Xu *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:41
DOI 10.1186/s13634-015-0225-y

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Noisy image magnification with total variation regularization and order-changed dictionary learning

Jian Xu[1,2*], Zhiguo Chang[3], Jiulun Fan[1], Xiaoqiang Zhao[1], Xiaomin Wu[1] and Yanzi Wang[1]

Abstract

Noisy low resolution (LR) images are always obtained in real applications, but many existing image magnification algorithms can not get good result from a noisy LR image. We propose a two-step image magnification algorithm to solve this problem. The proposed algorithm takes the advantages of both regularization-based method and learning-based method. The first step is based on total variation (TV) regularization and the second step is based on sparse representation. In the first step, we add a constraint on the TV regularization model to magnify the LR image and at the same time to suppress the noise in it. In the second step, we propose an order-changed dictionary training algorithm to train the dictionaries which is dominated by texture details. Experimental results demonstrate that the proposed algorithm performs better than many other algorithms when the noise is not serious. The proposed algorithm can also provide better visual quality on natural LR images.

Keywords: Image magnification; Super-resolution; Total variation regularization; Dictionary training; Sparse representation; Image denoising

1 Introduction

The technology of image magnification focuses on how to magnify a low resolution (LR) image and at the same time recover some high resolution (HR) details. The methods of this technology can be divided into three categories: the method based on up scaling [1], the method based on reconstruction [2-5], and the method based on learning [6]. Some methods based on up scaling, such as bilinear and bicubic interpolation (BI) [7], are popular since they have low computational complex, but they always produce blurring edges and suffer from artifacts since they use the invariant kernels for all kinds of local textures. Methods based on reconstruction aim at reconstructing the HR image by imitating the inverse process of degradation [2]. These methods rely on the rationality of the reconstructing model. The methods based on up scaling and reconstruction have smaller memory space costs than the learning-based methods in most of the cases. But it is difficult to use some simple mathematical models to fit the sophisticated natural conditions. This makes these methods can not recover many texture details. The learning-based methods are more flexible to deal with the problem [6]. They use the training images to learn the relationship between the HR and LR images, and many existing works have demonstrated their good effect for the high magnification factors.

There are two important aspects in the learning-based algorithms. The first is the feature extraction methods. The second is the learning models.

Many existing feature extraction methods can be utilized to extract features for image magnification problem. Gradient features [6,8], Gabor features [9], fields of experts (FoE) [10] features and histogram oriented gradients (HoG) [11] are developed. To deal with different texture features by different strategies, the input image can be separated into edge and texture components [12], shape and texture components [13,14], different texture regions [15], or different frequency bands [16,17].

The main idea of many existing learning-based models is to use some tools to learn the relationship between the

*Correspondence: xujian_paper@126.com
[1] School of Telecommunication and Information Engineering, Xi'an University of Posts and Telecommunications, Weiguo Road, 710121 Xi'an, China
[2] Image Processing and Recognition Center, Xi'an Jiaotong University, Xianning road, 710049 Xi'an, China
Full list of author information is available at the end of the article

LR and HR images. Neighbor embedding (NE) is based on the assumption that the LR and HR local patches have similar geometries in two distinct feature spaces [18]. However, finding neighbors in millions of data samples is a high time-exhaustive task for the NE-based algorithm. Canonical correlation analysis (CCA) [19-21] assumes that the corresponding HR and LR images have great inner product similarity after a transformation. Compared to the NE-based methods, CCA can accomplish the transformation with lower computational complexity. Sparse representation-based models are widely used [22] in image processing because of its good generalization ability. Yang et al. [6,8] proposed a classical model to transform the HR and LR images into a unified subspace. They suppose the HR and LR images should have the same sparse representations in the subspace. To accomplish the transformation, coupled dictionary training is an important step. Yang et al. proposed joint learning [6] and coupled learning [23] algorithms to train coupled dictionaries. The joint learning algorithm combines the LR and HR patch pairs together to convert the coupled dictionary training task into a single dictionary training task. However, the reliable sparse representations are not guaranteed to be found in the test phase. Yang's coupled learning algorithm [23] uses the alternately steepest descent algorithm to update the LR and HR dictionaries. Zeyde et al. [8] use a single dictionary training algorithm to train the LR dictionary and then generate the HR dictionary by solving a least square problem. Xu et al. [24] alternately update the LR and HR dictionaries with K-singular value decomposition (K-SVD). In these dictionary training algorithms, Zeyde's algorithm has the smallest time complexity. Since it is a too strict condition to let the LR and HR sparse representations to be exactly the same, some tools (such as the neural network [25] and linear transformation [26,27]) are employed to model the relationship between the two sparse representations. To accelerate the sparse representation-based algorithm, Timofte et al. group the dictionary atoms [28] or the training samples [29] to decrease the time complexity of calculating the sparse representations. Some algorithms can provide excellent results on some special image classes (such as face [30] and buildings [31]). Besides the abovementioned tools, support vector regression (SVR) [32], Kernel-based regression [33], deep convolutional neural network [34], and fuzzy rule-based prediction [35] are also used as the tools to solve the image magnification problem.

In real applications, the obtained LR images always contain noise (such as taking photos in low-light or strong interference conditions). Since some existing algorithm is not good at dealing with the noisy LR image, we propose an algorithm to cover the shortage. The destination of this algorithm is to reconstruct a clear HR image according to a noisy LR image. The proposed algorithm takes the advantage of both the regularization-based method and the learning-based method. We firstly use the regularization-based method to suppress the noise and then use the learning-based method to recover the details. To make it simple, we briefly call the proposed method total variation and order-changed dictionary training (TV-OCDT) algorithm.

Our contributions can be summarized as follows:

1) We propose a constraint for the total variation (TV) regularization-based image magnification model. The constraint is helpful to suppress the noise and recover sharp edges.
2) We propose an order-changed dictionary training algorithm to train the coupled dictionaries. The traditional dictionary training algorithm firstly trains the LR dictionary. Then, generate the HR dictionary according to the LR dictionary. But we firstly train the HR dictionary and then generate the LR dictionary according to the HR dictionary. This strategy changes the dominated content of the dictionaries so that the texture details can be recovered well. Experimental results show that the proposed algorithm is superior to others on the noisy images. The remainder of this paper is organized as follows. Section 2 describes the proposed algorithm. The experimental results are presented in Section 3. Section 4 concludes this paper.

2 The proposed algorithm

If the input LR image has noise, how could we deal with it? An idea that flashed into the mind may be firstly denoising the LR image and then magnify it. But it is difficult to be executed, since the textures are dense and incomplete in the LR image. Therefore, we propose an algorithm to solve this problem. The framework of the proposed algorithm is shown in Figure 1. A TV regularization-based algorithm is employed to simultaneously accomplish magnification and denoising at first. The details of the proposed TV regularization model will be described in Section 2.1. After the TV regularization, some texture details are damaged. We use an OCDT algorithm to compensate the texture details. The details of this step will be shown in Section 2.2.

2.1 TV regularization with LR constraint

In real applications, we often obtain the noisy LR images. If we directly use some magnification algorithm on these images, the noise will be magnified simultaneously. The strategy of denoising at first is not a good choice. Many existing denoising algorithm [36] works very well on HR images, but can not be executed on LR images because

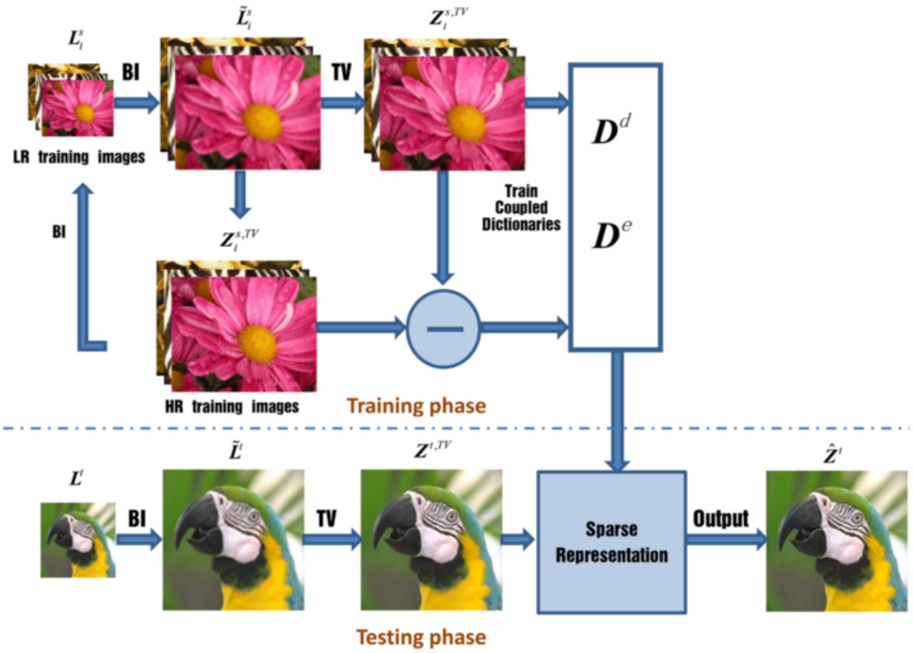

Figure 1 The framework of the proposed algorithm.

the textures are dense and incomplete (as shown in Figure 2).

To fit the recovered HR image to the initial input LR image $\mathbf{L}^s$, the famous iterative back projection (IBP) [38] algorithm is widely used in image magnification technology. It can be executed without storing any tools (such as data samples or dictionaries) and has low computational complexity.

The model of IBP is as follows:

$$\mathbf{Z}^{s,IBP} = \arg\min_{\mathbf{Z}} \left\| F(\mathbf{Z}) - \mathbf{L}^s \right\|_2^2 \tag{1}$$

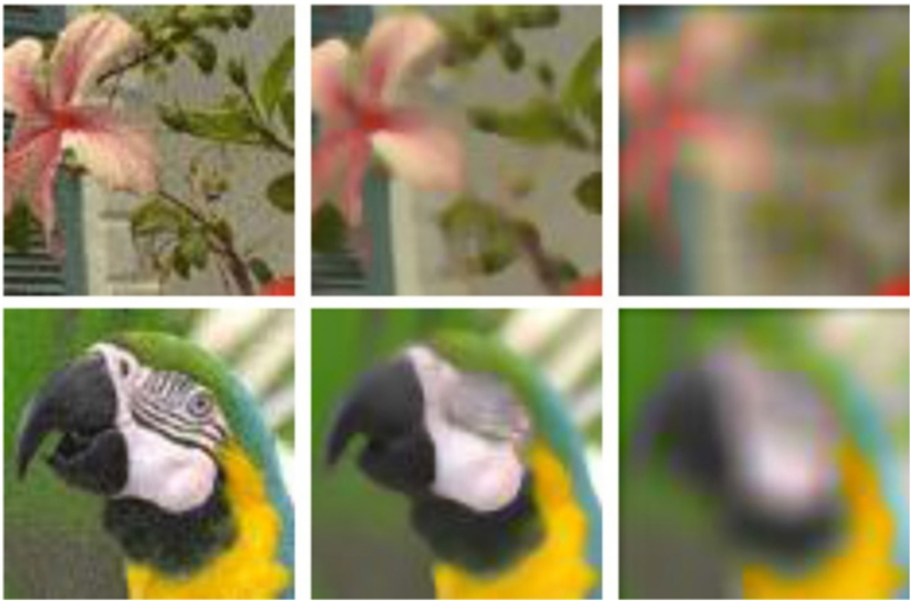

Figure 2 The results of applying some famous denoising algorithm on the LR images. The first column shows the noisy LR images. The second column shows the results of median filter [37]. The third column shows the results of non-local mean denoising [36].

where $\mathbf{Z}^{s,IBP}$ is the reconstructed HR image of IBP and '$F(\cdot)$' is the operation of down sampling by BI.

The model (1) can be solved in the following ways:

$$\mathbf{Z}^{J+1} = \mathbf{Z}^J + \lambda[\,U(\mathbf{L}^s - F(\mathbf{Z}^J))] \tag{2}$$

where $\mathbf{Z}^J$ is the output of the Jth iteration and '$U(\cdot)$' is the operation of up sampling by BI.

If the input is a noisy image, the model of IBP certainly will propagate the noise to the HR image and the noise will be even magnified (as shown in Figure 3). Therefore, some constraints should be added on the IBP model so that it can suppress the noise.

The traditional TV regularization [39] considers that the HR images have a small TV norm. The reconstructed image is obtained from :

$$\mathbf{Z}^{s,TV} = \arg\min_{\mathbf{Z}}\{\int |G(\mathbf{Z})| + \frac{\lambda}{2}\left\|F(\mathbf{Z}) - \mathbf{L}^s\right\|_2^2\} \tag{3}$$

where $\mathbf{Z}^{s,TV}$ is the reconstructed HR image of TV regularization and λ is a Lagrange multipliers, '$G(\cdot)$' is to calculate the gradients. But we found the traditional TV regularization is not effective enough to suppress the noise in image magnification (as shown in Figure 3).

To suppress the noise in LR images, we add another constraint. We propose the following model:

$$\mathbf{Z}^{s,TV} = \arg\min_{\mathbf{Z}}\left\{\int |G(\mathbf{Z})| + \lambda_1 \int |G\left(F(\mathbf{Z})\right)| + \frac{\lambda_2}{2}\left\|F(\mathbf{Z}) - \mathbf{L}^s\right\|_2^2\right\} \tag{4}$$

where λ_1 and λ_2 are the Lagrange multipliers. Obviously, the motivation of the constraint is to make the LR image also have a small TV norm. This motivation is inspired by the classical Rudin-Osher-Fatemi (ROF) TV denoising model [40].

Inspired by the reference [39], the model (4) is solved with the following method.

Define the following functions:

$$\begin{aligned}
\Phi_1(\mathbf{X}) &= [\,(\mathbf{X})_x^2 + (\mathbf{X})_y^2 + \epsilon\mathbf{1}]\cdot^{\frac{3}{2}}\\
\Phi_2(\mathbf{X}) &= (\mathbf{X})_{xx} \bullet ((\mathbf{X})_y^2 + \epsilon\mathbf{1})\\
\Phi_3(\mathbf{X}) &= (\mathbf{X})_{xy} \bullet (\mathbf{X})_x \bullet (\mathbf{X})_y\\
\Phi_4(\mathbf{X}) &= (\mathbf{X})_{yy} \bullet ((\mathbf{X})_x^2 + \epsilon\mathbf{1})
\end{aligned} \tag{5}$$

where $(\cdot)\cdot^{\frac{3}{2}}$ is to calculate the $\frac{3}{2}$ power for every element in the matrix and $\mathbf{1}$ stands for a matrix of ones of the proper size. '$\bullet$' is to multiply the corresponding elements in matrices. The gradient items are generated with the four gradient filters: $\mathbf{f}_1 = [-1\ 0\ 1]$, $\mathbf{f}_2 = \mathbf{f}_1^T$, $\mathbf{f}_3 = [-1\ 0\ 2\ 0\ 1]$, $\mathbf{f}_4 = \mathbf{f}_3^T$, where '$T$' is transposition.

$$\begin{aligned}
(\mathbf{X})_x &= \mathbf{X} * \mathbf{f}_1 \quad (\mathbf{X})_y = \mathbf{X} * \mathbf{f}_2\\
(\mathbf{X})_{xx} &= \mathbf{X} * \mathbf{f}_3 \quad (\mathbf{X})_{yy} = \mathbf{X} * \mathbf{f}_4
\end{aligned} \tag{6}$$

where '$*$' is convolution.

The model (4) is solved by the following iterative formula:

$$\begin{aligned}
\mathbf{Z}^{J+1} &= \mathbf{Z}^J + \mathbf{A}^J + \lambda_1\mathbf{B}^J + \lambda_2\mathbf{C}^J\\
\mathbf{A}^J &= (\Phi_2(\mathbf{Z}^J) - 2\Phi_3(\mathbf{Z}^J) + \Phi_4(\mathbf{Z}^J)) \bullet / \Phi_1(\mathbf{Z}^J)\\
\mathbf{B}^J &= U\big[\big(\Phi_2(F(\mathbf{Z}^J)) - 2\Phi_3(F(\mathbf{Z}^J))\\
&\quad + \Phi_4(F(\mathbf{Z}^J))\big) \bullet / \Phi_1(F(\mathbf{Z}^J))\big]\\
\mathbf{C}^J &= U[\,\mathbf{L}^s - F(\mathbf{Z}^J)]
\end{aligned} \tag{7}$$

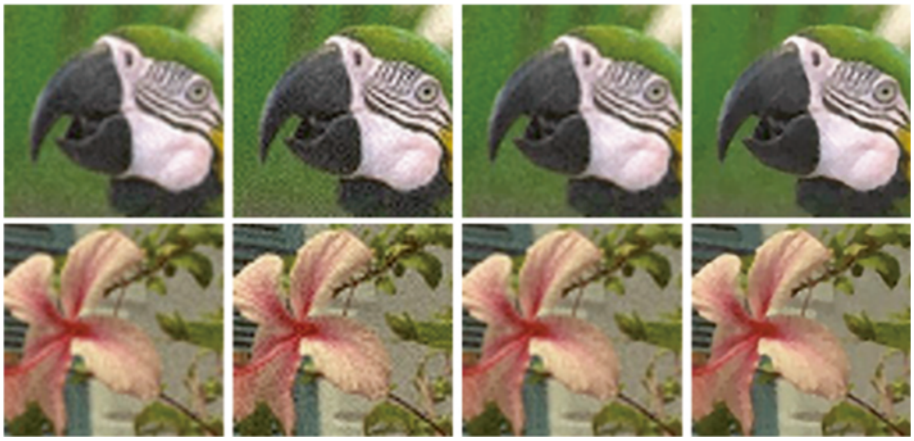

Figure 3 Visual comparison of the outputs of the traditional and the proposed TV regularization. The first column shows the outputs of BI without any other treatment. The second column shows the outputs of IBP. The third column shows the outputs of traditional TV regularization. The fourth column shows the outputs of the proposed TV regularization.

where '•/' is to do division on the elements of matrices, ϵ is a positive parameter to avoid singularity and is set to 1 according to [39].

Figure 3 compares the images before and after TV regularization. As shown, the proposed TV regularization constraint is benefit to suppress the noise and reconstruct sharp step edges. More experimental results shown in Section 3 will further demonstrate the effect of the proposed TV model.

2.2 OCDT algorithm

Since most of the smooth part and step edges have been recovered by the previous steps, we use the sparse representation-based algorithm to recover the missing texture details. Most of the sparse representation-based image magnification methods work on patches. We also use this strategy. As the operation in Zeyde's method [8], we produce four gradient maps of $\mathbf{Z}^{s,TV}$ with four filters $(\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4)$. These gradient maps are divided into patches, and four gradient patches in a given position are connected into a $4p^2$ dimensional column vector to form the data set $\{\mathbf{p}_i^{s,d}\}_{i=1}^N$, where $\mathbf{p}_i^{s,d}$ is the feature vector for the ith patch, N is the patch number. Difference between the desired HR image $\mathbf{Z}^s$ and $\mathbf{Z}^{s,TV}$ is calculated with:

$$\mathbf{Z}^{s,e} = \mathbf{Z}^s - \mathbf{Z}^{s,TV} \tag{8}$$

The data set $\left\{\mathbf{p}_i^{s,e}\right\}_{i=1}^N$ is generated by dividing $\mathbf{Z}^{s,e}$ into patches, where $\mathbf{p}_i^{s,e}$ and $\mathbf{p}_i^{s,d}$ are corresponding patches.

The classical coupled dictionary training model [23] is:

$$\begin{aligned}&\min_{\mathbf{D}^d,\mathbf{D}^e,\{\alpha_i\}} \textstyle\sum_{i=1}^N \left(\|\mathbf{p}_i^{s,e} - \mathbf{D}^e\alpha_i\|_2^2 + \|\mathbf{p}_i^{s,d} - \mathbf{D}^d\alpha_i\|_2^2\right)\\ &s.t\ \|\alpha_i\|_0 \le \hat{T}, \|\mathbf{d}_r^d\|_2 \le 1,\ \|\mathbf{d}_r^e\|_2 \le 1, r = 1, 2, \cdots, n\end{aligned} \tag{9}$$

where α_i is the sparse representation, $\mathbf{D}^d$ and $\mathbf{D}^e$ are dictionaries corresponding to $\{\mathbf{p}_i^{s,d}\}_{i=1}^N$ and $\{\mathbf{p}_i^{s,e}\}_{i=1}^N$, respectively, $\mathbf{d}_r^d$ and $\mathbf{d}_r^e$ are their rth dictionary atoms, $\hat{T}$ is the sparseness constraint.

This model can be easily solved with Zeyde's method [8]. But it is not proper to use the strategy to generate the $\mathbf{D}^d$ firstly and then calculate the $\mathbf{D}^e$ according to the $\mathbf{D}^d$. We should change the order of the dictionary training. According to the observation, we found $\mathbf{Z}^{s,TV}$ is dominated by smooth regions and step edges, but lacks texture details. The smooth regions can be well recovered by BI, so the smooth training patches are dropped in dictionary training stage as Yang's operation [23]. If we train the $\mathbf{D}^d$ firstly, $\mathbf{D}^d$ will be dominated by step edges. But we need coupled dictionaries dominated by texture details, since the step edges have been recovered by TV regularization in the previous steps. $\mathbf{Z}^{s,e}$ contains the lost texture details in $\mathbf{Z}^s$. Therefore, we firstly train $\mathbf{D}^e$ and then calculate $\mathbf{D}^d$ according to $\mathbf{D}^e$.

We firstly train $\mathbf{D}^e$. We solve the standard single dictionary training problem with K-SVD [41]:

$$\begin{aligned}&\min_{\mathbf{D}^e,\{\alpha_i\}} \sum_{i=1}^N \left\|\mathbf{p}_i^{s,e} - \mathbf{D}^e\alpha_i\right\|_2^2,\\ &s.t.\ \left\|\mathbf{d}_r^e\right\|_2 \le 1, \|\alpha_i\|_0 \le \hat{T}, r = 1, 2, \cdots, n\end{aligned} \tag{10}$$

The next issue is how to estimate the $\mathbf{D}^d$ that can provide similar sparse representation α_i for $\mathbf{p}_i^d$. We use the following model to calculate the representation of $\mathbf{d}_r^e$:

$$\beta_r = \arg\min_{\mathbf{z}} \|\mathbf{z}\|_2,\ s.t.\ \mathbf{d}_r^e = \mathbf{P}^{s,e}\mathbf{z},\ r = 1, 2, \cdots, n \tag{11}$$

where $\mathbf{P}^{s,e} = \left[\mathbf{p}_1^{s,e}, \mathbf{p}_2^{s,e}, \ldots, \mathbf{p}_N^{s,e}\right]$.

The solution of Equation 11 is:

$$\beta_r = \mathbf{P}^{s,e+}\mathbf{d}_r^e = \left(\mathbf{P}^{s,eT}\mathbf{P}^{s,e} + \varsigma\mathbf{I}\right)^{-1}\mathbf{P}^{s,eT}\mathbf{d}_r^e,\ r = 1, 2, \cdots, n \tag{12}$$

where ς (set to 0.1) is a small positive parameter to avoid singularity, $\mathbf{I}$ is an identity matrix.

$\mathbf{D}^d$ is calculated with:

$$\mathbf{D}^d = \mathbf{P}^{s,d}\mathbf{B} \tag{13}$$

where $\mathbf{P}^{s,d} = \left[\mathbf{p}_1^{s,d}, \mathbf{p}_2^{s,d}, \ldots, \mathbf{p}_N^{s,d}\right]$, $\mathbf{B} = [\beta_1, \beta_2, \cdots, \beta_n]$.

Figure 4 shows the visual quality of before and after OCDT. As shown, the output of TV lacks texture details. The step of OCDT recovers the texture details.

2.3 Summary of the proposed algorithm

The dictionary training scheme of the TV-OCDT algorithm is summarized in Algorithm 1. Since it is difficult to estimate the noise level for a natural image in real applications, we trained the dictionaries with clear training images. The test images with different noise levels are all recovered by same dictionaries.

Algorithm 1 Dictionary training algorithm

```
Input:
    HR training images $\mathbf{Z}_i^s, i = 1, 2, \cdots, Q$, where $Q$ is the number of training images.
Output:
    Coupled dictionaries $\mathbf{D}^d$ and $\mathbf{D}^e$.
    Down sample and up sample the $\mathbf{L}_i^s$ according to the magnification factor $\gamma$ with BI. Denote the output as $\tilde{\mathbf{L}}_i^s$. The size of $\tilde{\mathbf{L}}_i^s$ is the same as $\mathbf{Z}_i^s$.
    Enhance $\tilde{\mathbf{L}}_i^s$ with the TV regularization model (4). The initial value of the iteration is $\tilde{\mathbf{L}}_i^s$. Denote the output as $\mathbf{Z}_i^{s,TV}$. $\mathbf{Z}_i^{s,TV}$ can be considered as a degraded version of $\mathbf{Z}_i^s$.
    Calculate the difference image $\mathbf{Z}_i^{s,e}$ by $\mathbf{Z}_i^{s,e} = \mathbf{Z}_i^s - \mathbf{Z}_i^{s,TV}$.
    Produce four gradient maps of $\mathbf{Z}_i^{s,TV}$ with four filters $(\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4)$ as the operation in Zeyde's algorithm [8].
    Divide these gradient maps into patches and connect the four gradient patches in each position into a $4p^2$ dimensional column vector to form the training data set $\{\mathbf{p}_i^{s,d}\}_{i=1}^N$.
    Divide $\mathbf{Z}_i^{s,e}$ into $p \times p$ dimensional patches. Then, change the patches into $p^2$ dimensional column vectors to form the training sets $\{\mathbf{p}_i^{s,e}\}_{i=1}^N$. (The smooth patch is dropped. We only use $N$ patches which contain textures. The variance of patches is used to distinguish the smooth and texture patches. If the variance of $\mathbf{p}_i^{s,e}$ is small, we drop the $\mathbf{p}_i^{s,e}$ and $\mathbf{p}_i^{s,d}$. The similar operation has been used in Yang's work [6].)
    Train the HR dictionary $\mathbf{D}^e$ by solving model (10) with K-SVD dictionary training algorithm [41,42].
    Calculate $\beta_r$ according to model (11).
    Obtain the $\mathbf{D}^d$ by Equation 13.
```

The reconstruction stage of the proposed algorithm is summarized in Algorithm 2.

3 Experiments

In this section, we will first introduce the experimental settings and compare our algorithm with five state-of-the-art algorithms. Then, we will discuss two influential factors for the sparse representation stage (including the patch size and the dictionary size) and two parameters in TV regularization stage (including two Lagrange parameters and iteration number). Finally, we will show the time complexity of the proposed algorithm.

3.1 Experimental settings

In our experiments, we magnify the input LR image both by the factors of 3 and 4. We use the training

Algorithm 2 Reconstruction algorithm

```
Input:
    LR testing images $\mathbf{L}^t$.
Output:
    Reconstructed HR image $\hat{\mathbf{Z}}^t$.
    Up sample the $\mathbf{L}^t$ to the desired size $K_1 \times K_2$ with BI. Denote the output as $\tilde{\mathbf{L}}^t$.
    Enhance $\tilde{\mathbf{L}}^t$ with the TV regularization model (4). The initial value of the iteration is $\tilde{\mathbf{L}}_i^t$. Denote the output as $\mathbf{Z}^{t,TV}$.
    Produce four gradient maps of $\mathbf{Z}_i^{t,TV}$ with four filters $(\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4)$ as the operation in Zeyde's algorithm [8]. Divide these gradient maps into patches and connect the four gradient patches in each position into a $4p^2$ dimensional column vector to form the testing data set $\{\mathbf{p}_i^{t,d}\}_{i=1}^M$. Let $\mathbf{P}^{t,d} = \left[\mathbf{p}_1^{t,d}, \mathbf{p}_2^{t,d}, \cdots, \mathbf{p}_M^{t,d}\right]$, where $M$ is the number of testing patches.
    Calculate the sparse representations for each patch by solving the following problem with orthogonal matching pursuit (OMP) algorithm [43]:
        $$\min_{\{\alpha_i^t\}} \sum_{i=1}^{M} \left\|\mathbf{p}_i^{t,d} - \mathbf{D}^d \alpha_i^t\right\|_2^2, s.t. \left\|\alpha_i^t\right\|_0 \le \hat{T} \qquad (14)$$
    Reconstruct the HR patches $\mathbf{p}_i^t$ by:
        $$\mathbf{p}_i^t = \mathbf{p}_i^{t,d} + \mathbf{D}^e \alpha_i^t \qquad (15)$$
    Synthesis the output $\hat{\mathbf{Z}}^t$ by putting $\{\mathbf{p}_i^t\}_{i=1}^M$ into their proper locations and averaging the overlap regions.
```

images in the software package about Yang's algorithm [6]. Figure 5 shows some training images. We collected 100,000 coupled patches as the external training database. Figure 6 shows some LR test images. The color training images are transformed into gray images. We only use the patches which contain the texture information and the smooth patches are discarded. $\mathbf{p}_i^{s,e}$ whose variance exceeds 12 is discarded. Its corresponding $\mathbf{p}_i^{s,d}$ is also discarded.

For the overlapped regions between the adjacent patches, averaging fusion is usually used to obtain the final pixel values. The median value has been considered to be more robust than average value, since it is uneasy to be affected by a handful of bad values. In our algorithm, a Gaussian weighted average is employed for more accurate results. The median values are used as the mean μ of the Gaussian distribution, the variance θ^2 is calculated according to μ and the overlap pixel values $(u_1, u_2, \cdots, u_v)$, where v is the number of overlap pixels. The final pixel value u^* is calculated by Equation 16. Compared with the traditional average values, the Gaussian weighted averages can reduce the effect

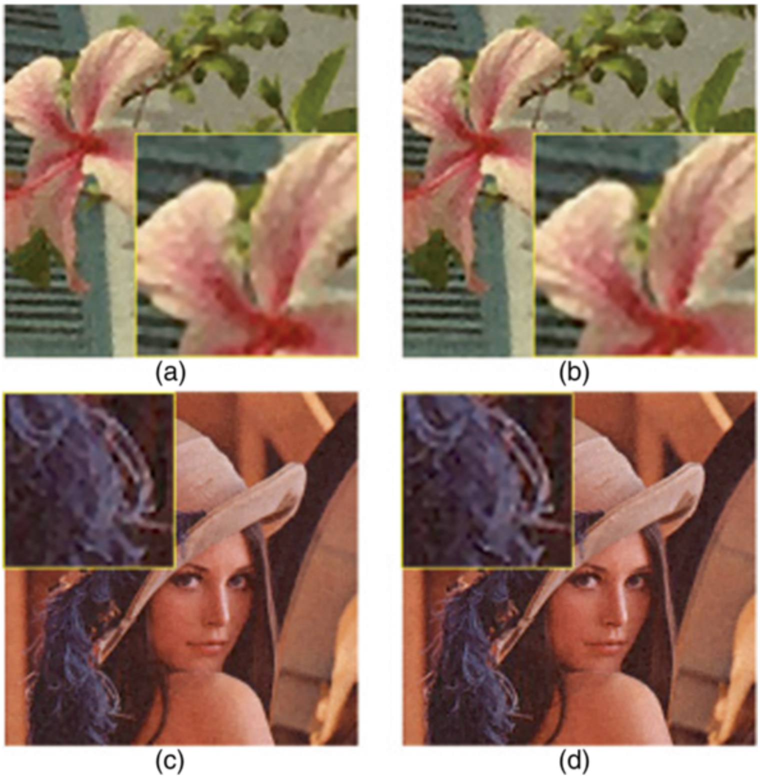

Figure 4 Visual quality of before and after OCDT. **(a)** TV output of flower image. **(b)** TV-OCDT output of flower image. **(c)** TV output of Lena image. **(d)** TV-OCDT output of Lena image.

of the very bad values which are far away from the desired values.

$$u^* = \frac{\sum_{i=1}^{\nu} \exp\left(-|u_i - \mu|^2/2\theta^2\right) u_i}{\sum_{i=1}^{\nu} \exp\left(-|u_i - \mu|^2/2\theta^2\right)} \tag{16}$$

$$\theta^2 = \frac{1}{\nu} \sum_{i=1}^{\nu} (u_i - \mu)^2 \tag{17}$$

3.2 Comparison with other methods

We compare our algorithm with five state-of-the-art algorithms, including: Zeyde's method [8], Anchored Neighborhood Regression (ANR) [28], ANR plus[29], Statistical Prediction Model (SPM) [25], and Deep Convolutional Network (DCN) [34]. The parameter settings about our method will be described in the following sections. In Tables 1 and 2, we compare the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [44] of the

Figure 5 Illustration of training images.

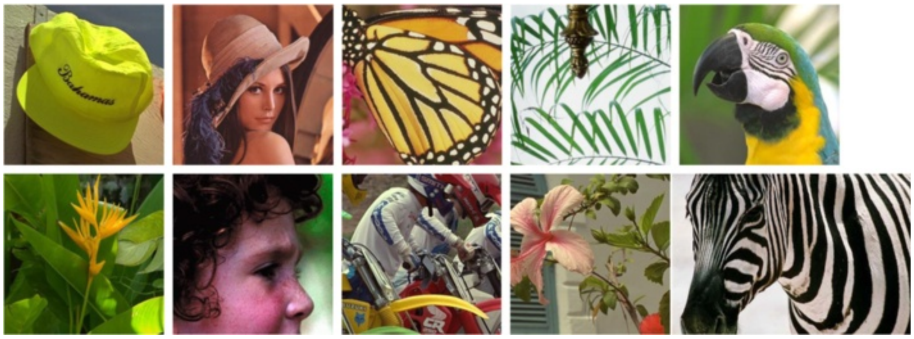

Figure 6 Gallery of test images. From left to right and from top to bottom, they are 'Hat', 'Lena', 'Butterfly', 'Leaves', 'Parrot', 'Plants', 'Head', 'Bike', 'Flower', and 'Zebra'.

reconstructed HR images referring to different noise standard deviation σ. PSNR and SSIM are all quantitative evaluations of the images. Figures 7 and 8 show the visual comparison. According to the comparison, we can see that though the proposed method obtains lower PSNR and SSIM values than others for images without noise, the reconstructed images have sharp edges and less artifacts than others. When the standard deviation of the noise is less than 10, the proposed algorithm obtains the highest PSNR and SSIM values in most of the cases. The average values are higher than others. From the visual comparison, we can see that the proposed algorithm can suppress the noise well when the noise is less than 10. When the variance of the noise is larger than 10, the PSNR and SSIM values of the proposed algorithm are lower than some of the existing algorithms. From the visual comparison, we can see that all of the algorithms can not obtain good visual quality. Therefore, how to magnify images with serious noise is a problem to be solved.

To test the robustness of the proposed algorithm, we test these algorithms on some natural images. Figure 9 shows the results. The test images are taken with a Coolpad 8750 mobile phone working on LR pattern and the flashlight is on. They are taken in a ground parking at 21:30 in the evening. From the results, we can see that the proposed algorithm provides sharper edges with less artifacts.

3.3 Effects of the parameters

3.3.1 Effects of the dictionary size and patch size

The dictionary and patch size greatly affect all the sparse representation-based image magnification methods. Since a larger overlap is good to denoising, we use the overlap of $p-1$ for each patch size. In TV regularization step, the iteration number is set to 200 and the

Table 1 The average PSNR and SSIM values for different methods (3×)

Method	Zeyde	ANR	ANR plus	SPM	DCN	Proposed
Without noise	28.762	27.545	29.376	29.080	29.255	29.238
	0.8539	0.8411	0.8535	0.8465	0.8453	0.8586
$\sigma=5$	27.044	26.851	27.181	27.089	27.191	27.673
	0.7273	0.7129	0.7189	0.7268	0.7223	0.7980
$\sigma=10$	24.460	24.101	24.020	23.714	23.996	24.836
	0.5662	0.5440	0.5381	0.5280	0.5376	0.6381
$\sigma=15$	22.129	21.698	21.420	20.912	21.281	22.079
	0.4436	0.4216	0.4110	0.3898	0.4047	0.4801
$\sigma=20$	20.238	19.762	19.403	18.811	19.190	19.838
	0.3588	0.3383	0.3260	0.3029	0.3174	0.3716

PSNR, peak signal-to-noise ratio; SSIM, structural similarity; ANR, Anchored Neighborhood Regression; SPM, Statistical Prediction Model; DCN, Deep Convolutional Network.

Table 2 The average PSNR and SSIM values for different methods (4×)

Method	Zeyde	ANR	ANR plus	SPM	DCN	Proposed
Without noise	26.812	26.680	27.172	26.951	27.059	27.129
	0.7841	0.7806	0.7860	0.7771	0.7780	0.7995
$\sigma=5$	25.494	25.335	25.623	25.535	25.766	26.055
	0.6796	0.6651	0.6774	0.6809	0.6916	0.7409
$\sigma=10$	23.476	23.138	23.115	22.863	23.337	23.762
	0.5445	0.5208	0.5194	0.5100	0.5409	0.6061
$\sigma=15$	21.485	21.055	20.858	20.428	20.964	21.413
	0.4350	0.4100	0.4032	0.3831	0.4171	0.4709
$\sigma=20$	19.810	19.323	19.045	18.503	19.005	19.428
	0.3539	0.3298	0.3200	0.2982	0.3283	0.3697

PSNR, peak signal-to-noise ratio; SSIM, structural similarity; ANR, Anchored Neighborhood Regression; SPM, Statistical Prediction Model; DCN, Deep Convolutional Network.

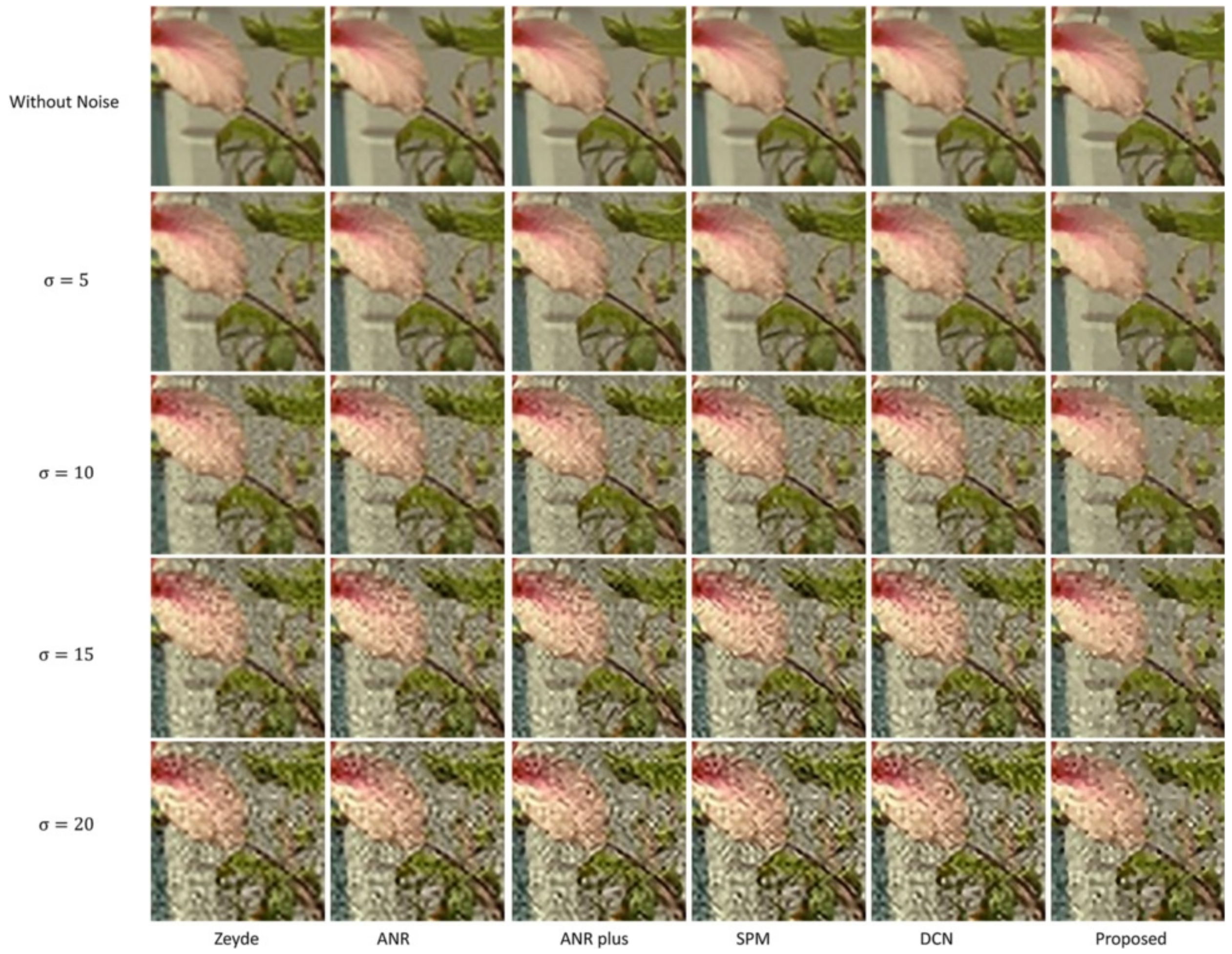

Figure 7 The visual comparison of different methods of the *flower* image (3×). To make the readers see them clearer, we only show a small part of every result image.

Lagrange multipliers are all set to 0.6 for the 3× magnification and λ_1 is set to 0.4 and λ_2 is set to 0.6 for the 4× magnification (we will discuss these parameters in the next section). Tables 3 and 4 show the average PSNR and SSIM values for different dictionary and patch sizes.

For the 3× magnification, all the best PSNR values are obtained when the dictionary size is 512. Most of the best SSIM values are also obtained when the dictionary size is 512. Therefore, we choose 512 as the dictionary size for the 3× magnification. When the noise standard deviation is 5, the best patch size is 3. When the noise standard deviation is 10 and 20, the best patch size is 5. When the noise standard deviation is 15, the best patch size is 7. Therefore, there are no best patch size which is suitable for every noise standard deviation. But the worst values appear when the patch size is 7. Furthermore, larger patch size will result in larger time complexity. Therefore, we choose 5 in our experiments since it is suitable for two standard deviation values.

For the 4× magnification, most of the best PSNR and SSIM values are obtained by the dictionary size 1,024. The patch size 7 gets the highest PSNR and SSIM values for most of the standard deviations. Therefore, we choose 1,024 as the best dictionary size and 7 as the best patch size.

3.3.2 Effects of the parameters in TV regularization

The iteration number $\hat{J}$ and two Lagrange multipliers are important parameters for TV regularization. Tables 5 and 6 show the change of the PSNR and SSIM values for different iteration numbers. It is obvious that 200 is the best iteration number for the 3× magnification. For the 4× magnification, the best iteration number is 200 when the noise standard deviation is 5 or 20. The best iteration number is 300 when the noise standard deviation is 10 and

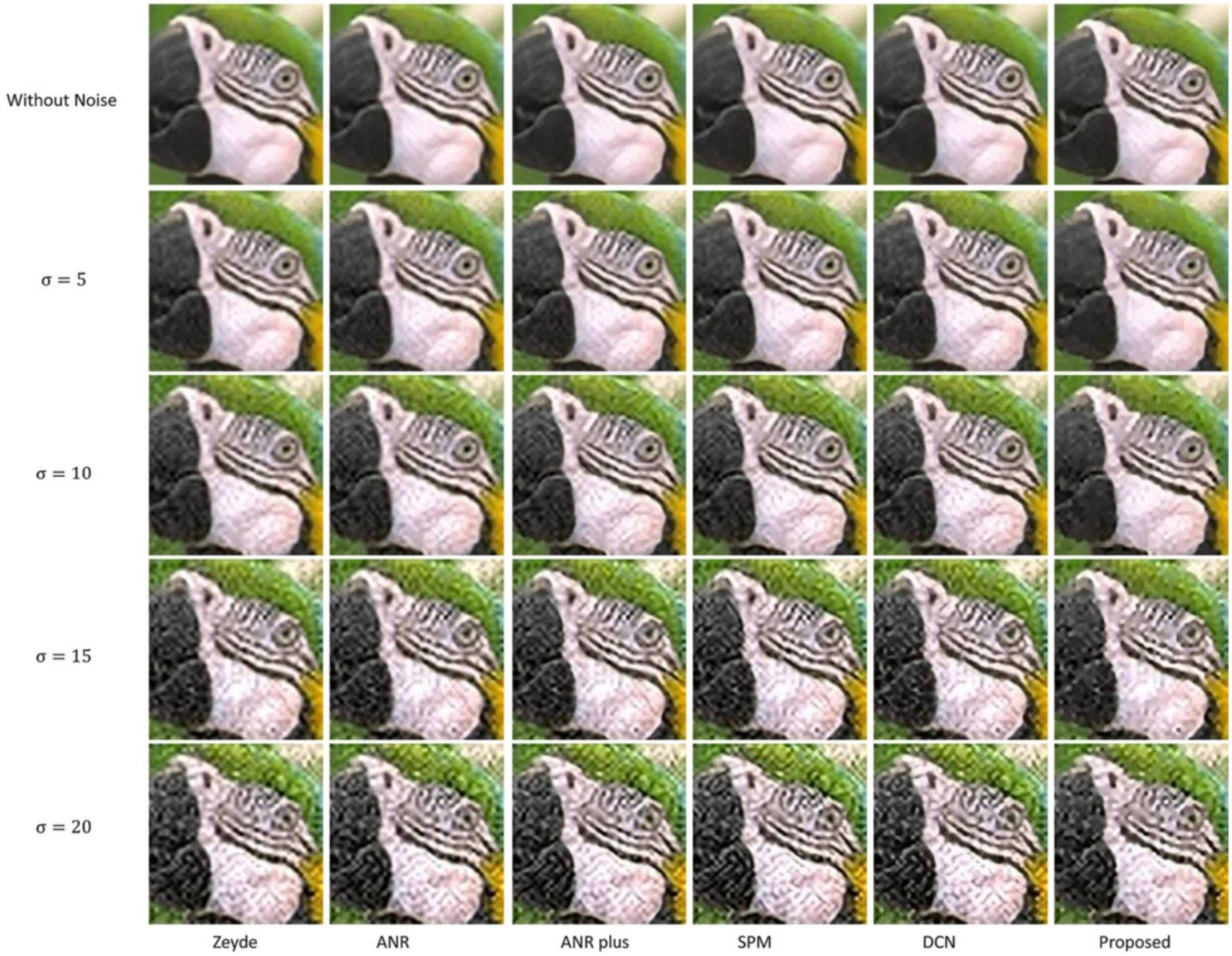

Figure 8 The visual comparison of different methods of the *parrot* image (3×). To make the readers see them clearer, we only show a small part of every result image.

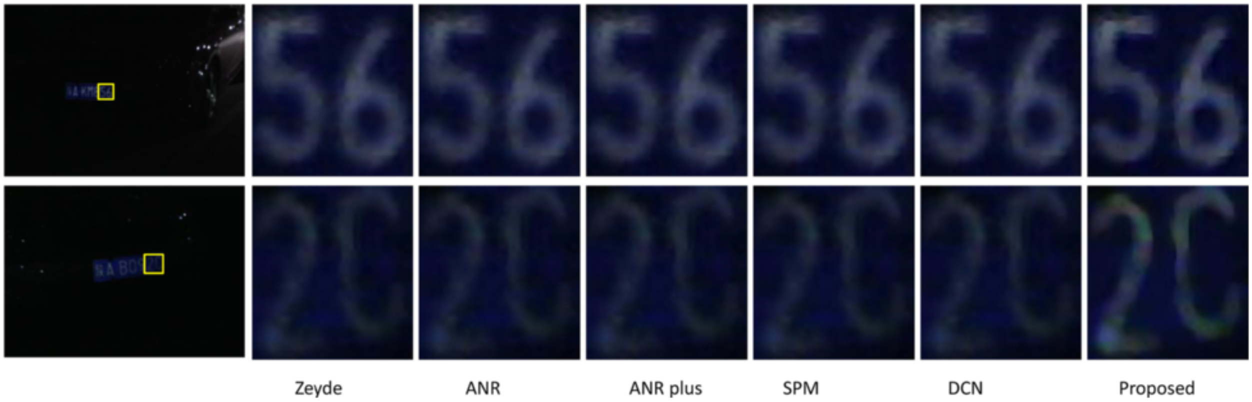

Figure 9 The visual comparison of different methods of the natural images (3×).

Table 3 The average PSNR and SSIM values for different patch and dictionary sizes (3×)

Patch size	Dictionary size					
	256	**512**	**1,024**	**256**	**512**	**1,024**
		$\sigma = 5$			$\sigma = 10$	
3	30.423	30.433	30.430	26.744	26.660	26.670
	0.8869	0.8857	0.8873	0.7198	0.7197	0.7179
5	30.422	29.883	30.437	26.660	27.567	26.572
	0.8858	0.8552	0.8879	0.7148	0.7570	0.7113
7	30.486	29.883	30.364	26.700	27.449	26.554
	0.8896	0.8576	0.8840	0.7173	0.7495	0.7076
		$\sigma = 15$			$\sigma = 20$	
3	22.928	22.823	22.856	19.379	19.356	19.325
	0.4474	0.4432	0.4449	0.3381	0.3346	0.3395
5	22.989	24.848	22.933	19.317	22.020	19.399
	0.4523	0.5323	0.4513	0.3405	0.4493	0.3444
7	22.853	24.881	22.821	19.357	22.013	19.237
	0.4390	0.5373	0.4417	0.3350	0.4502	0.3347

PSNR, peak signal-to-noise ratio; SSIM, structural similarity.

15. Since more iteration number will result in more time cost, we choose 200 in our experiments.

Tables 7 and 8 show the average PSNR and SSIM values for different Lagrange multipliers. As shown, the best Lagrange multipliers are different for different noise standard deviations and magnifications. According to the comparisons stated in Section 3.2, our method is superior to others when the noise standard deviation is less than 10. The advantage is more significant when the noise standard deviation is 5. Therefore, we choose the best Lagrange multipliers according to the data about noise standard deviation 5. For the 3× magnification, λ_1 and λ_2 are all set to 0.6. For the 4× magnification, λ_1 is set to 0.4 and λ_2 is set to 0.6.

Table 4 The average PSNR and SSIM values for different patch and dictionary sizes (4×)

Patch size	Dictionary size					
	256	**512**	**1,024**	**256**	**512**	**1,024**
		$\sigma = 5$			$\sigma = 10$	
3	28.803	28.730	28.831	25.688	25.623	25.710
	0.8333	0.8306	0.8354	0.6544	0.6575	0.6490
5	28.799	28.769	28.877	25.637	25.704	25.607
	0.8317	0.8289	0.8342	0.6498	0.6607	0.6505
7	28.850	28.382	28.417	25.639	26.372	26.314
	0.8328	0.8008	0.8031	0.6486	0.6780	0.6782
		$\sigma = 15$			$\sigma = 20$	
3	22.076	22.027	21.945	19.302	19.217	19.366
	0.3863	0.3892	0.3892	0.3472	0.3473	0.3455
5	22.123	21.885	22.151	19.458	19.173	19.353
	0.3971	0.3837	0.4024	0.3495	0.3413	0.3551
7	21.962	23.776	23.777	19.212	21.650	21.870
	0.3905	0.4734	0.4695	0.3397	0.4405	0.4628

PSNR, peak signal-to-noise ratio; SSIM, structural similarity.

Table 5 The average PSNR and SSIM values for different iteration numbers (3×)

Iteration numbers	σ			
	5	**10**	**15**	**20**
100	30.589	27.525	24.514	21.975
	0.8858	0.6994	0.5369	0.4179
200	30.654	27.574	24.510	22.112
	0.8881	0.7075	0.5428	0.4280
300	30.634	27.518	24.509	21.993
	0.8865	0.7061	0.5407	0.4204
400	30.623	27.514	24.437	22.002
	0.8859	0.7055	0.5335	0.4197

PSNR, peak signal-to-noise ratio; SSIM, structural similarity.

3.4 Time complex analysis

The time complexity is analyzed as follows. Suppose the time complexity of calculating an unknown pixel value with interpolation algorithm is c, the TV regularization requires order $O(cK_1K_2\hat{J})$ flops according to Equation 7. OMP algorithm [43,45] is used to calculate the sparse representations. OMP algorithm needs $O\left(4p^2n\hat{T}\right)$ flops when calculating sparse representation for a given feature vector [41]. Therefore, the whole time complexity of the testing phase is $O(4K_1K_2p^2n\hat{T} + cK_1K_2\hat{J})$. Our algorithm is tested on an AMD FX8150 CPU with 3.6 GHz and 16

Table 6 The average PSNR and SSIM values for different iteration numbers (4×)

Iteration numbers	σ			
	5	**10**	**15**	**20**
100	28.753	26.580	23.678	21.350
	0.8209	0.6646	0.5194	0.4075
200	28.791	26.588	23.704	21.621
	0.8225	0.6661	0.5217	0.4164
300	28.770	26.594	23.756	21.598
	0.8210	0.6725	0.5211	0.4166
400	28.788	26.609	23.807	21.533
	0.8212	0.6710	0.5213	0.4122

PSNR, peak signal-to-noise ratio; SSIM, structural similarity.

Table 7 The average PSNR and SSIM values for different Lagrange multipliers (3×)

λ_1	λ_2							
	0.2	**0.4**	**0.6**	**0.8**	**0.2**	**0.4**	**0.6**	**0.8**
	$\sigma = 5$				$\sigma = 10$			
0.2	30.391	30.329	30.368	30.061	27.615	27.660	27.763	27.480
	0.8763	0.8732	0.8714	0.8461	0.6927	0.6984	0.7048	0.6406
0.4	30.596	30.575	30.675	30.083	27.018	27.155	27.196	27.531
	0.8834	0.8833	0.8867	0.8463	0.6358	0.6513	0.6560	0.6449
0.6	30.623	30.653	30.677	30.077	26.770	26.809	26.842	27.583
	0.8783	0.8813	0.8826	0.8459	0.6162	0.6185	0.6218	0.6508
0.8	30.557	30.592	30.610	30.088	26.541	26.624	26.565	27.470
	0.8694	0.8746	0.8757	0.8485	0.5910	0.6017	0.6048	0.6422
	$\sigma = 15$				$\sigma = 20$			
0.2	23.833	24.115	24.271	24.459	20.997	21.136	21.327	22.030
	0.5460	0.5735	0.5851	0.5448	0.3961	0.4036	0.4117	0.4123
0.4	23.141	23.266	23.316	24.428	20.020	19.999	20.031	22.065
	0.4908	0.5003	0.5078	0.5411	0.3359	0.3317	0.3400	0.4158
0.6	22.785	22.795	22.931	24.415	19.602	19.754	19.723	22.157
	0.4673	0.4685	0.4835	0.5370	0.3075	0.3197	0.3132	0.4273
0.8	22.385	22.465	22.612	24.512	19.432	19.583	19.557	22.022
	0.4411	0.4515	0.4592	0.5526	0.3069	0.3108	0.3147	0.4108

PSNR, peak signal-to-noise ratio; SSIM, structural similarity.

Table 8 The average PSNR and SSIM values for different Lagrange multipliers (4×)

λ_1	λ_2							
	0.2	**0.4**	**0.6**	**0.8**	**0.2**	**0.4**	**0.6**	**0.8**
	$\sigma = 5$				$\sigma = 10$			
0.2	28.595	28.622	28.554	28.566	26.399	26.329	26.415	26.533
	0.8182	0.8215	0.8116	0.8073	0.6754	0.6745	0.6851	0.6889
0.4	28.795	28.748	28.824	28.719	25.908	25.972	26.048	26.056
	0.8284	0.8253	0.8306	0.8208	0.6334	0.6464	0.6500	0.6493
0.6	28.791	28.716	28.751	28.799	25.644	25.768	25.809	25.820
	0.8211	0.8219	0.8249	0.8216	0.6158	0.6158	0.6234	0.6282
0.8	28.759	28.760	28.694	28.807	25.529	25.631	25.539	25.651
	0.8168	0.8220	0.8188	0.8226	0.6040	0.6108	0.6038	0.6130
	$\sigma = 15$				$\sigma = 20$			
0.2	23.479	23.622	23.787	23.744	21.353	21.579	21.898	21.970
	0.5023	0.5123	0.5255	0.5292	0.3841	0.3954	0.4336	0.4300
0.4	22.777	22.824	22.984	23.110	20.667	20.689	20.985	20.886
	0.4488	0.4556	0.4658	0.4790	0.3484	0.3473	0.3627	0.3684
0.6	22.415	22.445	22.582	22.698	20.488	20.284	20.459	20.532
	0.4260	0.4260	0.4393	0.4407	0.3358	0.3267	0.3313	0.3399
0.8	22.227	22.241	22.529	22.399	20.271	20.269	20.210	20.357
	0.4150	0.4094	0.4297	0.4254	0.3174	0.3167	0.3146	0.3309

PSNR, peak signal-to-noise ratio; SSIM, structural similarity.

Table 9 Time cost comparison (Second)

Size	Zeyde	ANR	ANR plus	SPM	DCN	Proposed
256 × 256	1.53	0.88	0.89	31.89	6.57	4.17
512 × 512	6.24	4.10	4.42	143.73	40.48	24.21

ANR, Anchored Neighborhood Regression; SPM, Statistical Prediction Model; DCN, Deep Convolutional Network.

G memory PC at windows platform. Table 9 compares the time costs of different methods. Our time cost is larger than Zeyde's method, ANR, and ANR plus but smaller than SPM and DCN.

4 Conclusions

The capability of dealing with the noisy LR images is greatly related to the performance of an image magnification algorithm in real applications. In this paper, we propose an algorithm to magnify a noisy LR image. This algorithm combines the ideas of regularization and learning-based algorithm. The experimental results demonstrate that the proposed algorithm performs well when the standard deviation of noise is not very high. But some problems still need to be solved in the future. Firstly, the existing algorithms and the proposed algorithm all can not deal with LR images with serious noise. From Figures 7 and 8, we can see that when the noise is higher than 10, the visual quality is not ideal for all of these methods. Secondly, the performance on the natural images is not good enough. Many texture details still cannot be recovered. We should find better ways to deal with complex natural conditions in the future research.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

JX wrote Sections 1 and 2. ZC wrote Sections 3 and 4. JF revised the paper and added many analyses about the experimental results. XZ, XW, and YW did the experiments and collected the data. All authors read and approved the final manuscript.

Acknowledgements

This work was supported by the National Science Foundation of China (Grant no. 61340040, 61202183, 61102095), the Science and Technology Plan in Shannxi Province of China (No.2014KJXX-72), and Young Teachers Foundation of Xi'an University of Posts & Telecommunications.

Author details

[1]School of Telecommunication and Information Engineering, Xi'an University of Posts and Telecommunications, Weiguo Road, 710121 Xi'an, China. [2]Image Processing and Recognition Center, Xi'an Jiaotong University, Xianning road, 710049 Xi'an, China. [3]School of Information Engineering, Chang'an University, Erhuan Road, 710064 Xi'an, China.

Received: 7 August 2014 Accepted: 13 April 2015
Published online: 06 May 2015

References

1. X Liu, D Zhao, J Zhou, W Gao, H Sun, Image interpolation via graph-based bayesian label propagation. IEEE Trans. Image Process. **23**(3), 1084–1096 (2014)
2. W Dong, G Shi, X Li, L Zhang, X Wu, Image reconstruction with locally adaptive sparsity and nonlocal robust regularization. Signal Process. Image Commun. **27**(10), 1109–1122 (2012)
3. W Dong, L Zhang, R Lukac, G Shi, Sparse representation based image interpolation with nonlocal autoregressive modeling. IEEE Trans. Image Process. **22**(4), 1382–1394 (2013)
4. W Dong, L Zhang, G Shi, X Li, Nonlocally centralized sparse representation for image restoration. IEEE Trans. Image Process. **22**(4), 1620–1630 (2013)
5. K Zhang, X Gao, D Tao, X Li, Single image super-resolution with non-local means and steering kernel regression. IEEE Trans. Image Process. **21**(11), 4544–4556 (2012)
6. J Yang, J Wright, TS Huang, Y Ma, Image super-resolution via sparse representation. IEEE Trans. Image Process. **19**(11), 2861–2873 (2010)
7. H Hou, H Andrews, Cubic splines for image interpolation and digital filtering. IEEE Trans. Acous. Speech Signal Process. **26**(6), 508–517 (1978)
8. R Zeyde, M Protter, M Elad, On single image scale-up using sparse-representation. Lecture Notes Comput. Sci. **6920**(1), 711–730 (2010)
9. K Nguyen, S Sridharan, S Denman, C Fookes, in *IEEE Conference on Computer Vision and Pattern Recognition*. Feature-domain super-resolution framework for gabor-based face and iris recognition (IEEE Providence, RI, USA, 2012), pp. 2642–2649
10. Q Zhu, L Sun, C Cai, Non-local neighbor embedding for image super-resolution through foe features. Neurocomputing. **141**(2), 211–222 (2014)
11. X Gao, K Zhang, D Tao, X Li, Image super-resolution with sparse neighbor embedding. IEEE Trans. Image Process. **21**(7), 3194–3205 (2012)
12. J Li, W Gong, W Li, F Pan, Single-image super-resolution reconstruction based on global non-zero gradient penalty and non-local laplacian sparse coding. Digital Signal Process. **26**(1), 101–112 (2014)
13. A Akyol, M G?kmen, Super-resolution reconstruction of faces by enhanced global models of shape and texture. Pattern Recognit. **45**(12), 4103–4116 (2012)
14. JS Park, SW Lee, An example-based face hallucination method for single-frame, low-resolution facial images. IEEE Trans. Image Process. **17**(10), 1806–1816 (2008)
15. J Sun, J Zhu, MF Tappen, in *IEEE Conference on Computer Vision and Pattern Recognition*. Context-constrained hallucination for image super-resolution (IEEE San Francisco, CA, USA, 2010), pp. 231–238
16. H Chavez-Roman, V Ponomaryov, Super resolution image generation using wavelet domain interpolation with edge extraction via a sparse representation. IEEE Geoscience Remote Sensing Lett. **11**(10), 1777–1781 (2014)
17. M Nazzal, H Ozkaramanli, Wavelet domain dictionary learning-based single image superresolution. Signal Image Video Process. **1**, 1–11 (2014)
18. H Chang, DY Yeung, Y Xiong, in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* vol. 1. Super-resolution through neighbor embedding (IEEE Washington, DC, USA, 2004), pp. 275–282
19. H Huang, H He, Super-resolution method for face recognition using nonlinear mappings on coherent features. IEEE Trans. Neural Netw. **22**(1), 121–130 (2011)
20. L An, N Thakoor, B Bhanu, in *IEEE International Conference on Image Processing*. Vehicle logo super-resolution by canonical correlation analysis (IEEE Orlando, FL, USA, 2012), pp. 2229–2232
21. L An, B Bhanu, Face image super-resolution using 2d cca. Signal Process. **103**(1), 184–194 (2014)
22. T Ogawa, M Haseyama, Image inpainting based on sparse representations with a perceptual metric. EURASIP J. Adv. Signal Process. **2013**(1), 1–26 (2013)
23. J Yang, Z Wang, Z Lin, S Cohen, T Huang, Coupled dictionary training for image super-resolution. IEEE Trans. Image Process. **21**(8), 3467–3478 (2012)
24. J Xu, C Qi, Z Chang, in *IEEE International Conference on Image Processing*. Coupled k-svd dictionary training for super-resolution (IEEE Paris, France, 2014), pp. 3910–3914
25. T Peleg, M Elad, A statistical prediction model based on sparse representations for single image super-resolution. IEEE Trans. Image Process. **23**(6), 2569–2582 (2014)
26. H Li, Q Hairong, R Zaretzki, in *IEEE Conference on Computer Vision and Pattern Recognition*. Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution (Portland, Oregon, USA, 2013), pp. 345–352
27. S Wang, L Zhang, Y Liang, Q Pan, in *IEEE Conference on Computer Vision and Pattern Recognition*. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis (IEEE Providence, RI, USA, 2012), pp. 2216–2223
28. R Timofte, V De Smet, L Van Gool, in *IEEE International Conference on Computer Vision*. Anchored neighborhood regression for fast example-based super-resolution (IEEE Portland, Oregon, USA, 2013), pp. 1920–1927
29. R Timofte, V De Smet, L Van Gool, in *Asian Conference of Computer Vision*. A+: Adjusted anchored neighborhood regression for fast super-resolution (Singapore City, Singapore, 2014), pp. 1–15
30. J Shi, X Liu, C Qi, Global consistency, local sparsity and pixel correlation: A unified framework for face hallucination. Pattern Recognit. **47**(11), 3520–3534 (2014)
31. F-G Carlos, EJ Candes, in *IEEE International Conference on Computer Vision*. Super-resolution via transform-invariant group-sparse regularization (IEEE Sydney, Australia, 2013), pp. 3336–3343
32. KS Ni, TQ Nguyen, Image superresolution using support vector regression. IEEE Trans. Image Process. **16**(6), 1596–1610 (2007)
33. KI Kim, Y Kwon, Single-image super-resolution using sparse regression and natural image prior. IEEE Trans. Pattern Anal. Mach. Intell. **32**(6), 1127–1133 (2010)
34. C Dong, CC Loy, K He, X Tang, in *Proceedings of European Conference on Computer Vision*. Learning a deep convolutional network for image super-resolution (Zurich, Switzerland, 2014), pp. 1–16
35. P Purkait, NR Pal, B Chanda, A fuzzy-rule-based approach for single frame super resolution. IEEE Trans. Image Process. **23**(5), 2277–2290 (2014)
36. A Buades, B Coll, J-M Morel, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2*. A non-local algorithm for image denoising (IEEE San Diego, CA, USA, 2005), pp. 60–65
37. RC Gonzalez, RE Woods, *Digital Image Processing*. (Pearson Education, India, 2009)
38. M Irani, S Peleg, Improving resolution by image registration. CVGIP: Graphical Models Image Process. **53**(3), 231–239 (1991)
39. A Marquina, SJ Osher, Image super-resolution by tv-regularization and bregman iteration. J. Scientific Comput. **37**(3), 367–382 (2008)
40. LI Rudin, S Osher, E Fatemi, Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena. **60**(1), 259–268 (1992)
41. M Aharon, M Elad, A Bruckstein, K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. **54**(11), 4311–4322 (2006)
42. LN Smith, M Elad, Improving dictionary learning: Multiple dictionary updates and coefficient reuse. IEEE Signal Process. Lett. **20**(1), 79–82 (2013)
43. R Rubinstein, M Zibulevsky, M Elad, Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit, CS Technion (2008). http://www.cs.technion.ac.il/~ronrubin/Publications/KSVD-OMP-v2.pdf
44. Z Wang, AC Bovik, HR Sheikh, EP Simoncelli, Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
45. JA Tropp, AC Gilbert, Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inf. Theory. **53**(12), 4655–4666 (2007)