PLOS ONE

RESEARCH ARTICLE

Dictionary learning based noisy image super-resolution via distance penalty weight model

Yulan Han*, Yongping Zhao*, Qisong Wang

Department of Automatic Test and Control, Harbin Institute of Technology, Harbin, Heilongjiang, China

* hanyulanbox@126.com (YH); zhaoyp2590@126.com (YZ)

Abstract

In this study, we address the problem of noisy image super-resolution. Noisy low resolution (LR) image is always obtained in applications, while most of the existing algorithms assume that the LR image is noise-free. As to this situation, we present an algorithm for noisy image super-resolution which can achieve simultaneously image super-resolution and denoising. And in the training stage of our method, LR example images are noise-free. For different input LR images, even if the noise variance varies, the dictionary pair does not need to be retrained. For the input LR image patch, the corresponding high resolution (HR) image patch is reconstructed through weighted average of similar HR example patches. To reduce computational cost, we use the atoms of learned sparse dictionary as the examples instead of original example patches. We proposed a distance penalty model for calculating the weight, which can complete a second selection on similar atoms at the same time. Moreover, LR example patches removed mean pixel value are also used to learn dictionary rather than just their gradient features. Based on this, we can reconstruct initial estimated HR image and denoised LR image. Combined with iterative back projection, the two reconstructed images are applied to obtain final estimated HR image. We validate our algorithm on natural images and compared with the previously reported algorithms. Experimental results show that our proposed method performs better noise robustness.

Citation: Han Y, Zhao Y, Wang Q (2017) Dictionary learning based noisy image super-resolution via distance penalty weight model. PLoS ONE 12(7): e0182165. https://doi.org/10.1371/journal.pone.0182165

Editor: Zhao Zhang, Soochow University, CHINA

Received: November 10, 2016

Accepted: July 13, 2017

Published: July 31, 2017

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

1 Introduction

Single image super-resolution (SR) is a classical problem in computer vision. In general, it uses signal processing techniques to recover a high resolution (HR) image from only one low resolution (LR) image. SR methods can be broadly classified into three categories: interpolation-based methods, reconstruction-based methods, and example-based methods.

Interpolation-based SR such as [1, 2] has been proposed for in various applications and it demonstrates the advantage of fast computational simplicity. But they usually fail to generate fine details in discontinuous regions and often result in introducing blurring of edges and other high-frequency features in practice [3].

Reconstruction-based methods usually integrate one or more sophisticated priors such as gradient profile prior [4], edge prior [5], and total variation [6] into SR literature to estimate the missed details. Recently, sparse-based regularization [7–10] has also been shown to be

particularly effective for the ill-posed problems of SR. Usually, these methods achieved impressive results in preserving sharper edges and suppressing aliasing artifacts. However, the performance depends heavily upon a rational prior imposed on the up-sampled image [11].

Over the years, many example-based SR methods [12–14] have been proposed with demonstrated promising results and become the mainstream approaches of SR domain. The methods assume that the missing high frequency details can be estimated based on learning the mapping relationship from LR-HR patch pairs of external database and input LR patches. Two kinds of relationship models exist for these methods. One is that between LR patches and the corresponding HR patches in the database. After Freeman et al. [15] used Markov network to model the relationship, regression functions [16] are employed to exploit the relationship between HR and LR patch pairs. In addition, supervised or semi-supervised learning models are introduced into some of the algorithms [17–19]. Recently, a mapping of LR-HR image pairs was learned using a deep convolutional neural network [20], and has shown favorable results. D. Dai et al. [21] jointly learned a collection of regressors from LR to HR patches, which collectively yielded the smallest error for all training data. The other is that between LR example patches and input LR patches. Most of the methods [22, 23] is based on Nearest Neighbor Embedding (NNE). In these methods, a fixed number of nearest neighbors are extracted from database for each input LR patch, and then the corresponding HR patches are used to estimate the output HR patch by a linear combination determined by LR patch and its neighbors. Despite the algorithms are demonstrated by successful results, they highly depend on the number of neighbors which is difficult to determine. For this problem, [24] operates on a dynamic k-nearest neighbor algorithm, where k is small for test point with highly relevant neighbors and large others. Some researchers calculate the distance between input patch and its neighbors respectively. The neighbors will be abandoned when the distance is smaller than mean value. Yang [25] exploited sparse coding to perform image SR. The algorithm assumes that LR-HR patch pairs share the same sparse coefficients with respect to their respective dictionaries which are jointly learned from a set of external training images. It can be considered as neighbor embedding in sparse domain without choosing the number of neighbors. Since then, sparse coding is applied to SR problem [21–23], and achieves impressive results. Zeyde [26] used dimensionality reduction and orthogonal matching pursuit for sparse representation to improve efficiency. S. Wang [27], proposed a semi-coupled dictionary learning model, under which a pair of dictionaries and a mapping function describing the relationship between sparse coefficients of LR-HR patch pairs will be simultaneously learned. In [28], kernel ridge regression is employed to connect sparse coefficients of LR-HR patch pairs. Kaibing Zhang [29] determine the relationship between LR image patches and HR image patches by assuming that LR image patches and HR image patches are share the same sparse coefficients. R. Timofte et al. [30] proposed a fast image SR method called anchored neighbourhood regression (ANR) which learns sparse dictionaries and regressors anchored to dictionary atoms. This algorithm is faster, while making no compromise on quality. R. Timofte et al. [31] then produced an improved variant of ANR. The study in [31] enhanced these features and anchored regressors for ANR. Instead of learning the regressors on the dictionary, their method uses the full training material. It obtained improved quality, and became the fastest method indisputably. S. Gu [32] proposed a convolutional sparse coding based SR method to address consistency issue. In addition, researches show that image structures tend to repeat themselves within and across scales. [33–35] exploits the self-similarity of structures in nature image and extracts the database directly from the LR input image instead of the external database. Good reconstruction quality relies on much additional memory and running time to build counterparts across different scales in a recursive scheme. Therefore, its application is limited.

Although the algorithms can results in better performance, most of the SR algorithms including other learning-based methods assume that the input LR image is noise-free. Such assumption is not in accord with real applications. The algorithms are less robustness to noisy image SR. So another challenge is the super-resolution for noisy images. While compared with SR on clear LR input images, less attention has been paid to develop effective SR algorithms for noisy ones. J. Xie [36] first employs an adaptively regularized Shock filter to tackle the jagged noise, and then perform SR for depth image. The disadvantage of such scheme is that the artifacts can be created in denoising process and magnified in super-resolution process. Therefore, researchers started on simultaneously denoising and super-resolution. In [37], LR training images are magnified by a TV regularization model with a constraint before dictionaries training stage. However, the level of noise dealt with the method is small. Furthermore, it focuses on magnification only. Based on the current research status, we devote to design an algorithm to complete SR and denoising in the same framework to deal with noisy image patches.

Sparse representation makes the signal energy only concentrated in a few atoms. Because of the special nature, some sparse coding based SR algorithms such as [25] show certain robustness to noisy image. In addition, sparse representation has been successfully employed in image denoising [38, 39], image restoration [40, 41] and other processing [42, 43]. The dictionary plays an important role in the sparse representation process. A predefined analytical dictionary (e.g., wavelet dictionary, Gabor dictionary) make the coding fast and explicit, but it is less effective to model the complex local structures of natural images. A synthesis dictionary (e.g., K-SVD dictionary) can be learned from example natural images and has more expensive computation but can better model complex image local structures [44]. In recent years, lots of dictionary learning methods have been proposed and achieved obvious performance. Feng et al. [45] propose to learn jointly the projection matrix for dimensionality reduction and the discriminative dictionary for face representation. Zhang et al. [46] propose a semisupervised label consistent dictionary learning framework for machine fault classification. Inspired by these, we introduce sparse theory to our research. The synthesis procedure is illustrated in Fig 1. The input LR image and example images are firstly cropped into patches. The example images are noise-free. Then the features of example patch pairs are extracted, which will be learned for dictionary pair. For each input LR patch, according to its features, it is easy to achieve simultaneously similar dictionary atom pairs $(\mathbf{u}_i^h, \mathbf{u}_i^l)$ finding and calculating distance $\mathbf{b}_i$ between input LR patch and its similar atoms. Next, combined with the input LR image patch feature, LR dictionary atom $\mathbf{u}_i^l$ and distance $\mathbf{b}_i$ are used to compute weight $\boldsymbol{\omega}_i$. After the weight is computed, we can obtain estimated HR image patch and denoised LR image patch from $\mathbf{u}_i^h \omega_i$. Put all the estimated HR patches into an estimated HR image, which is computed by averaging in overlapping regions. In the same way, we obtain the denoised LR image from all the denoised LR patches. At last, combined with the iterative back projection (IBP), the estimated HR image and the denoised LR image are applied to obtain the final output HR image.

The contributions can be summarized as follows.

(1) Different from the conventional methods, the proposed algorithm can process noisy image, and present for simultaneously image superresolution and denoising. Furthermore, in the training stage of our method, LR example images are noise-free. For different input LR images, even if the noise variance varies, the dictionary pair does not need to be retrained.

(2) The core idea of our proposed method is that the estimated HR patch is weighted average of similar HR example patches. To reduce computational cost for finding similar patches from millions of examples, example patches are replaced by the learned sparse dictionary which makes the signal energy only concentrate in few atoms.

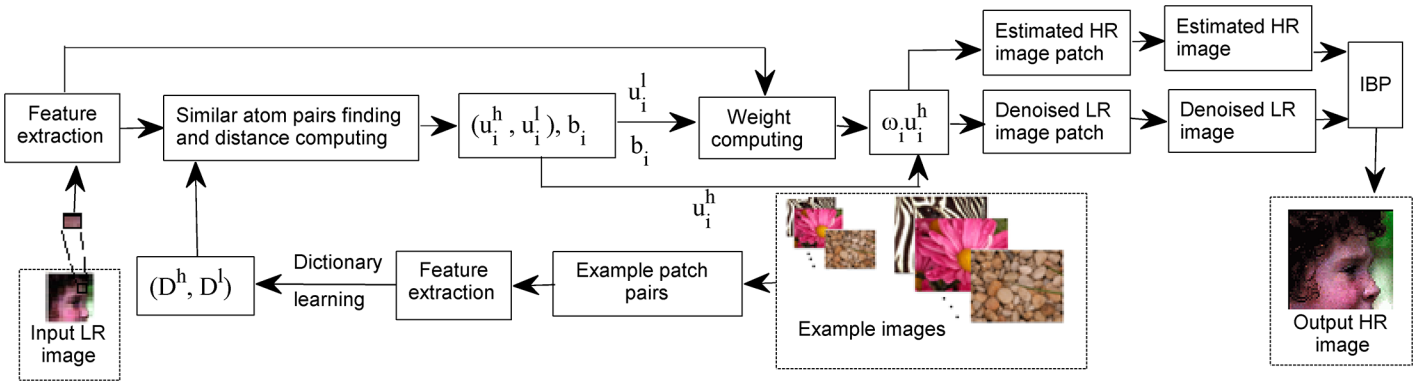

Fig 1. The flowchart of the proposed SR algorithm.

https://doi.org/10.1371/journal.pone.0182165.g001

(3) Penalty function is applied to least squares regression regularized by l_2-norm for modeling weight. It makes the objective function treat each similar atom unequally. The function is determined by the similarity between input LR patch and its similar atom of LR dictionary. When the similarity is strong, we make the penalty small, which forces large weight at the same time. Conversely, when the similarity is weak, we make the penalty large, which forces small or zero weight at the same time.

(4) LR example patches subtracted mean pixel value are used for training dictionary rather than just their gradient features like other literatures such as [25]. In the training stage, for each LR example patch, we first subtract its mean pixel value, then connect it to its corresponding HR example patch into a single vector. All the new vectors are used as new HR examples to learn HR dictionary. Thus, the HR dictionary represents textures of HR example patches, but also that of LR example patches which are noise-free. Therefore, in the reconstruction stage, the HR dictionary can also be used to recover denoised input LR patches. This is different from conventional learning methods. Combined with iterative back projection (IBP), the denoised LR patches are applied to enhance robustness to noise.

The remainder of this paper is organized as follows. The proposed algorithm is presented in detail in Section 2. Experimental results and comparisons are demonstrated in section 3. Section 4 concludes this paper.

2 The proposed method

Firstly, let us recall the image degradation model which is shown in Eq (1). Given an observed LR image $\mathbf{Y} \in R^M$ that is a degraded version of a HR image $\mathbf{X} \in R^N$ of the same scene

$$\mathbf{Y} = \mathbf{G}_s\mathbf{H}\mathbf{X} + v \tag{1}$$

Where, $\mathbf{G}_s$ is the down-sampling operator with scaling factor s; $\mathbf{H}$ is the blurring operator; v is the noise. It is the task of SR reconstruction to recover $\mathbf{X}$ from $\mathbf{Y}$ as accurate as possible. It is considered that the image is noise-free by conventional SR methods.

2.1 Example database

From the example images $\{\mathbf{I}_1^h, \mathbf{I}_2^h, ..., \mathbf{I}_N^h\}$, LR images $\{\mathbf{I}_1^l, \mathbf{I}_2^l, ..., \mathbf{I}_N^l\}$ are first obtained, which are considered as noise-free ones. For each image $\mathbf{I}_j^h$, its corresponding LR image $\mathbf{I}_j^l$ is determined

by

$$\mathbf{I}_j^l = \mathbf{G}_s \mathbf{H} \mathbf{I}_j^h \tag{2}$$

A set $\{\mathbf{p}_1^h, \mathbf{p}_2^h, ..., \mathbf{p}_n^h\}$ of vectorized HR patches of size $\sqrt{w} \times \sqrt{w}$ are taken from example HR images $\{\mathbf{I}_1^h, \mathbf{I}_2^h, ..., \mathbf{I}_N^h\}$ and a set $\{\mathbf{p}_1^l, \mathbf{p}_2^l, ..., \mathbf{p}_n^l\}$ of vectorized LR patches of size $\sqrt{w}/s \times \sqrt{w}/s$ are taken from example LR images $\{\mathbf{I}_1^l, \mathbf{I}_2^l, ..., \mathbf{I}_N^l\}$. Consequently, we obtain a database of HR-LR patch pairs

$$(\mathbf{p}^h, \mathbf{p}^l) = \{(\mathbf{p}_i^h, \mathbf{p}_i^l) \in R^w \times R^{w/s^2}, i = 1, 2, ..., n\} \tag{3}$$

2.2 Distance penalty weight model

For the super-resolution, given a LR image $\mathbf{Y}^L$, which is generated form HR image $\mathbf{X}^H$ by Eq (1), the task is to recover the unknown $\mathbf{X}^H$ from $\mathbf{Y}^L$ with the help of example patch pairs. The algorithm is performed with patch for the unit. Similar to [25], $\mathbf{Y}^L$ is firstly divided into over-lapping patches

$$\mathbf{Y}^L = \{\mathbf{y}_i^l, i = 1, 2, ..., N_y\} \tag{4}$$

Where, $\mathbf{y}_i^l$ is the vectorized LR image patch of size $\sqrt{w}/s \times \sqrt{w}/s$, N_y is the number of patches of $\mathbf{Y}^L$.

The estimated vectorized HR image $\mathbf{X}^H$ can be represented as

$$\mathbf{X}^H = \{\mathbf{x}_i^h, i = 1, 2, ..., N_y\} \tag{5}$$

Where,$\mathbf{x}_i^h$ is the estimated HR image patch of size $\sqrt{w} \times \sqrt{w}$.

According to Eq (1), the relationship can be described by

$$\mathbf{y}_i^l = \mathbf{G}_s \mathbf{H} \mathbf{x}_i^h + \mathbf{v}_i \tag{6}$$

Where,$\mathbf{v}_i$ is the noise. We assume that it is Gaussian noise with zero-mean and variance σ^2.

Thus, it become the purpose of super-resolution to estimate HR image patch $\mathbf{x}_i^h$ from input LR image patch $\mathbf{y}_i^l$.

As we known, for each $\mathbf{x}_i^h$, it can be approximated by HR example patches through weighted average, which have similar structures. Therefore, based on this core idea, the problem in this method is to find the similar patches of $\mathbf{x}_i^h$ in database and to calculate the weight.

Due to the repetition of local structures of images, a subset of patches $(\mathbf{u}_i^h, \mathbf{u}_i^l) \in (\mathbf{p}^h, \mathbf{p}^l)$ in which $\mathbf{u}_i^h$ has similar structures with $\mathbf{x}_i^h$ exists. That is

$$\mathbf{x}_i^h = \sum_{j=1}^{k} \mathbf{u}_{ij}^h \omega_{ij} = \mathbf{u}_i^h \omega_i \tag{7}$$

Where, weight vector is $\boldsymbol{\omega}_i = [\omega_{i1}, \omega_{i2}, \ldots, \omega_{ij}, \ldots, \omega_{ik}]^T$, k is the number of the patch pairs in this subset $(\mathbf{u}_i^h, \mathbf{u}_i^l)$.

There are many methods to determine the weight, such as set the weights to be inversely proportional to the distance between patches. These methods relying on number of similar patches heavily, and cannot suppress noise. Now, we discuss a new weight model in details.

According to the degradation model Eqs (1) and (7), we have

$$\mathbf{G}_s\mathbf{H}\mathbf{x}_i^h = \mathbf{G}_s\mathbf{H}\mathbf{u}_i^h\omega_i = \mathbf{u}_i^l\omega_i \tag{8}$$

From Eq (8), we can obtain

$$\mathbf{y}_i^l = \mathbf{G}_s\mathbf{H}\mathbf{x}_i^h + \mathbf{v}_i = \mathbf{u}_i^l\omega_i + \mathbf{v}_i \tag{9}$$

Where, $\mathbf{v}_i$ is assumed as Gaussian noise with zero-mean and variance σ^2.

Thus,

$$\mathbf{y}_i^l - \mathbf{u}_i^l\omega_i = \mathbf{v}_i \tag{10}$$

$$\|\mathbf{y}_i^l - \mathbf{u}_i^l\omega_i\|_2^2 \leq \boldsymbol{\varepsilon}_i \tag{11}$$

Where, $\boldsymbol{\varepsilon}_i$ is related to σ^2. We can see that the LR patch $\mathbf{y}_i^l$ can be represented by the same weight vector $\boldsymbol{\omega}_i$ over $\mathbf{u}_i^l$, with an error $\boldsymbol{\varepsilon}_i$. That is to say, we can get the weight from input LR image patch and similar LR example patches with a controlled error.

Based on the above discussions, We formulate the weight solution as a least squares regression regularized by l2-norm:

$$\boldsymbol{\omega}_i = argmin\|\boldsymbol{\omega}_i\|_2^2 \quad s.t.\|y_i^l - \mathbf{u}_i^l\boldsymbol{\omega}_i\|_2^2 \leq \boldsymbol{\varepsilon}_i, \sum_{j=1}^{k}\omega_{ij} = 1 \tag{12}$$

From Eq (12), the objective function treats the patches $\mathbf{u}_i^l$ equally. It is not flexible to obtain accurate weights for the input patch. Motivated by this, we introduce distance penalty to the least square problem

$$\boldsymbol{\omega}_i = argmin\|\mathbf{b}_i \cdot \boldsymbol{\omega}_i\|_2^2 \quad s.t.\|y_i^l - \mathbf{u}_i^l\boldsymbol{\omega}_i\|_2^2 \leq \boldsymbol{\varepsilon}_i, \sum_{j=1}^{k}\omega_{ij} = 1 \tag{13}$$

Where, $\cdot$ denotes a point wise vector product, $\mathbf{b}_i = [b_{i1}, b_{i2}, \ldots, b_{ij}, \ldots, b_{ik}]^T$. $\mathbf{b}_i$ is the distance between $\mathbf{y}_i^l$ and each similar example patch in $\mathbf{u}_i^l$. When the similarity between u_{ij}^l and $\mathbf{y}_i^l$ is strong, we make the b_{ij} small, which forces large ω_{ij} at the same time. Conversely, when the similarity is weak, we make b_{ij} large, which forces small or zero ω_{ij} at the same time. It is simply determined by the squared Euclidean distance.

Eq (13) can be written as

$$\boldsymbol{\omega}_i = argmin\|\mathbf{y}_i^l - \mathbf{u}_i^l\boldsymbol{\omega}_i\|_2^2 + \lambda\|\mathbf{b}_i \cdot \boldsymbol{\omega}_i\|_2^2 \quad s.t.\sum_{j=1}^{k}\omega_{ij} = 1 \tag{14}$$

Where, λ is a regularization parameter.

According to Eq (10), we have

$$\|\mathbf{y}_i^l - \mathbf{u}_i^l\boldsymbol{\omega}_i\|_2^2 \approx \|\mathbf{v}_i\|_2^2 \approx \gamma\sigma^2 \tag{15}$$

Where, γ is a positive constant. So we set $\lambda = \gamma\sigma^2$, when $\sigma \neq 0$.

Thus, the main task in reconstruction stage is to find the patches $\mathbf{u}_i^l$ from $\mathbf{p}^l$, which is similar to $\mathbf{y}_i^l$ and compute the weight. Squared Euclidean distance can be adopted in to quantify the similarity. The corresponding $\mathbf{u}_i^h$ is assumed to have similar structures with $\mathbf{x}_i^h$. But it is uneasy to find similar patches for each input patch from millions of example patch pairs. It will take lots of time for the repetitive computation. Sparse dictionary make the signal energy only

concentrate in few atoms, and some sparse coding based SR algorithm [25] show certain robustness to noisy image, so that we use a learned sparse dictionary instead of examples. We find similar patch pairs $(\mathbf{u}_i^h, \mathbf{u}_i^l)$ from dictionary atom pairs, meaning $(\mathbf{u}_i^h, \mathbf{u}_i^l) \in (\mathbf{D}^h, \mathbf{D}^l)$.

Two dictionaries $\mathbf{D}^h$ and $\mathbf{D}^l$ are trained to have the same sparse coding for each HR and LR patch pair. Similar to Yang [25] and Chang [22], we subtract the mean pixel value for each HR example patch, so that the dictionary $\mathbf{D}^h$ represents image textures rather than absolute intensities. In the reconstruction stage, the mean value for each estimated patch is then predicted by its LR version. Also we employ first- and second-order derivatives as the feature extraction for LR example patches to train. Thus, D^l represents the gradient feature of images rather than absolute intensities. The four filters used here are:

$$\mathbf{f}_1 = [-1, 0, 1], \mathbf{f}_2 = \mathbf{f}_1^T, \mathbf{f}_3 = [-1, 0, 2, 0, 1], \mathbf{f}_4 = \mathbf{f}_3^T \tag{16}$$

In addition, to enhance robustness to noise, we also subtract mean pixel value for each LR example patch, and connect the LR example patch to its corresponding HR example patch into a single vector, which is also used to learn $\mathbf{D}^h$. Thus, dictionary $\mathbf{D}^h$ represents textures of HR example patches, but also that of LR example patches which are noise-free. In the reconstruction stage, the $\mathbf{D}^h$ can also be used to recover denoised input LR patches. This is different from conventional learning methods.

From above, the training set is obtained by

$$(\mathbf{P}^H, \mathbf{P}^L) = \left\{ (\mathbf{P}_i^H, \mathbf{P}_i^L) = \left(\begin{bmatrix} \mathbf{p}_i^h - \bar{p}_i^h \\ \mathbf{p}_i^l - \bar{p}_i^l \end{bmatrix}, F(\mathbf{p}_i^l) \right), i = 1, 2, ..., n \right\} \tag{17}$$

Where, $(\mathbf{p}^h, \mathbf{p}^l)$ is original HR-LR patch pairs in Eq(3), $\bar{p}_i^h$ is the mean value of $\mathbf{p}_i^h$, $\bar{p}_i^l$ is the mean value of $\mathbf{p}_i^l$, $F(\cdot)$ is the operator to get four gradient vectors by Eq (16) and connect the four vectors into a single vector.

The set $(\mathbf{P}^H, \mathbf{P}^L)$ is used to jointly train the dictionaries as

$$(\mathbf{D}^h, \mathbf{D}^l) = \min_{\mathbf{D}^h, \mathbf{D}^l, \alpha} \left\{ \frac{1}{N} \|\mathbf{P}^H - \mathbf{D}^h \alpha\|_2^2 + \frac{1}{M} \|\mathbf{P}^L - \mathbf{D}^l \alpha\|_2^2 + \lambda_0 \|\alpha\|_1 \right\} \tag{18}$$

Where, N and M are the vector dimensions of $\mathbf{P}^H$ and $\mathbf{P}^L$, respectively.

To solve the problem easily, Eq (18) can be rewritten as

$$\tilde{\mathbf{D}} = \min_{\tilde{\mathbf{D}}, \alpha} \{ \|\tilde{\mathbf{P}} - \tilde{\mathbf{D}} \alpha\|_2^2 + \lambda_0 \|\alpha\|_1 \} \tag{19}$$

Where, $\tilde{\mathbf{D}} = \begin{bmatrix} \frac{1}{\sqrt{N}} \mathbf{D}^h \\ \frac{1}{\sqrt{M}} \mathbf{D}^l \end{bmatrix}, \tilde{\mathbf{P}} = \begin{bmatrix} \frac{1}{\sqrt{N}} \mathbf{P}^H \\ \frac{1}{\sqrt{M}} \mathbf{P}^L \end{bmatrix}$.

The minimization of Eq (19) is a typical patch-based sparse problem. Many methods can be used to solve it. Yang [25] proposed the framework and acquired good results. However, it takes a large amount of time to solve this sparse model. Zeyde [26] improve the execution speed by dimensionality reduction on the patches through PCA and Orthogonal Matching Pursuit for the Sparse coding. For sparse dictionaries learning, we use the approach of Zeyde [26].

Gradient features(see Eq (16)) of LR example patches are used to learn LR dictionary. D^l represents the image gradient feature and $u_i^l \in D^l$. Therefore, the weight model is rewritten by

$$\hat{\boldsymbol{\omega}}_i = argmin\|F(\mathbf{y}_i^l) - \mathbf{u}_i^l\hat{\boldsymbol{\omega}}_i\|_2^2 + \lambda\|\mathbf{b}_i \cdot \hat{\boldsymbol{\omega}}_i\|_2^2 \quad s.t. \sum_{j=1}^{k}\hat{\omega}_{ij} = 1 \tag{20}$$

Where, $\hat{\boldsymbol{\omega}}_i$ is the weight.

This problem Eq (20) is l_2-norm constraint. We solve it for $\hat{\boldsymbol{\omega}}_i$ by taking $\frac{\partial L}{\partial \hat{\boldsymbol{\omega}}_i} = 0$. The closed-form solution is

$$\hat{\boldsymbol{\omega}}_i = ((\mathbf{u}_i^l)^T\mathbf{u}_i^l + \lambda\mathbf{B}_i)^{-1}(\mathbf{u}_i^l)^T F(\mathbf{y}_i^l) \tag{21}$$

Where, $L = \|F(\mathbf{y}_i^l) - \mathbf{u}_i^l\hat{\boldsymbol{\omega}}_i\|_2^2 + \lambda\|\mathbf{b}_i \cdot \hat{\boldsymbol{\omega}}_i\|_2^2$, $\mathbf{B}_i$ is a $k \times k$ diagonal matrix,

$$\mathbf{B}_i(j,j) = b_{ij}(j = 1, 2, ...k) \tag{22}$$

The final optimal weight is obtained by rescaling it so that $\sum_{j=1}^{k}\hat{\omega}_{ij} = 1$.

2.3 Reconstruction

Based on the above discussions, for each input $\mathbf{y}_i^l$, we start by extracting its gradient features and finding k similar atom pairs $(\mathbf{u}_i^h, \mathbf{u}_i^l)$. Because the dictionary atoms are learned basis vectors, we find the similar atoms based on the correlation between the LR dictionary atoms and input LR patch rather than the Euclidean distance. Now, we describe how to compute the correlation.

$F(\mathbf{y}_i^l)$ can be represented by dictionary $\mathbf{D}^l = [\mathbf{d}_1^l, \mathbf{d}_2^l, ..., \mathbf{d}_j^l, ..., \mathbf{d}_{nd}^l]$ ($\mathbf{d}_j^l$ is the LR dictionary atom, nd is dictionary size)

$$F(\mathbf{y}_i^l) = \mathbf{D}^l\beta = \beta_1\mathbf{d}_1^l + \beta_2\mathbf{d}_2^l + ... + \beta_j\mathbf{d}_j^l + ... + \beta_{nd}\mathbf{d}_{nd}^l \tag{23}$$

Where, $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_j, \ldots, \beta_{nd}]$, β_j is the correlation between $\mathbf{d}_j^l$ and $F(\mathbf{y}_i^l)$.

Eq (23) shows that every dictionary atom makes its own contribution to representing the input patch. The contribution of the j_{th} atom $\mathbf{d}_j^l$ can be evaluated by β_j. In other words, β_j is a measurement of the similarity between the input patch and the j_{th} dictionary atom. We consider that the larger the β_j, the larger scale of similarity between input patch $F(\mathbf{y}_i^l)$ and dictionary atom $\mathbf{d}_j^l$; and a small β_j means that there is little similarity. We can solve $\boldsymbol{\beta}$ by

$$\beta = (\mathbf{D}^l)^T F(\mathbf{y}_i^l) \tag{24}$$

Thus, $(\mathbf{D}^l)^T F(\mathbf{y}_i^l)$ could return the correlation. In Eq (20), we use distance $\mathbf{b}_i$ as the penalty. When the similarity between $F(\mathbf{y}_i^l)$ and $\mathbf{d}_j^l$ is strong, we make the b_{ij} small, which forces large $\hat{\omega}_{ij}$ at the same time. Conversely, when the similarity is weak, we make b_{ij} large, which forces small or zero $\hat{\omega}_{ij}$ at the same time. Therefore, we use the reciprocal of β_j to compute the penalty. The atom pairs corresponding to the maximal k correlation coefficients constitute $(\mathbf{u}_i^h, \mathbf{u}_i^l)$. $\mathbf{b}_i$ in Eq (20) is determined by

$$\mathbf{b}_i = 1./Sort(abs((\mathbf{D}^l)^T F(\mathbf{y}_i^l)), k) \tag{25}$$

Where, $Sort(\mathbf{a}, num)$ is a function returning num top biggest values of vector $\mathbf{a}$, $abs(.)$ is absolute value operation. The scheme can achieve simultaneously similar atoms finding and distance computing. If $\sigma = 0$, after finding similar atoms, we set $\mathbf{b}_i = \mathbf{1}$.

After this, we can easily obtain the weight $\hat{\boldsymbol{\omega}}_i$ by Eq (20) and $\mathbf{u}_i^h\hat{\boldsymbol{\omega}}_i$. According to section 2.2, the reconstructed vector $\mathbf{u}_i^h\hat{\boldsymbol{\omega}}_i$ represents the estimated HR patch and the denoised LR patch correspondent to $\mathbf{y}_i^l$. And the estimated patch and the denoised patch are subtracted mean pixel value. Based on this, we have

$$\begin{bmatrix} \hat{\mathbf{x}}_i^h \\ \hat{\mathbf{y}}_i^l \end{bmatrix} = \mathbf{u}_i^h\hat{\boldsymbol{\omega}}_i + \begin{bmatrix} E(\hat{\mathbf{x}}_i^h)\mathbf{C}_1 \\ E(\hat{\mathbf{y}}_i^l)\mathbf{C}_2 \end{bmatrix} \tag{26}$$

Where, $\hat{\mathbf{x}}_i^h$ is the estimation of $\mathbf{x}_i^h$, $\hat{\mathbf{y}}_i^l$ is the denoised patch of $\mathbf{y}_i^l$, $\mathbf{C}_1 \in R^{w_1}$ is an all-one column vector, $\mathbf{C}_2 \in R^{w_2}$ is an all-one column vector, w_1 is the size of $\hat{\mathbf{x}}_i^h$, w_2 is the size of $\hat{\mathbf{y}}_i^l$, E(·) is the mean evaluation operator.

Noise here is assumed as zero-means, so

$$E(\mathbf{y}_i^l) = E(\mathbf{D}_s\mathbf{H}\mathbf{x}_i^h + \mathbf{v}_i) = E(\mathbf{D}_s\mathbf{H}\mathbf{x}_i^h) + E(\mathbf{v}_i) \approx E(\mathbf{D}_s\mathbf{H}\mathbf{x}_i^h) \tag{27}$$

We can see that the noise has little effect on image mean. The mean of $\hat{\mathbf{y}}_i^l$ and $\hat{\mathbf{x}}_i^h$ could be estimated by the mean of $\mathbf{y}_i^l$. Eq (26) can be written by

$$\begin{bmatrix} \hat{\mathbf{x}}_i^h \\ \hat{\mathbf{y}}_i^l \end{bmatrix} = \mathbf{u}_i^h\hat{\boldsymbol{\omega}}_i + \begin{bmatrix} E(\mathbf{y}_i^l)\mathbf{C}_1 \\ E(\mathbf{y}_i^l)\mathbf{C}_2 \end{bmatrix} \tag{28}$$

Put all estimated patches $\hat{\mathbf{x}}_i^h$ into a HR image $\hat{\mathbf{X}}^H$, which is computed by averaging in overlapping regions. In the same way, we obtain a denoised image $\hat{\mathbf{Y}}^L$ from $\hat{\mathbf{y}}_i^l$. In order to strengthen the reconstruction constraint Eq (1), we compute the final estimated HR image $\mathbf{X}^*$ by

$$\mathbf{X}^* = \|\mathbf{X}^* - \hat{\mathbf{X}}^H\|_2^2 \quad s.t. \mathbf{D}_s\mathbf{H}\mathbf{X}^* = \hat{\mathbf{Y}}^L \tag{29}$$

The iterative back-projection (IBP) method [32] is used to solve this optimization problem

$$\mathbf{X}_{t+1}^* = \mathbf{X}_t^* + ((\hat{\mathbf{Y}}^L - \mathbf{D}_s\mathbf{H}\mathbf{X}_t^*)\uparrow_s) * p \tag{30}$$

Where, $\mathbf{X}_t^*$ is the estimate of the HR image at the t_{th} iteration, $\uparrow_s$ denote up-scaling by factor s, p is a symmetric Gaussian filter.

The entire SR process is summarized as Algorithm 1.

Algorithm 1: The Proposed SR Algorithm

Input: the sparse dictionaries $\mathbf{D}^h$ and $\mathbf{D}^l$; input LR image $\mathbf{Y}$; number of similar atoms k; a positive constant γ;
output: HR image $\mathbf{X}^*$;
1: **for** each patch $\mathbf{y}_i^l$ of $\mathbf{Y}$ **do**
2: Extract the gradient features for $\mathbf{y}_i^l$ by Eq (16).
3: Find k similar atom pairs $(\mathbf{u}_i^h, \mathbf{u}_i^l)$ and compute $\mathbf{b}_\perp$ by Eq (25).
4: Solve Eq (21) for $\hat{\boldsymbol{\omega}}_i$.
5: Generate estimated HR patch $\hat{\mathbf{x}}_i^h$ and denoised patch $\hat{\mathbf{y}}_i^l$ by Eq (28).
6: **end for**
7: Put the patches $\hat{\mathbf{x}}_i^h$, $i = 1, 2, ..., N_y$ and $\hat{\mathbf{y}}_i^l$, $i = 1, 2, ..., N_y$ into an image $\hat{\mathbf{X}}^H$ and $\hat{\mathbf{Y}}^L$, respectively.
8: Perform IBP Eq (30) to obtain a HR image $\mathbf{X}^*$.

3 Experiments

In this section, we will show the robustness of the proposed algorithm to noise and compare the state-of-the-art methods [20, 22, 25, 26, 31, 32]. In the training stage, we used 77 standard natural images as training set. For testing, we used Set5 [20, 31], Set14 [20, 31] and B100 [20, 31] to evaluate the performance of upscaling factors ×2, ×3 and ×4, respectively. Set5 and Set14 contain 5 and respectively 14 images for super-resolution evaluation. B100 contains 100 testing images of Berkeley Segmentation Dataset called BSDS300.

All LR images (training or test images) are generated from the original HR images. Firstly, the original HR images are directly blurred and down-sampled. The MATLAB function "imresize" is used here to complete the process. The function "imresize" involved a smooth filtering before down-sampling. Similar to [7], the noise is generated by MATLAB function "randn", and σ times noise is added to the blurred and down-sampled test images. It should be noted that LR example images for training dictionary are noise-free. For color images used in experiments, SR algorithms are performed only on luminance channel, because humans are more sensitive to illuminant changes. Therefore, we first changes channels into YCbCr ones and then apply our method to the Y channel. We interpolate the color layers (Cb, Cr) using bicubic interpolation.

3.1 Parameters

In this section, we analyze the main parameters of our algorithm. The standard settings we use are Set5 [20, 31] database, dictionary size 1024, $\gamma = 0.08$ and $k = 24$ for upscaling factor ×2, $k = 8$ for upscaling factor ×3, ×4. Peak signal-to-noise ratio (PSNR) and reconstruction time were used as the objective criteria.

3.1.1 Regularization parameter. γ is a key regularization parameter of our method. Here, we validate the effectiveness of using different γ, and choose an appropriate one. The results of Set5 are shown in Fig 2. Experimental setting is dictionary size 1024 and $k = 24$ for upscaling factor ×2, $k = 8$ for upscaling factor ×3, ×4. We can see that the curves are not monotonic, and PSNR peaks at $\gamma = 0.08$. For different datasets, the optimal γ is slightly different (0.06 of Set14 and B100 compared to 0.08 of Set5) for reconstruction quality. The results of Set14 and B100 are shown in S1–S6 Figs. Therefore, we suggest determining γ to be around 0.08 in practice. Here, in all of our following experiments, we set γ as 0.08 for convenience.

3.1.2 Dictionary size. In this experiments, dictionary size is varied from 32 up to 2048, while the training samples are extracted from the same training images previously mentioned. In Fig 3, we present the results that show the relation between our method's performance and the dictionary size when $\gamma = 0.08$ and $k = 24$ for upscaling factor ×2, $k = 8$ for upscaling factor ×3, ×4. Actually, noise has little effect on reconstruction time. So we only show the reconstruction time when $\sigma = 10$. We can see that the larger we learn the dictionary, the better reconstruction quality becomes. However, this comes with a higher computational cost. The result is the same as that of [25, 47]. Other datasets Set14 and B100 can also achieve similar results. The results of Set14 and B100 are shown in S7–S12 Figs. In practice, we suggest choosing the appropriate dictionary size as a tradeoff between reconstruction quality and computation. Dictionary size here is 1024 in our following experiments.

3.1.3 Number of similar atoms. The proposed method finds the similar atom pairs for each input patch. The performance of the method depends on the number of similar atoms k. The effect of k is shown in Fig 4 when dictionary size is 1024 and $\gamma = 0.08$. Here, we also only

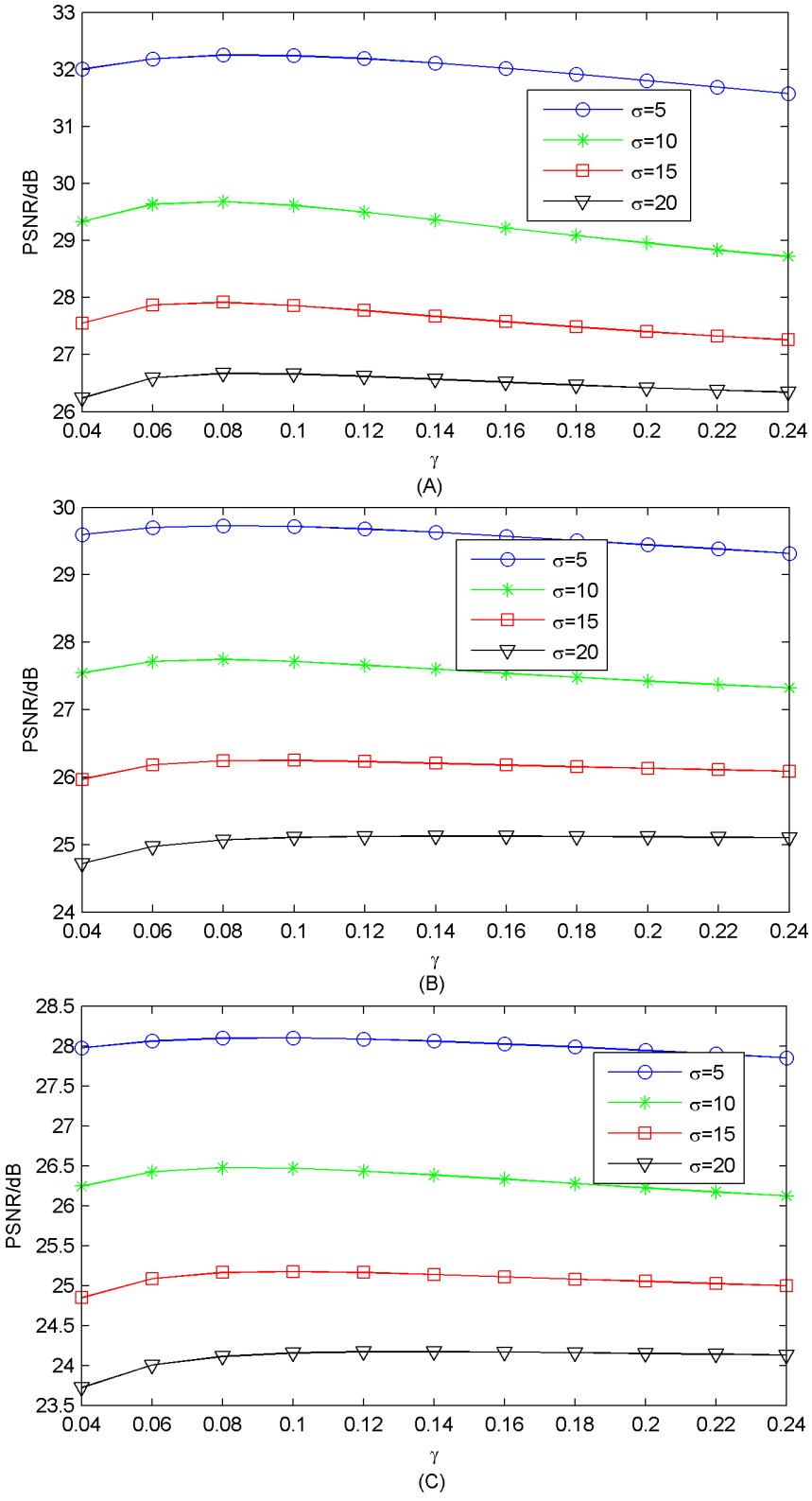

Fig 2. *γ* versus average PSNR on Set5. (A) upscaling factor ×2; (B) upscaling factor ×3; (C) upscaling factor ×4.

https://doi.org/10.1371/journal.pone.0182165.g002

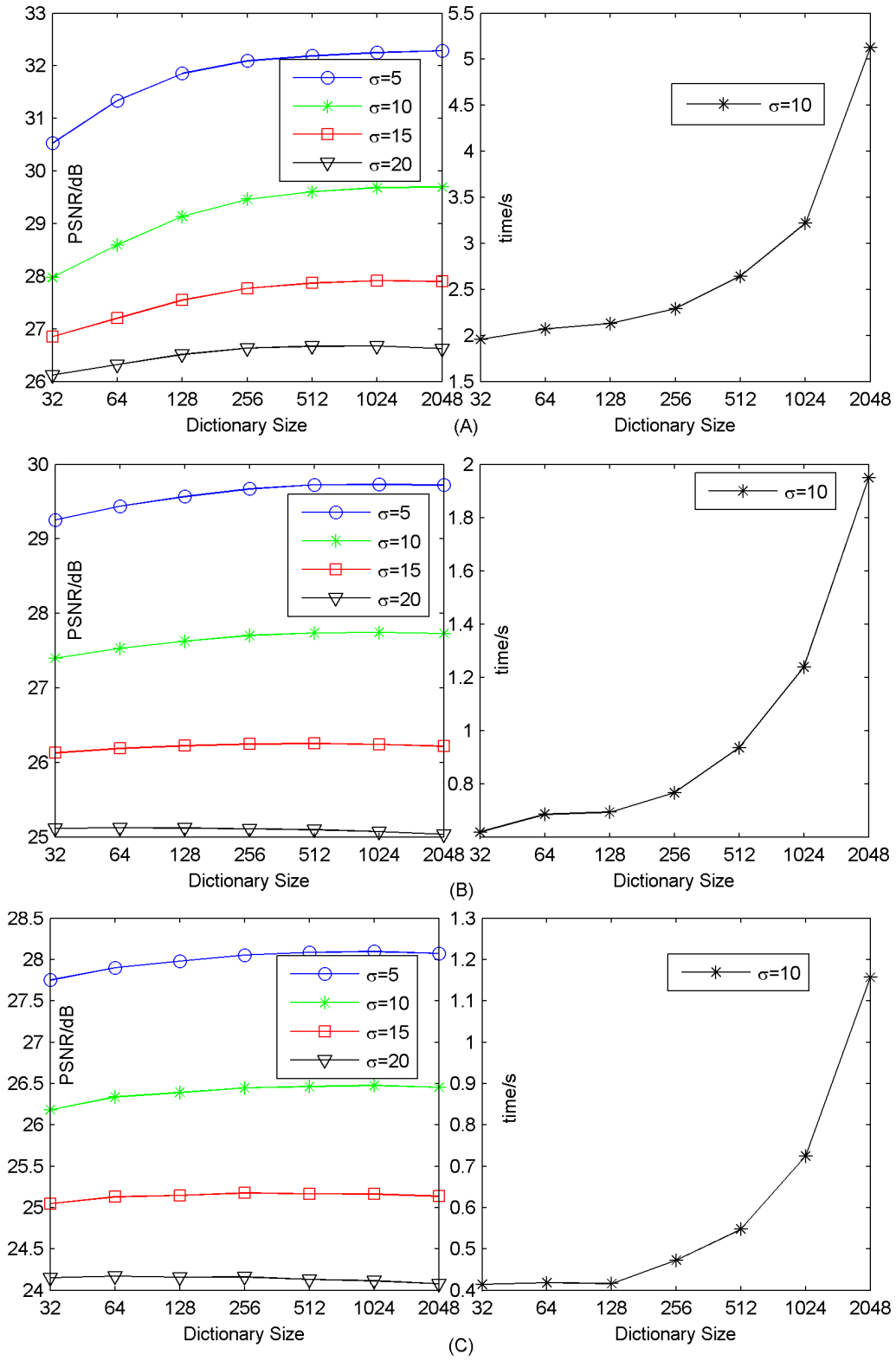

Fig 3. Dictionary size influence on performance on average on Set5. (A) upscaling factor ×2; (B) upscaling factor ×3; (C) upscaling factor ×4.

https://doi.org/10.1371/journal.pone.0182165.g003

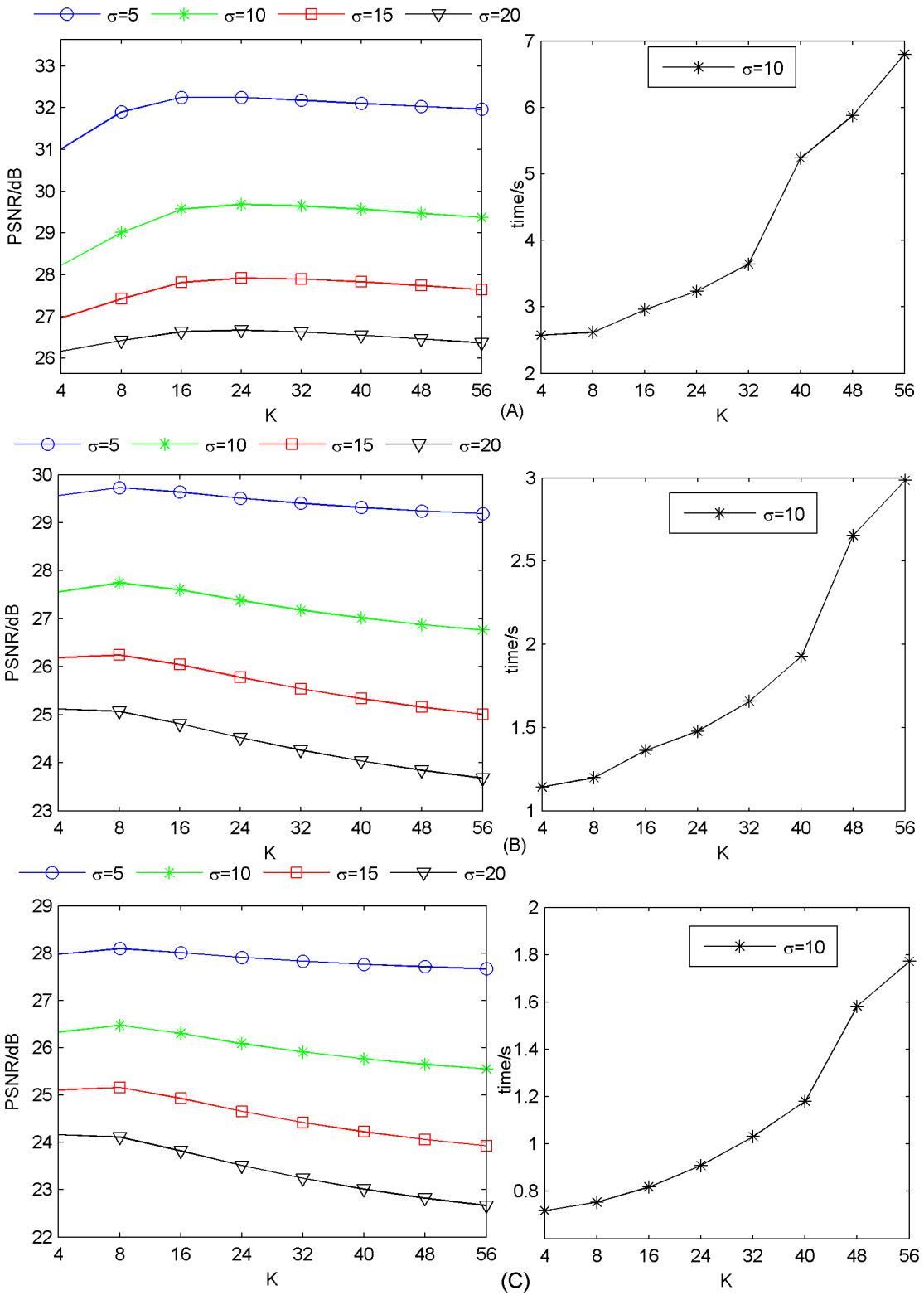

Fig 4. Number of similar atoms influence on performance on average on Set5. (A) upscaling factor ×2; (B) upscaling factor ×3; (C) upscaling factor ×4.

https://doi.org/10.1371/journal.pone.0182165.g004

show the reconstruction time when $\sigma = 10$. We can see that $k = 24$ is best for reconstruction quality when upscaling factor is ×2. The PSNR peaks at $k = 8$ when upscaling factor is ×3 or ×4. Moreover, average reconstruction time increases distinctly as k increases. It is due to the fact that by having a larger k, the computation of matrix inversion in Eq (21) increases. Other datasets Set14 and B100 can also achieve similar results. The results of Set14 and B100 are shown in S13–S18 Figs. Therefore, in resource-limited systems, a reasonable selection of k depends on the tradeoff between reconstruction quality and computational time. We will use $k = 24$ when upscaling factor is ×2, $k = 8$ when upscaling factor is ×3 or ×4 in our further experiments.

3.1.4 Patch size and overlap. Intuitively, using a too large or too small patch size tends to produce a smooth or unwanted artifact as noticed also in [25, 29] and a larger overlapping leads to a better SR results [25]. Therefore, patch size is set as 6×6, 6×6 and 8×8 for upscaling factor ×2, ×3 and ×4, respectively, and overlap is set as 4, 3 and 4 for upscaling factor ×2, ×3 and ×4, respectively.

3.2 Performance evaluation

In this section we analyze the performance of our algorithm in quantitative and qualitative comparison with the state-of-the-art methods including NE [22], SCSR [25], Zeyde [26], A+ [31], SRCNN [20], and CSC [32]. We also show the reconstruction times of the algorithms. The code of the compared method was downloaded from the authors' homepage. Peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) were used as the objective criteria. The parameters are analyzed in the previous section. Besides the patch size and overlap(see section 3.1.4), the other parameter are unified ($\gamma = 0.08$, dictionary size = 1024, $k = 24$ for upscaling factor ×2, $k = 8$ for upscaling factor ×3 and ×4).

3.2.1 Quality. Tables 1–3 list the PSNR and SSIM comparisons. When $\sigma = 0$, the approach CSC [32] achieves the best performance. But it is not in accord with real application. When $\sigma \neq 0$, as repeatedly shown, the results demonstrate the superiority of our proposed algorithm over other approaches on Set5, Set14 and B100. The average PSNR of the recent method CSC [32] is 0.24 dB (Set14, upscaling factor ×4, $\sigma = 5$) and 7.4 dB (Set5, upscaling factor ×2, $\sigma = 20$) behind our method. Compared with CSC, for dataset B100, the average PSNR improvement is

Table 1. Comparisons of average PSNR (dB) and SSIM ($\sigma = 0$).

dataset	Scale	NE [22]		SCSR [25]		Zedye [26]		A+ [31]		SRCNN [20]		CSC [32]		ours	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Set5	×2	35.77	0.949	36.04	0.951	35.78	0.949	36.55	0.954	36.34	0.952	**36.62**	**0.955**	35.65	0.948
	×3	31.84	0.896	31.40	0.887	31.90	9.897	32.59	0.909	32.39	0.887	**32.66**	**0.909**	31.57	0.895
	×4	29.61	0.840	-	-	29.69	0.843	30.28	0.860	30.09	0.853	**30.36**	**0.859**	29.49	0.841
Set14	×2	31.76	0.899	31.71	0.903	31.81	0.899	32.28	0.906	32.18	0.904	**32.31**	**0.907**	31.71	0.901
	×3	28.60	0.808	28.07	0.803	28.67	0.808	29.13	0.819	29.00	0.815	**29.15**	**0.821**	28.26	0.811
	×4	26.81	0.733	-	-	26.88	0.734	**27.32**	0.749	26.61	0.725	27.30	**0.750**	26.55	0.738
B100	×2	30.41	0.871	31.04	0.884	30.40	0.868	30.77	0.877	31.14	0.885	**31.27**	**0.888**	30.76	0.881
	×3	27.85	0.771	27.81	0.772	27.87	0.770	28.18	0.780	28.21	0.780	**28.31**	**0.786**	27.85	0.778
	×4	26.47	0.697	-	-	26.55	0.697	26.77	0.709	26.71	0.702	**26.83**	**0.711**	26.51	0.703

https://doi.org/10.1371/journal.pone.0182165.t001

Table 2. The results of PSNR (dB) and SSIM on the set5 dataset.

Scale	σ	Set5 images	NE [22]		SCSR [25]		Zedye [26]		A+ [31]		SRCNN [20]		CSC [32]		ours	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
×2	5	baby	31.5	0.7556	32.96	0.8287	31.9	0.7753	31.11	0.7399	33.47	0.8482	30.63	0.7190	**34.19**	**0.8760**
		bird	31.84	0.8063	32.21	0.8683	32.25	0.8226	31.48	0.7918	33.34	0.8836	30.89	0.7704	**34.46**	**0.9134**
		butterfly	28.35	0.8372	**29.05**	0.8819	28.70	0.8492	28.97	0.8385	26.87	0.8588	28.43	0.8222	28.08	**0.8936**
		head	30.71	0.7022	32.04	0.763	31.08	0.7193	30.42	0.6898	32.34	0.774	29.99	0.6693	**32.87**	**0.7915**
		woman	30.39	0.7867	31.33	0.8465	30.68	0.803	30.3	1.7762	30.6	0.8575	29.82	0.7575	**31.65**	**0.8913**
		average	30.56	0.7776	31.52	0.8377	30.92	0.7939	30.46	0.9672	31.32	0.8444	29.95	0.7477	**32.25**	**0.8732**
	10	baby	26.17	0.5044	28.54	0.6364	26.66	0.5307	25.73	0.4834	25.43	0.4713	25.21	0.4587	**31.41**	**0.7946**
		bird	26.30	0.5702	28.51	0.6943	26.77	0.5946	25.84	0.5487	25.52	0.5373	25.24	0.5208	**30.99**	**0.8345**
		butterfly	25.04	0.6845	**26.33**	0.7572	25.41	0.7002	24.95	0.6777	24.52	0.6597	24.3	0.6547	26.08	**0.8381**
		head	25.92	0.4649	28.34	0.5936	26.4	0.4899	25.53	0.4463	25.44	0.4454	24.99	0.4192	**30.84**	**0.7124**
		woman	25.83	0.5698	27.68	0.6765	26.23	0.5912	25.44	0.5528	25.14	0.5433	24.92	0.5311	**29.11**	**0.8199**
		average	25.85	0.5588	27.88	0.6716	26.29	0.5813	25.50	0.5418	25.21	0.5314	24.93	0.517	**29.69**	**0.7999**
	15	baby	22.85	0.3496	25.62	0.4863	23.35	0.3726	22.39	0.3303	21.91	0.3108	21.83	0.3081	**29.71**	**0.7343**
		bird	22.93	0.4126	25.52	0.5495	23.41	0.4358	22.45	0.3915	22.03	0.3753	21.81	0.3643	**28.98**	**0.7723**
		butterfly	22.29	0.5741	23.88	0.6497	22.64	0.5892	21.92	0.5626	21.37	0.5399	21.3	0.5392	**24.23**	**0.7802**
		head	22.81	0.3150	25.64	0.4546	23.3	0.3372	22.4	0.2983	22.18	0.2933	21.7	0.2709	**29.50**	**0.6567**
		woman	22.69	0.4311	24.98	0.5442	23.12	0.4505	22.24	0.4137	21.81	0.3971	21.67	0.3933	**27.18**	**0.7592**
		average	22.71	0.4165	25.13	0.5369	23.164	0.4371	22.28	0.3993	21.86	0.3833	21.66	0.3752	**27.92**	**0.7406**
	20	baby	20.5	0.2556	23.46	0.3792	20.98	0.2741	20.03	0.2389	19.45	0.2196	19.39	0.2191	**28.51**	**0.6856**
		bird	20.55	0.3105	23.34	0.4398	21.02	0.3304	20.05	0.2914	19.57	0.2754	19.35	0.2665	**27.67**	**0.7219**
		butterfly	20.12	0.4935	21.95	0.5666	20.46	0.5071	19.64	0.4785	19.04	0.4555	19.03	0.4557	**22.87**	**0.7262**
		head	20.58	0.2253	23.58	0.3529	21.05	0.2429	20.15	0.211	19.82	0.2049	19.29	0.185	**28.47**	**0.6121**
		woman	20.41	0.3415	22.93	0.4475	20.83	0.3576	19.94	0.3247	19.42	0.3062	19.28	0.3051	**25.86**	**0.7077**
		average	20.43	0.3253	23.05	0.437	20.87	0.3424	19.96	0.3089	19.46	0.2923	19.27	0.2863	**26.67**	**0.6907**
×3	5	baby	30.70	0.7523	30.42	0.7483	31.06	0.7713	30.33	0.7349	30.5	0.7495	29.95	0.7142	**32.05**	**0.8407**
		bird	30.53	0.8043	30.22	0.8024	30.86	0.8214	30.39	0.793	30.45	0.8084	29.96	0.7724	**31.43**	**0.8756**
		butterfly	24.97	0.7859	24.8	0.7846	25.20	0.8006	**25.94**	0.8088	26.15	0.8044	25.53	0.7841	24.97	**0.8280**
		head	30.08	0.6777	30.01	0.6748	30.4	0.6938	29.78	0.6638	30.09	0.6871	29.42	0.6436	**31.40**	**0.7442**
		woman	28.33	0.7781	28.08	0.776	28.61	0.7943	28.6	0.7719	28.51	0.7823	28.2	0.7483	**28.80**	**0.8534**
		average	28.92	0.7597	28.71	0.7572	29.23	0.7763	29.01	0.7545	29.14	0.7663	28.61	0.7325	**29.73**	**0.8284**
	10	baby	26.08	0.5299	25.85	0.5257	26.54	0.5576	25.53	0.5034	25.59	0.5091	25.05	0.4762	**29.78**	**0.7625**
		bird	26.08	0.5952	25.83	0.5913	26.51	0.6215	25.55	0.5685	25.58	0.5798	25.03	0.5394	**28.99**	**0.7993**
		butterfly	23.22	0.6601	23.06	0.6602	23.5	0.6784	23.46	0.6665	23.45	0.6573	22.92	0.6341	**23.42**	**0.7635**
		head	25.89	0.4726	25.81	0.468	26.36	0.498	25.39	0.4488	25.61	0.4677	24.87	0.4199	**29.64**	**0.6749**
		woman	25.17	0.5863	24.94	0.5839	25.54	0.6100	24.86	0.5673	24.80	0.5734	24.34	0.5371	**26.89**	**0.7812**
		average	25.29	0.5688	25.10	0.5658	25.69	0.5931	24.96	0.5509	25.01	0.5575	24.44	0.5213	**27.74**	**0.7563**
	15	baby	22.95	0.3812	22.72	0.3763	23.41	0.406	22.34	0.3541	22.25	0.3506	21.79	0.3285	**28.13**	**0.6977**
		bird	23.00	0.4435	22.75	0.4375	23.44	0.4683	22.35	0.4125	22.23	0.4114	21.78	0.3835	**27.33**	**0.7353**
		butterfly	21.34	0.5608	21.11	0.5585	21.62	0.5781	21.12	0.556	20.99	0.5446	20.57	0.5246	**22.10**	**0.6994**
		head	22.98	0.3324	22.86	0.3273	23.44	0.3556	22.39	0.3083	22.34	0.3114	21.71	0.2778	**28.22**	**0.6178**
		woman	22.53	0.4519	22.28	0.4478	22.91	0.4733	21.99	0.4278	21.84	0.4257	21.43	0.4002	**25.43**	**0.7168**
		average	22.56	0.434	22.34	0.4295	22.96	0.4563	22.04	0.4117	21.93	0.4087	21.46	0.3829	**26.24**	**0.6934**
	20	baby	20.68	0.2856	20.43	0.2805	21.11	0.3056	20.04	0.2606	19.83	0.2528	19.4	0.2375	**26.78**	**0.6397**
		bird	20.73	0.3403	20.47	0.3337	21.16	0.4998	20.04	0.3109	19.8	0.3023	19.39	0.2825	**26.07**	**0.6782**
		butterfly	19.63	0.4847	19.36	0.4802	19.9	0.262	19.15	0.4708	18.98	0.4612	18.62	0.4435	**21.18**	**0.6453**
		head	20.83	0.243	20.69	0.238	21.27	0.3791	20.21	0.2215	19.96	0.2154	19.35	0.1913	**26.97**	**0.5650**
		woman	20.47	0.3614	20.19	0.3558	20.82	0.3615	19.82	0.3361	19.58	0.3286	19.19	0.3108	**24.35**	**0.6598**
		average	20.47	0.3430	20.23	0.3376	20.85	0.3616	19.85	0.3200	19.63	0.3121	19.19	0.2931	**25.07**	**0.6376**

(*Continued*)

Table 2. (*Continued*)

Scale	σ	Set5 images	NE [22]		SCSR [25]		Zedye [26]		A+ [31]		SRCNN [20]		CSC [32]		ours	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
×4	5	baby	29.79	0.7405	-	-	30.1	0.7579	29.57	0.7276	29.95	0.7566	29.18	0.7045	**30.65**	**0.8066**
		bird	29.19	0.7878	-	-	29.43	0.8029	29.22	0.7823	29.38	0.8053	28.86	0.7608	**29.67**	**0.8355**
		butterfly	22.92	0.7246	-	-	23.13	0.7414	**23.70**	**0.7624**	24.22	0.7734	23.46	0.7344	23.05	0.7582
		head	29.4	0.6561	-	-	29.69	0.6717	29.27	0.6494	29.67	0.6788	28.93	0.6309	**30.39**	**0.7093**
		woman	26.52	0.7496	-	-	26.76	0.7654	26.96	0.7522	26.83	0.7688	26.71	0.7282	**26.82**	**0.8062**
		average	27.56	0.7317	-	-	27.82	0.7479	27.74	0.7348	28.01	0.7566	27.43	0.7118	**28.10**	**0.7832**
	10	baby	25.73	0.5442	-	-	26.14	0.5697	25.25	0.5196	25.67	0.5523	24.72	0.4864	**28.66**	**0.7342**
		bird	25.61	0.6081	-	-	25.93	0.6310	25.16	0.5832	25.51	0.6208	24.61	0.5504	**27.56**	**0.7612**
		butterfly	21.78	0.6258	-	-	22.00	0.6427	22.02	0.6405	**22.41**	0.6546	21.62	0.6038	21.96	**0.7053**
		head	25.63	0.4798	-	-	26.04	0.5038	25.24	0.4619	25.77	0.5052	24.7	0.431	**28.82**	**0.6504**
		woman	24.21	0.5843	-	-	24.49	0.6056	24.05	0.5707	24.15	0.599	23.59	0.5376	**25.39**	**0.7394**
		average	24.59	0.5684	-	-	24.92	0.5906	24.34	0.5552	24.70	0.5864	23.85	0.5218	**26.48**	**0.7181**
	15	baby	22.79	0.4029	-	-	23.19	0.4264	22.21	0.3755	22.37	0.3935	21.59	0.3433	**27.19**	**0.6783**
		bird	22.82	0.4648	-	-	23.17	0.488	22.23	0.434	22.40	0.4625	21.58	0.3985	**26.11**	**0.7021**
		butterfly	20.38	0.5401	-	-	20.57	0.5544	20.20	0.5375	20.43	0.549	19.73	0.5025	**20.83**	**0.6482**
		head	22.85	0.3508	-	-	23.25	0.3722	22.36	0.3301	22.60	0.3628	21.63	0.2942	**27.53**	**0.6037**
		woman	21.98	0.4594	-	-	22.28	0.4788	21.53	0.438	21.57	0.4564	20.98	0.4057	**24.16**	**0.6822**
		average	22.16	0.4436	-	-	22.49	0.4640	21.71	0.4230	21.87	0.4448	21.10	0.3888	**25.16**	**0.6629**
	20	baby	20.59	0.3081	-	-	20.96	0.3271	19.97	0.2812	19.91	0.2885	19.25	0.2515	**25.98**	**0.6304**
		bird	20.68	0.3633	-	-	21.01	0.3819	20.03	0.3308	20.01	0.3485	19.28	0.2957	**25.01**	**0.6514**
		butterfly	18.99	0.4705	-	-	19.16	0.4819	18.54	0.4563	18.65	0.4652	18.03	0.4251	**19.97**	**0.5984**
		head	20.77	0.264	-	-	21.15	0.2819	20.21	0.2444	20.19	0.2643	19.31	0.2065	**26.38**	**0.5609**
		woman	20.1	0.3704	-	-	20.41	0.3865	19.53	0.3461	19.45	0.3570	18.87	0.3154	**23.22**	**0.6335**
		average	20.23	0.3553	-	-	20.54	0.3719	19.66	0.3318	19.64	0.3446	18.95	0.2988	**24.11**	**0.6149**

https://doi.org/10.1371/journal.pone.0182165.t002

from the minimum 0.52 dB (upscaling factor ×4, $\sigma = 5$) to the maximum 6.18 dB (upscaling factor ×2, $\sigma = 20$). In addition, our method improves on average 3.62 dB (Set5, upscaling factor ×2, $\sigma = 20$) over the next top robustness method SCSR [25]. Figs 5–8 provide a visual assessment. We can see that our method gets similar quality performance as the top methods it was compared to when $\sigma = 0$, and it has the strongest robustness.

3.2.2 Reconstruction time. Average reconstruction time of test images in Set5 was compared when $\sigma = 10$. Actually, noise has little effect on test results. The experiments were conducted on the same computer. The results are summarized in Table 4. The reconstruction

Table 3. The results of average PSNR (dB) and SSIM on the Set14 and B100 dataset.

dataset	Scale	σ	NE [22]		SCSR [25]		Zedye [26]		A+ [31]		SRCNN [20]		CSC [32]		ours	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Set14	×2	5	28.74	0.7514	29.31	0.7981	29.01	0.7647	28.71	0.7737	28.61	0.7435	28.36	0.7275	**29.69**	**0.8205**
		10	25.08	0.5551	26.59	0.6478	25.46	0.5750	24.78	0.5400	24.45	0.5264	24.31	0.5180	**27.80**	**0.7381**
		15	22.29	0.4204	24.31	0.5213	22.71	0.4394	21.89	0.4309	21.42	0.3848	21.35	0.3821	**26.38**	**0.6732**
		20	20.15	0.3302	22.47	0.4259	20.57	0.3466	19.70	0.3140	19.14	0.2945	19.10	0.2933	**25.35**	**0.6220**
	×3	5	26.86	0.6903	26.55	0.6891	27.08	0.7035	26.96	0.6859	26.99	0.6942	26.70	0.6703	**27.16**	**0.7220**
		10	24.19	0.5240	23.95	0.5215	24.52	0.5441	23.92	0.5079	23.90	0.5014	23.53	0.4856	**25.78**	**0.6663**
		15	21.89	0.4032	21.64	0.3990	22.24	0.4221	21.43	0.3827	21.29	0.3781	20.96	0.3606	**24.67**	**0.6075**
		20	19.99	0.3196	19.72	0.3142	20.35	0.3355	19.43	0.2981	19.20	0.2900	18.88	0.2767	**23.77**	**0.5579**
	×4	5	25.57	0.6398	-	-	25.76	0.6526	25.76	0.6416	25.89	0.6575	25.49	0.6241	**25.73**	**0.6788**
		10	23.42	0.4985	-	-	23.42	0.5174	23.24	0.4865	23.45	0.5078	22.84	0.4607	**24.64**	**0.6171**
		15	21.39	0.3896	-	-	21.69	0.4076	21.01	0.3713	21.08	0.3836	20.51	0.3448	**23.70**	**0.5686**
		20	19.66	0.3115	-	-	19.96	0.3265	19.15	0.2910	19.08	0.2958	18.57	0.2655	**22.91**	**0.5283**
B100	×2	5	28.00	0.7264	28.81	0.7719	28.19	0.7380	27.96	0.7196	28.02	0.721	27.83	0.7076	**28.95**	**0.7917**
		10	24.66	0.5279	26.28	0.6200	25.02	0.5480	24.36	0.5123	24.17	0.5059	24.07	0.4997	**27.29**	**0.7037**
		15	22.01	0.3951	24.10	0.4941	22.42	0.4136	21.63	0.3792	21.25	0.3661	21.23	0.3654	**26.08**	**0.6378**
		20	19.95	0.3077	22.32	0.4000	20.37	0.3233	19.52	0.2925	19.03	0.277	19.02	0.2779	**25.20**	**0.5873**
	×3	5	26.32	0.6518	26.79	0.6728	26.49	0.6638	26.34	0.6463	26.46	0.6586	26.20	0.6351	**26.85**	**0.7010**
		10	23.86	0.4878	23.74	0.4874	24.15	0.5067	23.58	0.4716	23.60	0.4767	23.27	0.4538	**25.64**	**0.6273**
		15	21.66	0.3703	21.47	0.3674	21.99	0.3882	21.22	0.3510	21.10	0.3481	20.81	0.3326	**24.66**	**0.5702**
		20	19.82	0.2902	19.59	0.2855	20.17	0.3051	19.29	0.2706	19.07	0.2635	18.78	0.2526	**23.84**	**0.5223**
	×4	5	25.3	0.6015	-	-	25.46	0.6133	25.36	0.5991	25.53	0.6171	25.2	0.5857	**25.72**	**0.6414**
		10	23.23	0.4615	-	-	23.49	0.4800	23.01	0.4478	23.23	0.4690	22.69	0.4269	**24.74**	**0.5815**
		15	21.26	0.3562	-	-	21.56	0.3736	20.86	0.3378	20.95	0.3500	20.43	0.3161	**23.89**	**0.5358**
		20	19.56	0.2820	-	-	19.86	0.2966	19.06	0.2622	18.99	0.2669	18.52	0.2412	**23.15**	**0.4980**

https://doi.org/10.1371/journal.pone.0182165.t003

time varies a lot for different upscaling factors. Our algorithm cost fewer than 10s. The reconstruction time of our algorithm is comparable to that of SCSR, CSC, and SRCNN. SCSR is the slowest method.

3.3 Effect of IBP

Combined with iterative back projection (IBP), the denoised LR patches are applied to improve SR performance in our algorithm. According to [47], IBP has an important role to improve SR performance. But if the input is a noisy image, the model of IBP will propagate the noise to the HR image. Experimental results show that if we use IBP algorithm directly on the

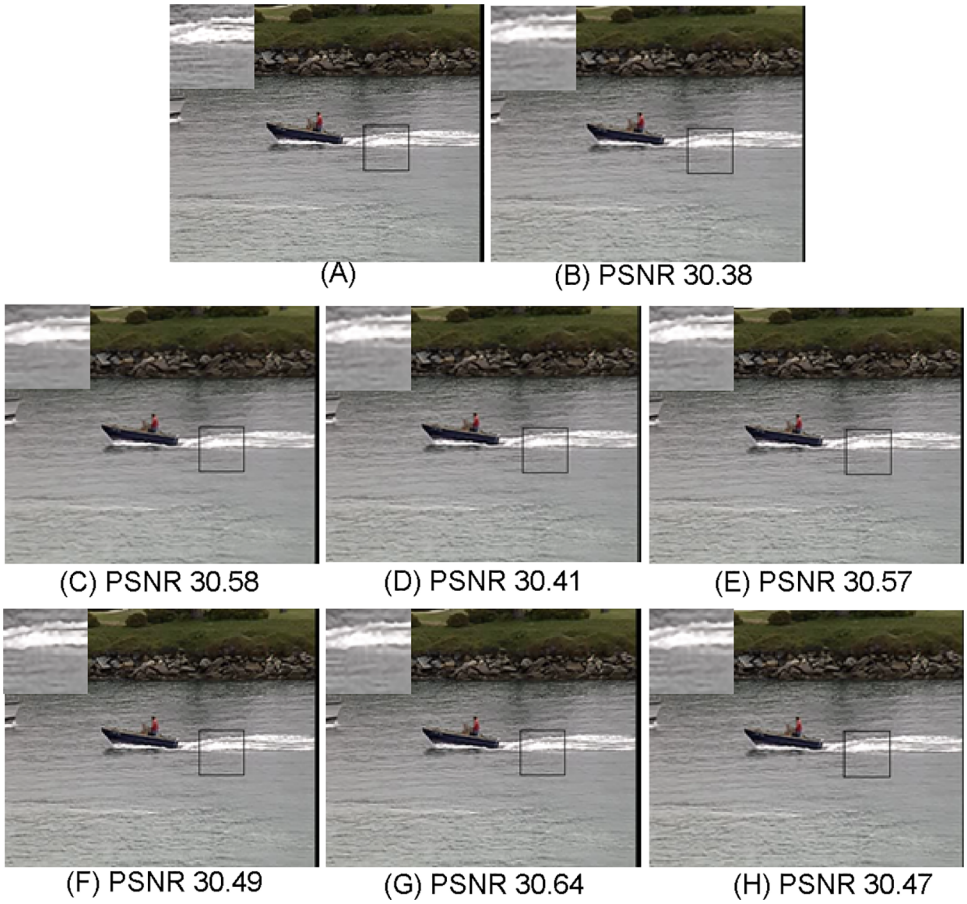

Fig 5. Comparisons with various image super-resolution methods on "coastguard" from Set14 with upscaling factor ×2 ($\sigma = 0$, PSNR in dB). (A) Ground truth HR; (B) NE [22]; (C) SCSR [25]; (D) Zedye [26]; (E) A+ [31]; (F) SRCNN [20]; (G) CSC [32]; (H) ours.

https://doi.org/10.1371/journal.pone.0182165.g005

input LR image, the performance will become worse. The results are listed in Table 5. The iteration number of IBP here is 20. From this comparison, we can see that the superiority of our method is obvious. Other datasets Set14 and B100 can also achieve similar results. The results of Set14 and B100 are shown in S1 Table.

3.4 Effect of distance penalty

Distance penalty is applied to model the weight. To check the effect of the penalty for improving SR performance, we perform our method with and without the penalty respectively on

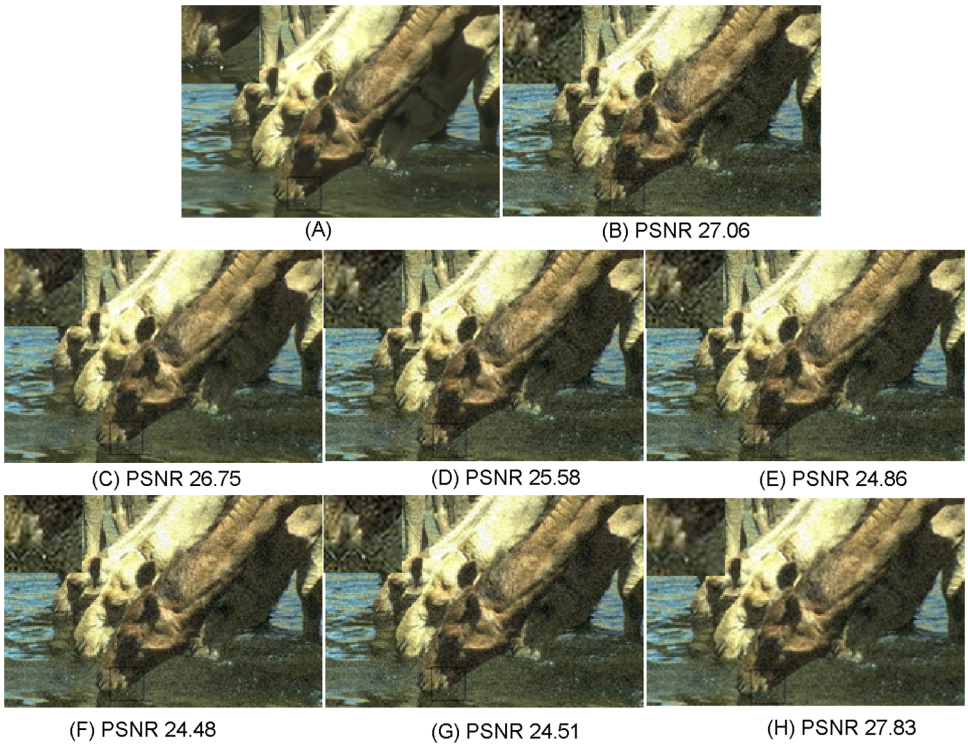

Fig 6. Comparisons with various image super-resolution methods on "16077" from B100 with upscaling factor ×2 ($\sigma = 10$, PSNR in dB). (A) Ground truth HR; (B) NE [22]; (C) SCSR [25]; (D) Zedye [26]; (E) A+ [31]; (F) SRCNN [20]; (G) CSC [32]; (H) ours.

https://doi.org/10.1371/journal.pone.0182165.g006

Set5 database. The experiments are done in different γ. The results are shown in in Fig 9. We can see that our method with distance penalty obtains better performance and the superiority of our method with distance penalty is obvious. Other datasets Set14 and B100 can also achieve similar results. The results of Set14 and B100 are shown in S19–S24 Figs.

4 Conclusion

In this research, we proposed an algorithm of noisy image super-resolution based on sparse representation. For the problem of noisy image super-resolution, most of the existing methods will become less effective because they assume that the input LR image is noise-free. The

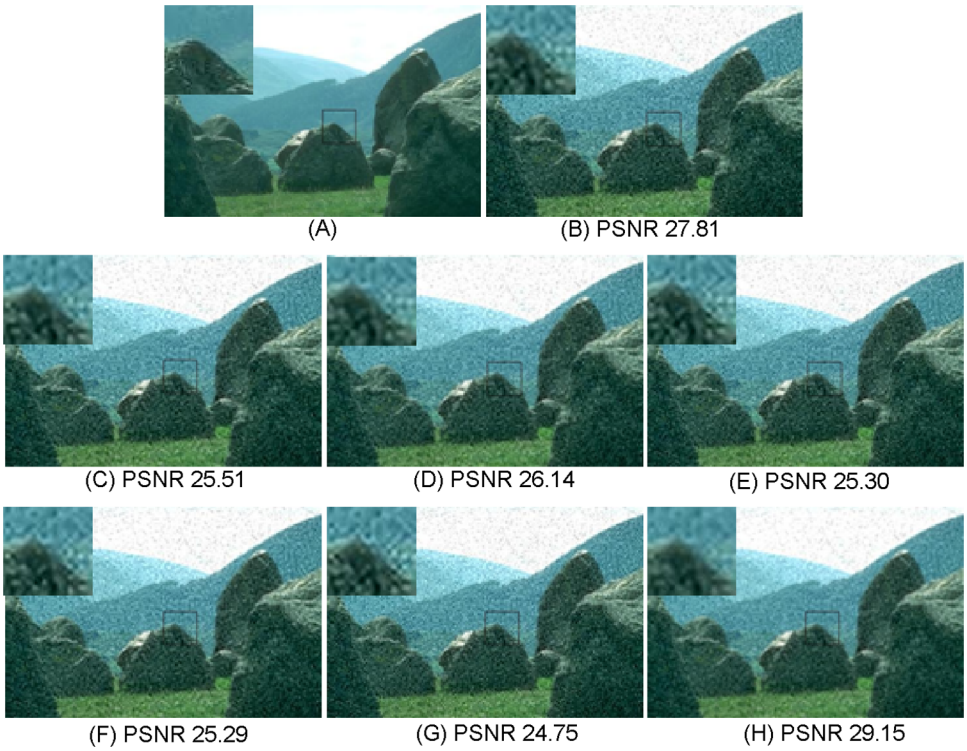

Fig 7. Comparisons with various image super-resolution methods on "241004" from B100 with upscaling factor ×3 ($\sigma = 10$, PSNR in dB). (A) Ground truth HR; (B) NE [22]; (C) SCSR [25]; (D) Zedye [26]; (E) A+ [31]; (F) SRCNN [20]; (G) CSC [32]; (H) ours.

https://doi.org/10.1371/journal.pone.0182165.g007

proposed algorithm can achieve simultaneously image super-resolution and denoising. For different input LR images, even if the noise variance varies, the dictionary pair does not need to be retained. The core idea of the proposed algorithm is that HR image patch is reconstructed through weighted average of similar HR example patches. In particular, atoms of learned sparse dictionary are used to compute the weight and reconstruct HR patch instead of example patches. This strategy can reduce time computation and suppress noise. In addition, LR example patches subtracted mean pixel value are also used to learn dictionary rather than just their gradient features, which will help IBP to further improve the SR performance. The experimental results show that our method performs better noise robustness.

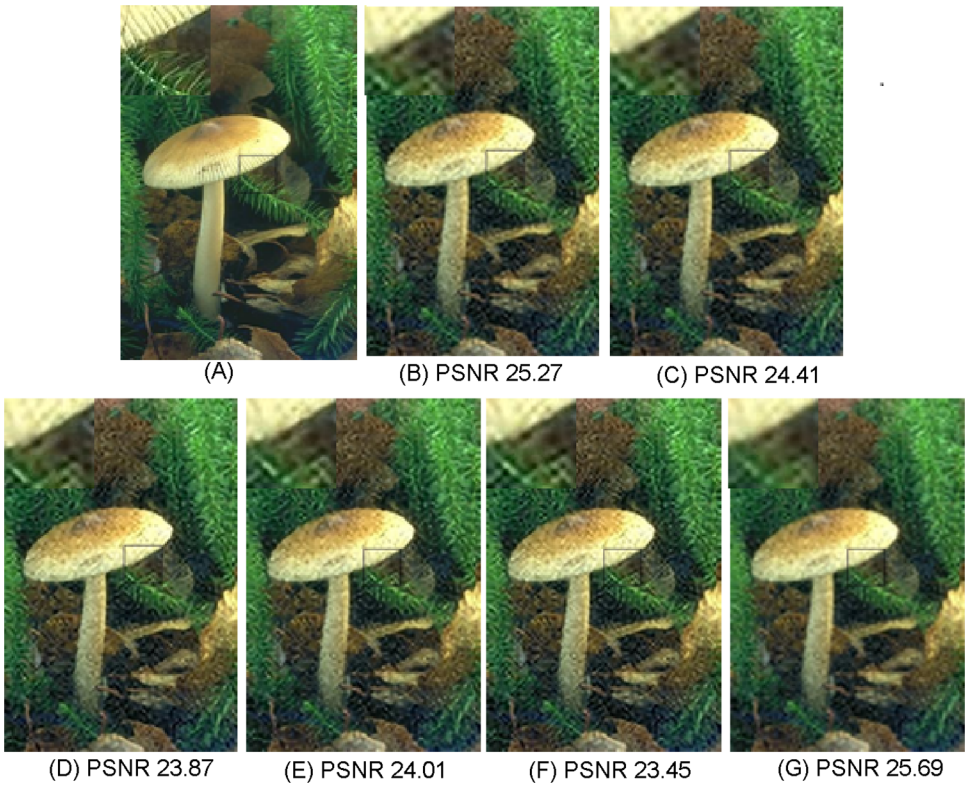

Fig 8. Comparisons with various image super-resolution methods on "208001" from B100 with upscaling factor ×4 ($\sigma = 10$, PSNR in dB). (A) Ground truth HR; (B) NE [22]; (C) Zedye [26]; (D) A+ [31]; (E) SRCNN [20]; (F) CSC [32]; (G) ours.

https://doi.org/10.1371/journal.pone.0182165.g008

Table 4. Comparisons of average reconstruction time (s)on Set5.

Scale	NE [22]	SCSR [25]	Zedye [26]	A+ [31]	SRCNN [20]	CSC [32]	ours
×2	4.78	193.26	6.82	0.88	7.54	139.03	3.21
×3	2.78	44.31	3.01	0.57	7.47	78.46	1.24
×4	1.63	-	1.96	0.42	6.39	48.24	0.75

https://doi.org/10.1371/journal.pone.0182165.t004

Table 5. Effect of IBP on average PSNR(dB) and SSIM (Set 5).

Scale	IBP	$\sigma = 5$		$\sigma = 10$		$\sigma = 15$		$\sigma = 20$	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
×2	×	31.48	0.831	27.76	0.665	25.03	0.531	22.93	0.432
	√	29.93	0.753	25.16	0.526	21.95	0.383	19.58	0.293
	ours	32.49	0.873	29.69	0.800	27.92	0.741	26.67	0.691
×3	×	29.19	0.801	26.59	0.660	24.33	0.537	22.47	0.442
	√	28.39	0.730	24.52	0.523	21.62	0.385	19.39	0.296
	ours	29.72	0.828	27.74	0.756	26.24	0.693	25.07	0.638
×4	×	27.65	0.765	25.58	0.646	23.64	0.537	21.97	0.449
	√	27.19	0.706	23.93	0.524	21.28	0.392	19.17	0.303
	ours	28.10	0.783	26.48	0.718	25.16	0.663	24.11	0.615

https://doi.org/10.1371/journal.pone.0182165.t005

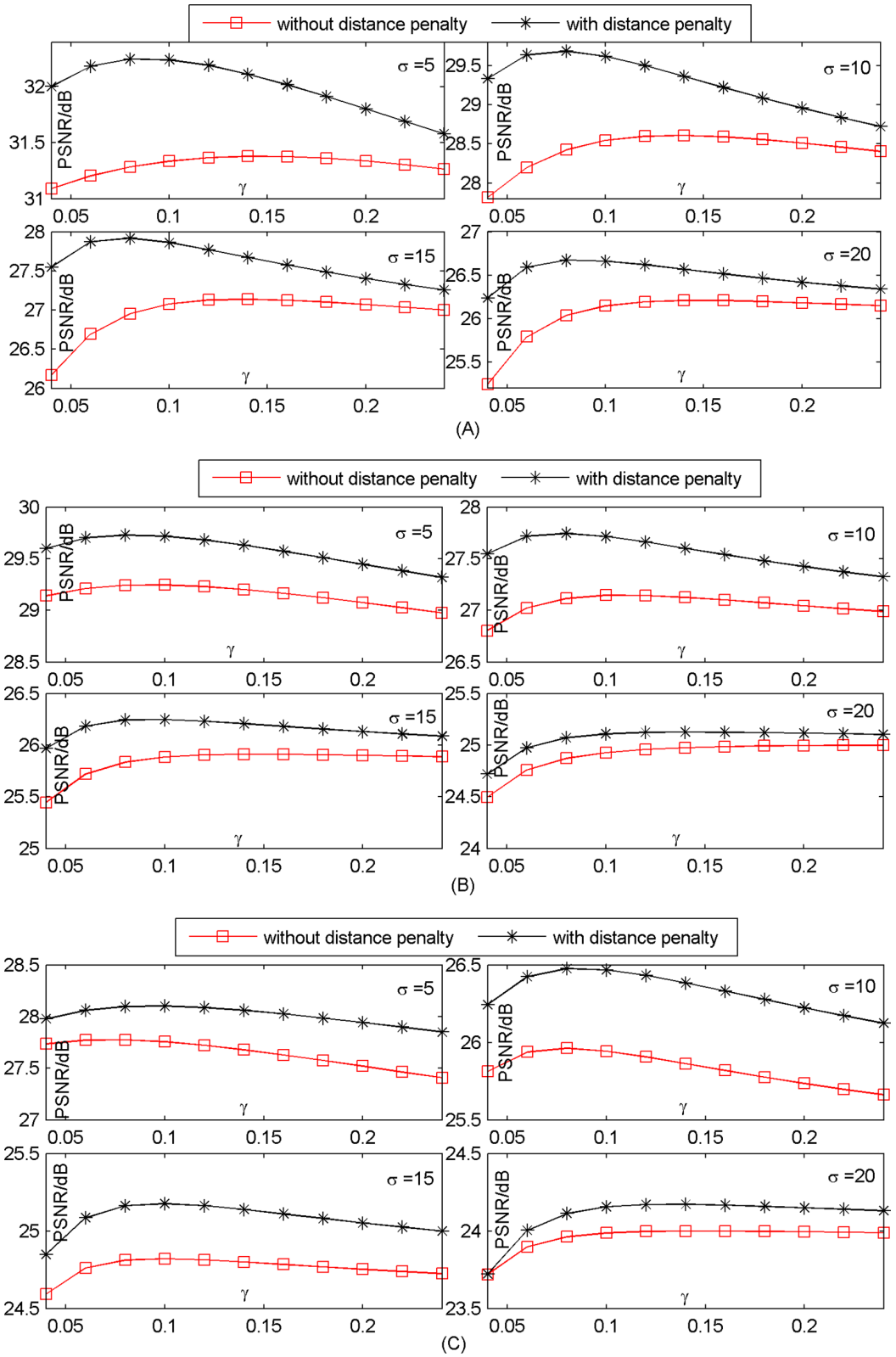

Fig 9. Effect of distance penalty on average PSNR (dB)(Set 5). (A) upscaling factor ×2; (B) upscaling factor ×3; (C) upscaling factor ×4.

https://doi.org/10.1371/journal.pone.0182165.g009

Supporting information

S1 Fig. γ versus average PSNR on Set14. (upscaling factor ×2).
(TIF)

S2 Fig. γ versus average PSNR on Set14. (upscaling factor ×3).
(TIF)

S3 Fig. γ versus average PSNR on Set14. (upscaling factor ×4).
(TIF)

S4 Fig. γ versus average PSNR on B100. (upscaling factor ×2).
(TIF)

S5 Fig. γ versus average PSNR on B100. (upscaling factor ×3).
(TIF)

S6 Fig. γ versus average PSNR on B100. (upscaling factor ×4).
(TIF)

S7 Fig. Dictionary size influence on performance on average on Set14. (upscaling factor ×2).
(TIF)

S8 Fig. Dictionary size influence on performance on average on Set14. (upscaling factor ×3).
(TIF)

S9 Fig. Dictionary size influence on performance on average on Set14. (upscaling factor ×4).
(TIF)

S10 Fig. Dictionary size influence on performance on average on B100. (upscaling factor ×2).
(TIF)

S11 Fig. Dictionary size influence on performance on average on B100. (upscaling factor ×3).
(TIF)

S12 Fig. Dictionary size influence on performance on average on B100. (upscaling factor ×4).
(TIF)

S13 Fig. Number of similar atoms influence on performance on average on Set14. (upscaling factor ×2).
(TIF)

S14 Fig. Number of similar atoms influence on performance on average on Set14. (upscaling factor ×3).
(TIF)

S15 Fig. Number of similar atoms influence on performance on average on Set14. (upscaling factor ×4).
(TIF)

S16 Fig. Number of similar atoms influence on performance on average on B100. (upscaling factor ×2).
(TIF)

S17 Fig. Number of similar atoms influence on performance on average on B100. (upscaling factor ×3).
(TIF)

S18 Fig. Number of similar atoms influence on performance on average on B100. (upscaling factor ×4).
(TIF)

S19 Fig. Effect of Distance Penalty on Average PSNR (dB) on average on Set14. (upscaling factor ×2).
(TIF)

S20 Fig. Effect of Distance Penalty on Average PSNR (dB) on average on Set14. (upscaling factor ×3).
(TIF)

S21 Fig. Effect of Distance Penalty on Average PSNR (dB) on average on Set14. (upscaling factor ×4).
(TIF)

S22 Fig. Effect of Distance Penalty on Average PSNR (dB) on average on B100. (upscaling factor ×2).
(TIF)

S23 Fig. Effect of Distance Penalty on Average PSNR (dB) on average on B100. (upscaling factor ×3).
(TIF)

S24 Fig. Effect of Distance Penalty on Average PSNR (dB) on average on B100. (upscaling factor ×4).
(TIF)

S1 Table. Effect of IBP on Average PSNR (dB) and SSIM (Set14 and B100).
(PDF)

Author Contributions

Conceptualization: Yulan Han.

Data curation: Yulan Han.

Formal analysis: Yulan Han.

Methodology: Yulan Han, Yongping Zhao, Qisong Wang.

Project administration: Yongping Zhao.

Software: Yulan Han, Qisong Wang.

Supervision: Yongping Zhao.

Writing – original draft: Yulan Han.

Writing – review & editing: Yulan Han, Yongping Zhao, Qisong Wang.

References

1. Li X, Orchard MT. New edge-directed interpolation. IEEE Trans. Image Process. 10 (2001) 1521–1527. https://doi.org/10.1109/83.951537 PMID: 18255495
2. Zhang X, Wu X, Image interpolation by adaptive 2-d autoregressive modeling and soft-decision estimation, IEEE Trans. Image Process. 17 (2008) 887–896. https://doi.org/10.1109/TIP.2008.924279 PMID: 18482884
3. Lia Xiaoyan, Hea Hongjie, Yina Zhongke, Chena Fan, Cheng Jun, KPLS-based Image Super-resolution using Clustering and Weighted Boosting, Neurocomputing. 149 (2015) 940–948. https://doi.org/10.1016/j.neucom.2014.07.040
4. Sun J, Xu Z, Shum H-Y, Image super-resolution using gradient profile prior, IEEE Conf. Computer Vision and Pattern Recognition. (2008), 1–8.
5. Tai YW, Liu S, Brown MS, and Lin S. Super resolution using edge prior and single image detail synthesis, IEEE Conf. Comput. Vis. Pattern Recognit. (2010) 2400–2407.
6. Marquina A and Osher SJ, Image super-resolution by TV-regularization and Bregman iteration, J. Sci. Comput. 37(3) (2008) 367–382. https://doi.org/10.1007/s10915-008-9214-8
7. Dong W, Zhang L, Shi G, et al., Nonlocally Centralized Sparse Representation for Image Restoration, IEEE Trans. On Image Processing. 22(4) (2013) 1620–1630. https://doi.org/10.1109/TIP.2012.2235847
8. Dong W, Zhang L, Sparse Representation based Image Interpolation with Nonlocal Autoregressive Modeling, IEEE Trans. on Image Processing. 4(22) (2013) 1382–1394. https://doi.org/10.1109/TIP.2012.2231086
9. Zhang J, Zhao D, Gao W, Group-based Sparse Representation for Image Restoration, IEEE Transactions on Image Processing. 8(23) (2014) 3336–3351. https://doi.org/10.1109/TIP.2014.2323127
10. Zhang J, Zhao D, Xiong R, et al., Image Restoration Using Joint Statistical Modeling in a Space-Transform Domain, IEEE Transactions on Circuits System and Video Technology. 6(24) (2014) 915–928. https://doi.org/10.1109/TCSVT.2014.2302380
11. Lin Z, Shum H, Fundamental limits of reconstruction-based super-resolution algorithms under local translation, IEEE Trans. Pattern Anal. Mach. Intell. 26(1) (2004) 83–97. https://doi.org/10.1109/TPAMI.2004.1261081 PMID: 15382688
12. Lee Hui Jung, Choi Dong-Yoon, Song Byung Cheol, Learning-based superresolution algorithm using quantized pattern and bimodal postprocessing for text images, Journal of Electronic Imaging. 24(6) (2015) 063011. https://doi.org/10.1117/1.JEI.24.6.063011
13. Zhang Kaibing, Gao Xinbo, Tao Dacheng, et al., Single Image Super-Resolution With Multiscale Similarity Learning, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. 24 (10) (2013) 1648–1659. https://doi.org/10.1109/TNNLS.2013.2262001
14. Trinh Dinh-Hoan, Luong Marie et al., Novel Example-Based Method for Super-Resolution and Denoising of Medical Images, IEEE Trans on Image Processing. 23(4) (2014) 1882–1895. https://doi.org/10.1109/TIP.2014.2308422
15. Freeman WT, Jones TR, Pasztor EC, Example-based superresolution, IEEE Comput. Graph. 22(2) (2002) 56–65. https://doi.org/10.1109/38.988747
16. Kim KI, Kwon Y, Example-based learning for single-image super-resolution, Proc. DAGM. (2008) 456–465.
17. Ni Karl S, Nguyen TQ, Image superresolution using support vector regression, IEEE Trans. Image Process. 16(6) (2007) 1596–1610. https://doi.org/10.1109/TIP.2007.896644 PMID: 17547137
18. Wei Zhao, Tao Feng, Jun Wang, Kalman filter based method for image superresolution using a sequence of low-resolution images, Journal of Electronic Imaging. 23(1) (2014). https://doi.org/10.1117/1.JEI.23.1.013008
19. Tang Songze, Xiao Liang, Liu Pengfei, Zhang Jun, Huang Lili, Edge and color preserving single image superresolution, Journal of Electronic Imaging. 23(3) (2014). https://doi.org/10.1117/1.JEI.23.3.033002
20. Dong Chao, Loy Chen Change, He Kaiming and Tang Xiaoou, Learning a Deep Convolutional Network for Image Super-Resolution, European Conference on Computer Vision. (2014) 184–199.
21. Dai D, Timofte R, Van Gool L, Jointly Optimized Regressors for Image Super-resolution, Image and Video Processing. 34(2) (2015) 95–104.

22. Chang H, Yeung D-Y, Y Xiong, Super-resolution through neighbor embedding, IEEE Conf. Comput. Vis. PatternRecog. (2004) 275–282.
23. Gao X, Zhang K, Li X, Tao D, Image super-resolution with sparse neighbor embedding, IEEE Trans. Image Process. 21(7) (2014) 3194–3205.
24. Ni Karl S, Nguyen Truong Q. An Adaptable K-Nearest Neighbors Algorithm for MMSE Image Interpolation, IEEE Trans. Image Process. 18(9) (2009) 1976–1987. https://doi.org/10.1109/TIP.2009.2023706 PMID: 19473939
25. Yang J et al., Image super-resolution via sparse representation, IEEETrans. Image Process. 19(11) (2010) 861–2873.
26. Zeyde R, Elad M, Protter M, On single image scale-up using sparse-representations, in Proc. 7th Int. Conf. Curves Surf.. (2010) 711–730.
27. Wang S, Zhang L, Liang Y, Pan Q, Semi-Coupled Dictionary Learning with Applications to Image Super-Resolution and Photo-Sketch Image Synthesis, In CVPR 2012.
28. Zhou Fei, Yuan Tingrong, Yang Wenming, et al., Single-Image Super-Resolution Based on Compact KPCA Coding and Kernel Regression, IEEE Signal Processing Letters. 3(22) (2015) 336–340. https://doi.org/10.1109/LSP.2014.2360038
29. Zhang Kaibing, Tao Dacheng, Gao Xinbo, Learning Multiple Linear Mappings for Efficient Single Image Super-Resolution, IEEE Trans. Image Process. 3(24) (2015) 846–861. https://doi.org/10.1109/TIP.2015.2389629
30. Timofte Radu, Vincent De Smet, Luc Van Gool, Anchored Neighborhood Regression for Fast Example-Based Super-Resolution, IEEE International Conference on Computer Vision. (2013) 1920–1927.
31. Timofte Radu, Vincent De Smet, Luc Van Gool, A+: Adjusted Anchored Neighborhood Regression for Fast Super-Resolution, ACCV. 3(24) (2014) 111–126.
32. Gu S, Zuo W, Xie Q, Meng D, Feng X, Zhang L, Convolutional Sparse Coding for Image Super-resolution, ICCV. 3(24) (2015).
33. Yang Min chun, Wang Yuchiang, A self-learning Approach to single Image Super-resolution, IEEE Trans. Multimedia. 15(3) (2013) 498–508. https://doi.org/10.1109/TMM.2012.2232646
34. Singh A and Ahuja N, Super-resolution using sub-band self-similarity, ACCV. (2014) 1–8.
35. Huang Jia-Bin, Singh Abhishek, and Ahuja Narendra, Single Image Super-resolution From Transformed self-exemplars, CVPR. (2015) 5197–5206.
36. Xie Jun, Feris RS, Yu Shiaw-Shian, Sun Ming-Ting, Joint Super Resolution and Denoising From a Single Depth Image, Transactions on Multimedia. 17(9) (2015) 1525–1537. https://doi.org/10.1109/TMM.2015.2457678
37. Xu Jian, Chang Zhiguo, Fan Jiulun, Zhao Xiaoqiang, Wu Xiaomin, Wang Yanzi, Noisy image magnification with total variation regularization and order-changed dictionary learning, Journal on Advances in Signal Processing. 41 (2015).
38. Elad M, Aharon M, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. Image Process. 15(2) (2006) 3736–3745. https://doi.org/10.1109/TIP.2006.881969 PMID: 17153947
39. Chen Chun Lung Philip, Liu Licheng, Chen Long, Yan Tang Yuan, Zhou Yicong, Weighted Couple Sparse Representation With Classified Regularization for Impulse Noise Removal, IEEE Trans. Image Process. 24(1) (2015) 4014–4026. https://doi.org/10.1109/TIP.2015.2456432 PMID: 26186781
40. Liu Yu, Wang Zengfu, Simultaneous image fusion and denoising with adaptive sparse representation, IET Image Processing. 9(9) (2015) 347–357. https://doi.org/10.1049/iet-ipr.2014.0311
41. Dong Weidsheng, Zhang Lei, Nonlocally Centralized Sparse Representation for Image Restoration, IEEE Trans. Image Process. 4(22) (2013) 1620–1630. https://doi.org/10.1109/TIP.2012.2235847
42. Zhang Zhao, Jiang Weiming, Li Fanzhang, Zhao Mingbo, Bing Li, Zhang Li, Structured Latent Label Consistent Dictionary Learning for Salient Machine Faults Representation-Based Robust Classification, IEEE Trans. On Industrial Informatics. 13(2) (2017) 644–656. https://doi.org/10.1109/TII.2017.2653184
43. Zhang Zhao, Fanzhang Li, Chow Tommy WS, Zhang Li, Yan Shuicheng, Sparse Codes Auto-Extractor for Classification: A Joint Embedding and Dictionary Learning Framework for Representation, IEEE Trans. On Signal Processing. 64(14) (2016) 3790–3805. https://doi.org/10.1109/TSP.2016.2550016
44. Gu Shuhang, Zhang Lei, Zuo Wangmeng, Feng Xiangchu, Projective dictionary pair learning for pattern classification, Advances in Neural Information Processing Systems. (1) (2014) 793–801.

45. Feng Zhizhao, Yang Meng, Zhang Lei, Liu Yan, Zhang David, Joint discriminative dimensionality reduction and dictionary learning for face recognition, Pattern Recognition. 46(8) (2013) 2134–2143. https://doi.org/10.1016/j.patcog.2013.01.016

46. Jiang Weiming, Zhang Zhao, Fanzhang Li, Zhang Li, Zhao Mingbo, Jin Xiaohang, Joint Label Consistent Dictionary Learning and Adaptive Label Prediction for Semisupervised Machine Fault Classification, IEEE Trans. On Industrial Informatics. 12(1) (2016) 248–256. https://doi.org/10.1109/TII.2015.2496272

47. Timofte Radu, Rothe Rasmus, Luc Van Gool, Seven ways to improve example-based single image super resolution, CVPR. (2016).