# Image Restoration with Group Sparse Representation and Low-rank Group Residual Learning

Zhaoyuan Cai[1], Xianghua Xie[2], Jingjing Deng[3], Zengfa Dou[4], Bo Tong[5], Xiaoke Ma[1*]

[1] School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi, China
[2] Department of Computer Science, Swansea University, Swansea, United Kingdom
[3] Department of Computer Science, Durham University, Durham, United Kingdom
[4] 20th Research Institute, China Electronic Science and Technology Group Co., Ltd. Shaanxi Xi'an, China
[5] Xi'an Thermal Power Research Institute Co., Ltd., Xi'an, China
[*] corresponding: xkma@xidian.edu.cn

**Abstract: Image restoration, as a fundamental research topic of image processing, is to reconstruct the original image from degraded signal using the prior knowledge of image. Group sparse representation (GSR) is powerful for image restoration, it however often leads to undesirable sparse solutions in practice. In order to improve the quality of image restoration based on GSR, the sparsity residual model expects the representation learned from degraded images to be as close as possible to the true representation. In this article, a group residual learning based on low-rank self-representation is proposed to automatically estimate the true group sparse representation. It makes full use of the relation among patches and explores the subgroup structures within the same group, which makes the sparse residual model have better interpretation, furthermore, results in high-quality restored images. Extensive experimental results on two typical image restoration tasks (image denoising and deblocking) demonstrate that the proposed algorithm outperforms many other popular or state-of-the-art image restoration methods.**

**Key Words:** Image Restoration, Group Sparse Representation, Low-rank Self-Representation, Group Residual Learning.

## 1. Introduction

Image restoration is one of elementary task in image processing to reconstruct or recovery the original image from the degraded or corrupted signal [1]. It has been extensively studied, in general, can be formulated as

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X} + \boldsymbol{E} \tag{1}$$

where $\boldsymbol{X} \in \mathbb{R}^{h\times w}$, $\boldsymbol{Y} \in \mathbb{R}^{h\times w}$ and $\boldsymbol{E} \in \mathbb{R}^{h\times w}$ are the original, degraded and noise of image respectively, and $\boldsymbol{H}$ is a degradation operator. The restoration problem represented in Eq. (1) can differ greatly in terms of the degradation operator $\boldsymbol{H}$. For example, $\boldsymbol{H}$ as an identity matrix corresponds to image denoising [2], a diagonal masking corresponds to image inpainting [3], and a blurring operator corresponds to image deblurring [4].

Evidence demonstrates that image priors are the foundation for image restoration, including total variation (TV) [5-7], sparsity [2, 8], low-rank [9-11], and deep image prior [12-20]. Particularly, sparsity prior is considered as one of the most remarkable for natural images [2, 8, 21-24]. On the basis of the strategies for manipulating sparsity prior, current algorithms are roughly divided into two classes, i.e., patch- [2, 25, 26] and group-based approaches [27-29, 8, 22], where the former ones independently perform image restoration for each patch, and the latter ones execute restoration task for each group of patches.

In the past decades, patch-based image restoration has attracted great attention from researchers [2, 30], and those algorithms are devoted to identify the low-dimensional representation (also called patch code) under the assumption that each patch can be modeled with a linear combination of learned basis elements, known as dictionary [2]. The typical dictionary strategies are classified into two categories, i.e., analytic and learning ones, where former one includes discrete cosine transform (DCT), and wavelet and curvelet [31]. In comparison of the traditional analytic approach, dictionary learned from images is more adoptive and accurate since it can depict the local structure of images comprehensively. For example, the well-known dictionary learning method K-SVD [30] is of strong adaptability, which has been successfully applied to image denoising and other tasks [2, 30]. Furthermore, by imposing sparse constraint onto patch representation, patch-based sparse representation (PSR) achieves an excellent performance for image restoration, where each patch is represented with a linear combination of a few atoms of the learned dictionary.

However, those algorithms are criticized for independently learning the dictionary and representation for each patch, resulting in two significant limitations. First, the patch-based methods are computationally time-consuming, hampering its application for large-scale image dataset. Second, these algorithms only exploit the intrinsic structure of each patch but ignores the correlation among various patches [26, 32], namely non-local self-similarity (NSS). To address the named issues, group-based approaches, such as group sparse representation (GSR) [33, 27, 28], learn the sparse coding and dictionary from a group of similar patches, where the strong correlation amongst them can be captured.

Compared to patch-based methods, the GSR models [25, 26] achieves an outstanding performance in image restoration. For example, BM3D [26] performs collaborative filtering on groups of 3D patches. Mairal *et al.* [33] proposed learned simultaneous sparse coding (LSSC), which given a certain transform domain and simultaneously sparse encodes similar patches to enforce them have similar coefficients. Zhang *et al.* [27] proposed a GSR-based model for image restoration, which designs the self-adaptive dictionary for image patch group and solves sparse coding with $\ell_0$ minimization. Xu *et al.* [34] learned an NSS prior

for patch groups based on external image databases before image denoising, which can achieve excellent results when the distribution of external patch groups and target image patch groups is similar. To preserve the characteristics of the target image itself, a series of models combining internal and external priors are proposed in [35, 36]. To obtain the more correct sparsity solution under image restoration problem, Wang *et al.* [29] incorporated the nonconvex weighted $\ell_p$ minimization into GSR framework for image denoising. To avoid learning dictionary from patches of images, principal component analysis (PCA) is adopted to construct dictionary [27, 29]. Recently, Zha *et al.* [37] proposed low-rank guided GSR model, which utilizes low-rankness to guide dictionary learning. However, due to the degradation of the observed image, the sparse representation obtained by the above methods can't reconstruct the original image faithfully.

To tackle this problem, the residual models [8, 22] assume that the group of similar patches exist the truth representation, where the learned representation should not deviate from the truth one. In contrast to GSR, the residual model is much more difficult to train since it needs to estimate the truth representation, which is also the significant difference among various algorithms. For example, nonlocally centralized sparse representation model (NCSR) [8] utilizes the weighted average of group sparse representation to estimate the true sparse code, whereas NSSRC (for nonconvex structural sparsity residual constraint) [22] integrates structural sparse representation and non-convex sparsity residual constraint for the estimation of truth coding. In addition, the low-rank GSR model (LRGSC) [38] which estimates the true sparse code with low-rank prior is proved to be effective for image compressed sensing in [39].These methods significantly improve the accuracy of algorithms for image restoration, implying that residual model is promising for this issue.

Even though great efforts have been devoted to the residual model for image restoration, there still many unsolved problems. Firstly, the relation among patches of the same groups is neglected since they only depict the distance between the patches and the centers of groups, which decreases the performance and interpretability of patterns. Secondly, conventional residual models employ the weighted linear function to obtain true sparse representation, where the weights of patches are difficult to select since the relative importance between them and exemplars is hard to measure. Thirdly, the existing algorithms ignore the specificity of patches when estimating the truth representation since they assume that patches within the same group share the same truth representation.

To address these problems, a flexible and interpretable image restoration algorithm based on group sparse representation and residual learning is proposed in this paper. To make use of the relation among patches in the same group and remove the dependence on the exemplars, we define the estimate of the true sparse representation as the product of the learned sparse code and a weight matrix. To adaptively determine the weight matrix, we integrate the learning of weight matrix into self-representation learning, which estimates the weight matrix automatically. Each column of the weight matrix can be different, which enables preservation of the specificity for each patch. In addition, we impose the low-rank constraint on the weight matrix using the nuclear norm, which can further explore the sub-group structures of each patch group. Finally, the proposed algorithm jointly learns the sparse representation, sub-group structure and group residual by combining group sparse representation and residual learning. The experiments imply that the proposed method outperforms many mainstream SOTA methods of image restoration.

The following is a summary of this research's main contributions.

1. To enhance the quality of learned sparse representation, we propose a residual learning model based on self-representation, which makes full use of the relation among patches within the same group. The true sparse representation is estimated by the self-representation of the learned sparse representation, which improves the interpretability of the estimate.

2. To preserve the specificity of the patches within the same group well, we impose the low-rank constraint on the weight matrix using the nuclear norm, which can further capture the sub-group structures of each patch group.

3. The proposed algorithm jointly learns the patch representation, sub-group structure and group residual by combining group sparse representation and residual learning. In this case, the residual learning assists sparse representation to learn better patch representation, and the sub-group structure of each patch group can ensure that the learned representation preserves the specificity of patches.

4. The experimental results on image denoising and deblocking, imply that the proposed algorithm outperforms many popular image restoration baselines in various quality measurements.

The remaining sections of this article is arranged as follows. Section 2 introduces the preliminaries, Section 3 elaborates the proposed algorithm for image restoration in detail, Section 4 presents the experimental results, and conclusions are drawn in Section 5.

## 2. Preliminaries

In this section, we will present the notations and preliminaries that are going to be used for the rest of the paper.

### *2.1. Notations*

Let the bold upper, bold lower, and lower-case letters denote matrices, vectors, and scalars, respectively. Let $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ be a $n \times m$ matrix, and $\boldsymbol{x} \in \mathbb{R}^{d}$ be a vector with $d$ elements, respectively. $\boldsymbol{X}'$ is the transpose of matrix $\boldsymbol{X}$.

The Frobenius norm of matrix $\mathbf{X}$ is defined as

$$||\boldsymbol{X}|| = \sqrt{tr(\boldsymbol{X}'\boldsymbol{X})} = \sqrt{tr(\boldsymbol{X}\boldsymbol{X}')}, \tag{2}$$

where $tr(\boldsymbol{X})$ is the trace of matrix $\boldsymbol{X}$. $\ell_0$-norm of vector $\boldsymbol{x}$ is defined as the sum of non-zero elements in $\boldsymbol{x}$, i.e.,

$$||\boldsymbol{x}||_0 = \sum_i |x_i|^0. \tag{3}$$

$\ell_1$-norm of vector $\boldsymbol{x}$ is the sum of absolute values of elements in $\boldsymbol{x}$, i.e.,

$$||\boldsymbol{x}||_0 = \sum_i |x_i|. \quad (4)$$

$\ell_p$-norm $(0 < p < 1)$ of vector $\boldsymbol{x}$ is defined as

$$||\boldsymbol{x}||_p = (\sum_i |x_i|^p)^{1/p}. \quad (5)$$

$||\boldsymbol{X}||_0$, $||\boldsymbol{X}||_1$ and $||\boldsymbol{X}||_p$ denotes imposing $\ell_0$-norm, $\ell_1$-norm, and $\ell_p$-norm on each column of matrix $\boldsymbol{X}$, respectively.

Nuclear norm of matrix $\boldsymbol{X}$ is defined as

$$||\boldsymbol{X}||_* = \sum_{i=1}^{min(m,n)} |\lambda_i|, \quad (6)$$

where $\lambda_i$ is the $i$-th singular value of matrix $\boldsymbol{X}$.

### 2.2. Image Restoration

To simplify the model in Eq. (1), we set the degradation matrix $\boldsymbol{H}$ as the identity matrix. Then, given a degraded image $\boldsymbol{Y}$, image restoration is formulated as

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E}, \quad (7)$$

where $\boldsymbol{X} \in \mathbb{R}^{h \times w}$ and $\boldsymbol{E} \in \mathbb{R}^{h \times w}$ denote the original image and additive noise, respectively. Without loss of generality, image prior is denoted by $\theta$ and then maximum a posteriori (MAP) framework [8, 27, 40] is employed, i.e., a posteriori function of the form $log\, p\,(\boldsymbol{X}|\boldsymbol{Y}, \theta)$ is maximized

$$log\, p\,(\boldsymbol{X}|\boldsymbol{Y}) = log\, p\,(\boldsymbol{Y}|\boldsymbol{X}, \theta) + log\, p\,(\boldsymbol{X}|\theta). \quad (8)$$

The likelihood term is the Gaussian distribution [8]

$$p(\boldsymbol{Y}|\boldsymbol{X}, \theta) = \frac{1}{\sqrt{2\pi}\sigma_E} exp(-\frac{1}{2\sigma_E^2}||\boldsymbol{Y} - \boldsymbol{X}||^2), \quad (9)$$

where the $\sigma_E^2$ is the variance of noise. And then Eq. (8) is equal to

$$\min_{\boldsymbol{X}} \frac{1}{2}||\boldsymbol{Y} - \boldsymbol{X}||^2 + \sigma_E^2 \Theta(\boldsymbol{X}), \quad (10)$$

where $\Theta(\boldsymbol{X})$ is regularization term derived from prior $\theta$.

### 2.3. Sparse Representation

Given features $\boldsymbol{d}_1, \dots, \boldsymbol{d}_n$, representation learning for a vector $\boldsymbol{x}$ aims to obtain a linear function such that

$$\boldsymbol{x} \approx a_1\boldsymbol{d}_1 + \cdots + a_n\boldsymbol{d}_n, \quad (11)$$

where $a_i$ is the coefficient for feature $\boldsymbol{x}_i$.

Eq. (11) is solved by minimizing approximation, i.e.,

$$min\, \frac{1}{2}||\boldsymbol{x} - \boldsymbol{D}\boldsymbol{a}||^2, \quad (12)$$

where $D = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]$, and $\boldsymbol{a} = (a_1, \dots, a_n)'$, respectively.

The sparse representation learning expects most of coefficients are 0, where Eq. (12) is formulated as

$$min\, \frac{1}{2}||\boldsymbol{x} - \boldsymbol{D}\boldsymbol{a}||^2 + \alpha||\boldsymbol{a}||_0, \quad (13)$$

where $\alpha$ is a parameter.

Furthermore, extension for sparse representation learning is needed. When multiple objects involve, i.e., $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n]$, GSR simultaneously handles $n$ objects into an objective function, where Eq. (13) is re-written as

$$min\, \frac{1}{2}||\boldsymbol{X} - \boldsymbol{D}\boldsymbol{A}||^2 + \alpha||\boldsymbol{A}||_0, \quad (14)$$

where $||\boldsymbol{A}||_0$ is regularization item, denotes imposing $\ell_0$-norm on each column of $\boldsymbol{A}$.

There are various strategies for constructing sparsity, i.e., $\ell_1$-norm [41, 42], and $\ell_p$-norm $(0 < p < 1)$, to bridge $\ell_0$ and $\ell_1$ [43, 44].

## 3. Proposed Method

In this section, we will present the proposed method in detail, including restoration model, optimization, parameter selection, and discussion on its computational complexity.

The overview of the proposed algorithm is shown in Fig. 1, which consists of three major components, patch grouping, sparse representation learning and group residual learning. Patch grouping divides sub-blocks of the original images into different classes, where patches within the same groups are highly similar. Sparse representation learning projects each group of patches into a subspace that spanned by columns of dictionary matrix to obtain the representation of patches, whereas group residual learning aims to automatically learn the true code of patch groups.

### 3.1. Restoration Model

In the patch grouping block, like other GSR-based restoration models [33, 27, 28], a patch-matching based approach is utilized. Specifically, the degraded image $\boldsymbol{Y}$ is divided into patches, where the size $\sqrt{d} \times \sqrt{d}$ of patches varies with downstream applications. For each reference patch, the closest $m$ patches within window of $l \times l$ are selected as a group, where patches belonging to multiple groups are allowed. To ensure the quality of groups, the step size of selected reference patches is small, where window size is large. In general, we set step size of selection reference patches as 3 or 4, and that of windows as $25 \times 25$. By stacking pixels each reference patch is denoted as $\overline{\boldsymbol{y}}_i$, and the corresponding patch group is $\boldsymbol{Y}_i \in \mathbb{R}^{d \times m}$, where each column corresponds to a patch within the group.

In the sparse representation learning block, the most intuitive strategy is to project each group of patches into a subspace, where the low-dimensional representation of patches is obtained. Specifically, given patch group $\boldsymbol{Y}_i$, the low-dimensional representation of patches is learned by minimizing the approximation, which is formulated as

$$\mathcal{O}(\boldsymbol{Y}_i) = \frac{1}{2}||\boldsymbol{Y}_i - \boldsymbol{D}_i\boldsymbol{A}_i||^2, \quad (15)$$

where $\boldsymbol{D}_i \in \mathbb{R}^{d \times d}$ and $\boldsymbol{A}_i \in \mathbb{R}^{d \times m}$ denotes the dictionary and coefficient matrix of $\boldsymbol{Y}_i$, respectively. Sparse representation learning [30] expects the learned $\boldsymbol{A}_i$ is sparse, i.e., the most elements are 0, which improves computational efficiency and interpretability of solutions. By imposing $\ell_1$-norm constraint to coefficient matrix $\boldsymbol{A}_i$, Eq. (15) is reformulated as

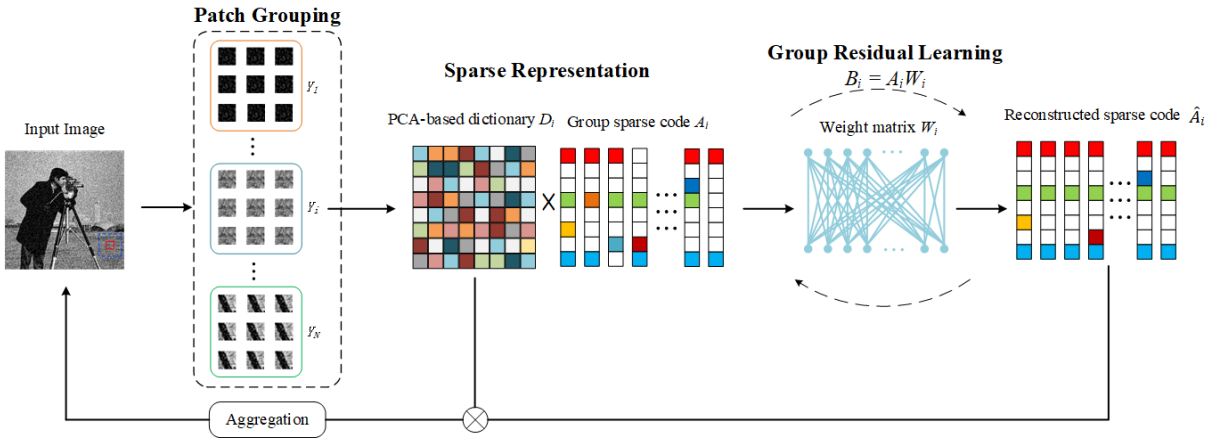


***Fig. 1.*** *The overview of our proposed image restoration algorithm, which consists of three major parts, namely patch grouping, sparse representation learning and group residual learning.*

$$\mathcal{O}(\boldsymbol{Y}_i) = \frac{1}{2}||\boldsymbol{Y}_i - \boldsymbol{D}_i\boldsymbol{A}_i||^2 + \beta||\boldsymbol{A}_i||_1, \quad (16)$$

where parameter $\beta$ determines the relative importance of sparsity constraint. Recently, evidence [43, 44] demonstrates that $\ell_p$-norm overcomes limitation of $\ell_1$-norm to fulfill sparsity of representation. Therefore, Eq. (16) is re-written as

$$\mathcal{O}(\boldsymbol{Y}_i) = \frac{1}{2}||\boldsymbol{Y}_i - \boldsymbol{D}_i\boldsymbol{A}_i||^2 + \beta||\boldsymbol{A}_i||_p. \quad (17)$$

However, the coefficient matrix $\boldsymbol{A}_i$ in Eq. (17) is very likely to deviate from the true representation of the corresponding group of patches, due to various degradations (noise, compression, etc.) on $\boldsymbol{Y}_i$. Thus, the residual model [8, 22] enforces the learned representation $\boldsymbol{A}_i$ to be in consistent with the truth representation of $\boldsymbol{Y}_i$. Given the truth representation $\boldsymbol{B}_i \in \mathbb{R}^{d\times m}$ of patch group $\boldsymbol{Y}_i$, the residual is defined as

$$\boldsymbol{R}_i = \boldsymbol{A}_i - \boldsymbol{B}_i. \quad (18)$$

Obviously, the learned sparse representation $\boldsymbol{A}_i$ should be as close as possible to the truth sparse representation $\boldsymbol{B}_i$, that is, $\boldsymbol{R}_i$ should be as small as possible. Thus, Eq. (17) can be extended as

$$\mathcal{O}(\boldsymbol{Y}_i) = \frac{1}{2}||\boldsymbol{Y}_i - \boldsymbol{D}_i\boldsymbol{A}_i||^2 + \beta||\boldsymbol{A}_i||_p + \frac{\gamma}{2}||\boldsymbol{A}_i - \boldsymbol{B}_i||^2, \quad (19)$$

where $\gamma$ determines importance of residual.

The typical advantage of the residual models in Eq. (19) is that it learns the low-dimensional representation of patches by preserving the truth distribution of patch group. However, solving the Eq. (19) is a great challenge. In other words, the dictionary matrix $\boldsymbol{D}_i$ and truth representation $\boldsymbol{B}_i$ are pre-requisites. Usually, principal component analysis (PCA) [27, 8] is a widely adopted strategy to construct dictionary $\boldsymbol{D}_i$. Specifically, each column of the dictionary $\boldsymbol{D}_i$ consists of the eigenvectors of the $\boldsymbol{Y}_i$'s covariance matrix. However, the truth sparse representation $\boldsymbol{B}_i$ is difficult to estimate, due to we always only have degraded images without ground truth.

To address this problem, existing algorithms assume that patches within each group share the common representation that can be learned from coefficient matrix $\boldsymbol{A}_i$. The most intuitive and straightforward strategy is to combine the representation of patches with a linear function to obtain the shared truth representation $\boldsymbol{b} \in \mathbb{R}^d$ as [8, 22]

$$\boldsymbol{b} = \frac{1}{m}\sum_{j=1}^{m} w_j \boldsymbol{A}_{i,j}, \quad (20)$$

where $w_i$ is the weight for patch $\boldsymbol{Y}_i$ as Euclidean distance between the $i$-th patch and the center of group, and $\boldsymbol{A}_{i,j}$ denotes the $j$-th column of $\boldsymbol{A}_i$, respectively. Then, the truth representation set as $\boldsymbol{B}_i = [\boldsymbol{b}, \dots, \boldsymbol{b}]$.

In comparison to the sparse representation model, the residual model of Eq. (19) and Eq. (20) is more effective and efficient for image restoration. However, it also has three limitations on the estimation of $\boldsymbol{B}_i$. First of all, it assumes that patches within the same group share the same truth representation, ignoring the specificity of patches. In other words, Eq. (20) achieves the desirable performance if and only if patches within the same group are homogeneous. Actually, patches groups can be further divided into sub-groups, where each sub-group has a unique representation, indicating that there are multiple representations for the original patch group. Second, the relation among patches of the same groups is also neglected since Eq. (20) only depicts the distance between patches and centers of groups decreasing the performance and interpretability of patterns. Furthermore, Eq. (20) employs the weighted linear function to obtain $\boldsymbol{B}_i$, where weights of patches are difficult to select since the relative importance between them and exemplars is hard to measure.

To overcome these issues, we hypothesize that, for each patch group $\boldsymbol{Y}_i$, there is a close relation between the learned and truth representation of patches, i.e., $\boldsymbol{B}_i$ can be learned from $\boldsymbol{A}_i$. Specifically, the proposed algorithm automatically learns $\boldsymbol{B}_i$ from $\boldsymbol{A}_i$ as

$$\boldsymbol{B}_i = \boldsymbol{A}_i\boldsymbol{W}_i, \quad (21)$$

where the $\boldsymbol{W}_i \in \mathbb{R}^{m\times m}$ is a weight matrix. Eq. (21) avoids sharing the unique center in the residual model in Eq. (20), where multiple centers are allowed. In this case, the relations among patches within the same groups are explicitly exploited, which provides a better strategy to characterize and model the truth representation of patches.

Substituting Eq. (21) into Eq. (19), we have

$$\mathcal{O}(\boldsymbol{Y}_i) = \frac{1}{2}||\boldsymbol{Y}_i - \boldsymbol{D}_i\boldsymbol{A}_i||^2 + \beta||\boldsymbol{A}_i||_p + \frac{\gamma}{2}||\boldsymbol{A}_i - \boldsymbol{A}_i\boldsymbol{W}_i||^2. \quad (22)$$

Eq. (22) brings out three consequent advantages. First, the last term corresponds to the self-representation of learned representation $\boldsymbol{A}_i$, i.e., $\boldsymbol{B}_i$ is represented with linear combination of close patches of groups, improving interpretability of $\boldsymbol{B}_i$. Second, the relation among patches is further exploited for learning the low-dimensional representation for patches, thereby, enhancing quality of representation in return. Third, the weight matrix $\boldsymbol{W}_i$ is automatically learned under the guidance of the representation of patches, extending the application of models.

Moreover, we also want to explore sub-groups of each patch group with an immediate purpose to further exploit intrinsic structure of representation. Evidence [45] demonstrates the nuclear-norm of matrix ensures the block structure of matrices. Therefore, we fulfil the sub-groups of patch group by imposing constraint on $\boldsymbol{W}_i$ with nuclear norm, then Eq. (22) can be reformulated as

$$\begin{aligned}\mathcal{O}(\mathbf{Y}_i) = &\tfrac{1}{2}||\mathbf{Y}_i - \mathbf{D}_i\mathbf{A}_i||^2 + \beta||\mathbf{A}_i||_p \\ &+\tfrac{\gamma}{2}||\mathbf{A}_i - \mathbf{A}_i\mathbf{W}_i||^2 + \tau||\mathbf{W}_i||_* \\ &s.t.\quad \mathbf{W}_i = \mathbf{W'}_i.\end{aligned} \tag{23}$$

where $\tau$ determines the importance of low-rank constraint. The constraint ensures the symmetry of matrix $\boldsymbol{W}_i$, because each element of $\boldsymbol{W}_i$ can be regarded as similarity between corresponding sparse representations. Eq. (24) jointly learns the representation of patches, sub-group structure, and residuals for each patch groups. By summing all patch groups, the proposed algorithm learns representation of patches that is formulated as

$$\mathcal{O}(\boldsymbol{Y}) = \textstyle\sum_i \mathcal{O}(\boldsymbol{Y}_i). \tag{25}$$

To restore the degraded image, the main model is to minimize the difference between the degraded and restored images, i.e., $||\boldsymbol{Y} - \boldsymbol{X}||$. Thus, the overall objective function of the proposed algorithm with the prior regulation terms mentioned above can formulated as

$$\begin{aligned}\mathcal{O} = &\tfrac{1}{2}||\mathbf{Y} - \mathbf{X}||^2 + \mathcal{O}(\mathbf{X}) \\ = &\tfrac{1}{2}||\mathbf{Y} - \mathbf{X}||^2 + \tfrac{\alpha}{2}\textstyle\sum_i||\mathbf{Q_i}\mathbf{X} - \mathbf{D_i}\mathbf{A_i}||^2 + \beta\sum_i||\mathbf{A_i}||_p \\ &+\tfrac{\gamma}{2}\textstyle\sum_i||\mathbf{A_i} - \mathbf{A_i}\mathbf{W_i}||^2 + \tau\sum_i||\mathbf{W_i}||_* \\ &s.t.\quad \mathbf{W}_i = \mathbf{W'_i},\ \ \forall i \in \{1,2,\dots,n\},\end{aligned} \tag{26}$$

where $\boldsymbol{Q}_i$ represents the matrix operator for the extraction of the $i$-th patch group for $\boldsymbol{X}$, i.e., $\boldsymbol{Q}_i\boldsymbol{X} = \boldsymbol{X}_i$. In the next subsection, we will discuss the optimization technique for the objective function in Eq. (26).

### 3.2. Optimization

The nuclear norm of $\boldsymbol{W}_i$ and $\ell_p$ norm of $\boldsymbol{A}_i$ result in the non-convexity of problem in Eq. (26), which cannot be solved directly with analytical solutions. Thus, an alternative iterative strategy is adopted by optimizing one variable by fixing the others until the algorithm converges or termination criteria are reached.

*Step 1. Update weight matrix $\boldsymbol{W}_i$:* By fixing $\boldsymbol{X}$ and $\boldsymbol{A}_i$, and removing irrelevant items, the objective function of Eq. (26) is equivalent to the following one with respect to $\boldsymbol{W}_i$, i.e.,

$$\tfrac{\gamma}{2}||\boldsymbol{A}_i - \boldsymbol{A}_i\boldsymbol{W}_i||^2 + \tau||\boldsymbol{W}_i||_* \ \ s.t. \mathbf{W}_i = \boldsymbol{W'}_i. \tag{27}$$

According to [46], Eq. (27) can be effectively solved with singular value decomposition (SVD) of matrix $\boldsymbol{A}_i$, i.e., $\boldsymbol{A}_i = \boldsymbol{U}_i\Lambda_i\boldsymbol{V'}_i$, where $\Lambda_i = diag(\{\lambda_i\})$ is a diagonal matrix with singular values on the diagonal, the $\boldsymbol{U}_i$ and $\boldsymbol{V}_i$ are the left and right singular vector matrix, respectively. The optimal solution to Eq. (27) is formulated as [46]

$$\widehat{\boldsymbol{W}}_i = \boldsymbol{V}_i\mathcal{P}_{\tau/\gamma}(\lambda_i)\boldsymbol{V'}_i, \tag{28}$$

where $\mathcal{P}_{\tau/\gamma}$ is defined as

$$\mathcal{P}_{\tau/\gamma}(\lambda_i) = \begin{cases}1 - \frac{\tau}{\gamma\lambda_i^2}, & \lambda_i > \sqrt{\tau/\gamma}, \\ 0, & otherwise.\end{cases} \tag{29}$$

*Step 2. Update group sparse representation $\boldsymbol{A}_i$:* By removing other items, the objective function of Eq. (26) in terms of $\boldsymbol{A}_i$ is reformulated as

$$\tfrac{\alpha}{2}||\boldsymbol{X}_i - \boldsymbol{D}_i\boldsymbol{A}_i||^2 + \tfrac{\gamma}{2}||\boldsymbol{A}_i - \boldsymbol{A}_i\boldsymbol{W}_i||^2 + \beta||\boldsymbol{A}_i||_p. \tag{30}$$

Even though $\ell_p$-norm of $\boldsymbol{A}_i$ leads to non-convexity of Eq. (30), the generalized soft-thresholding (GST) algorithm [47] provides an efficiently iterative strategy to achieve desired solution. Specifically, the update rule for $\boldsymbol{A}_i$ is formulated as

$$\boldsymbol{A}_i = GST(\boldsymbol{S}_i, \mu, p, t), \tag{31}$$

where $t$ is the number of iterations, $\boldsymbol{S}_i$ and $\mu$ are defined as

$$\begin{cases}\boldsymbol{S}_i = \alpha(\boldsymbol{A}_i - \boldsymbol{D}_i{}^T\boldsymbol{X}_i) + \gamma\boldsymbol{A}_i(\boldsymbol{I} - \boldsymbol{W}_i - \boldsymbol{W'}_i + \boldsymbol{W}_i\boldsymbol{W'}_i), \\ \mu = \frac{\beta}{\alpha + \gamma\|\boldsymbol{I} - \boldsymbol{W}_i\|^2}.\end{cases}$$

*Step 3. Update restored image $\boldsymbol{X}$:* By fixing matrix $\boldsymbol{A}_i$ and $\boldsymbol{W}_i$, the restoration of images is formulated as

$$\tfrac{1}{2}||\boldsymbol{Y} - \boldsymbol{X}||^2 + \tfrac{\alpha}{2}\textstyle\sum_i||\boldsymbol{Q}_i\boldsymbol{X} - \boldsymbol{D}_i\boldsymbol{A}_i||^2. \tag{32}$$

Since Eq. (32) is convex with respect to $\boldsymbol{X}$. By setting the partial derivative of $\boldsymbol{X}$ to zero, the closed-form solution for Eq. (32) is deduced as

$$\widehat{\boldsymbol{X}} = (\boldsymbol{I} + \alpha\textstyle\sum_i \boldsymbol{Q'}_i\boldsymbol{Q}_i)^{-1}(\boldsymbol{Y} + \alpha\textstyle\sum_i \boldsymbol{Q'}_i\boldsymbol{D}_i\boldsymbol{A}_i), \tag{33}$$

where $\boldsymbol{D}_i\boldsymbol{A}_i$ denotes the reconstructed patch for $\widehat{\boldsymbol{X}}_i$, $\boldsymbol{Q'}_i$ is treated as the operator to put the restored patch back into the original image. Actually, $(\boldsymbol{I} + \alpha\sum_i \boldsymbol{Q'}_i\boldsymbol{Q}_i)$ is a diagonal matrix, whose inverse can be solved at the element-wise division.

Therefore, Eq. (33) is interpreted as the weighted average of overlapped reconstructed patches and degraded image to reconstruct the image.

### 3.3. Parameter Selection

In order to achieve optimal performance results, we also employ the adaptive parameter adjustment strategy to adapt the proposed algorithm to various image structures. Firstly, we update the noise variance $\sigma_E^2$ using the iterative regularization strategy [6]

$$\sigma_E^{(k)} = c_0\sqrt{\left(\sigma_E^2 - \left\|\boldsymbol{Y} - \widehat{\boldsymbol{X}}^{(k)}\right\|^2\right)}, \tag{34}$$

where $k$ denotes the $k$-th iteration and $c_0$ is a positive constant.

Furthermore, inspired by the maximum a posteriori (MAP) framework [8], we assume that sparse code $\boldsymbol{A}_i$ obeys

a hyper-Laplacian distribution [29, 48] and the sparsity residual $\boldsymbol{R}_i$ follows a Gaussian distribution, then we have

$$\beta = \frac{\sigma_E^2}{\delta_i + \epsilon}, \tag{35}$$

$$\gamma = \frac{\sigma_E^2}{\delta_i^2 + \epsilon}, \tag{36}$$

where $\delta_i$ is the standard variance of $\boldsymbol{A}_i$, estimated as in [10], and U is a very small constant to avoid dividing by zero.

We also dynamically adjust the parameters $\alpha$ and $\tau$ as

$$\alpha = c_1 \sigma_E^2, \tag{37}$$

$$\tau = c_2 \delta_i, \tag{38}$$

where $c_1$ and $c_2$ are predefined positive constants. It means that the parameters $\alpha$ and $\tau$ are proportional to the noise variance $\sigma_E^2$ and standard variance $\delta_i$ of $\boldsymbol{A}_i$, respectively.

### 3.4. Method Overview

In summary, the proposed algorithm for image restoration can be implemented through the above alternating updating steps and parameter adjustment mechanism. The pseudocode of our proposed algorithm can be found in Algorithm 1.

### 3.5. Computational Complexity Analysis

We further analyze the computational complexity of our proposed method in theory in this subsection. For the space complexity of the proposed algorithm, it requires space $(m^2 n)$. The space for matrix $\boldsymbol{W}_i$ is $O(m^2)$ for each group. The space for matrix $\boldsymbol{A}_i$ and $\boldsymbol{X}_i$ is $O(dm)$ for each group, where $d$ is the number of rows of $\boldsymbol{A}_i$ and $\boldsymbol{X}_i$. Therefore, the total space complexity of our proposed algorithm is $O(m^2 n)$, where $n$ is the number of groups. For the time complexity analysis of our proposed algorithm, it consists of three major components, that is: 1) group residual learning, 2) sparse representation and 3) reconstruction. The time complexity for updating $\boldsymbol{W}_i$ is $O(tnm^3)$, where $t$ is the number of iterations. The time complexity for updating $\boldsymbol{A}_i$ is $O(tndm)$. The time complexity for group reconstruction is $O(tnb^2 m)$. Thus, the total time complexity of the proposed algorithm is $O(tnm^3)$.

**Algorithm 1** The proposed algorithm for image restoration.

**Input:** The degraded image $\boldsymbol{Y}$.
**Output:** The restored image $\widehat{\boldsymbol{X}}$.

Initialize $\widehat{\boldsymbol{X}}^{(0)} = \boldsymbol{Y}$, $k = 0$, $\sigma_E^{(0)}$.
Set the parameters $c_0$, $c_1$, $c_2$ and $p$.
**while** ($k$ <= Max-Iter) **do**
  **for** each reference patch $\boldsymbol{x}_i$ in $\widehat{\boldsymbol{X}}^{(k)}$ **do**
    Search similar patches to construct patch group $\boldsymbol{X}_i$.
    Build dictionary $\boldsymbol{D}_i$ by $\boldsymbol{X}_i$ using PCA.
    Update $\boldsymbol{A}_i$ by $\boldsymbol{A}_i = \boldsymbol{D}'_i \boldsymbol{X}_i$.
    Update $\tau$ by Eq. (38).
    Update $\boldsymbol{W}_i$ by Eq. (28).
    Update $\alpha$, $\beta$ and $\gamma$ by Eqs. (37), (35) and (36).
    Update $\boldsymbol{A}_i$ by Eq. (31).
  **end for**
  Update $\widehat{\boldsymbol{X}}^{(k)}$ by Eq. (33).
  Update $\sigma_E^{(k)}$ by Eq. (34).
  Until the convergence condition is met.
**end while**

## 4. Experimental Results

To fully validate the performance of the proposed algorithms, extensive experiments are conducted on two typical image restoration tasks: denoising and deblocking. Please find the source code for this research at https://github.com/xkmaxidian/GSR_SRLR_IR .

### 4.1. Experimental Setting

*1) Benchmark:* As the proposed algorithm is self-supervised learning, only testing data is needed to verify the proposed algorithm performance.

For image denoising task, the benchmark dataset Set12 [49] (including 12 grayscale images) is used to validate the performance with comparisons. The noisy images are synthesized by adding Gaussian noise given a noise level.

For image deblocking, we adopt two widely used datasets: LIVE1 dataset [50] and Classic5 dataset [51], including 29 and 5 natural images, respectively, Compress images are obtained by encoding each test image with the MATLAB jpeg encoder under different compression quality $Q$ and then the JPEG deblocking input images can be generated by a standard JPEG decoder. In addition to LIVE1 and Classic5, we also use 8 fingerprint images to further verify the superiority of the proposed algorithm.

*2) Parameter Setting:* For image denoising, the parameter settings of the proposed algorithm are as follows. The patch size is set to $7 \times 7$, $8 \times 8$ and $9 \times 9$ for $\sigma_E \leq 30$, $30 < \sigma_E \leq 50$ and $50 < \sigma_E \leq 100$. The number of similar patches in a group is set to 60, 70, 80, 90, 100 for $\sigma_E \leq 30$, $30 < \sigma_E \leq 40$, $40 < \sigma_E \leq 50$, $50 < \sigma_E \leq 75$ and $75 < \sigma_E \leq 100$. The parameter $p$ is set to 0.8, 0.85 and 0.9 for $\sigma_E \leq 30$, $30 < \sigma_E \leq 40$ and $40 < \sigma_E \leq 100$.

For image deblocking, the patch size is set to $7 \times 7$. The number of similar patches in a group is set to 60. The parameter $p$ is set to 0.9, 0.8 and 0.2 for $Q \leq 10$, $10 < Q \leq 20$ and $20 < Q \leq 40$.

### 4.2. Compared Methods

In image denoising task, we first compared the proposed algorithm with many SOTA image denoising baselines, including BM3D [26], EPLL [52], NCSR [8], PGPD [34], aGMM [53], AST-NLS [54], GSRC [55], NSSRC [22] and LGSR [37]. These methods are based on sparse representation or image non-local self-similarity prior. Among these methods, NCSR, AST-NLS, GSRC and NSSRC are models based on group sparsity residual, and NSSRC is the SOTA of these methods. In addition, we also compared our proposed algorithm with several deep learning image denoising models, including TRND [56], DnCNN [49], S2S [57], where TRND and DnCNN are supervised learning-based image denoising benchmark algorithms, and S2S is self-supervised learning algorithm.

For image deblocking, we compared our proposed method against BM3D [26], SA-DCT [58], PC-LRM [59], WNNM [60], ANCE [61], SSR-QC [62], COGL [63], JPG-SR [64], NSSRC [22]. To further demonstrate the superiority of our proposed method, we also compared it with several DNN-based image deblocking baselines, including AR-CNN [65], TRND [56], DnCNN [49], DCSC [66], and MDDU (for model-driven deep unfolding method) [67]. Among these competing methods, AR-CNN is the most common deep learning model benchmark for compressed image artifacts removal, the TRND and DnCNN are universal image restoration methods, and the DCSC and MDDU are SOTA image deblocking methods.

It should be emphasized that all comparison methods were tested using the default parameters set by the original authors. In particular, deep learning-based methods were tested using the official pre-training model.

### 4.3. Image Denoising

Image denoising is the most common and basic image restoration task. In order to verify the effectiveness of our proposed model for image denoising, we use MATLAB's random number generator to synthesize Gaussian White Noise (GWN) images for testing. In addition, we also selected some real images for denoising.

*1) Comparison with Classic Image Denoising Methods:* We evaluated all the comparison classic denoising methods using six noise levels: $\sigma_E = 20, 30, 40, 50, 75,$ and 100. To quantify the effectiveness of the algorithms, we employ two quality measurements for restored image: PSNR and structural similarity (SSIM) [68]. The image denoising average results on dataset Set12 are summarized in Table 1 (PSNR) and Table 2 (SSIM). The best results are highlighted in bold and it is obvious that our proposed algorithm outperforms all other competing methods on both PSNR and SSIM. It is worth noting that our method achieves better performance than other sparsity residual-based methods, especially the NSSRC, where the rationale is two folds. On one hand, the proposed residual learning considers the relation among patches, where the quality of sparse representation in return is enhanced. On the other hand, the weight matrix automatically learned with the low-rank constraint can explore sub-groups of each patch group, which enables the sparse representation learned to preserve the specificity of each patch within the same group.

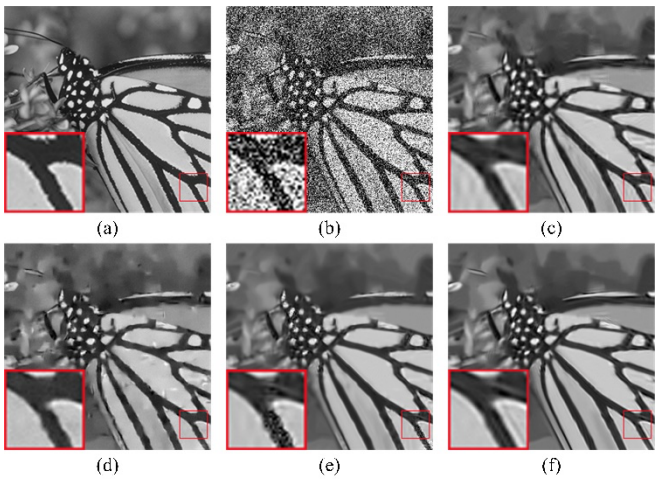


**Fig. 2** The visual comparison denoising results of image Monarch on the Set12 dataset with $\boldsymbol{\sigma_E = 75}$. (a) Original image. (b) Noisy image. (c) BM3D (PSNR = 23.91 dB, SSIM = 0.7557). (d) EPLL (PSNR = 23.72 dB, SSIM = 0.7396). (e) NCSR (PSNR = 23.68 dB, SSIM = 0.7657). (f) **Ours (PSNR = 24.38 dB, SSIM = 0.7790).**

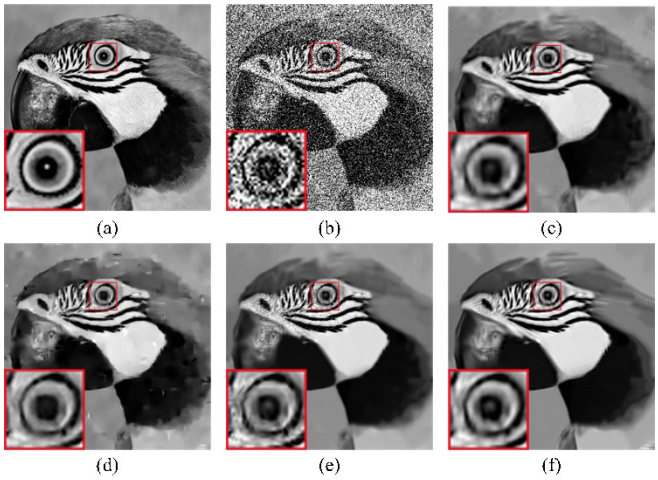


**Fig. 3** The visual comparison denoising results of image *Parrot* on the Set12 dataset with $\boldsymbol{\sigma_E = 75}$. (a) Original image. (b) Noisy image. (c) BM3D (PSNR = 24.19 dB, SSIM = 0.7307). (d) EPLL (PSNR = 24.04 dB, SSIM = 0.7029). (e) NCSR (PSNR = 23.89 dB, SSIM = 0.7363). (f) **Ours (PSNR = 24.48 dB, SSIM = 0.7469).**

**Table 1** Average PSNR (dB) Results of Comparison with Classic Methods for Image Denoising on the Set12 Dataset

| $\sigma_E$ | BM3D | EPLL | NCSR | PGPD | aGMM | AST-NLS | GSRC | NSSRC | LGSR | Ours |
|---|---|---|---|---|---|---|---|---|---|---|
| **20** | 31.01 | 30.70 | 31.02 | 31.02 | 31.02 | 31.16 | 30.93 | 31.19 | 31.27 | **31.28** |
| **30** | 29.14 | 28.75 | 29.04 | 29.12 | 29.08 | 29.22 | 29.09 | 29.28 | 29.36 | **29.40** |
| **40** | 27.65 | 27.40 | 27.65 | 27.82 | 27.67 | 27.81 | 27.80 | 27.97 | 28.04 | **28.07** |
| **50** | 26.72 | 26.35 | 26.60 | 26.81 | 26.62 | 26.86 | 26.81 | 27.00 | 27.00 | **27.09** |
| **75** | 24.91 | 24.48 | 24.65 | 24.98 | 24.67 | 24.98 | 25.04 | 25.15 | 25.25 | **25.28** |
| **100** | 23.61 | 23.21 | 23.29 | 23.69 | 23.37 | 23.71 | 23.67 | 23.84 | 23.95 | **23.97** |
| **Average** | 27.17 | 26.82 | 27.04 | 27.24 | 27.07 | 27.29 | 27.22 | 27.41 | 27.48 | **27.52** |

**Table 2** Average SSIM Results of Comparison with Classic Methods for Image Denoising on the Set12 Dataset

| $\sigma_E$ | BM3D | EPLL | NCSR | PGPD | aGMM | AST-NLS | GSRC | NSSRC | LGSR | Ours |
|---|---|---|---|---|---|---|---|---|---|---|
| **20** | 0.8719 | 0.8681 | 0.8711 | 0.8674 | 0.8716 | 0.8705 | 0.8660 | 0.8736 | 0.8738 | **0.8740** |
| **30** | 0.8320 | 0.8223 | 0.8308 | 0.8286 | 0.8290 | 0.8260 | 0.8279 | 0.8339 | 0.8347 | **0.8350** |
| **40** | 0.7944 | 0.7827 | 0.7951 | 0.7965 | 0.7909 | 0.7847 | 0.7965 | 0.8030 | 0.8044 | **0.8051** |
| **50** | 0.7681 | 0.7475 | 0.7673 | 0.7666 | 0.7569 | 0.7618 | 0.7664 | 0.7780 | **0.7798** | 0.7784 |
| **75** | 0.7065 | 0.6738 | 0.7095 | 0.7070 | 0.6832 | 0.7003 | 0.7101 | 0.7219 | 0.7236 | **0.7248** |
| **100** | 0.6566 | 0.6154 | 0.6642 | 0.6525 | 0.6230 | 0.6578 | 0.6575 | 0.6807 | 0.6821 | **0.6827** |
| **Average** | 0.7716 | 0.7516 | 0.7730 | 0.7698 | 0.7591 | 0.7669 | 0.7707 | 0.7819 | 0.7831 | **0.7833** |

For further intuitive demonstration, we have made visual comparison with three representative methods: BM3D [26], EPLL [52] and NCSR [8]. BM3D is based on collaborative filtering after block matching, EPLL is based on statistical model (GMM), and NCSR is based on sparse residual model. The visual results on images Monarch and Parrot in Set12 with $\sigma_E = 75$ are presented in Fig. 2 and Fig. 3, respectively. It can be seen that the images restored by BM3D are blurred or oversmoothed, and the images restored by EPLL and NCSR suffer from undesired visual artifacts. The proposed method not only reduces the noise more effectively than other methods, but also preserves many image details like edges and textures.

*2) Comparison with DNN-based Image Denoising Models:* Deep neural networks (DNN) have achieved great success in various downstream tasks for both high level image understanding and low-level image processing. We therefore compared the proposed algorithm with several mainstream DNN-based image denoising models, including TRND [56], DnCNN [49], S2S [57]. The PSNR and SSIM results on the Set12 dataset are summarized in Table 3.

Overall, the results show that our proposed method achieves better results even competing with DNN-based methods in image denoising tasks. Selected qualitative results with $\sigma_E = 50$ are illustrated in Fig. 4 and Fig. 5, in comparison with TRND, DnCNN and S2S. It can be seen that the deep learning-based methods are prone to produce some artifacts or to be over-smoothing, while our method can restore image with more clearly details. Therefore, we conclude that although supervised deep learning can fit diverse image structures with the large training dataset, they hardly to capture the prior of the image itself, such as sparsity and NSS, while our algorithm can make good use of these priors.

To fully verify the effectiveness of our proposed algorithm, we also selected two real noisy images for experiments. Since our model requires the noise variance of noisy image as a priori, we adopt a fast noise estimation method [69] to obtain the noise variance of the real image in advance. The denoising results are shown in Fig. 6, Fig. 7 and Fig. 8. We compare our method with the deep learning-based S2S [57] which is also a self-supervised model. It can be clearly seen that the restoration result of S2S is over-smoothed, while our method preserved more image details.

**Table 3** Average PSNR (dB) / SSIM Results of Comparison with DNN-based Models for Image Denoising on the Set12 Dataset.

| **Methods** | $\sigma_E = 15$ | $\sigma_E = 25$ | $\sigma_E = 50$ | **Average** |
|---|---|---|---|---|
| TRND | 32.51 | 30.04 | 26.78 | 29.78 |
| | 0.8970 | 0.8523 | 0.7672 | 0.8388 |
| DnCNN | 32.50 | 30.17 | 26.98 | 29.88 |
| | 0.8966 | **0.8549** | 0.7700 | 0.8405 |
| S2S | 32.07 | 29.94 | 26.12 | 29.38 |
| | 0.8891 | 0.8475 | 0.7382 | 0.8249 |
| **Ours** | **32.65** | **30.24** | **27.09** | **29.99** |
| | **0.8980** | 0.8535 | **0.7784** | **0.8433** |

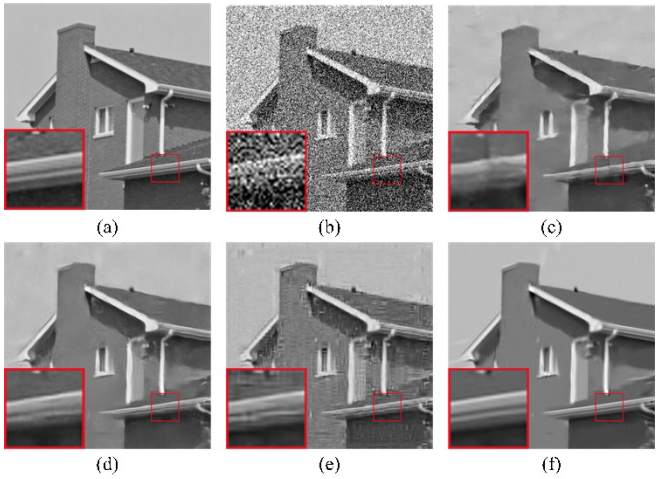


**Fig. 4** The visual comparison denoising results of image *House* on the Set12 dataset with $\boldsymbol{\sigma_E = 50}$. (a) Original image. (b) Noisy image. (c) TRND (PSNR = 29.40 dB, SSIM = 0.8058). (d) DnCNN (PSNR = 29.74 dB, SSIM = 0.8059). (e) S2S (PSNR = 27.47 dB, SSIM = 0.7032). (f) **Ours (PSNR = 30.36 dB, SSIM = 0.8221).**

**Fig. 5** The visual comparison denoising results of image *Barbara* on the Set12 dataset with $\boldsymbol{\sigma_E = 50}$. (a) Original image. (b) Noisy image. (c) TRND (PSNR = 25.78 dB, SSIM = 0.7450). (d) DnCNN (PSNR = 25.53 dB, SSIM = 0.7361). (e) S2S (PSNR = 26.82 dB, SSIM = 0.7840). (f) **Ours (PSNR = 27.92 dB, SSIM = 0.8231).**

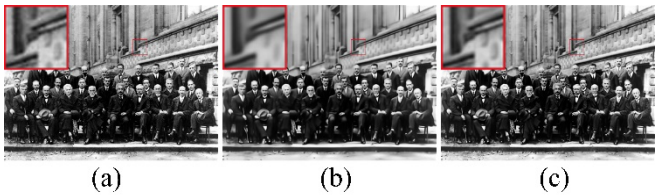

**Fig. 6** The visual results of real image 1 denoising. (a) Real image 1. (b) The result of S2S. (c) The result of our method.

**Fig. 7** The visual results of real image 2 denoising. (a) Real image 2. (b) The result of S2S. (c) The result of our method.

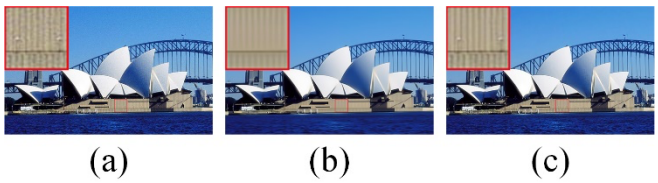

**Fig. 8** The visual results of real image 3 denoising. (a) Real image 2. (b) The result of S2S. (c) The result of our method.

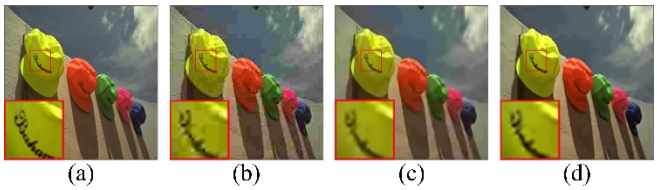

**Fig. 9** The visual comparison deblocking results of image *Caps* on the LIVE1(image size: 256×256 ) dataset at $\boldsymbol{Q = 10}$. (a) Original image. (b) JPEG compressed image (PSNR = 30.75 dB, SSIM = 0.8232). (c) SA-DCT (PSNR = 31.24 dB, SSIM = 0.8634). (d) **Ours (PSNR = 31.70 dB, SSIM = 0.8734).**

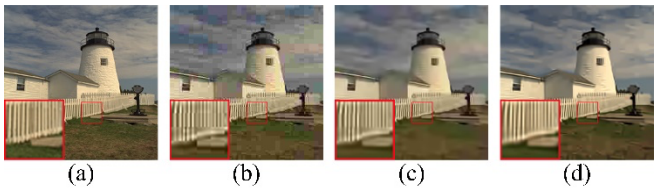

**Fig. 10** The visual comparison deblocking results of image *LightHouse3* on the LIVE1(image size: 256×256 ) dataset at $\boldsymbol{Q = 10}$. (a) Original image. (b) JPEG compressed image (PSNR = 27.52 dB, SSIM = 0.7561). (c) SA-DCT (PSNR = 28.33 dB, SSIM = 0.7786). (d) **Ours (PSNR = 28.78 dB, SSIM = 0.7951).**

### *4.4. Image Deblocking*

To further comprehensively verify the effectiveness of our proposed algorithm, we investigated the JPEG deblocking problem [58, 62, 65], reducing the block artifacts in the JPEG compressed images. Different to image denoising, the additive noise $E$ is quantization noise in image deblocking. Thus, we employ a classical Gaussian model [58] which characterizes the noise quantization by estimating the noise standard variance $\sigma_E$.

**Table 4** Average PSNR (dB) Results of Comparison with Classic Methods for Image Deblocking on the LIVE1 Dataset (image size: $\boldsymbol{256 \times 256}$) and the Classic5 Dataset (image size: $\boldsymbol{256 \times 256}$).

| **LIVE1 dataset** (image size: 256 × 256) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Q | **JPEG** | **BM3D** | **SA-DCT** | **PC-LRM** | **ANCE** | **WNNM** | **SSR-QC** | **COGL** | **JPG-SR** | **NSSRC** | **Ours** |
| **10** | 26.37 | 27.16 | 27.23 | 27.24 | 27.24 | 27.25 | 27.26 | 27.38 | 27.29 | 27.43 | **27.45** |
| **20** | 28.55 | 29.21 | 29.24 | 29.28 | 29.29 | 29.29 | 29.33 | 29.46 | 29.37 | 29.53 | **29.54** |
| **30** | 29.86 | 30.45 | 30.48 | 30.54 | 30.57 | 30.55 | 30.60 | 30.74 | 30.75 | 30.85 | **30.87** |
| **40** | 30.80 | 31.35 | 31.37 | 31.45 | 31.51 | 31.46 | 31.57 | 31.66 | 31.71 | 31.82 | **31.84** |
| **Average** | 28.90 | 29.54 | 29.58 | 29.63 | 29.65 | 29.64 | 29.69 | 29.81 | 29.78 | 29.91 | **29.93** |
| **Classic5 dataset** (image size: 256 × 256) | | | | | | | | | | | |

| **LIVE1 dataset** (image size: 256 × 256) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Q | **JPEG** | **BM3D** | **SA-DCT** | **PC-LRM** | **ANCE** | **WNNM** | **SSR-QC** | **COGL** | **JPG-SR** | **NSSRC** | **Ours** |
| **10** | 27.57 | 28.69 | 28.72 | 28.79 | 28.77 | 28.78 | 28.83 | 28.93 | 28.78 | 28.97 | **29.03** |
| **20** | 29.90 | 30.87 | 30.89 | 30.98 | 30.96 | 30.98 | 31.07 | 31.13 | 31.12 | 31.23 | **31.26** |
| **30** | 31.21 | 32.07 | 32.09 | 32.21 | 32.22 | 32.21 | 32.34 | 32.39 | 32.50 | 32.55 | **32.58** |
| **40** | 32.14 | 32.94 | 32.96 | 33.09 | 33.16 | 33.10 | 33.30 | 33.29 | 33.46 | **33.54** | **33.54** |
| **Average** | 30.21 | 31.14 | 31.17 | 31.27 | 31.28 | 31.27 | 31.39 | 31.43 | 31.47 | 31.57 | **31.60** |

**Table 5** Average SSIM Results of Comparison with Classic Methods for Image Deblocking on the LIVE1 Dataset (image size: $\mathbf{256 \times 256}$)) and the Classic5 Dataset (image size: $\mathbf{256 \times 256}$).

| **LIVE1 dataset** (image size: 256 × 256)) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Q | **JPEG** | **BM3D** | **SA-DCT** | **PC-LRM** | **ANCE** | **WNNM** | **SSR-QC** | **COGL** | **JPG-SR** | **NSSRC** | **Ours** |
| **10** | 0.7611 | 0.7877 | 0.7869 | 0.7835 | 0.7879 | 0.7824 | 0.7859 | 0.7957 | 0.7931 | 0.7956 | **0.7979** |
| **20** | 0.8423 | 0.8591 | 0.8571 | 0.8550 | 0.8585 | 0.8542 | 0.8576 | 0.8642 | 0.8630 | 0.8645 | **0.8658** |
| **30** | 0.8791 | 0.8917 | 0.8903 | 0.8892 | 0.8913 | 0.8888 | 0.8913 | 0.8952 | 0.8967 | 0.8963 | **0.8976** |
| **40** | 0.8998 | 0.9103 | 0.9093 | 0.9089 | 0.9102 | 0.9087 | 0.9099 | 0.9129 | 0.9145 | 0.9148 | **0.9155** |
| **Average** | 0.8456 | 0.8622 | 0.8609 | 0.8592 | 0.8620 | 0.8585 | 0.8612 | 0.8670 | 0.8668 | 0.8678 | **0.8692** |
| **Classic5 dataset** (image size: 256 × 256)) | | | | | | | | | | | |
| Q | **JPEG** | **BM3D** | **SA-DCT** | **PC-LRM** | **ANCE** | **WNNM** | **SSR-QC** | **COGL** | **JPG-SR** | **NSSRC** | **Ours** |
| **10** | 0.7715 | 0.8087 | 0.8060 | 0.8043 | 0.8081 | 0.8033 | 0.8094 | 0.8134 | 0.8134 | 0.8168 | **0.8199** |
| **20** | 0.8519 | 0.8753 | 0.8728 | 0.8723 | 0.8730 | 0.8714 | 0.8740 | 0.8751 | 0.8796 | 0.8802 | **0.8811** |
| **30** | 0.8844 | 0.9018 | 0.9002 | 0.9003 | 0.9002 | 0.8998 | 0.9017 | 0.9012 | 0.9063 | 0.9060 | **0.9065** |
| **40** | 0.9036 | 0.9178 | 0.9168 | 0.9170 | 0.9172 | 0.9167 | 0.9180 | 0.9175 | 0.9225 | **0.9226** | 0.9225 |
| **Average** | 0.8529 | 0.8759 | 0.8740 | 0.8735 | 0.8746 | 0.8728 | 0.8758 | 0.8768 | 0.8805 | 0.8814 | **0.8825** |

*1) Comparison with Classic Image Deblocking Methods:* We evaluated the performance of all competing classic deblocking methods on two public benchmarks: the LIVE1 dataset [50] and the Classic5 dataset [51]. Similar to image denoising, we adopted two evaluation metrics, PSNR and SSIM, and the results are shown in Table 4 and Table 5. It is clear that our method outperforms other classical methods apart from SSIM metric on Classic5 with $Q = 40$. In particular, the proposed method is significantly superior to other competing methods on low compression quality images ($Q = 10,20,30$), and close to or even exceeds SOTA methods on high compression quality images ($Q = 40$).

The SA-DCT [58] is the most popular image deblocking method, thus we present the visual comparisons shown in Fig. 9 and Fig. 10. From the qualitative results, we can conclude that the proposed method can effectively remove the block artifacts and preserve more image details.

*2) Comparison with DNN-based Image Deblocking Models:* To further demonstrate the advantage of our proposed method on image deblocking task, we also compared it against several DNN-based methods, including AR-CNN [65], TRND [56], DnCNN [49], DCSC [66], and MDDU [67]. The evaluation was carried out on the Classic5 [51] dataset, a popular benchmark for image deblocking. The results of the average PSNR and SSIM with different compression quality $Q$ are listed in the Table 6. It shows that our proposed method achieves better result than AR-CNN and TRND, meanwhile it is comparable to DnCNN, DCSC, and MDDU.

It is worth noting that these supervised deep learning methods require the large-scale image datasets to train image deblocking model. We notice that if the distribution of the training image dataset and the test images is identical or similar, then the deep learning model can effectively fit diverse image structures.

However, we found that the deep learning method is prone to over-smoothing, especially for images with rich textures, as shown in Fig. 11. To further verify this finding, we collected eight fingerprint images from the NIST dataset as a testing benchmark. The eight fingerprint images are shown in Fig. 12 and the results for fingerprint image deblocking are summarized in

Table 7. Our proposed method achieves better results than all the other competing deep learning-based image deblocking method. The visual comparison examples are shown in Fig. 13, where our proposed method can reconstruct better texture details than others.

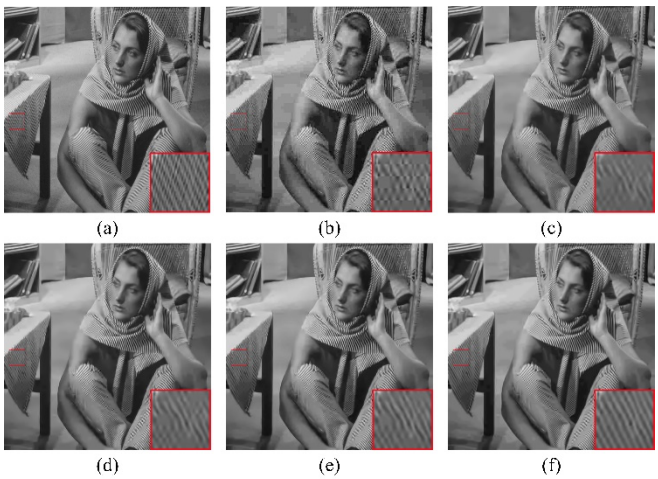

**Fig. 11** The visual comparison deblocking results of image *Barbara* on the Classic5 dataset at $\boldsymbol{Q = 10}$. (a) Original image. (b) JPEG compressed image (PSNR = 25.78 dB, SSIM = 0.7621). (c) ARCNN (PSNR = 26.89 dB, SSIM = 0.7934). (d) TRND (PSNR = 27.24 dB, SSIM = 0.8104). (e) DnCNN (PSNR = 27.59 dB, SSIM = 0.8161). (f) **Ours (PSNR = 27.91 dB, SSIM = 0.8287).**

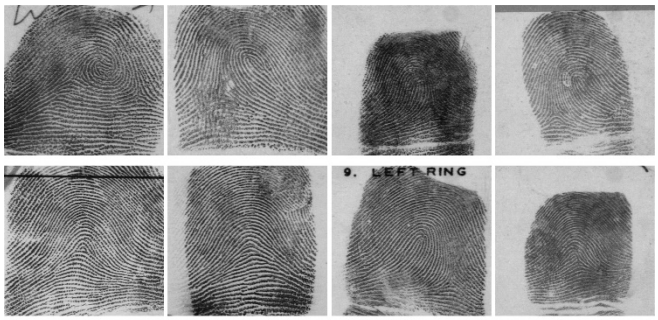

**Fig. 12** The eight fingerprint test images from NIST dataset.

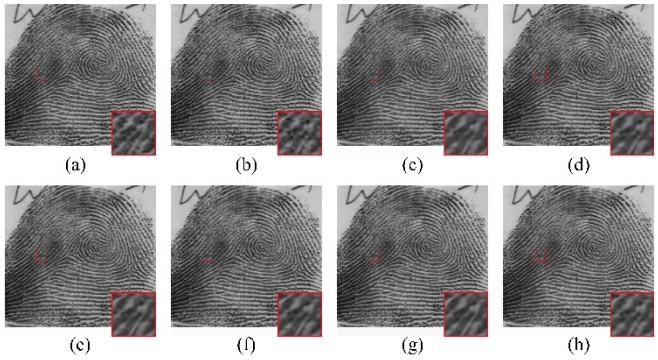

**Fig. 13** The visual comparison deblocking results of *image 01* on the fingerprint dataset at $\boldsymbol{Q = 10}$. (a) Original image. (b) JPEG compressed image (PSNR = 28.41 dB, SSIM = 0.8737). (c) ARCNN (PSNR = 29.57 dB, SSIM = 0.8969). (d) TRND (PSNR = 29.73 dB, SSIM = 0.9008). (e) DnCNN (PSNR = 29.72 dB, SSIM = 0.9019). (f) DCSC (PSNR = 29.82 dB, SSIM = 0.9045). (g) MDDU (PSNR = 29.82 dB, SSIM = 0.9081). (h) **Ours (PSNR = 30.17 dB, SSIM = 0.9089).**

**Table 6** Average PSNR (dB) / SSIM Results of Comparison with DNN-based Models for Image Deblocking on the Classic5 Dataset

| **Methods** | $Q = 10$ | $Q = 20$ | $Q = 30$ | **Average** |
|---|---|---|---|---|
| AR-CNN | 29.08 | 31.25 | 32.60 | 30.98 |
| | 0.7909 | 0.8514 | 0.8808 | 0.8410 |
| TRND | 29.29 | 31.48 | 32.79 | 31.19 |
| | 0.7996 | 0.8581 | 0.8841 | 0.8473 |
| DnCNN | 29.40 | 31.63 | 32.91 | 31.31 |
| | 0.8026 | 0.8610 | 0.8861 | 0.8499 |
| DCSC | 29.62 | 31.81 | 33.06 | 31.50 |
| | 0.8096 | 0.8641 | 0.8882 | 0.8540 |
| MDDU | 29.95 | 32.11 | 33.33 | 31.80 |
| | 0.8171 | 0.8689 | 0.8916 | 0.8592 |
| **Ours** | 29.36 | 31.52 | 32.88 | 31.25 |
| | 0.8041 | 0.8602 | 0.8857 | 0.8500 |

**Table 7** Average PSNR (dB) / SSIM Results of Comparison with DNN-based Models for Image Deblocking on the Fingerprint Images Dataset

| **Methods** | $Q = 10$ | $Q = 20$ | $Q = 30$ | **Average** |
|---|---|---|---|---|
| AR-CNN | 30.23 | 33.04 | 34.76 | 32.68 |
| | 0.8859 | 0.9291 | 0.9480 | 0.9210 |
| TRND | 30.42 | 33.19 | 34.87 | 32.83 |
| | 0.8899 | 0.9317 | 0.9492 | 0.9236 |
| DnCNN | 30.31 | 33.07 | 34.73 | 32.70 |
| | 0.8894 | 0.9308 | 0.9485 | 0.9229 |
| DCSC | 30.52 | 33.13 | 34.78 | 32.81 |
| | 0.8934 | 0.9330 | 0.9497 | 0.9254 |
| MDDU | 30.45 | 32.95 | 34.35 | 32.58 |
| | 0.8961 | **0.9349** | **0.9508** | 0.9273 |
| **Ours** | **30.84** | **33.59** | **35.19** | **33.21** |
| | **0.8971** | 0.9348 | 0.9506 | **0.9275** |

**Table 8** The Average Computational Time (in seconds) of Ours Method for Image Denoising and Deblocking.

| Denoising on 13 Widely Used Test Images ($256 \times 256$) | | | | |
|---|---|---|---|---|
| $\sigma_E$ | 20 | 40 | 75 | 100 |
| **Time** | 136.50 | 116.49 | 171.02 | 249.60 |
| Deblocking on LIVE1 Dataset ($256 \times 256$) | | | | |
| **Q** | 10 | 20 | 30 | 40 |
| **Time** | 81.16 | 61.68 | 90.25 | 69.61 |

### *4.5. Convergence*

It is difficult to provide a theoretical proof for local convergence of our proposed algorithm, due to its patch

grouping operations, non-convex optimization, and parameter updates. Therefore, the empirical evidence is provided to verify the convergence of our proposed algorithm. We selected 5 test images from the Set12 dataset, and recorded the process of restoring these images. Fig. 14(a) and (b) shows variation curve of PSNR value during algorithm iteration process for image denoising with noise level $\sigma_E = 50$ and image deblocking with compression quality $Q = 10$, respectively. It can be clearly seen that as the algorithm iterates, all PSNR curves of restored images are initially monotonically increasing and then gradually stabilizing. Therefore, the proposed algorithm possesses a good convergence property.

### 4.6. Computational Time

It is worth noting that our proposed method is based on a group sparse representation model, and each group can be processed independently. Therefore, we implement the algorithm with parallel code in MATLAB. All experiments were run on a laptop with Intel(R) Xeon(R) W-2223 CPU at 3.60GHz with 32-GB memory. We recorded the average time (in seconds) consumed by our algorithm for denoising on the 13 images of size $256 \times 256$ shown in Fig. 15 and the average time consumed for deblocking on the LIVE1 dataset [50]. The statistical results are shown in Table 8. Our algorithm can complete image denoising in 2-4 minutes, and image deblocking in about 1.5 minutes. Like other self-supervised algorithms, our method is difficult to achieve real-time, but it is suitable for scenes lacking ground truth (clean images) and high demand for restoration quality.

### 4.7. Ablation Study

To investigate the effectiveness of these different modules in our algorithm, we present the ablation study in this subsection, by removing the group residual constraint ($\gamma = 0$) and the sparsity constraint ($\beta = 0$), respectively. Its variants are simplified into group sparse representation-based restoration model (denoted as SR) and group residual learning-based restoration model (denoted as RL). In order to examine the contributions of SR, RL and our proposed model (SR+RL), we selected 13 test images which are widely used (as shown in Fig. 15) and applied these variants to image denoising. The average PSNR results are shown in Table 9. One can observe that both the group sparse representation and group residual learning play crucial roles in the success of our proposed model.

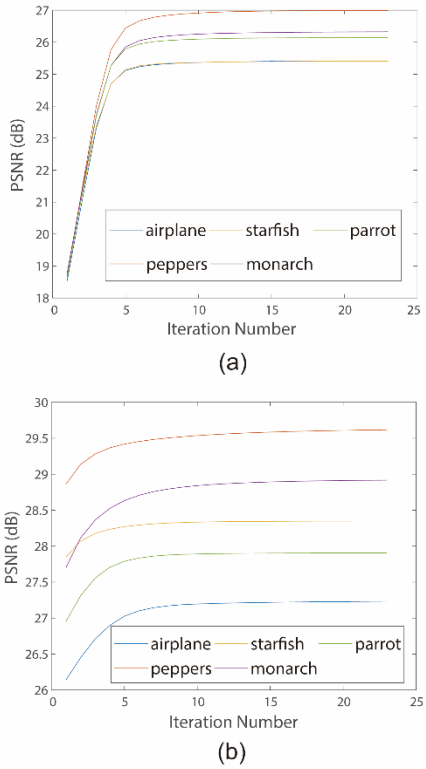


**Fig. 14** Convergence behaviour. (a) PSNR values curve with the number of iterations for image denoising with noise level $\boldsymbol{\sigma_E = 50}$. (b) PSNR values curve with the number of iterations for image deblocking with compression quality $\boldsymbol{Q = 10}$.

**Table 9** Average PSNR (dB) Results of Ablation Study on Image Denoising with the 13 Widely Used Test Images.

| **Modules** | $\sigma_E = 20$ | $\sigma_E = 30$ | $\sigma_E = 40$ | $\sigma_E = 50$ | $\sigma_E = 75$ | $\sigma_E = 100$ | **Average** |
|---|---|---|---|---|---|---|---|
| **SR** | 31.17 | 28.57 | 27.35 | 27.24 | 24.87 | 23.74 | 27.16 |
| **RL** | 31.08 | 29.81 | 28.32 | 25.90 | 22.73 | 23.33 | 26.86 |
| **SR+RL** | **32.23** | **30.24** | **28.81** | **27.73** | **25.72** | **24.40** | **28.79** |

**Fig. 15** The 13 test images used for computational time and ablation study.

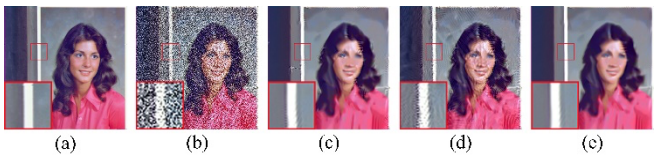


**Fig. 16** The visual comparison denoising results of image *Miss* with $\boldsymbol{\sigma_E = 75}$. (a) Original image. (b) Noisy image. (c) SR (PSNR = 26.38 dB, SSIM = 0.7646). (d) RL (PSNR = 23.52 dB, SSIM = 0.5758). (e) **SR+RL (PSNR = 27.61 dB, SSIM = 0.7944).**

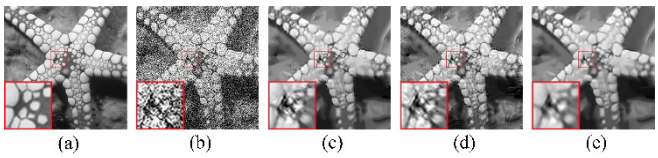

**Fig. 17** The visual comparison denoising results of image *Starfish* with $\boldsymbol{\sigma_E = 75}$. (a) Original image. (b) Noisy image. (c) SR (PSNR = 23.12 dB, SSIM = 0.6716). (d) RL (PSNR = 21.67 dB, SSIM = 0.5711). (e) **SR+RL (PSNR = 23.47 dB, SSIM = 0.6788).**

To further reveal the role of each component of the proposed algorithm, we illustrated some image examples reconstructed by SR model, RL model and the proposed model (SR+RL). As shown in Fig. 16(c) and Fig. 17(c), the SR model is indeed an ideal image restoration tool, but it can be affected by noise leading to undesirable results. As shown in Fig. 16(d) and Fig. 17(d), the group residual learning model we proposed can also achieve the purpose of image restoration, but there are many artifacts in the results. Therefore, we consider that SR model provides initial denoising, while RL model provides constraints for the SR model. The mutual promotion of these two components makes the proposed algorithm successful in image restoration.

## 5. Conclusion

In this paper, a novel algorithm was proposed for image restoration, where group sparse representation and sparsity residual are simultaneously learned. Unlike conventional sparsity residual models which estimate the true sparse representation using the weighted average method, we first define the estimate of the true sparse representation as the product of the learned sparse representation and a weight matrix, and then our proposed group sparsity residual learning model automatically learns adaptive weight matrix via self-representation learning. Moreover, we explore the sub-group structure of each patch group using the low-rank constraint to better leverage the relation among patches within the same group. The proposed algorithm is comprehensively validated and compared with many other popular or SOTA image restoration methods on public benchmarks. The results demonstrate that, in image restoration tasks such as denoising and deblocking, our suggested approach performs better than those baselines.

However, there are still shortcomings in this work, as it can only handle Gaussian additive noise. In future work, we will consider other distributions of noise, even multiplicative noise.

## Acknowledgments

This work was supported by the Shaanxi Key Research and Development Program (Program No. 2021ZDLGY02-02).

## References

[1] Banham MR, Katsaggelos AK. Digital image restoration. IEEE signal processing magazine. 1997;14(2):24–41.

[2] Elad M, Aharon M. Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image processing. 2006;15(12):3736–3745.

[3] Yang J, Wright J, Huang T, Ma Y. Image super-resolution as sparse representation of raw image patches. In: 2008 IEEE conference on computer vision and pattern recognition. IEEE; 2008. p. 1–8.

[4] Jung C, Jiao L, Qi H, Sun T. Image deblocking via sparse representation. Signal Processing: Image Communication. 2012;27(6):663–677.

[5] Rudin LI, Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena. 1992;60(1-4):259–268.

[6] Osher S, Burger M, Goldfarb D, Xu J, Yin W. An iterative regularization method for total variation-based image restoration. Multiscale Modeling & Simulation. 2005;4(2):460–489.

[7] Bappy D, Jeon I. Combination of hybrid median filter and total variation minimisation for medical X-ray image restoration. IET Image Processing. 2016;10(4):261–271.

[8] Dong W, Zhang L, Shi G, Li X. Nonlocally centralized sparse representation for image restoration. IEEE transactions on Image Processing. 2012;22(4):1620–1630.

[9] Ji H, Huang S, Shen Z, Xu Y. Robust video restoration by joint sparse and low rank matrix approximation. SIAM Journal on Imaging Sciences. 2011;4(4):1122–1142.

[10] Dong W, Shi G, Li X. Nonlocal image restoration with bilateral variance estimation: a low-rank approach. IEEE transactions on image processing. 2012;22(2):700–711.

[11] Gu S, Zhang L, Zuo W, Feng X. Weighted nuclear norm minimization with application to image denoising. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2014. p. 2862–2869.

[12] Liu D, Wen B, Fan Y, Loy CC, Huang TS. Non-local recurrent network for image restoration. Advances in neural information processing systems. 2018;31.

[13] Mastan ID, Raman S. DeepCFL: Deep contextual features learning from a single image. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2021. p. 2897–2906.

[14] Zha Z, Yuan X, Zhou JT, Zhou J, Wen B, Zhu C. The power of triply complementary priors for image compressive sensing. In: 2020 IEEE International Conference on Image Processing (ICIP). IEEE; 2020. p. 983–987.

[15] Mataev G, Milanfar P, Elad M. Deepred: Deep image prior powered by red. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops; 2019. p. 0–0.

[16] Zamir SW, Arora A, Khan S, Hayat M, Khan FS, Yang MH. Restormer: Efficient transformer for high-resolution image restoration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2022. p. 5728–5739.

[17] Shen L, Pauly J, Xing L. NeRP: implicit neural representation learning with prior embedding for sparsely sampled image reconstruction. IEEE Transactions on Neural Networks and Learning Systems. 2022;.
[18] Chen L, Chu X, Zhang X, Sun J. Simple baselines for image restoration. In: European Conference on Computer Vision. Springer; 2022. p. 17–33.
[19] Zha Z, Wen B, Yuan X, Zhou J, Zhu C. Simultaneous nonlocal low-rank and deep priors for poisson denoising. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2022. p. 2320–2324.
[20] Wu W, Chen M, Xiang Y, Zhang Y, Yang Y. Recent progress in image denoising: A training strategy perspective. IET Image Processing. 2023;.
[21] Deeba F, Kun S, Ali Dharejo F, Zhou Y. Sparse representation based computed tomography images reconstruction by coupled dictionary learning algorithm. IET image Processing. 2020;14(11):2365–2375.
[22] Zha Z, Yuan X, Wen B, Zhang J, Zhu C. Nonconvex structural sparsity residual constraint for image restoration. IEEE Transactions on Cybernetics. 2021;.
[23] Yuan W, Liu H, Liang L. Image restoration via exponential scale mixture-based simultaneous sparse prior. IET Image Processing. 2022;16(12):3268–3283.
[24] Xu W, Zhu Q, Qi N, Chen D. Deep sparse representation based image restoration with denoising prior. IEEE Transactions on Circuits and Systems for Video Technology. 2022;32(10):6530–6542.
[25] Buades A, Coll B, Morel JM. A non-local algorithm for image denoising. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). vol. 2. Ieee; 2005. p. 60–65.
[26] Dabov K, Foi A, Katkovnik V, Egiazarian K. Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Transactions on image processing. 2007;16(8):2080–2095.
[27] Zhang J, Zhao D, Gao W. Group-based sparse representation for image restoration. IEEE transactions on image processing. 2014;23(8):3336–3351.
[28] Li X, Shen H, Li H, Zhang L. Patch matching-based multitemporal group sparse representation for the missing information reconstruction of remote-sensing images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2016;9(8):3629–3641.
[29] Wang Q, Zhang X, Wu Y, Tang L, Zha Z. Nonconvex Weighted $\ell_p$ Minimization Based Group Sparse Representation Framework for Image Denoising. IEEE Signal Processing Letters. 2017;24(11):1686–1690.
[30] Aharon M, Elad M, Bruckstein A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on signal processing. 2006;54(11):4311–4322.
[31] Daubechies I. The wavelet transform, time-frequency localization and signal analysis. IEEE transactions on information theory. 1990;36(5):961–1005.
[32] Guo L, Zha Z, Ravishankar S, Wen B. Exploiting non-local priors via self-convolution for highly-efficient image restoration. IEEE Transactions on Image Processing. 2022;31:1311–1324.
[33] Mairal J, Bach F, Ponce J, Sapiro G, Zisserman A. Non-local sparse models for image restoration. In: 2009 IEEE 12th international conference on computer vision. IEEE; 2009. p. 2272–2279.
[34] Xu J, Zhang L, Zuo W, Zhang D, Feng X. Patch group based nonlocal self-similarity prior learning for image denoising. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 244–252.
[35] Zha Z, Wen B, Yuan X, Zhou J, Zhu C, Kot AC. A hybrid structural sparsification error model for image restoration. IEEE Transactions on Neural Networks and Learning Systems. 2021;33(9):4451–4465.
[36] Yuan W, Liu H, Liang L. Joint group dictionary-based structural sparse representation for image restoration. Digital Signal Processing. 2023;137:104029.
[37] Zha Z, Wen B, Yuan X, Zhou J, Zhu C, Kot AC. Low-rankness guided group sparse representation for image restoration. IEEE Transactions on Neural Networks and Learning Systems. 2022;.
[38] Zha Z, Wen B, Yuan X, Zhou J, Zhu C. Image restoration via reconciliation of group sparsity and low-rank models. IEEE Transactions on Image Processing. 2021;30:5223–5238.
[39] Zha Z, Wen B, Yuan X, Ravishankar S, Zhou J, Zhu C. Learning nonlocal sparse and low-rank models for image compressive sensing: Nonlocal sparse and low-rank modeling. IEEE Signal Processing Magazine. 2023;40(1):32–44.
[40] Niknejad M, Rabbani H, Babaie-Zadeh M. Image restoration using Gaussian mixture models with spatially constrained patch clustering. IEEE Transactions on Image Processing. 2015;24(11):3624–3636.
[41] Candes EJ, Tao T. Near-optimal signal recovery from random projections: Universal encoding strategies? IEEE transactions on information theory. 2006;52(12):5406–5425.
[42] Donoho DL. For most large underdetermined systems of linear equations the minimal $\ell_1$-norm solution is also the sparsest solution. Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences. 2006;59(6):797–829.
[43] Chartrand R, Yin W. Iteratively reweighted algorithms for compressive sensing. In: 2008 IEEE international conference on acoustics, speech and signal processing. IEEE; 2008. p. 3869–3872.
[44] Mazumder R, Friedman JH, Hastie T. Sparsenet: Coordinate descent with nonconvex penalties. Journal of the American Statistical Association. 2011;106(495):1125–1138.
[45] Wang L, Huang J, Yin M, Cai R, Hao Z. Block diagonal representation learning for robust subspace clustering. Information Sciences. 2020;526:54–67.
[46] Vidal R, Favaro P. Low rank subspace clustering (LRSC). Pattern Recognition Letters. 2014;43:47–61.
[47] Zuo W, Meng D, Zhang L, Feng X, Zhang D. A generalized iterated shrinkage algorithm for non-convex sparse coding. In: Proceedings of the IEEE international conference on computer vision; 2013. p. 217–224.
[48] Krishnan D, Fergus R. Fast image deconvolution using hyper-Laplacian priors. Advances in neural information processing systems. 2009;22.
[49] Zhang K, Zuo W, Chen Y, Meng D, Zhang L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE transactions on image processing. 2017;26(7):3142–3155.

[50] Sheikh H. LIVE image quality assessment database release 2. http://live ece utexas edu/research/quality. 2005;.
[51] Zeyde R, Elad M, Protter M. On single image scale-up using sparse-representations. In: International conference on curves and surfaces. Springer; 2010. p. 711–730.
[52] Zoran D, Weiss Y. From learning models of natural image patches to whole image restoration. In: 2011 international conference on computer vision. IEEE; 2011. p. 479–486.
[53] Luo E, Chan SH, Nguyen TQ. Adaptive image denoising by mixture adaptation. IEEE transactions on image processing. 2016;25(10):4489–4503.
[54] Xiong R, Liu H, Zhang X, Zhang J, Ma S, Wu F, et al. Image denoising via bandwise adaptive modeling and regularization exploiting nonlocal similarity. IEEE Transactions on Image Processing. 2016;25(12):5793–5805.
[55] Zha Z, Zhang X, Wang Q, Bai Y, Chen Y, Tang L, et al. Group sparsity residual constraint for image denoising with external nonlocal self-similarity prior. Neurocomputing. 2018;275:2294–2306.
[56] Chen Y, Pock T. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. IEEE transactions on pattern analysis and machine intelligence. 2016;39(6):1256–1272.
[57] Quan Y, Chen M, Pang T, Ji H. Self2self with dropout: Learning self-supervised denoising from single image. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2020. p. 1890–1898.
[58] Foi A, Katkovnik V, Egiazarian K. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. IEEE transactions on image processing. 2007;16(5):1395–1411.
[59] Ren J, Liu J, Li M, Bai W, Guo Z. Image blocking artifacts reduction via patch clustering and low-rank minimization. In: 2013 Data Compression Conference. IEEE; 2013. p. 516–516.
[60] Gu S, Xie Q, Meng D, Zuo W, Feng X, Zhang L. Weighted nuclear norm minimization and its applications to low level vision. International journal of computer vision. 2017;121(2):183–208.
[61] Zhang X, Xiong R, Fan X, Ma S, Gao W. Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity. IEEE transactions on image processing. 2013;22(12):4613–4626.
[62] Zhao C, Zhang J, Ma S, Fan X, Zhang Y, Gao W. Reducing image compression artifacts by structural sparse representation and quantization constraint prior. IEEE Transactions on Circuits and Systems for Video Technology. 2016;27(10):2057–2071.
[63] Young SI, Naman AT, Taubman D. COGL: Coefficient graph laplacians for optimized JPEG image decoding. IEEE Transactions on Image Processing. 2018;28(1):343–355.
[64] Zha Z, Yuan X, Wen B, Zhang J, Zhou J, Zhu C. Image restoration using joint patch-group-based sparse representation. IEEE Transactions on Image Processing. 2020;29:7735–7750.
[65] Dong C, Deng Y, Loy CC, Tang X. Compression artifacts reduction by a deep convolutional network. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 576–584.
[66] Fu X, Zha ZJ, Wu F, Ding X, Paisley J. Jpeg artifacts reduction via deep convolutional sparse coding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019. p. 2501–2510.
[67] Fu X, Wang M, Cao X, Ding X, Zha ZJ. A model-driven deep unfolding method for jpeg artifacts removal. IEEE Transactions on Neural Networks and Learning Systems. 2021;.
[68] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing. 2004;13(4):600–612.
[69] Immerkaer J. Fast noise variance estimation. Computer vision and image understanding. 1996;64(2):300–302.