm 1	32.9679	0.7790	72 s	4.2 s
Algorithm	Test image: parrot			
	Statistics		Computing time	
	PSNR	SSIM	Training	Recovering
Chang's algorithm	28.2989	0.8563	885.1 s	
Tang's algorithm	28.4388	0.8644	795.6 s	
Yang's algorithm	28.7858	0.8774	≥ 12 h	22.1 s
Algorithm 1	28.3348	0.8781	72 s	2.5 s
Algorithm	Test image: words			
	Statistics		Computing time	
	PSNR	SSIM	Training	Recovering
Chang's algorithm	23.6598	0.8457	298.5 s	
Tang's algorithm	23.779	0.8483	226.5 s	
Yang's algorithm	24.1268	0.8479	≥ 12 h	11.3 s
Algorithm 1	26.6039	0.8543	72 s	0.6 s
Algorithm	Test image: tablecloth			
	Statistics		Computing time	
	PSNR	SSIM	Training	Recovering
Chang's algorithm	25.7480	0.7123	467.3 s	
Tang's algorithm	25.9348	0.7344	352.7 s	
Yang's algorithm	26.0670	0.7454	≥ 12 h	25.4 s
Algorithm 1	26.6039	0.7884	72 s	0.9 s

The high-resolution estimation $\hat{y}$ satisfies

$$\hat{y} = \sum_{i=1}^{n} y_i S(x_i, x). \tag{9}$$

According to Eq. (9), the matrix-value operator $S(x_i, x)$ measures the similarity between test image patch and the i-th training image patch. Different from traditional real value similarity measure, $S(x_i, x)$ records two-dimensional similarity information between two matrices. In other words, the matrix-value similarity measure $S(x_i, x)$ records not only the energy of difference between both matrices, but also the structure of difference. Therefore, the novel matrix-value similarity measure provides more information than traditional real value measures.

Secondly, the matrix-value similarity measure implies that a novel matrix-value feature mapping is used in our algorithm. Noticing the formula (8), the inverse matrix $(\sum_{i=1}^{n} x_i x_i^T)^{-1}$ implicitly defines a matrix-value feature mapping $\phi : R^{d\times d} \mapsto R^{d\times d}$. In fact, the feature mapping ϕ could be generated by eigenvalue decomposition technique, that is, $\phi(x) = xM$ where M satisfies $(\sum_{i=1}^{n} x_i x_i^T)^{-1} = MM^T$. By using ϕ, the similarity measure $S(x_i, x)$ could be represented as $S(x_i, x) = \phi(x_i\phi(x))$, and the high-resolution estimation $\hat{y}$ satisfies

$$\hat{y} = \sum_{i=1}^{n} y_i \phi(x_i)\phi(x). \tag{10}$$

Following the idea of Eq. (10), our algorithm could be understood as a novel matrix-value operator based kernel algorithm.

4. Experiments

4.1. Samples and settings

MIRFLICKR08-25K is used as the training image set. Samples of the training images and four test images are respectively shown in Fig. 3(a) and (b). For comparing the performance of different super-resolution algorithms, three traditional example-based super-resolution algorithms including Chang's nearest neighbor embedding algorithm [12], Tang's vector-value regression algorithm [32], and Yang's dictionary-based algorithm [18] are used. Moreover, the statistical performance of these super-resolution algorithms is measured by the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM).

YCbCr channels are used to represent all of color images, and only Y channel is used in super-resolution algorithms because it is more sensitive to human vision.

All low-resolution training samples used in Algorithm 1 are generated by down-sampling high-resolution images and then up-sampling them with the same zooming parameter. In all of our experiments, the zooming parameter is 3. By dividing the low-resolution training images and their high-resolution counterparts into $d \times d$ image patches, the training set is generated by using

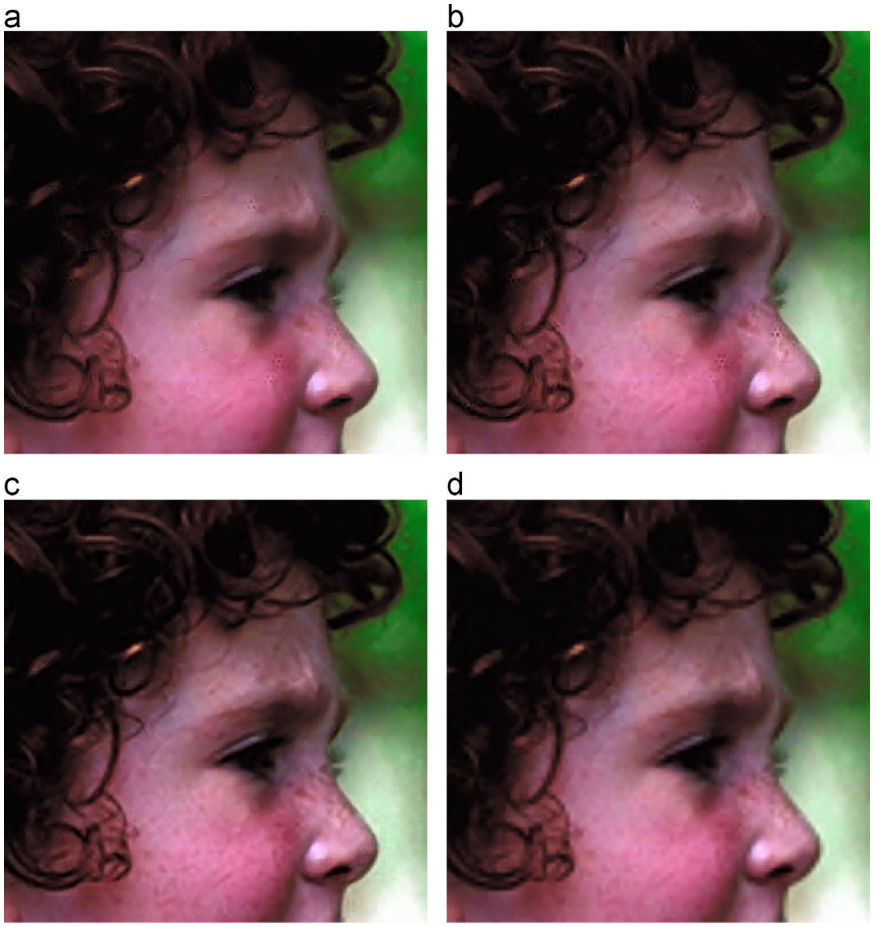

Fig. 9. Super-resolved images: Girl. (a) Chang's algorithm. (b) Tang's algorithm. (c) Yang's algorithm. (d) Algorithm 1.

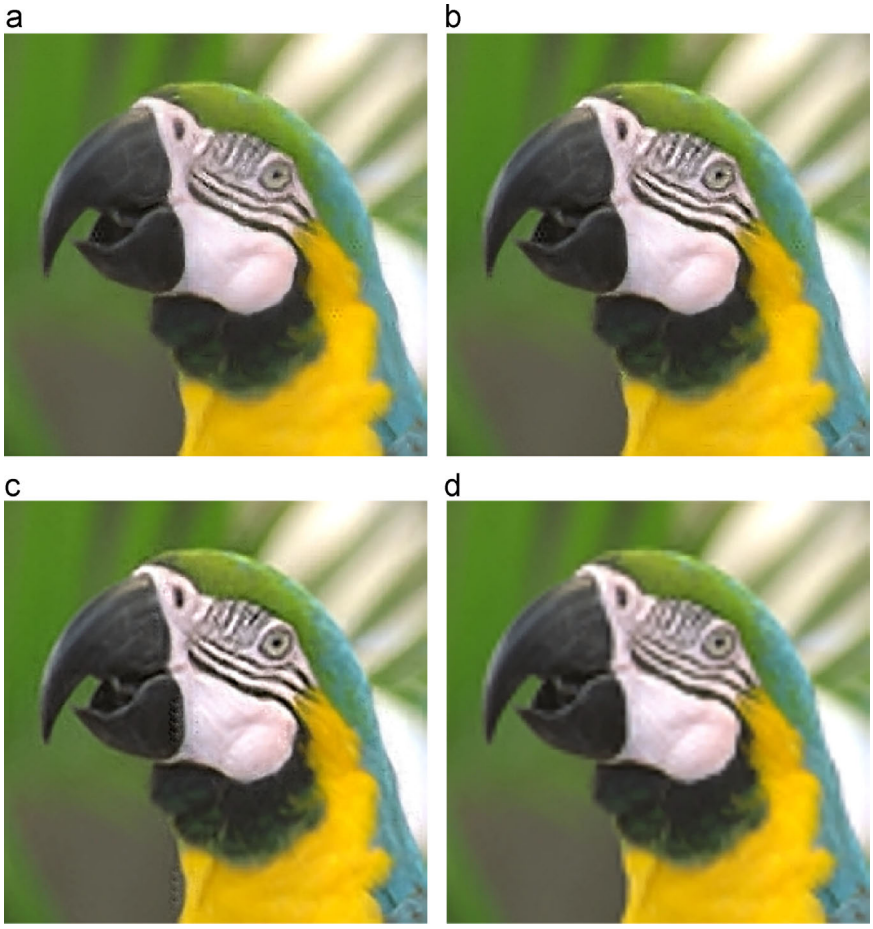

Fig. 10. Super-resolved images: Parrot. (a) Chang's algorithm. (b) Tang's algorithm. (c) Yang's algorithm. (d) Algorithm 1.

Fig. 11. Super-resolved images: Words. (a) Chang's algorithm. (b) Tang's algorithm. (c) Yang's algorithm. (d) Algorithm 1.

social images. For fair comparison, the patch size d sets to be 3. For other super-resolution algorithms, the low- and high-resolution image features are generated as their original reporting.

4.2. Results

The performance of matrix-value regression is tested with a big training set which consists of 4,035,993 image patch pairs generated according to the settings in Section 4.1.

The efficiency of the matrix-value regression algorithm reported in Algorithm 1 is firstly tested. The matrix-value regression algorithm is smoothly and quickly applied without using any accelerated algorithm even when all of 4,035,993 training pairs are considered. In fact, the optimal matrix operator is obtained by the matrix-value regression algorithm in 367.5376 s when a laptop with 2.40 GHz CPU and 2.00G memory is used. The relation between the number of training samples and the computing time is summarized in Fig. 4 where the horizontal and vertical axes represent respectively the logarithms of the number of training samples and computing time. It shows that it is linear for the relation between the number of training samples and the computing time. Therefore, it could be expected that the matrix-value regression algorithm can be smoothly applied even when bigger training set is used.

The performance of Algorithm 1 in super-resolution is also tested. By shifting the number of training samples, a set of matrix-value operators is generated by using the training sets with different sizes. The super-resolved images are shown in Figs. 5–8 where the images from left to right correspond to #1 training set to #6 training set where the size of training sets gets larger from #1 to #6 training sets. The corresponding statistical results including PSNRs and SSIMs are reported in Table 1. All of these results show that the matrix-value regression algorithm is effective in super-resolving nature images.

Computing time of training algorithms and recovering super-resolution images is used to compare the efficiency of different algorithms. Because Chang's and Tang's algorithms are local learning algorithms, the processes of training algorithms and recovering super-resolution images cannot be separated. The time from training algorithms to recovering a super-resolution image is used to measure the efficiency of these algorithms. For Yang's algorithm and Algorithm 1, the processes of training algorithms and recovering super-resolution images are independent. The time of training algorithms and

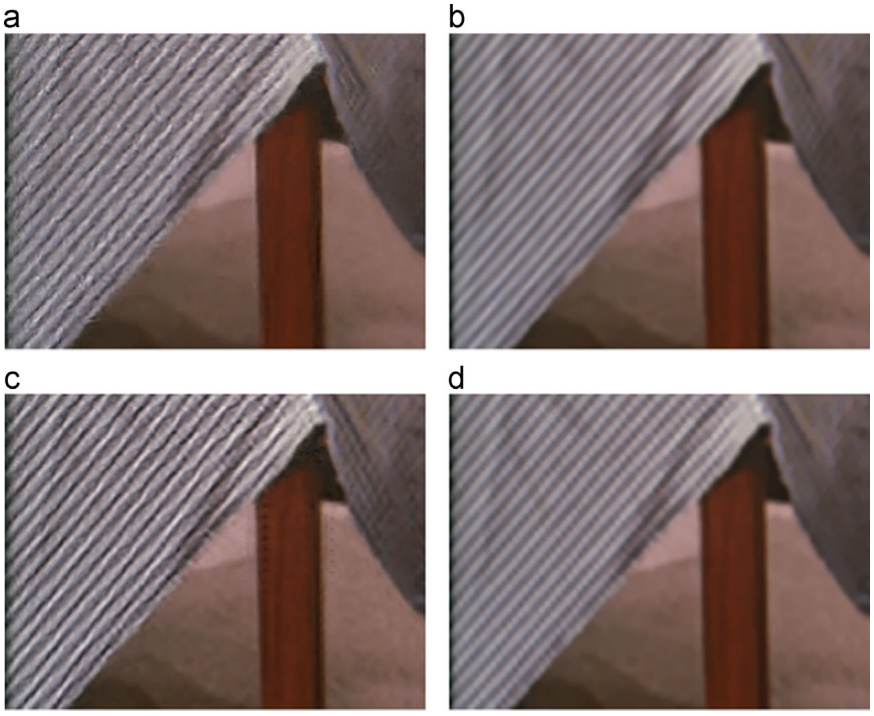

Fig. 12. Super-resolved images: Tablecloth. (a) Chang's algorithm. (b) Tang's algorithm. (c) Yang's algorithm. (d) Algorithm 1.

recovering super-resolution images is separately compared. A training set with 10^5 samples is used to test the computing time. The experimental results are shown in Table 2. Meanwhile, the super-resolved images are shown in Figs. 9–12, and the PSNRs and SSIMs of these super-resolved images are also reported in Table 2.

Based on the images shown in Figs. 9–12 the results generated by Algorithm 1 are cleaner than both local algorithms and slightly weaker than the results of Yang's algorithm. Similarly, PSNRs and SSIMs reported in Table 2 also support the observations. However, the computing time of Algorithm 1 is much less than any of these algorithms according to the results in Table 2. Therefore, the performance of Algorithm 1 in super-resolving low-resolution images is competitive to most of popular super-resolution algorithms, and the efficiency of Algorithm 1 is much superior than all of other mentioned algorithms.

5. Conclusion

In this paper, a novel matrix-value operator regression algorithm is proposed for super-resolving images with the help of social images. The proposed super-resolution algorithm is efficient in extracting social images information with much less cost of computation because matrix-value operator regression model is used to describe the relation between low- and high-resolution image patches. All of our theoretical and experimental results show the efficiency and the effectiveness of matrix-value operator in representing the image-pair information. Therefore, we believe that the image-pair information of social images will be more efficiently and effectively learned by employing matrix-value operator regression model.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China (Grant nos. 61105051, 61379094, and 61105012), the Fundamental Research Funds for the Central Universities (Grant nos. 3102014JC02020G07, 2011PY025, 2013G1241111, and 2014G1321035), the Natural Science Foundation Research Project of Shaanxi Province (Grant nos. 2012JQ8006, 2012JM8011, and 2012JM8024), the Science and Technology Plan Project of Yunnan Province (Grant no. 2014FB148), and the Open Research Fund of Key Laboratory of Spectral Imaging Technology, Chinese Academy of Sciences.

References

[1] W. Freeman, T. Jones, E. Pasztor, Example-based super-resolution, IEEE Comput. Graph. Appl. 22 (2) (2002) 56–65.
[2] Z.J. Zha, L. Yang, T. Mei, M. Wang, Z. Wang, T.S. Chua, X.-S. Hua, Visual query suggestion: towards capturing user intent in internet image search, ACM Trans. Multimed. Comput. Commun. Appl. (TOMMCAP) 6 (3) (2010) (Article no. 13).
[3] Z.J. Zha, L. Yang, T. Mei, M. Wang, Z. Wang, Visual query suggestion, in: The 17th ACM International Conference on Multimedia, 2009, pp. 15–24.
[4] Q. Wang, P. Yan, Y. Yuan, X. Li, Multi-spectral saliency detection, Pattern Recognit. Lett. 34 (1) (2013) 34–41.
[5] Q. Wang, Y. Yuan, P. Yan, X. Li, Saliency detection by multiple-instance learning, IEEE Trans. Cybern. 43 (2) (2013) 660–672.
[6] H. Chen, Y. Tang, L. Li, Y. Yuan, X. Li, Y.Y. Tang, Error analysis of stochastic gradient descent ranking, IEEE Trans. Image Process. 43 (3) (2013) 898–909.
[7] H. Chen, J. Peng, Y. Zhou, L. Li, Z. Pan, Extreme learning machine for ranking: generalization analysis and applications, Neural Netw. 53 (2014) 119–126.
[8] H. Chen, Z. Pan, L. Li, Learning performance of coefficient-based regularized ranking, Neurocomputing 133 (2014) 54–62.
[9] Y. Luo, D. Tao, B. Geng, C. Xu, S.J. Maybank, Manifold regularized multitask learning for semi-supervised multilabel image classification, IEEE Trans. Image Process. 22 (2) (2013) 523–536.
[10] Z.J. Zha, X.S. Hua, T. Mei, J. Wang, G.J. Qi, Z. Wang, Joint multi-label multi-instance learning for image classification, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[11] H. Chen, Z. Pan, L. Li, Y.Y. tang, Learning rates of coefficient-based regularized classifier for density level detection, Neural Comput. 25 (4) (2013) 1107–1121.
[12] H. Chang, D. Yeung, Y. Xiong, Super-resolution through neighbor embedding, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, vol. 1, 2004, pp. 275–282.
[13] Y. Tang, X. Pan, Y. Yuan, P. Yan, L. Li, X. Li, Single-image super-resolution based on semi-supervised learning, in: Proceedings of Asian Conference on Pattern Recognition, 2011, pp. 52–56.
[14] X. Gao, K. Zhang, X. Li, D. Tao, Image super-resolution with sparse neighbor embedding, IEEE Trans. Image Process. 21 (7) (2012) 3194–3205.
[15] M. Aharon, M. Elad, A. Bruckstein, K-SVD: an algorithm for designing over-complete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (11) (2006) 4311–4322.

[16] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Supervised dictionary learning, in: Advances in Neural Information Processing Systems, 2008.
[17] M.J. Gangeh, A. Ghodsi, M.S. Kamel, Kernelized supervised dictionary learning, IEEE Trans. Signal Process. 61 (19) (2013) 4753–4767.
[18] J. Yang, J. Wright, T. Huang, Y. Ma, Image super-resolution as sparse representation of raw image patches, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[19] X. Lu, H. Yuan, P. Yan, Y. Yuan, X. Li, Geometry constrained sparse coding for single image super-resolution, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2012, pp. 1648–1655.
[20] H. Chen, Y. Zhou, Y.Y. Tang, L. Li, Z. Pan, Convergence rate of semi-supervised greedy algorithm, Neural Netw. 44 (2013) 40–50.
[21] K. Ni, T. Nquyen, Image superresolution using support vector regression, IEEE Trans. Image Process. 16 (6) (2007) 1596–1610.
[22] K. Kim, Y. Kwon, Single-image super-resolution using sparse regression and natural image prior, IEEE Trans. Pattern Anal. Mach. Intell. 32 (6) (2010) 1127–1133.
[23] Y. Tang, Y. Yuan, P. Yan, X. Li, Greedy regression in sparse coding space for single-image super-resolution, J. Vis. Commun. Image Represent. 24 (2) (2012) 148–159.
[24] H. Burger, C. Schuler, S. Harmeling, Image denoising: can plain neural networks compete with BM3D?, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2012, pp. 2392–2399.
[25] Y. Tang, H. Chen, Matrix-value regression for single-image super-resolution, in: International Conference on Wavelet Analysis and Pattern Recognition, 2013, pp. 215–220.
[26] B. Li, H. Chang, S. Shan, X. Chen, Locality preserving constraints for super-resolution with neighbor embedding, in: Proceedings of IEEE International Conference on Image Processing, 2009, pp. 1189–1192.
[27] X. Gao, K. Zhang, X. Li, D. Tao, Joint learning for single-image super-resolution via a coupled constraint, IEEE Trans. Image Process. 21 (2) (2012) 469–480.
[28] Y. Tang, P. Yan, Y. Yuan, X. Li, Single-image super-resolution via local learning, Int. J. Mach. Learn. Cybern. 2 (1) (2011) 15–23.
[29] K. Zhang, X. Gao, D. Tao, X. Li, Multi-scale dictionary for single image super-resolution, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2012, pp. 1114–1121.
[30] R. Rubinstein, M. Zibulevsky, M. Elad, Efficient Implementation of the K-SVD Algorithm Using Batch Orthogonal Matching Pursuit, Technical Report, Technion, 2008.
[31] R. Rubinstein, T. Peleg, M. Elad, Analysis K-SVD: a dictionary-learning algorithm for the analysis sparse model, IEEE Trans. Signal Process. 61 (3) (2013) 661–677.
[32] Y. Tang, P. Yan, Y. Yuan, X. Li, Single-image super-resolution via local learning, Int. J. Mach. Learn. Cybern. 2 (2011) 15–23.

Yi Tang received the B.S., the M.S. and Ph.D. degrees in mathematics from Hubei University, Wuhan, China, in 2001, 2006, and 2010 respectively. He is currently an associate professor with the Department of Mathematics and Computer Science, Yunnan Minzu University, Kunming, China. His research interests include machine learning, statistical learning theory and pattern recognition.

Hong Chen received the B.Sc. and the Ph.D. degrees from Hubei University, Wuhan, China, in 2003 and 2009 respectively. He is an associate professor with the Department of Mathematics and Statistic Sciences, College of Science, Huazhong Agricultural University, China. His current research interests include statistical learning theory, approximation theory, and machine learning.

Zhanwen Liu received the B.Sc. degree from Northwestern Polytechnical University in 2006, the M.Sc. and the Ph.D degrees in Traffic Information Engineering and Control from Chang'an University in 2009 and 2014 respectively. She is a lecturer in School of Information Engineering, Chang'an University, Xi'an, China. Her research interests include image and video processing, computer vision, pattern recognition and image segmentation.

Biqin Song is currently a Ph.D. candidate of Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China. Her research interests include computer vision, pattern recognition and machine learning.

Qi Wang received the B.E. degree in automation and Ph.D. degree in pattern recognition and intelligent system from the University of Science and Technology of China, Hefei, China, in 2005 and 2010 respectively. He is currently an associate professor with the School of Computer Science and the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and pattern recognition.