Pairwise Operator Learning for Patch Based Single-image Super-resolution

Yi Tang and Ling Shao *Senior Member, IEEE*

***Abstract*—Motivated by the fact that image patches could be inherently represented by matrices, single-image super-resolution is treated as a problem of learning regression operators in a matrix space in this paper. The regression operators that map low-resolution image patches to high-resolution image patches are generally defined by left and right multiplication operators. The pairwise operators are respectively used to extract the raw and column information of low-resolution image patches for recovering high-resolution estimations. The patch based regression algorithm possesses three favorable properties. Firstly, the proposed super-resolution algorithm is efficient during both training and testing, because image patches are treated as matrices. Secondly, the data storage requirement of the optimal pairwise operator is far less than most popular single-image super-resolution algorithms because only two small sized matrices need to be stored. Lastly, the super-resolution performance is competitive with most popular single-image super-resolution algorithms because both raw and column information of image patches is considered. Experimental results show the efficiency and effectiveness of the proposed patch-based single-image super-resolution algorithm.**

***Index Terms*—Single-image super-resolution, matrix space, matrix-value operator regression, left and right multiplication operators.**

I. Introduction

Single-image super-resolution [1] can be understood as a problem of learning a regression function which maps low-resolution image patches to high-resolution image patches from the perspective of machine learning. Low- and high-resolution images are transformed into a training set in which each training sample consists of low- and high-resolution image features extracted from training image patches. For super-resolving a test image, we need to model the relation between low- and high-resolution features with a regression function which maps low-resolution image features to high-resolution image features. Therefore, different single-image super-resolution algorithms could be developed by choosing different features to represent low- and high-resolution image patches and choosing different regression methods.

The common strategy of transforming training image patches into training feature vectors is adopted by most of the existing single-image super-resolution algorithms. High-resolution training image patches are mostly represented as feature vectors by vectorizing the corresponding gray level image patches. Low-resolution training image patches are represented as feature vectors according to different motivations. For example, the low-resolution feature vectors used in [2]–[4] are generated by vectorizing difference images of low-resolution training images because difference images could highlight the high-frequency features of low-resolution images. And HOGs (histograms of oriented gradients) [5] are used in [6] for generating low-resolution feature vectors because local geometric structures of low-resolution image patches could be emphasized by HOGs. Another special case could be found in [7] where low- and high-resolution image patches are all transformed into sparse coding vectors by using coupled dictionaries because local geometric structures of low- and high-resolution training image patches could be characterized by the learned coupled dictionaries.

Transforming low- and high-resolution image patches into feature vectors also has some shortcomings, even though many promised results have been reported. It is clear that the low- and medium-frequency information of low-resolution image patches is abandoned when difference image based or HOG based feature vectors are used. Even the training image patches are just transformed into vectors without any feature extraction, the information about the relations between rows or columns of these training image patches will disappear. Therefore, it is valuable to consider the single-image super-resolution by preserving the inherent expression of image patches.

Recently, some works such as [8], [9] try to consider single-image super-resolution without changing the inherent expression of image patches. Matrices, the inherent expression of image patches, are used to represent low- and high-resolution image patches. It is clear that all raw information about low- and high-resolution image patches is preserved without the processing of vectorizing these image patches. In [8], [9], regression method is employed to learn the matrix-value operator which maps low-resolution image patches to high-resolution image patches. According to [9], the proposed single-image super-resolution algorithm is so efficient that it could be trained with a very large training set because of introducing the matrices to represent image patches. Meanwhile, it is asserted in [8] that the way of representing image patches as matrices offers more effective method to representing the local relations between training low- and high-resolution image patches. In fact, each high-resolution training image patches could be exactly recovered from its counterpart low-resolution training image patches by using the image-pair operators proposed in [8].

However, the mentioned patch based single-image super-

Yi Tang is with Key Laboratory of IOT Application Technology of Universities in Yunnan Province, School of Mathematics and Computer Science, Yunnan Minzu University, 134 Yieryi Ave, Kunming 650500, Yunnan, P. R. China. Email: tangyi@opt.ac.cn.

Ling Shao is with the School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, U.K.. Email: ling.shao@ieee.org.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2016.2639440, IEEE Transactions on Image Processing

resolution algorithms cannot make full advantages of the information of low- and high-resolution image patches in the view of matrix algebra. It should be noticed that only left multiplication operators are used to define the regression function mapping low-resolution image patches to high-resolution image patches in [8], [9]. According to the fact in matrix algebra, left multiplication operators means applying row transformations to a given matrix. In the field of single-image super-resolution, it means just the information of rows of low-resolution image patches is used in estimating high-resolution image patches.

For making the best of the raw information of training image patches, a novel pairwise operator learning algorithm is proposed in this paper for developing more efficient and effective single-image super-resolution algorithm. Specifically, left and right multiplication operators are simultaneously used to define a regression function which generates high-resolution image patches according to the observed low-resolution image patches. Clearly, the row and column information of low-resolution image patches can be well used during estimating high-resolution image patches because both of row and column transformations are appeared in the regression function.

The main contributions of the novel patch based single-image super-resolution algorithm include the followings.

- Taking advantages of the row and column information of image patches
  Row and column information of image patches is extracted by introducing left and right multiplication operators into single-image super-resolution algorithm. Therefore, right multiplication operators substantially enrich the information used by the reported algorithm with additional column information of image patches.
- An effective and efficient patch based single-image super-resolution algorithm
  The explicit formulation of left and right multiplication operators enable the reported algorithm can be trained and tested with low computational cost. The learned left and right multiplication operators can also be efficiently stored because both operators are represented by two small sized matrices. Meanwhile, the performance of the novel patch based single-image super-resolution algorithm is competitive because the row and column information of image patches are all considered.

The rest of this paper is organized as follows. A brief review about some existing single-image super-resolution algorithms are reported in Sec. II. The main algorithm and some remarks on the novel patch based single-image super-resolution algorithm are reported in Sec. III. Experimental results are shown in Sec. IV. This paper ends with a conclusion in Sec. V.

II. A BRIEF REVIEW ON SINGLE-IMAGE SUPER-RESOLUTION

Single-image super-resolution [10], also named example-based super-resolution [1], is a problem of estimating a high-resolution image from a given low-resolution image with the help of training samples that consist of low- and high-resolution image pairs. The task of single-image super-resolution is modeling the information containing in these training samples for super-resolving a test low-resolution image. We will briefly review some existing single-image super-resolution algorithms according to different methods of modeling these training samples.

Most of existing single-image super-resolution algorithms adopt the patch based method to analyze the information of training samples. In the patch based method, an image is treated as a set of image patches. Then a pair of low- and high-resolution training images is thought as a set of low- and high-resolution image patches. These image patches are often represented as feature vectors in many single-image super-resolution algorithms such as [2], [3], [11], because these feature vectors can highlight the high-frequency information of low-resolution image patches. Different from the vector-based representation, the matrix-based representation has been recently used by some single-image super-resolution algorithms such as [8], [9] where an image patch is represented by a matrix. It is clear that more raw information of image patches is preserved because these image patches are represented in their original form without any transformation.

For utilizing the information of these low- and high-resolution image patches, different strategies are used by different single-image super-resolution algorithms. Nearest neighbor embedding is mostly used in single-image super-resolution. For example, Chang *et al.* [2] proposed an algorithm to estimate super-resolution estimations according to the manifold information [12] of K nearest neighbors of test image patches. Gao *et al.* [6] proposed a sparse neighbor embedding algorithm by introducing the sparse representation assumption into nearest neighbor embedding. Refer to [4], [13]–[15] for more related works. Dictionary based methods are also used in designing single-image super-resolution algorithms. In the framework of dictionary based methods, the training pairs of low- and high-resolution image patches are summarized by dictionary learning [16], [17] as a coupled dictionary which consists of a low-resolution dictionary and its high-resolution counterpart. Yang *et al.* [3] proposed the first dictionary based single-image super-resolution algorithm by assuming low- and high-resolution image patches share common sparse codings [18], [19]. Lu *et al.* [20] introduced geometry restriction on the sparse codes into dictionary based single-image super-resolution algorithms. Dong *et al.* [21] introduced the multi-dictionary based method into the field of single-image super-resolution. For more references, please see [22]–[25]. Regression based methods are also popular in modeling the relation between low- and high-resolution image patches. For example, kernel based regression [26], [27] was used to learn nonlinear regression functions for mapping low-resolution feature vectors to high-resolution feature vectors in [11], [28]. Steering kernel regression was used by [29], [30]. Wang *et al.* [31] used active-sampling Gaussian process regression for super-resolution. He *et al.* [32] used Gaussian process regression to model the relation of self-similarity among multi-scale test images. Different from these vector-value regression methods, matrix-value linear regression was introduced into single-image super-resolution in [8], [9], [33].

Different learning strategies make the cost of training and testing these single-image super-resolution algorithms various.

For example, nearest neighbor embedding based algorithms can be tested without any training, but they need a large enough training set to ensure the better performance of super-resolving test images. It is clear that searching nearest neighbors in a large training set will make these algorithms time-consuming in estimating super-resolution images. Meanwhile, it is also space-consuming for the nearest neighbor embedding based algorithms because a large training set must be stored. Dictionary based algorithms release the burden of storing the training set by summarizing a large training set into a coupled dictionary. But the computational cost of learning a coupled dictionary from a large training set is very high, which makes these dictionary based algorithms time-consuming during the training process. However, the learned coupled dictionary heavily reduces the computational time during the testing process because of the existence of efficient sparse coding algorithms such as [19]. Similar to dictionary based algorithms, kernel regression based algorithms also suffer from time-complexity during the training process because the computational cost is very high to train a support vector machine on a large training set. Fortunately, the burden of storing the learned kernel based regression function is low enough because just a small set of training samples, namely a set of support vectors [26], [27], is needed during the testing process. Meanwhile, the explicit formula of the learned kernel based regression function makes the kernel regression based algorithms very efficient during testing.

Beside these patch based algorithms, some image based algorithms were recently proposed based on deep convolutional neural networks [34]. For example, Dong *et al.* [35], [36] designed a convolutional neural network for directly mapping low-resolution images to their super-resolution counterparts. Liu *et al.* [37] introduced the sparse prior into deep networks for generating robust super-resolution estimations. Yan and Shao [38] also used deep learning technique to estimate image blur blindly. For more reference about the application of deep learning, see [39], [40] Taking advantages of deep learning, these image based algorithms offer more promising super-resolution estimations than most of patch based algorithms. However, the huge burden of training a convolutional neural network makes these image based algorithms time-consuming during the training process.

Therefore, more and more effort is paid to balancing the performance and the cost of training, testing and storing learned data in designing novel single-image super-resolution algorithms. For example, Zhang *et al.* [41] proposed a novel single-image super-resolution algorithm where multiple linear mappings are used to model the relation between low- and high-resolution image features. Because all of these linear mappings are learned based on sub-dictionaries, the multiple linear mappings based algorithm well balances the performance and the computational complexity for both training and testing. Similarly, Tang *et al.* [9] also proposed a fast single-image super-resolution algorithm based on learning matrix-value operators. Zhu *et al.* [42] proposed a fast single image hazing removal algorithm based on color attention prior. For more references on fast single-image super-resolution algorithms, please see [43], [44].

III. Pairwise operators learning

A. Linear regression for patch based single-image super-resolution

Let $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^{w \times w}$ be separately the low- and high-resolution image patches spaces, where $w > 0$ is the size of each image patches. Let training set be

$$S_n = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} | i = 1, 2, \cdots, n\}, \tag{1}$$

where $n > 0$ is the number of training samples. Let $A : \mathcal{X} \mapsto \mathcal{Y}$ be the operator which recovers high-resolution image patches from the observed low-resolution image patches, and $\mathcal{A}$ be the hypothesis space of matrix-value operators.

The optimal operator in the hypothesis space $\mathcal{A}$ could be learned by minimizing the regularized empirical super-resolution error [9], [33], that is

$$A^* = \arg\min_{A \in \mathcal{A}} \sum_{i=1}^{n} \|y_i - A(x_i)\|_F^2 + \lambda \Omega(A), \tag{2}$$

where $\|\cdot\|_F$ means Frobenius norm of matrices, $\lambda > 0$ is a regularization coefficient, and $\Omega(A)$ is a regularization term for measuring the complexity of an operator A. Because of the advantages of efficient training and testing, $\mathcal{A}$ is assumed as a linear operator space in this paper. Different from the linear operators defined on vector space, the linear operators defined on matrix space consist of left and right multiplication operators, such as $A(x) = LxR$, where $L \in \mathbb{R}^{w \times w}$ and $R \in \mathbb{R}^{w \times w}$ are respectively left and right multiplication operator. The complexity of the pair of left and right multiplication operators (L, R) is defined as

$$\Omega(A) = \Omega(L, R) = \|L\|_F^2 + \|R\|_F^2 . \tag{3}$$

Therefore, the optimal linear operator could be learned by searching the optimal pair of left and right multiplication operators, that is,

$$(L^*, R^*) = \arg\min_{L \in \mathcal{L}, R \in \mathcal{R}} \sum_{i=1}^{n} \|y_i - Lx_iR\|_F^2 + \lambda \Omega(L, R), \tag{4}$$

where $\mathcal{L}$ and $\mathcal{R}$ are respectively the hypothesis spaces of left and right multiplication operators.

Left and right multiplication operator bear specific meaning in patch based single-image super-resolution. According to matrix algebra, left and right multiplication operators respectively correspond to left and right transforms of a matrix x. In the field of single-image super-resolution, the fact means that the row and column information of x could be used by the left and right multiplication operator L and R. Therefore, left and right multiplication operators represent respectively the relation between rows and columns of low- and high-resolution image patches.

B. Iteration optimization

The matrix based regression (4) has no closed formula because left and right multiplication operators should be simultaneously learned. In this paper, the near optimal pair of left and right multiplication operators is learned by using iteration optimization.

Noticed the optimization problem (4), left multiplication operator has closed-form expression given right multiplication operator, and vice versa. Therefore, right multiplication operator R is initialized as a unit matrix I which means mapping low-resolution image patch x to itself. By using the initialization, the left multiplication operator could be generated by using necessary conditions of the optimization, which means the first learned left multiplication operator satisfies

$$\frac{\partial}{\partial L}\bigg|_{L_1} \sum_{i=1}^{n} \|y_i - Lx_i\|_F^2 + \lambda \|L\|_F^2 = 0, \tag{5}$$

where L_1 is the optimal left multiplication operator learned from the first iteration. It is clear that the formula (5) is equal to

$$\begin{aligned} & \sum_{i=1}^{n} \frac{\partial}{\partial L}\bigg|_{L_1} \|y_i - Lx_i\|_F^2 + \lambda \frac{\partial}{\partial L}\bigg|_{L_1} \|L\|_F^2 \\ = & \sum_{i=1}^{n} (\frac{\partial}{\partial L}\bigg|_{L_1} \|Lx_i\|_F^2 - 2\frac{\partial}{\partial L}\bigg|_{L_1} \langle y_i, Lx_i \rangle_F) + 2\lambda L_1 \\ = & \sum_{i=1}^{n} (L_1 x_i x_i' - y_i x_i') + 2\lambda L_1 \\ = & L_1(\sum_{i=1}^{n} x_i x_i' + 2\lambda I) - \sum_{i=1}^{n} y_i x_i' = 0 \\ \Leftrightarrow & L_1 = (\sum_{i=1}^{n} y_i x_i')(\sum_{i=1}^{n} x_i x_i' + 2\lambda I)^{-1}. \end{aligned} \tag{6}$$

Given the k-th left multiplication operator L_k, the corresponding k-th right multiplication operator R_k could be deduced from the formula

$$\frac{\partial}{\partial R}\bigg|_{R_k} (\sum_{i=1}^{n} \|y_i - L_k x_i R\|_F^2 + \lambda \|R\|_F^2) = 0. \tag{7}$$

Similar to the closed-form expression (5), the k-th right multiplication operator R_k becomes

$$R_k = (\sum_{i=1}^{n} x_i' L_k' L_k x_i + 2\lambda I)^{-1} (\sum_{i=1}^{n} x_i' L_k' y_i). \tag{8}$$

Then the $k+1$-th left multiplication operator L_{k+1} could be generated when the k-th right multiplication operator R_k is learned.

$$\begin{aligned} & \frac{\partial}{\partial L}\bigg|_{L_{k+1}} (\sum_{i=1}^{n} \|y_i - Lx_i R_k\|_F^2 + \lambda \|L\|_F^2) = 0 \Rightarrow \\ & L_{k+1} = (\sum_{i=1}^{n} y_i R_k' x_i')(\sum_{i=1}^{n} x_i R_k R_k' x_i' + 2\lambda I)^{-1}. \end{aligned} \tag{9}$$

The process of iteration optimization is summarized in Alg. 1.

IV. Experiments

In this section, the characters of POL are firstly discussed by analyzing the experimental results. Specifically, the impacts of the size of image patches are shown by testing over 300 images.

Algorithm 1 Pairwise operators learning for patch based single-image super-resolution (POL)

```
Require:
    Training set $S_n = \{(x_i, y_i) | i = 1, 2, \cdots, n\}$, a regularized parameter $\lambda > 0$ and the number of iterations $K$.
    Initializing right multiplication operator
    $R_1 = I$
    Generating the first left multiplication operator as (6)
    $L_1 = (\sum_{i=1}^{n} y_i x_i')(\sum_{i=1}^{n} x_i x_i' + 2\lambda I)^{-1}$.
    Iteration optimization
    for $k = 2, \cdots, K$ do
        Generating the $k$-th right multiplication operator as the formula (8)
            $R_k = (\sum_{i=1}^{n} x_i' L_k' L_k x_i + 2\lambda I)^{-1} (\sum_{i=1}^{n} x_i' L_k' y_i)$;
        Generating the $k+1$-th left multiplication operator as the formula (9)
            $L_k = (\sum_{i=1}^{n} y_i R_k' x_i')(\sum_{i=1}^{n} x_i R_k R_k' x_i' + 2\lambda I)^{-1}$.
    end for
    return The optimal pair of left and right multiplication operators $(L_K, R_K)$;
```

The convergence of POL is also shown by modeling the relation between the number of iterations and the peak signal-to-noise ratio (PSNR) or the structural similarity (SSIM) [45]. According to Zhou *et al.* [45], SSIM is defined as

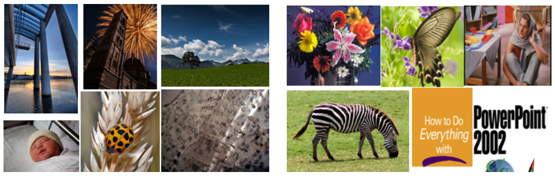
$$SSIM(x, y) = \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)},$$

where x, y are two images whose pixels are respectively denoted as x_i and y_i, C_1, C_2 are constants, $\mu_x = \frac{1}{W*H} \sum_{i=1}^{W*H} x_i$, $\mu_y = \frac{1}{W*H} \sum_{i=1}^{W*H} y_i$, and $\sigma_x = (\frac{1}{W*H-1} \sum_{i=1}^{W*H} (x_i - \mu_x)^2)^{1/2}$, $\sigma_y = (\frac{1}{W*H-1} \sum_{i=1}^{W*H} (y_i - \mu_y)^2)^{1/2}$, $\sigma_{xy} = \frac{1}{W*H-1} \sum_{i=1}^{W*H} (x_i - \mu_x)(y_i - \mu_y)$.

Then the performance of POL is firstly compared with some of existing single-image super-resolution algorithms for showing its effectiveness and efficiency in super-resolution. The effectiveness is measured by the visual experience, PSNR, SSIM, feature-similarity (FSIM) index [46] and visual saliency induced (VSI) index [47]. Both image quality indexes of FSIM and VSI pay more attentions on simulating human visual system which offer more possibility to unveil the difference among super-resolved images. The efficiency is measured not only by the CPU time of training algorithms and recovering high-resolution images but also the size of storing the learned super-resolution model. All of the algorithms compared with POL are summarized in Tab. I. The codes of these compared algorithms are all obtained from their original authors except for the code of MLM, which we re-implemented ourselves.

TABLE I
COMPARATIVE ALGORITHMS

Abbreviation	Algorithm	Strategy
BINT	Bilinear interpolation	pointwise
MF	Manifold-based algorithm [2]	vectorwise
DL	Dictionary learning algorithm [3]	
RDL	regularized dictionary learning algorithm [20]	
MLM	multiple linear mapping algorithm [41]	
SNE	Sparse neighbor embedding algorithm [6]	
MVOR	Matrix-value operator regression algorithm [9]	patchwise
IPO	Image-pair operator based algorithm [8]	
POL	The proposed Algorithm 1	



(a) training images (b) test images

Fig. 1. Samples of training and test images

A. Experiment settings

Training and test images are natural images randomly downloaded from internet. Some of samples of training and test images are respectively shown in Figs. 1(a) and 1(b). The low-resolution images are generated by down-sampling these downloaded images with a zooming factor $\kappa = 3$, and all of these downloaded images are treated as high-resolution images. All of these pairs of low- and high-resolution images are respectively processed for generating training sets as the original settings reported in corresponding reference for all algorithms compared with POL.

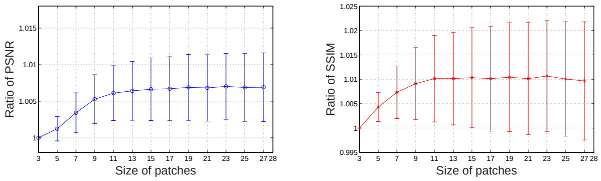
For POL, the training set is generated as following settings. Firstly, low-resolution images are interpolated with bicubic interpolation operator to the same size of its high-resolution counterparts. The interpolated images are serviced as low-resolution images in following steps. These low- and high-resolution images are represented by using the YCbCr channels, and only Y channel of these images are used during training and testing POL. Secondly, all of low- and high-resolution images are blocked into pairs of image patches for training and testing POL. The size of each image patch is denoted as $w \times w$. The value of w will be specified in different experiments.

B. Characters of POL

1) Effects of the patch size: The effects of the size of image patches is shown by considering the variation of PSNRs and SSIMs of more than 300 super-resolved images. The sizes of image patches vary from 3×3 to 27×27 with a step of 2 pixels, and the overlap equals to $w - 2$. Both ratios of PSNRs and SSIMs ρ_{PSNR} and ρ_{SSIM} defined as

$$\rho_{PSNR} = \frac{\text{PSNRs of } w \times w}{\text{PSNRs of } 3 \times 3}, \quad (10)$$

$$\rho_{SSIM} = \frac{\text{SSIMs of } w \times w}{\text{SSIMs of } 3 \times 3}, \quad (11)$$



(a) Curve of ρ_{PSNR} (b) Curve of ρ_{SSIM}

Fig. 2. Effects of the patch sizes: The patch sizes corresponding to the horizontal axis vary from 3 pixels to 27 pixels with a step of 2 pixels. ρ_{PSNR} and ρ_{SSIM} corresponding to the vertical axis show the degree of improvement when different patch sizes are used. The PSNR and SSIM of POL corresponding to the patch size of 3 pixels are separately used as the baseline.

are used to show the performance of POL given different sizes of image patches, where w varies from 3 to 27 with a step of 2. The curves of ρ_{PSNR} and ρ_{SSIM} are shown in Fig. 2 where horizontal axis corresponds to the size of image patches, and vertical axis corresponds to the values of ρ_{PSNR} and ρ_{SSIM}. According to both curves of ρ_{PSNR} and ρ_{SSIM}, POL could benefit from the larger size of patches because ρ_{PSNR} and ρ_{SSIM} almost get larger along with the enlargement of the sizes of patches. Meanwhile, the improvement of ρ_{PSNR} and ρ_{SSIM} gets slight and the standard deviation gets larger when the size of patches w is larger than 11 pixels. We think these facts show that POL could adopt more complex information from image patches because larger image patches will contain more complex structures than the smaller ones, however, POL cannot be well trained by too complex structure information from too large image patches. Therefore, the size of patches w sets to be 11 for balancing the performance of PSNR and SSIM of POL in the following.

2) Effects of the pairwise operators: According to the definition of matrix multiplication, pairwise operators learned by POL contain respectively column and row information for describing the relation between low- and high-resolution image patches. The column and row information connected with left and right multiplication operators is detected by observing the ratios of PSNR and SSIM which are defined as

$$\tau_{PSNR} = \frac{\text{PSNRs of } (L_k, R_k)}{\text{PSNRs of baseline}}, \quad (12)$$

$$\tau_{SSIM} = \frac{\text{SSIMs of } (L_k, R_k)}{\text{SSIMs of baseline}}, \quad (13)$$

where the baseline means the mapping defined by the first pair of left and right multiplication operators. It should be noticed that the first right multiplication operator R_1 is defined as an identity operator according to Alg. 1, which means no more than trivial column information of low-resolution image patches is introduced into the process of super-resolution. Therefore, the variation of both ratios of τ_{PSNR} and τ_{SSIM} could reflect if more information is introduced into the process of super-resolution by using nontrivial right multiplication operators. Because of the symmetry of left and right multiplication operators, the effects of left multiplication will coincide with the effects of right ones.

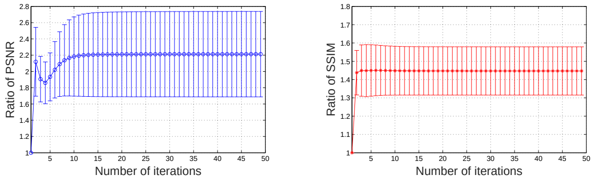
Similarly, more than 300 test images are used to measure the effects of left and right multiplication operators learned by different optimization iterations. All experimental results are shown in Fig. 3 where the vertical axis correspond respectively τ_{PSNR} in subfig. 3(a) and τ_{SSIM} in subfig. 3(b), and both horizontal axis correspond to the number of optimization iterations.

According to Fig. 3, the means of τ_{PSNR} will increase from 1 to 2.2, and the means of τ_{SSIM} will increase from 1 to 1.4 along with the increase of the number of iterations. This demonstrates that some meaningful information is gradually learned by the iteration optimization method reported in Alg. 1 because the performance of the learned pairwise operators (L_k, R_k) is gradually improved in terms of PSNR and SSIM along with the increase of the number of optimization iterations. It could be also observed that the significant improvement of the means of τ_{PSNR} and τ_{SSIM} appears after the second iteration. It should be noticed that the first pair of nontrivial left and right multiplication operators was learned after two iterations. Considering the algebraic meaning of the right multiplication matrix, this pair of learned left and right multiplication operators brings row and column information of image patches into the matrix-value regression model. Because row and column information is simultaneously used in the learned matrix-value regression model, the performance of Alg. 1 is more promising.

The convergence of the means of τ_{PSNR} and τ_{SSIM} shows the convergence of Alg. 1. It is clear in Fig. 3 that the curves of the means of τ_{PSNR} and τ_{SSIM} approximate a line when the number of iterations gets large enough. Meanwhile, the variances of τ_{PSNR} and τ_{SSIM} get larger when the number of iterations gets larger. Both facts show that the power of learning more meaningful information is decaying along with the increase of the number of iterations. According to the experimental results shown in Fig. 3, the means of τ_{PSNR} and τ_{SSIM} cannot change significantly after 10 iterations. The experimental evidence shows that Alg. 1 will converge at the very beginning. Therefore, Alg. 1 will be stopped after 10 iterations in the following experiments.

C. Performance of POL

1) Quality of super-resolved images: Three groups of super-resolved images are shown in Figs. 4, 6, 8, 10 and 12 for comparing the performance of POL with the other known super-resolution algorithms in term of vision. For showing the visual quality of super-resolved images more clearly, the corresponding local images are also shown in Figs. 5, 7, 9, 11 and 13. Related experimental results about PSNR/SSIM and FSIM/VSI are reported in Tab. II.



(a) Curve of τ_{PSNR}

(b) Curve of τ_{SSIM}

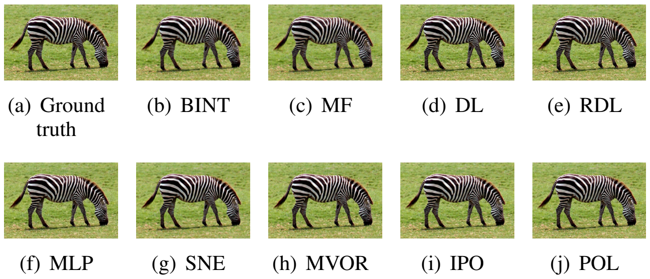
Fig. 3. Effects of the pairwise operators: The horizontal axis corresponds to the number of iterations which varies from 1 to 49 with a step of 1. τ_{PSNR} and τ_{SSIM} show the change of PSNR and SSIM against to the baselines when the number of iterations becomes larger where the first-round experimental results of PSNR and SSIM used separately as the baselines.

As shown in Tab. I, the algorithms compared with POL use different strategies to training themselves and reconstructing super-resolved images. BINT uses pixel-based information, and MF, DL and SNE employ the vector-based information which relates with the feature vectors extracted from images. MVOR and IPO designed with the similar strategy of POL focus on patch-based information which directly relates with the image patches generated from training and test images. Because of the difference in training strategies, the different performance of recovering high-resolution images with different super-resolution algorithms could be found from the following super-resolved images and the corresponding quantitative experimental results. Generally, all of these qualitative and quantitative experimental results show the patch-based information is more attractive than the other two.

In Fig. 4, the picture Zebra is used to test the performance of different super-resolution algorithm in recovering black and white stripe. According to the local images shown in Fig. 5, it is clear that the performance of the patch-based methods including MVOR, IPO and POL is better than the vector-based and pixel-based methods because less black diffusion is found in the white area of the zebra, for example the subfigures 5(h), 5(i), and 5(j). Comparing the performance of these triple patch-based algorithms, the super-resolved image generated by MVOR is more obscure than IPO's and POL's. Meanwhile, the quantitative results shown in Tab. II are shown that the performance of POL is better than IPO's in terms of PSNR/SSIM and FSIM/VSI. We think that all of these experimental results show that the image patches could offer more information for training a super-resolution algorithm, and the pairwise operator used in POL is more effective than single left multiplication operator used in MVOR or IPO in extracting the information of image patches.

Additional experimental results show the similar phenomena. The super-resolved images of the picture Butterfly (Fig. 6) are used to show the performance of preserving the structure constructing by slim lines. In Fig. 7, the super-resolved image (Fig.7(j)) generated by POL preserve more structure of the butterfly's wing similar to the ground truth shown in Fig. 7(a). Especially, comparing the low right part of these local images shown in Fig. 7, the edges of the texture shown in Fig. 7(j) are clearer and sharper. The experimental results shown in Fig. 8

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2016.2639440, IEEE Transactions on Image Processing



(a) Ground truth (b) BINT (c) MF (d) DL (e) RDL

(f) MLP (g) SNE (h) MVOR (i) IPO (j) POL

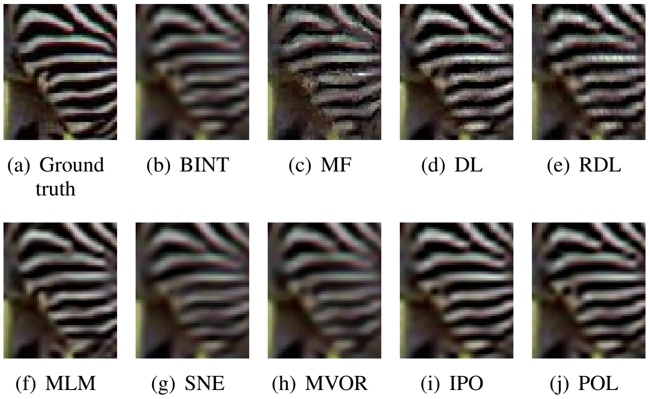
Fig. 4. Zebra: Ground truth and super-resolved results of different single-image super-resolution algorithms.

are used to comparing the performance of all considered super-resolution algorithms in preserving the structure consisted of little points. According to the results shown in Fig. 9, it is hard to identify the little points on the super-resolved lily generated by the point-based method (BINT) and the vector-based methods (MF, DL and SNE). However, more point structures could be recognized on the images generated by patch-based methods (MOVR, IPO and POL). Among these super-resolved images generated by MOVR, IPO and POL, the performance of POL is better than the other two in terms of PSNR, SSIM, FSIM and VSI because of using pairwise operator to extracting row and column information of image patches. The test image of Tablecloth (Fig.10(a)) is used to test the performance of preserving the structure of mesh. All super-resolved results of Tablecloth are shown in Fig. 10. Just as the super-resolved results shown in Fig. 11, only the super-resolved results generated by IPO and POL show a mesh-style structure which is similar to the ground truth structure. The performance of super-resolving characters is also tested even though the images of characters are out of the range of the training images. The super-resolved results of Characters are shown in Fig. 12. Observing the local images shown in Fig. 13, the arc structure of the character 'g' is well preserved by the super-resolved result of POL. Though similar phenomena could be found in the super-resolved results of SNE, MVOR, and IPO, POL's result is better than the others in terms of PSNR and SSIM.

The quantitative experimental results including PSNR, SSIM, FSIM and VSI of all super-resolved results shown in Tab. II support the qualitative experimental results shown above in most cases. According to Tab. II, the optimal experimental results belong to POL with high probability.

2) CPU time of training and test: All of these super-resolution algorithms are trained and tested on the personal computer with an Intel(R) Core(TM) i3-3217U CPUs(1.80GHz) and 2.00GB RAM. No parallel technique is used in all super-resolution algorithms tested by us. All experimental results on CPU time are reported in Tab. III.

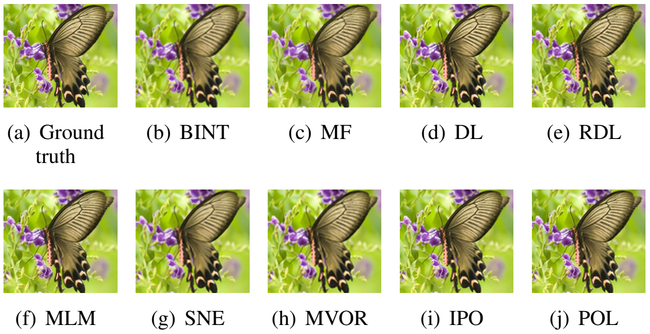
According to Tab. III, the patch-based algorithms have more advantages than vector-based algorithms in training and testing CPU time because of replacing the vector calculation by the matrix calculation. More detailed discussions on the efficiency of using matrix calculation in the field of single-image super-resolution have reported in [9] or [8]. The CPU time of training POL is slightly longer than the time of training the fastest



(a) Ground truth (b) BINT (c) MF (d) DL (e) RDL

(f) MLM (g) SNE (h) MVOR (i) IPO (j) POL

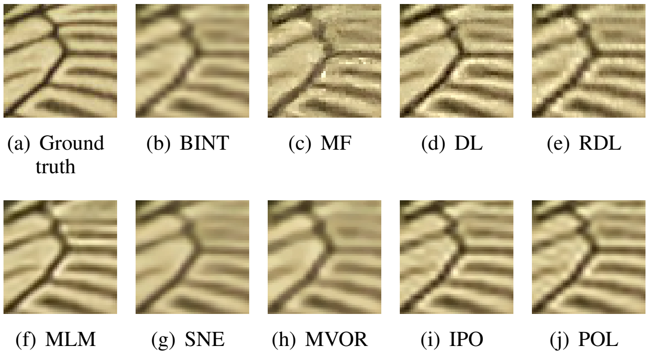
Fig. 5. Local images of the ground truth and super-resolved Zebra



(a) Ground truth (b) BINT (c) MF (d) DL (e) RDL

(f) MLM (g) SNE (h) MVOR (i) IPO (j) POL

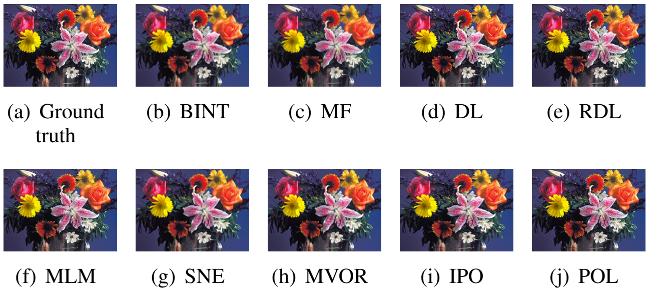
Fig. 6. Butterfly: Ground truth and super-resolved results of different single-image super-resolution algorithms.



(a) Ground truth (b) BINT (c) MF (d) DL (e) RDL

(f) MLM (g) SNE (h) MVOR (i) IPO (j) POL

Fig. 7. Local images of the ground truth and super-resolved Butterfly



(a) Ground truth (b) BINT (c) MF (d) DL (e) RDL

(f) MLM (g) SNE (h) MVOR (i) IPO (j) POL

Fig. 8. Flowers: Ground truth and super-resolved results of different single-image super-resolution algorithms.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2016.2639440, IEEE Transactions on Image Processing

TABLE II
PSNR/SSIM AND FSIM/VSI OF SUPER-RESOLVED RESULTS

Fig. / Alg.	Zebra	Butterfly	Flowers	Tablecloth	Characters
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
BINT	24.4384/0.788871	22.6693/0.863462	15.6736/0.662267	17.6842/0.685531	14.5375/0.809692
MF	25.7421/0.733008	27.1119/0.838649	26.4377/0.743447	25.6724/0.706499	23.0025/0.851883
DL	26.9204/0.82603	28.9778/0.899462	27.7657/0.816394	26.067/0.745366	23.6447/0.833726
RDL	26.7247/0.81703	28.1662/0.881333	27.3581/0.802569	26.2163/0.742701	23.5181/0.843319
MLM	26.8085/0.827681	28.2182/0.915265	27.7872/0.80445	26.2015/0.75388	23.9364/0.899195
SNE	26.0683/0.764756	27.6428/0.865422	27.0133/0.779194	25.8126/0.719243	23.4302/0.871481
MOVR	26.729/0.800548	28.2125/0.884023	27.4539/0.806092	26.0899/0.736519	23.5517/0.884761
IPO	27.4663/0.841975	29.7314/0.917275	27.7891/0.822953	26.8912/0.775717	24.6545/0.879675
POL	**28.1477/0.851719**	**29.7752/0.924526**	**28.213/0.830801**	**26.9117/0.782724**	**24.7798/0.885434**
	FSIM/VSI	FSIM/VSI	FSIM/VSI	FSIM/VSI	FSIM/VSI
BINT	0.937812/0.97591	0.888631/0.963502	0.844818/0.952489	0.951883/0.984445	0.79902/0.932565
MF	0.930198/0.972148	0.872506/0.956238	0.831981/0.945701	0.944816/0.982696	0.794457/0.932518
DL	0.959794/0.985435	0.902297/0.967317	0.865558/0.956997	0.958477/0.987229	0.79982/0.928281
RDL	0.951038/0.98.213	0.887987/0.96317	0.869205/0.958313	0.958173/0.986899	0.775652/0.922208
MLM	0.97516/0.989241	**0.923941/0.971373**	0.860927/0.959041	**0.969587**/0.988804	**0.845651/0.948148**
SNE	0.921663/0.967473	0.881482/0.95959	0.838037/0.948869	0.940772/0.980121	0.825859/0.94249
MOVR	0.941763/0.976147	0.882229/0.960313	0.844301/0.951027	0.952567/0.9838	0.820513/0.939563
IPO	0.955805/0.984365	0.911859/0.970025	0.860528/0.956934	0.969569/0.989993	0.810684/0.934826
POL	**0.966494/0.987869**	0.910226/0.969887	**0.868662/0.959302**	0.968868/**0.989866**	0.807559/0.934364

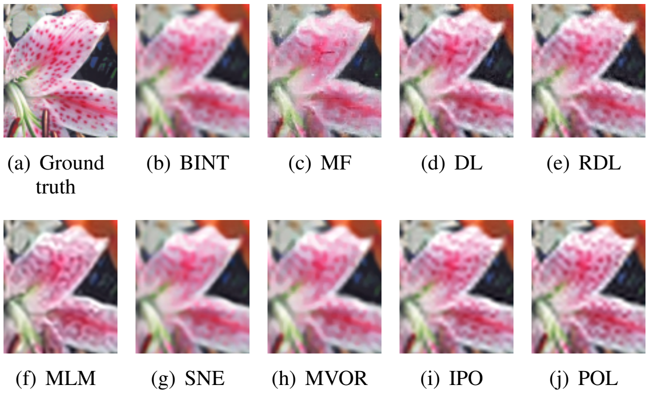

(a) Ground truth (b) BINT (c) MF (d) DL (e) RDL
(f) MLM (g) SNE (h) MVOR (i) IPO (j) POL

Fig. 9. Local images of the ground truth and super-resolved Flowers

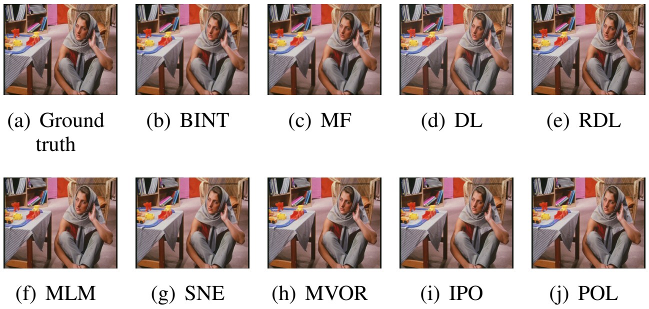

(a) Ground truth (b) BINT (c) MF (d) DL (e) RDL
(f) MLM (g) SNE (h) MVOR (i) IPO (j) POL

Fig. 10. Tablecloth: Ground truth and super-resolved results of different single-image super-resolution algorithms.

algorithm MVOR, however its testing CPU time is much faster than MOVR's because a larger patch size is used in POL. Similar to POL, a larger patch size is also used in IPO, however the CPU time of training IPO is significantly longer than POL's because square optimization is used in IPO. Therefore, by employing pairwise operator to describe the relation between low- and high-resolution image patches, POL could well balance the CUP time of training and testing.

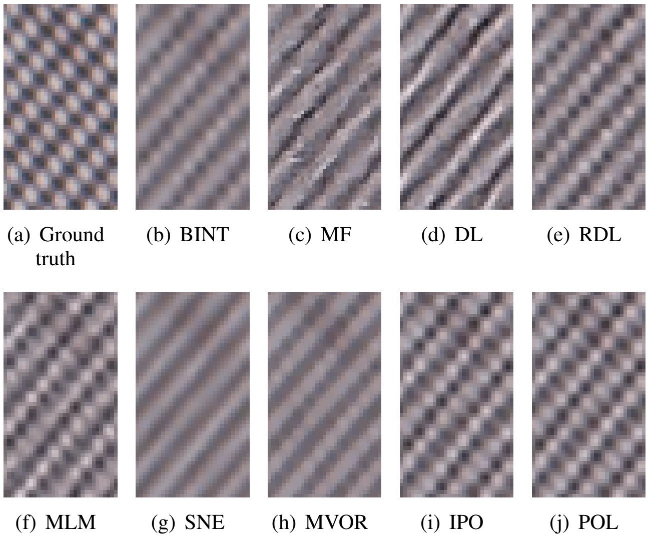
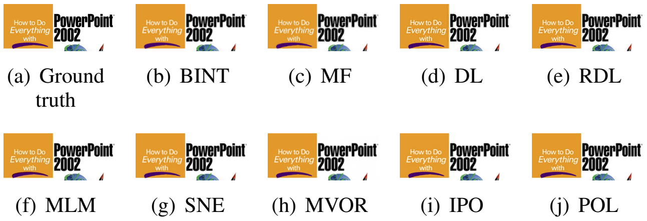

(a) Ground truth (b) BINT (c) MF (d) DL (e) RDL
(f) MLM (g) SNE (h) MVOR (i) IPO (j) POL

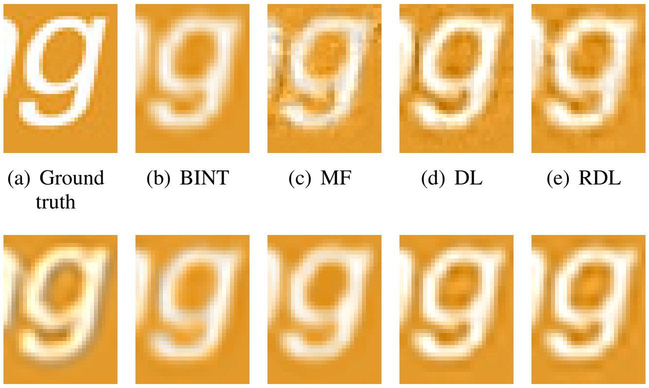
Fig. 11. Local images of the ground truth and super-resolved Tabelcloth



(a) Ground truth (b) BINT (c) MF (d) DL (e) RDL
(f) MLM (g) SNE (h) MVOR (i) IPO (j) POL

Fig. 12. Characters: Ground truth and super-resolved results of different single-image super-resolution algorithms.

3) Size of learned models: In Tab. III, the sizes of storing learned models by different algorithms are reported. It is clear that all training samples should be stored for MF which means very large size of data is needed. In fact, the largest storage size belongs to MF. All patch-based algorithms including MOVR, IPO and POL just need significantly small



(a) Ground truth (b) BINT (c) MF (d) DL (e) RDL

(f) MLM (g) SNE (h) MVOR (i) IPO (j) POL

Fig. 13. Local images of the ground truth and super-resolved Characters

size to store the learned models. Especially, the smallest size belongs to MOVR. Therefore, patch-based algorithms are also more efficient in storing learned models than other mentioned algorithms.

V. Conclusion

In this paper, we discussed a novel operator learning method for single-image super-resolution which describes the relation between low- and high-resolution image patches with a pair of left and right multiplication operators. By using left and right multiplication simultaneously, the raw and column information between low- and high-resolution image patches which are represented by matrices could be separately described. In subsection IV-B, we empirically showed that the pairwise operator learning method is more effective than the existing left operator learning methods which just benefit from column information of image patches. Additional experimental results shown in subsection IV-C also indicated that the performance of POL is not only competitive to some existing vector- and matrix-based single-image super-resolution algorithms but also efficient in training and testing processes. Moreover, the size of storing the learned model is far smaller than most existing single-image super-resolution algorithms.

Acknowledgment

This work was partly supported by National Natural Science Foundation of China (Grant no. 61462096) and the Science and Technology Plan Project of Yunnan Province (Grant no. 2014FB148). The corresponding author is Ling Shao.

References

[1] W. Freeman, T. Jones, and E. Pasztor, “Example-based super-resolution,” *IEEE Computer Graphics and Applications*, vol. 22, pp. 56–65, 2002.

[2] H. Chang, D. Yeung, and Y. Xiong, “Super-resolution through neighbor embedding,” in *Proc. IEEE CVPR’04*, vol. 1, Jun. 2004, pp. 275–282.

[3] J. Yang, J. Wright, T. Huang, and Y. Ma, “Image super-resolution as sparse representation of raw image patches,” in *Proc. IEEE CVPR’08*, Aug. 2008, pp. 1–8.

[4] X. Gao, K. Zhang, X. Li, and D. Tao, “Joint learning for single-image super-resolution via a coupled constraint,” *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 469–480, 2012.

[5] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[6] X. Gao, K. Zhang, X. Li, and D. Tao, “Image super-resolution with sparse neighbor embedding,” *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3194–3205, 2012.

[7] Y. Tang, Y. Yuan, P. Yan, and X. Li, “Greedy regression in sparse coding space for single-image super-resolution,” *Journal of Visual Communication and Image Representation*, vol. 24, no. 2, pp. 148–159, 2012.

[8] Y. Tang and Y. Yuan, “Image pair analysis with matrix-value operator,” *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2042–2050, 2015.

[9] Y. Tang, H. Chen, Z. Liu, B. Song, and Q. Wang, “Example-based super-resolution via social images,” *Neurocomputing*, vol. 172, pp. 38–47, 2016.

[10] W. Freeman, E. Pasztor, and O. Carmichael, “Learning low-level vision,” *International Journal of Computer Vision*, vol. 40, no. 1, pp. 25–47, 2000.

[11] K. Kim and Y. Kwon, “Single-image super-resolution using sparse regression and natural image prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1127–1133, Jun. 2010.

[12] S. T. Roweis and L. K. Saul, “Locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[13] B. Li, H. Chang, S. Shan, and X. Chen, “Locality preserving constraints for super-resolution with neighbor embedding,” in *Proc. IEEE ICIP’09*, Nov. 2009, pp. 1189–1192.

[14] K. Zhang, X. Gao, X. Li, and D. Tao, “Partially supervised neighbor embedding for example-based image super-resolution,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 2, pp. 230–239, 2011.

[15] L. Pan, G. Peng, W. Yan, and H. Zheng, “Single image super resoltuion based on multiscale local simialrity and neighbor embedding,” *Neurocomputing*, vol. 207.

[16] M. Lewicki and T. Sejnowski, “Learning overcomplete representations,” *Neural Computation*, vol. 12, pp. 337–365, 2000.

[17] J. Mairal, M. Elad, and G. Sapiro, “Sparse representation for color image restoration,” *IEEE Transactions on Image Processing*, vol. 17, pp. 53–69, 2008.

[18] M. Aharon, M. Elad, and A. Bruckstein, “K-svd: an algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[19] H. Lee, A. Battle, and A. Y. Ng, “Efficient sparse coding algorithms,” in *Advances in Neural Information Processing Systems*, 2007, pp. 801–808.

[20] X. Lu, H. Yuan, P. Yan, Y. Yuan, and X. Li, “Geometry constrained sparse coding for single image super-resolution,” in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1648–1655.

[21] W. Dong, L. Zhang, G. Shi, and X. Wu, “Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization,” *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1838–1857, 2011.

[22] K. Zhang, X. Gao, D. Tao, and X. Li, “Multi-scale dictionary for single image super-resolution,” in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1114–1121.

[23] G. Mu, X. Gao, K. Zhang, X. Li, and D. Tao, “Single image super resolution with high resolution dictionary,” in *proceedings of IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 1141–1144.

[24] S. Ye, C. Deng, J. Xu, and X. Gao, “Coupled fisher discrimination dictionary learning for single image super-resolution,” in *ICASSP 2015*.

[25] J. Yu, X. Gao, D. Tao, X. Li, and K. Zhang, “A unified learning framework for single image super-resolution,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 4, pp. 780–792, 2014.

[26] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[27] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge: Cambridge Univeersity Press, 2004.

[28] K. Ni and T. Nquyen, “Image superresolution using support vector regression,” *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1596–1610, Jun. 2007.

[29] K. Zhang, X. Gao, D. Tao, and X. Li, “Single image super-resolution with non-local means and steering kernel regression,” *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4544–4556, 2012.

[30] K. Zhang, X. Gao, J. Li, and H. Xia, “Single image super-resolution using regularization of non-local steering kernel regression,” *Signal Processing*, vol. 123.

[31] H. Wang, X. Gao, K. Zhang, and J. Li, “Single image super-resolution using active-sampling ’g’aussian process regression,” *Signal Processing*, vol. 123.

TABLE III
CPU TIME AND THE SIZE OF LEARNED MODELS

Fig. / Alg.	Testing					Training	Size of the learned model
	Zebra	Butterfly	Flowers	Tablecloth	Characters		
MF	350.124s	189.38s	285.147s	675.449s	225.962s	0s	27.5MB
DL	22.3746s	11.1113s	20.8126s	37.4243s	6.70069s	>24h	1.68MB
RDL	86.2296s	36.053s	71.4358s	146.135s	28.0431s	>24h	29.768MB
MLM	35.0793s	14.1187s	25.4407s	60.2670s	16.68s	>2h	6.875MB
SNE	1187.18s	574.794s	923.632s	2371.09s	650.252s	4370s	54.7MB
MOVR	5.57762s	2.29097s	4.13453s	9.25232s	2.68356s	**0.4629**s	**397B**
IPO	0.0786006s	0.0304336s	0.0751676s	1.29031s	0.385514s	3.7870s	4.87KB
POL	**0.0430029**s	**0.013236**s	**0.0308841**s	**0.673918**s	**0.259661**s	1.5837s	9.42KB

[32] H. He and W. Siu, “Single image super-resolution using gaussian process regression,” in *Proc. IEEE CVPR’11*, 2011, pp. 449–456.

[33] Y. Tang and H. Chen, “Matrix-value regression for single-image super-resolution,” in *ICWAPR2013*, 2013, pp. 215–220.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, 2012, pp. 1097–1105.

[35] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in *Proceedings of European Conference on Computer Vision (ECCV)*, 2014, pp. 184–199.

[36] ——, “Image super-resolution using deep convolutional networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 295–307, 2016.

[37] D. Liu, Z. Wang, B. Wen, J. Yang, W. Han, and T. S. Huang, “Robust single image super-resolution via deep networks with sparse prior,” *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3194–3207, 2016.

[38] R. Yan and L. Shao, “Blind image blur estimation via deep learning,” *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1910–1921, 2016.

[39] W. Zhang, J. Han, J. Han, and L. Shao, “Cosaliency detection based on intrasaliency prior transfer and deep intersaliency mining,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 6, pp. 1163–1176, 2016.

[40] D. Wu, L. Pigou, P.-J. Kindermans, N. Le, L. Shao, J. Dambre, and J.-M. Odobez, “Deep dynamic neural networks for multimodal gesture segmentation and recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1583–1597, 2016.

[41] K. Zhang, D. Tao, X. Gao, X. Li, and Z. Xiong, “Learning multiple linear mappings for efficient single image super-resolution,” *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 846–861, 2015.

[42] Q. Zhu, J. Mai, and L. Shao, “A fast single image haze removal algorithm using color attenuation prior,” *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3522–3533, 2015.

[43] N. Zhao, Q. Wei, A. Basarab, N. Dobigeon, D. Kouame, and J. Tourneret, “Fast single image super-resolution using a new analytical solution for ℓ_2-ℓ_2 problems,” *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3683–3697, 2016.

[44] N. Kumar and A. Sethi, “Fast learning-based single image super-resolution,” *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1504–1515, 2016.

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

[46] L. Zhang, L. Zhang, X. Mou, and D. Zhang, “FSIM: a feature similarity index for image quality assessment,” *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2383, 2011.

[47] L. Zhang, Y. Shen, and H. Li, “VSI: a visual saliency induced index for perceptual image quality assessment,” *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4270–4281, 2014.

Yi Tang is currently an Associate Professor with Key Laboratory of IOT Application Technology of Universities in Yunnan Province and the department of mathematics and computer Science, Yunnan Minzu University, Kunming, China. He received the B.S., the M.S. and Ph.D. degrees in mathematics from Hubei University, Wuhan, China, in 2001, 2006, and 2010, respectively. His research interests include machine learning, statistical learning theory, image processing, and pattern recognition.

PLACE
PHOTO
HERE

Ling Shao (M’09-SM’10) is a professor with the School of Computing Sciences at the University of East Anglia, Norwich, UK. Previously, he was a professor (2014-2016) with Northumbria University, a senior lecturer (2009-2014) with the University of Sheffield and a senior scientist (2005- 2009) with Philips Research, The Netherlands. His research interests include computer vision, image/video processing and machine learning. He is an associate editor of IEEE Transactions on Image Processing, IEEE Transactions on Neural Networks and Learning Systems and several other journals. He is a Fellow of the British Computer Society and the Institution of Engineering and Technology. He is a senior member of the IEEE.