

RESEARCH

Open Access

Clustering K-SVD for sparse representation of images



Jun Fu^{1,2}, Haikuo Yuan^{1,2}, Rongqiang Zhao^{1,2*}  and Luquan Ren^{1,2}

Abstract

K-singular value decomposition (K-SVD) is a frequently used dictionary learning (DL) algorithm that iteratively works between sparse coding and dictionary updating. The sparse coding process generates sparse coefficients for each training sample, and the sparse coefficients induce clustering features. In the applications like image processing, the features of different clusters vary dramatically. However, all the atoms of dictionary jointly represent the features, regardless of clusters. This would reduce the accuracy of sparse representation. To address this problem, in this study, we develop the clustering K-SVD (CK-SVD) algorithm for DL and the corresponding greedy algorithm for sparse representation. The atoms are divided into a set of groups, and each group of atoms is employed to represent the image features of a specific cluster. Hence, the features of all clusters can be utilized and the number of redundant atoms are reduced. Additionally, two practical extensions of the CK-SVD are provided. Experimental results demonstrate that the proposed methods could provide more accurate sparse representation of images, compared to the conventional K-SVD and its existing extended methods. The proposed clustering DL model also has the potential to be applied to the online DL cases.

Keywords: Dictionary learning, Sparse representation, Image processing

1 Introduction

Sparse representation aims to model signals as sparse linear combinations of the atoms in a dictionary, and this technique is widely used in various fields of image processing [1–4]. Let $\mathbf{z} \in \mathbb{R}^n$ and $\mathbf{D} \in \mathbb{R}^{n \times q}$, $q \geq n$ denote a signal and an over-complete dictionary, respectively. The sparse representation of \mathbf{z} with respect to the dictionary \mathbf{D} is expressed as $\mathbf{z} \approx \mathbf{D}\mathbf{s}$. The sparse coefficients vector $\mathbf{s} \in \mathbb{R}^q$ satisfies $\|\mathbf{s}\|_0 \leq k$ and $\|\mathbf{z} - \mathbf{D}\mathbf{s}\|_2 \leq \epsilon$, where $\|\cdot\|_0$ denotes the number of non-zero entries of a vector, k and ϵ represent the maximum number of sparse coefficients and sparse representation error, respectively. In general, the dictionaries used for sparse representation can be divided into two categories: analytical dictionaries and learned dictionaries. The analytical dictionaries like wavelet dictionaries can be universally applied, and they are easy to obtain. However, the moderate sparse representation accuracy limits their applications. For better performance,

the over-complete dictionary \mathbf{D} is commonly obtained from the DL process using a set of training samples $\mathbf{Z} \in \mathbb{R}^{n \times \xi}$, expressed as:

$$\arg \min_{\mathbf{D}, \mathbf{S}} \|\mathbf{Z} - \mathbf{D}\mathbf{S}\|_F^2 \text{ s.t. } \|\mathbf{s}_t\|_0 \leq k, \|\mathbf{d}_i\|_2 = 1, \quad (1)$$

where $t \in \{1, 2, \dots, \xi\}$, $i \in \{1, 2, \dots, q\}$, and $\|\cdot\|_F$ denote the Frobenius norm. The notation \mathbf{d}_i denotes the i th column of the dictionary \mathbf{D} , which is also referred to as the i th atom. \mathbf{S} is the sparse coefficients with respect to \mathbf{Z} and \mathbf{D} , and it is obtained with \mathbf{D} in the DL process; \mathbf{s}_t is the t th column of \mathbf{S} .

To date, researchers have proposed various DL algorithms. In [5], Engan et al. propose the well-known DL method, named “the method of optimal directions (MOD).” The MOD contained two iterative process, sparse coefficients computing and dictionary updating. The dictionary updating is globally realized by the least squares (LS) computation in terms of training samples and sparse coefficients. In [6], Aharon et al. propose another LS-based algorithm for DL, referred to as the K-SVD. Different from the global update strategy, for K-SVD, the atoms of dictionary is updated separately. The

*Correspondence: sharmmi@126.com

¹Key Laboratory of Bionic Engineering, Ministry of Education, Jilin University, 130022 Changchun, China

²College of Biological and Agricultural Engineering, Jilin University, 130022 Changchun, China

MOD, the K-SVD, and their extended methods are used for batch DL, i.e., the training samples input simultaneously. However, when the training samples can not be obtained all at one, the online learning is required. In [7], Mairal et al. propose the online DL (ODL) algorithm, aiming to update the atoms by using only the newly input samples. This algorithm allows training samples to be input successively and realizes the online learning. Additionally, a set of DL methods that are extended from MOD, K-SVD, and ODL have also been proposed, in order to improve the sparse representation accuracy or reduce the computational complexity [8–13]. Among these algorithms, the K-SVD is frequently used in the fields of image processing due to its generality and low complexity. The K-SVD algorithm consists two processes, sparse coding and dictionary updating, which are executed alternately. In the sparse coding process, at most k sparse coefficients for each training sample are computed via greedy algorithms, inducing clustering features [14–16]. For the K-SVD algorithm, all the atoms of dictionary jointly represent the training images, regardless of clusters. While representing different training samples, an atom may be employed by different clusters of features. In the applications of image processing, the features of different clusters vary dramatically, and therefore, the above phenomenon may reduce the accuracy of sparse representation. In [9], Nazzari et al. utilize the residual of training samples to train a set of sub-dictionaries. However, the sub-dictionaries are not distinguished by different clusters. In [11], Smith and Elad improve the K-SVD by considering only the used atoms in dictionary updating process. In [31], Tariyal et al. propose the deep DL by combining the concepts of DL and deep learning. The multiple DL framework is developed for multiple levels of dictionaries. In [30], Yi et al. build a hierarchical sparse representation framework that consists of the local histogram-based model, the weighted alignment pooling model, and the sparsity-based discriminative model. In [28], Rubinstein et al. propose the approximate K-SVD method to reduce the computational complexity, which can be regarded as another implementation of the K-SVD. In [29], Mairal et al. develop a multiscale DL framework based on an efficient quadtree decomposition of the learned dictionary.

In this study, we aim to utilize the clusters of features of training samples. We divide the atoms of learned dictionary into a set of groups, and each group serves for a specific cluster of features. This strategy improves the DL process from two aspects. First, besides the image features of the original training samples, we also consider the features of residuals of different clusters, which reduces the number of redundant atoms. Second, we develop the strategy to ensure that an arbitrary atom of a dictionary is utilized for only a specific

feature of a cluster. Hence, the atom would not be influenced by the features of other clusters. Based on the strategy, we propose the CK-SVD algorithm, as well as the corresponding greedy recovery algorithm for computing sparse representations. Compared to the conventional K-SVD, the CK-SVD improves the sparse reconstruction accuracy without increasing the requirements or computational complexity of DL process. Based on the clustering DL model, we also provide two practical extended methods of the CK-SVD, which achieve the adaptive sparsity and the dynamic refinement of atoms, respectively.

The remainder of this paper is organized as follows. Section 2 describes the aim of this study and introduces the proposed method. Section 3 provides the extended methods of the CK-SVD. Section 4 presents the experimental results. Section 5 discusses the proposed clustering model and its potential to be applied to online learning. Section 6 draws a conclusion.

2 Proposed method

In this section, we primarily review the conventional K-SVD algorithm and describe the problem that needs to be addressed. Next, we introduce the proposed CK-SVD algorithm.

2.1 Problem formulation

Given the training samples $\mathbf{Z} \in \mathbb{R}^{n \times \xi}$ and the initialized dictionary $\mathbf{D} \in \mathbb{R}^{n \times q}$, $q \geq n$, the sparse coding process is given by:

$$\arg \min_{\mathbf{s}} \|\mathbf{Z} - \mathbf{D}\mathbf{s}\|_F^2 \text{ s.t. } \|\mathbf{s}_t\|_0 \leq k, \quad (2)$$

where $t = 1, 2, \dots, \xi$. The process is executed by computing the sparse representation for each training sample expressed as:

$$\arg \min_{\mathbf{s}_t} \|\mathbf{z}_t - \mathbf{D}\mathbf{s}_t\|_2^2 \text{ s.t. } \|\mathbf{s}_t\|_0 \leq k, \quad (3)$$

where the vector $\mathbf{z}_t \in \mathbb{R}^n$ denotes an arbitrary training sample. The above problem is commonly solved by greedy algorithms like orthogonal matching pursuit (OMP) [17]. Specifically, for each iteration of the process for solving (3), the atom that leads the largest inner product with the residual of \mathbf{z}_t is selected. Thus, k atoms are selected successively, inducing k clusters of features. For the image samples, the objective features of different clusters vary greatly. But the atoms jointly represent the features, without considering the clusters. Hence, an arbitrary atom of the dictionary may be interfered by different features.

Here, we provide an example to describe this issue by employing the test image “Koala” with the size of 480×320 from the standard image dataset, Berkeley dataset [18]. We divided the images into patches with the size of 4×4 ,

i.e., $n = 16$. We vectorized the patches and used as the training samples for the K-SVD algorithm. The dictionary $\mathbf{D} \in \mathbb{R}^{n \times q}$ was initialized as the Gaussian random matrix. We set the total number of atoms and the maximum number of sparse coefficients for each sample to $q = 3n$ and $k = 3$, respectively. After ten iterations, the outputted dictionary was obtained and it was utilized for sparse coding of the image “Koala” via the OMP algorithm. We divided the image into a set of patches and coded these patches respectively by using the learned dictionary. The original image, the clusters of image features, and the residual images are presented in Fig. 1. We obtain the residual image \mathbf{R}_1 by removing the first cluster of features \mathbf{H}_1 from the original image \mathbf{X} , and \mathbf{H}_1 can be regarded as the sparse representation by using the first selected atom in the greedy recovering process. Similarly, we can obtain \mathbf{R}_2 and \mathbf{R}_3 by removing \mathbf{H}_2 and \mathbf{H}_3 from \mathbf{R}_1 and \mathbf{R}_2 , respectively.

An atom of the learned dictionary can be used to represent several patches. When an atom is used to represent a patch, it may be invoked in the first (\mathbf{H}_1), the second (\mathbf{H}_2), or the third (\mathbf{H}_3) cluster (as indicated in Fig. 1). Thus, in Table 1 and Fig. 2, we summarize how many times that a specific atom is invoked by different cluster of features. Most atoms are invoked by more than one cluster of features.

To further illustrate this issue, we also selected an atom, the 12th atom of the learned dictionary and collected the features that invoked the atom. The result is displayed in Fig. 3. The features invoked the 12th atom belong to two different clusters. It can be noted that different cluster of features vary greatly. In other word, the 12th atom of the dictionary is employed to represent two different type of features. Obviously, one atom cannot provide the accuracy represents for both two types of features. Hence, the atom has to compromise among these different features to achieve the global minimum representation error. As a result, the performance of learned dictionary will be influenced. The graphical representation of the atom is presented in Fig. 3, indicating it contains part of characteristic of the first cluster of features and part of characteristic of the second cluster of features. It implies the learned atom is a compromise of the two types of features.

To address this problem, we propose the CK-SVD algorithm for DL and the corresponding greedy algorithm for sparse recovery, which will be introduced in the following section.

2.2 CK-SVD for sparse representation of images

The proposed DL algorithm is also composed of two iterative process. As shown in Fig. 4, for the sparse coding

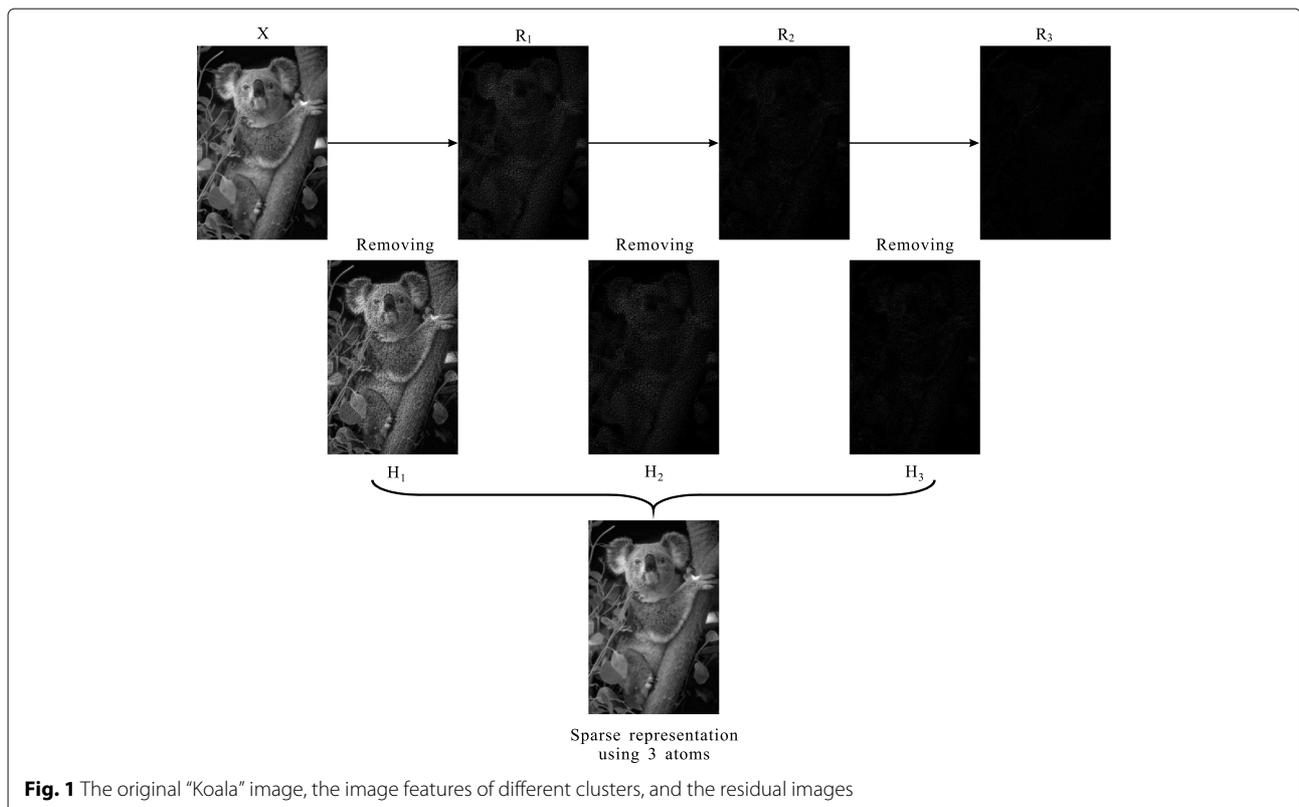


Fig. 1 The original “Koala” image, the image features of different clusters, and the residual images

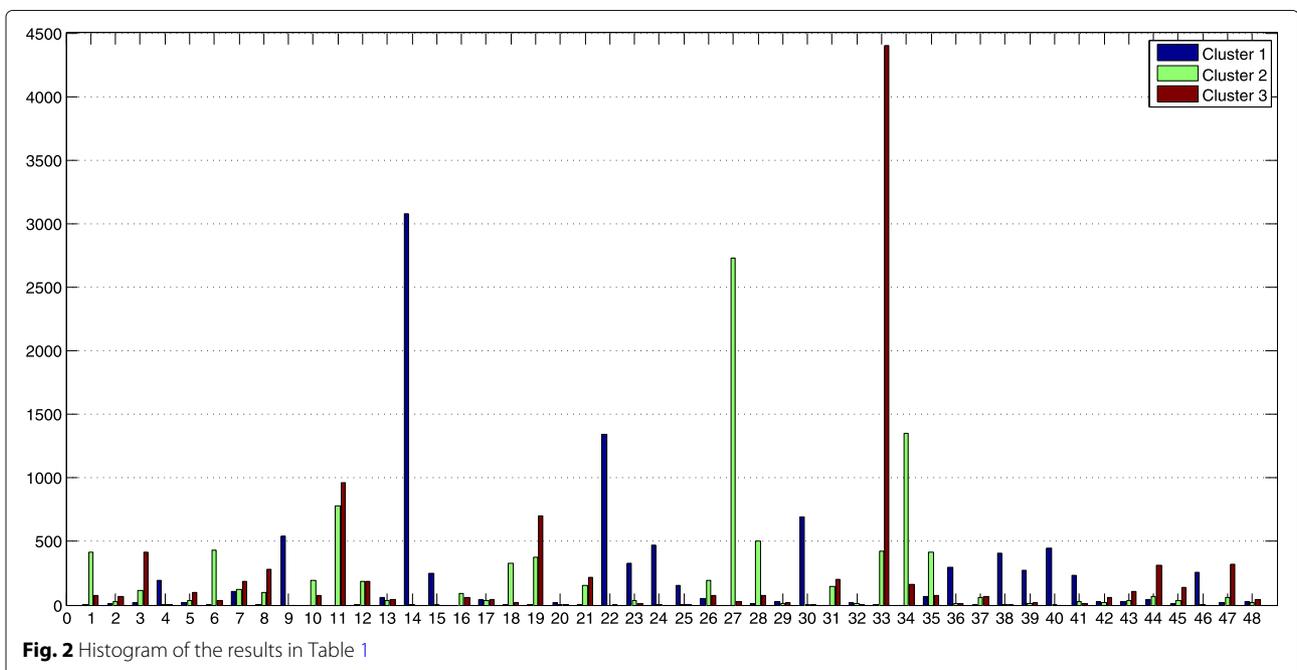
Table 1 Summary on how many times that a specific atom is invoked by different cluster of features

atom	Usage count			atom	Usage count			atom	Usage count		
	H_1	H_2	H_3		H_1	H_2	H_3		H_1	H_2	H_3
1st	2	416	76	17th	44	35	40	33rd	3	420	4398
2nd	9	24	70	18th	4	332	20	34th	0	1349	163
3rd	17	114	416	19th	1	375	704	35th	65	418	73
4th	195	1	1	20th	20	5	2	36th	294	10	9
5th	16	38	102	21st	6	156	215	37th	7	57	64
6th	2	433	39	22nd	1339	0	1	38th	410	1	1
7th	108	119	190	23rd	331	39	8	39th	271	8	20
8th	6	96	282	24th	475	1	0	40th	445	2	0
9th	545	0	0	25th	152	7	5	41st	231	26	14
10th	0	193	72	26th	48	197	76	42nd	27	20	61
11th	0	779	965	27th	0	2729	26	43rd	30	35	108
12th	3	184	152	28th	12	505	78	44th	40	66	310
13th	58	35	42	29th	25	9	20	45th	12	39	139
14th	3079	2	0	30th	697	3	1	46th	258	3	0
15th	249	3	0	31st	0	150	205	47th	20	56	323
16th	0	93	58	32nd	17	10	6	48th	27	17	45

process, we divide the atoms into k groups, which serve for each cluster of features.

In other words, we divide the dictionary into k sub-dictionaries, expressed as $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_k]$. We propose the greedy algorithm to solve the sparse recovery problem, which is described in Algorithm 1. For the l th iterative cycle, the features of the l th cluster are

considered, and therefore, we only search the atoms in the l th sub-dictionary. In other words, only the atoms of \mathbf{D}_l have the opportunity to be selected in the l th iterative cycle. Among these atoms, the one that is most relative to the residual obtained in the $(l - 1)$ th iteration is selected to represent the feature of the l th cluster. We compute the sparse coefficient for the current cluster based on the LS



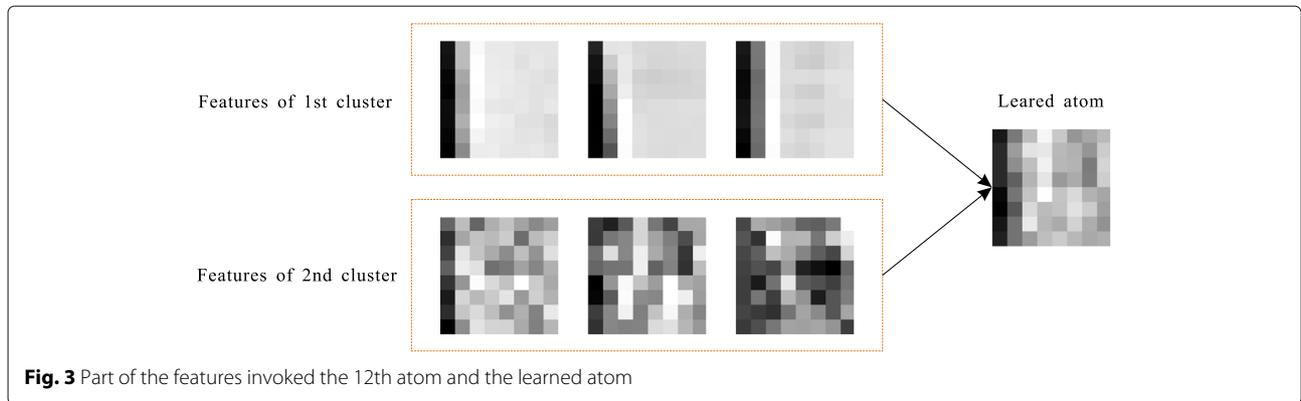


Fig. 3 Part of the features invoked the 12th atom and the learned atom

method:

$$c = (\mathbf{b}^* \mathbf{b})^{-1} \mathbf{b}^* \mathbf{z}, \tag{4}$$

where \mathbf{b} is the atom of \mathbf{D} indexed by ω , and ω is obtained by steps 3 and 4 in Algorithm 1. Then, the residual of the objective sample is updated by:

$$\mathbf{r} \leftarrow \mathbf{r} - \mathbf{c} \mathbf{b}. \tag{5}$$

The above process is executed until the maximum number of coefficients is reached or the residual is small enough.

The Algorithm 1 is employed for computing the sparse coefficients of each training sample \mathbf{z}_l . Next, the dictionary updating process is executed. Different from the conventional K-SVD, for the proposed method, the sub-dictionaries $\{\mathbf{D}_l \in \mathbb{R}^{n \times q_l}\}$ are initialized. The larger parameter q leads better performance of the dictionary but increases the complexity of the DL process and subsequent applications. For simplicity and without loss of generality, in this study, we assume the number of atoms

Algorithm 1 Sparse recovery for CK-SVD

Require: A training sample $\mathbf{z} \in \mathbb{R}^n$, sparse level k , dictionary $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_k] \in \mathbb{R}^{n \times q}$, threshold ϵ .

- 1: Initialize $\mathbf{r} = \mathbf{z}, l = 1$;
- 2: **while** $l < k$ and $\|\mathbf{r}\|_2 > \epsilon$
- 3: Compute the inner product by $\mathbf{p} = \mathbf{D}_l^* \mathbf{r}$;
- 4: Find the maximum magnitude entry of \mathbf{p} , the index of which is denoted as ω ;
- 5: Denote $\mathbf{b} = \mathbf{D}(:, \omega)$;
- 6: Compute the LS coefficient by $c = (\mathbf{b}^* \mathbf{b})^{-1} \mathbf{b}^* \mathbf{z}$;
- 7: Compute s_l by $s_l(\omega) = c$;
- 8: Update \mathbf{s} by $\mathbf{s} \leftarrow [\mathbf{s}; s_l]$;
- 9: Update the residual by $\mathbf{r} \leftarrow \mathbf{r} - \mathbf{c} \mathbf{b}$;
- 10: $l = l + 1$;
- 11: **end while**

Ensure: Sparse coefficients \mathbf{s} ;

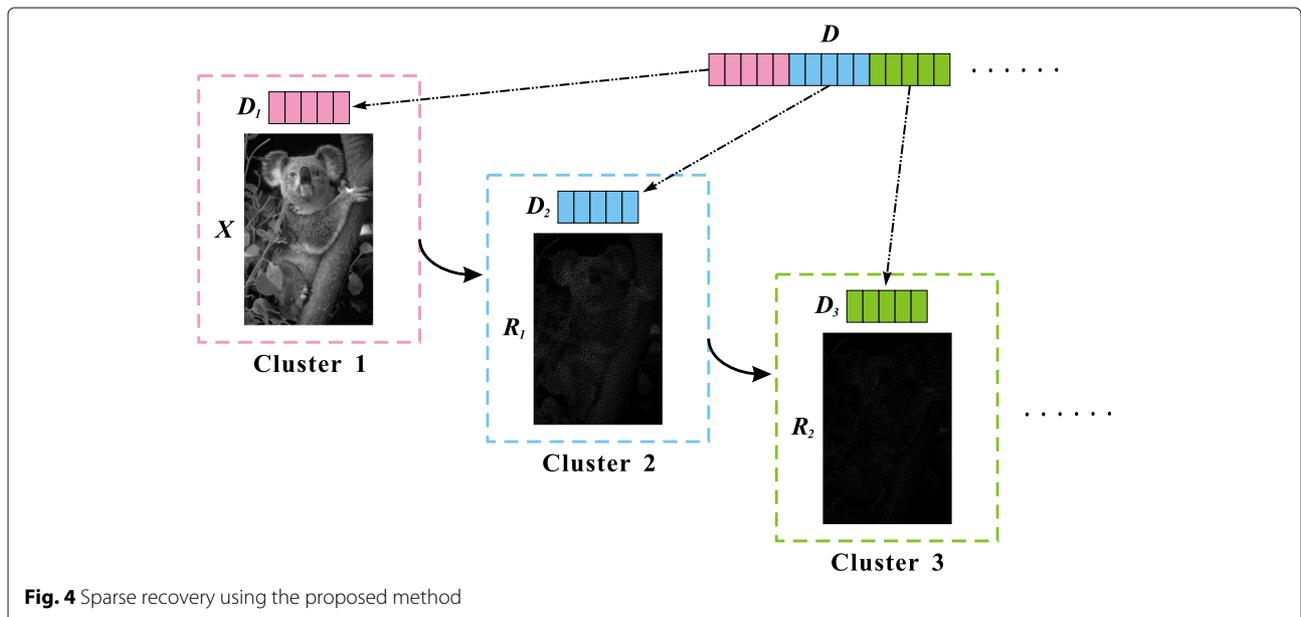


Fig. 4 Sparse recovery using the proposed method

of each sub-dictionary is the same. For an arbitrary atom \mathbf{d}_i , we first find the training samples that have used \mathbf{d}_i , and their indexes are denoted as γ_i . Then, we focus on the training samples indexed by γ_i , i.e., \mathbf{Z}_{γ_i} , and compute the residual of these samples by excluding the atom \mathbf{d}_i that is expressed as:

$$\mathbf{R}_{\gamma_i} = \mathbf{Z}_{\gamma_i} - \sum_{j \neq i} \mathbf{d}_j \tilde{\mathbf{s}}_{\gamma_i}^j, \quad (6)$$

where \mathbf{R}_{γ_i} denotes the mentioned residual, $\tilde{\mathbf{s}}_{\gamma_i}^j$ represents the j th row of \mathbf{S}_{γ_i} . In fact, while excluding \mathbf{d}_i , other used atoms for \mathbf{Z}_{γ_i} would not belong to the group that contains \mathbf{d}_i , and this property could be utilized to reduce the computational complexity. Next, we apply the singular value decomposition (SVD) to the residual \mathbf{R}_{γ_i} , expressed as:

$$\mathbf{R}_{\gamma_i} = \mathbf{U} \mathbf{\Delta} \mathbf{V}^*. \quad (7)$$

We update the atom \mathbf{d}_i to be the first column of \mathbf{U} , denoted as \mathbf{u}_1 , and update the sparse coefficients row $\tilde{\mathbf{s}}_{\gamma_i}^j$ by multiplying $\Delta_{1,1}$ and the first column of \mathbf{V} , denoted as \mathbf{v}_1 .

Algorithm 2 CK-SVD for DL

Require: Training samples $\mathbf{Z} \in \mathbb{R}^{n \times \xi}$, sparse level k , threshold ϵ .

- 1: Initialize each sub-dictionary $\mathbf{D}_l \in \mathbb{R}^{n \times q_l}$, $l = 1, 2, \dots, k$, $\mathbf{D} = [\mathbf{D}_1 \mathbf{D}_2 \dots \mathbf{D}_k]$;
- 2: **while** the termination condition is not reached
- 3: Solve the sparse representation \mathbf{s}_t for each training sample \mathbf{z}_t by using the Algorithm 1;
- 4: **for** each atom \mathbf{d}_i
- 5: Find the samples that use \mathbf{d}_i , their indexes are denoted as γ_i ;
- 6: Derive \mathbf{Z}_{γ_i} and \mathbf{S}_{γ_i} from \mathbf{Z} and \mathbf{S} , respectively;
- 7: Compute the residual by $\mathbf{R}_{\gamma_i} = \mathbf{Z}_{\gamma_i} - \sum_{j \neq i} \mathbf{d}_j \tilde{\mathbf{s}}_{\gamma_i}^j$;
- 8: Apply SVD by $\mathbf{R}_{\gamma_i} = \mathbf{U} \mathbf{\Delta} \mathbf{V}^*$;
- 9: Update the atom \mathbf{d}_i to be \mathbf{u}_1 ;
- 10: Update the sparse coefficients $\tilde{\mathbf{s}}_{\gamma_i}^i$ to be $\Delta_{1,1} \mathbf{v}_1$;
- 11: **end for**
- 12: **end while**

Ensure: Sub-dictionaries $\{\mathbf{D}_l\}$;

3 Extension of cK-SVD

The proposed idea not only leads to the CK-SVD method but also builds a framework for DL. In other words, the CK-SVD can still be extended for better performance. Next, we introduce two practical extensions.

3.1 Sparsity-wise CK-SVD

For the standard CK-SVD, we fix the sparsity level, i.e., the number of sparse coefficients, for each training sample. However, this may lead to underfitting or overfitting of sparse representation. To address this problem, we develop the sparsity-wise CK-SVD (SwCK-SVD). We employ multi-atoms to represent a cluster of features instead of a single atom. To obtain the SwCK-SVD from the CK-SVD, we set termination conditions to determine the number of used atoms for a training sample. The sparse recovery strategy is summarized in Algorithm 3. For each cluster, the sparse coding is realized via an iterative process. The parameter a_{\max} controls the maximum expected number of used atoms. In step 9, it is determined whether more atoms are required by examining if the residual is relative enough to the remained atoms. This operation allows the sparsity to be adaptive to each training sample, aiming to achieve a satisfied representation accuracy by using as few atoms as possible. The parameter ρ controls the threshold of the termination condition, and we empirically suggest $\rho = 0.4 \sim 0.6$.

3.2 Dynamic CK-SVD

Although the sparse coding strategies for the K-SVD, the CK-SVD, and the SwCK-SVD are different, their dictionary updating strategies are the same. They all use the

Algorithm 3 Sparse recovery for SwCK-SVD

Require: A training sample $\mathbf{z} \in \mathbb{R}^n$, maximum number of clusters l_{\max} , dictionary $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_{l_{\max}}] \in \mathbb{R}^{n \times q}$, threshold ϵ , parameters ρ , a_{\max} .

- 1: Initialize $\mathbf{r} = \mathbf{z}$, $l = 1$;
- 2: **while** $l < l_{\max}$ and $\|\mathbf{r}\|_2 > \epsilon$
- 3: Compute the inner product by $\mathbf{p} = \mathbf{D}_l^* \mathbf{r}$;
- 4: $\Omega = \emptyset$;
- 5: **for** $a = 1$ to a_{\max}
- 6: Find the maximum magnitude entry of \mathbf{p} , the index of which is denoted as ω ;
- 7: $\Omega = [\Omega, \omega]$;
- 8: Denote $\mathbf{B} = \mathbf{D}(:, \Omega)$;
- 9: Compute the LS coefficient by $\mathbf{c} = (\mathbf{B}^* \mathbf{B})^{-1} \mathbf{B}^* \mathbf{z}$;
- 10: Compute \mathbf{s}_l by $\mathbf{s}_l(\Omega) = \mathbf{c}$;
- 11: **if** $\max |\mathbf{p}| < \rho \|\mathbf{r}\|_2$ **then**
- 12: $\mathbf{r} \leftarrow \mathbf{r} - \mathbf{B} \mathbf{c}$;
- 13: Break the iteration;
- 14: **end if**
- 15: $\mathbf{r} \leftarrow \mathbf{r} - \mathbf{B} \mathbf{c}$;
- 16: **end for**
- 17: Update \mathbf{s} by $\mathbf{s} \leftarrow [\mathbf{s}; \mathbf{s}_l]$;
- 18: $l = l + 1$;
- 19: **end while**

Ensure: Sparse coefficients \mathbf{s} ;

first principal component of the SVD result to update dictionary and sparse coefficients (see steps 9 and 10 in Algorithm 2), while ignoring other components. Under the framework of CK-SVD, the first cluster contributes most to the representation, and later clusters make less contributions. For instance, it is possible that a second principal component of the SVD result of a residual R_{γ_i} in the first cluster, expressed as $u_2 \Delta v_2^*$, may be more significant than a first principal component of the SVD results of a residual in the second cluster. Based on this consideration, we extend the CK-SVD to the dynamic CK-SVD (DCK-SVD), for which the atoms of different clusters are refined after each iterative cycle. The dictionary updating strategy is provided in Algorithm 4. For each iterative cycle, we use the second component of the SVD results with respect to the most used atom in D_l , to replace the least used atom in the next cluster of dictionary. This operation makes the dictionaries dynamic that means the atoms will be refined after an iterative cycle and those contribute little to the representation will be abandoned.

4 Results

In this section, we provide the experimental results and analysis. The Berkeley dataset is employed for the exper-

Algorithm 4 DCK-SVD

Require: Training samples $Z \in \mathbb{R}^{n \times \xi}$, sparse level k , threshold ϵ .

- 1: Initialize each sub-dictionary $D_l \in \mathbb{R}^{n \times q_l}$, $l = 1, 2, \dots, k$, $D = [D_1 D_2 \dots D_k]$;
- 2: **while** the termination condition is not reached
- 3: Solve the sparse coefficients s_t for each training sample z_t by using Algorithm 1;
- 4: **for** each cluster l
- 5: **for** each atom d_i of D_l
- 6: Find the samples that use d_i , their indexes are denoted as γ_i ;
- 7: Derive Z_{γ_i} and S_{γ_i} from Z and S , respectively;
- 8: Compute the residual by $R_{\gamma_i} = Z_{\gamma_i} - \sum_{j \neq i} d_j s_{\gamma_i}^j$;
- 9: Apply SVD by $R_{\gamma_i} = U \Delta V^*$;
- 10: Update the atom d_i to be u_1 ;
- 11: Update the sparse coefficients $s_{\gamma_i}^j$ to be $\Delta_{1,1} v_1$;
- 12: **if** d_i is the most used atom in D_l **then**
- 13: Set $g_{l+1} = u_2$;
- 14: **end if**
- 15: **end for**
- 16: **end for**
- 17: Find the least used atom in D_l and replace it by g_l , $l = 2, 3, \dots, k$;
- 18: **end while**

Ensure: Sub-dictionaries $\{D_l\}$;

iments [18]. The experiments are organized as follows. First, we conducted the standard CK-SVD-based DL process and the conventional K-SVD-based DL process, respectively, by using the training dataset, in order to verify the improvement of the CK-SVD over the conventional K-SVD. Second, we applied different dictionaries to the compressive sensing, which is the typical application in the field of image processing. Besides the K-SVD and the standard CK-SVD, we also employed the SwCK-SVD, the DCK-SVD, and two existing methods extended from the K-SVD, proposed in [28] and [29], respectively.

4.1 Experiments on sparse representation

The Berkeley dataset contains various images, and they are composed of the training dataset and test dataset. For this part of experiments, we first used the training dataset to execute the DL process based on the proposed method and the conventional K-SVD method, respectively. We divided the training dataset into 19,194 patches with a size of 4×4 , i.e., $n = 16$, and set the initialized dictionaries to Gaussian random matrices. We considered different number of atoms, different maximum number of sparse coefficients, and different number of iterative cycles in this part of experiments. The parameters are the same for both methods in each set of experiments. After the DL process, we used the obtained dictionaries for the sparse representation of test dataset. The test dataset contains 100 test images with the size of 480×312 . The images are represented based on the trained dictionaries by being divided into the image patches. The accuracy is evaluated by the peak signal to noise ratio (PSNR), and the results are presented in Figs. 5, 6, and 7.

It could be noted that, with the same parameters, the dictionaries trained by the CK-SVD can provide more accurate sparse representations to the test images. As the number of iterations increases, the performance of dictionaries is improved. When the number of iterations exceeds 10, the increasing trend is slow. Similarly, the accuracy of sparse representations can benefit from the increase of q_0 the larger q_0 . However, the growth trend slows down ceaselessly, and a too large q_0 would increase the computational complexity of sparse coding and dictionary updating. The larger number of sparse coefficients could also improve the performance of dictionaries. But it is not suggested to set a too large k , as it would reduce the sparsity of images.

4.2 Applied to compressive sensing

Compressive sensing (CS) is the technique that compressively samples and reconstructs signals from fewer measurements, in order to reduce the cost of signal transmission and storage [19–23]. The signal x can be compressively sampled by a sensing matrix Φ expressed as:

$$y = \Phi x, \quad (8)$$

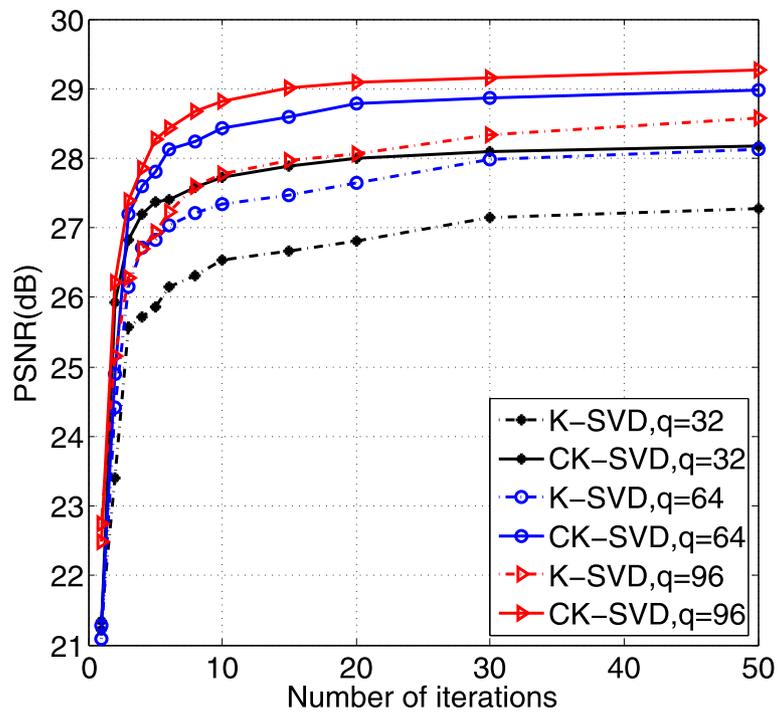


Fig. 5 PSNR of the sparse representations of test images versus the number of iterations, $k = 2$

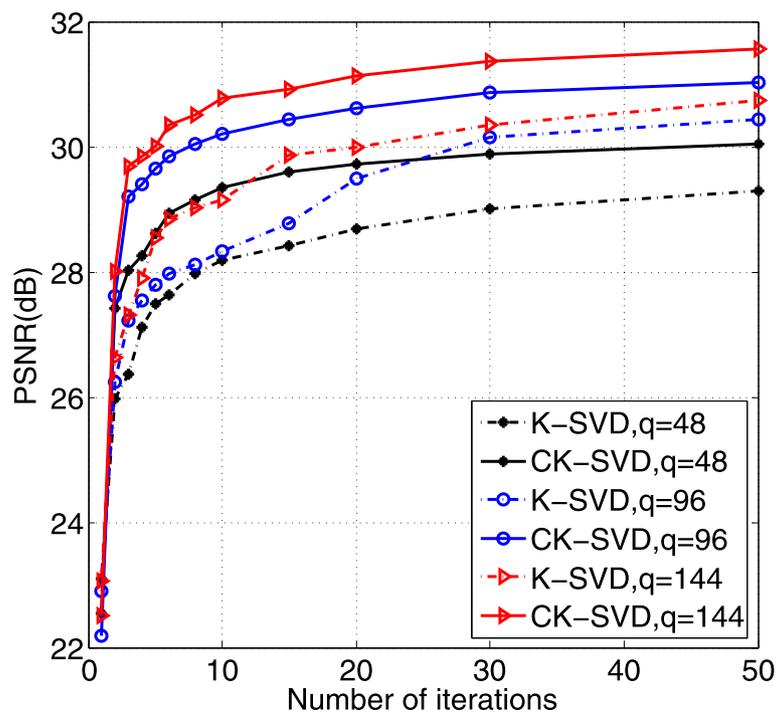


Fig. 6 PSNR of the sparse representations of test images versus the number of iterations, $k = 3$

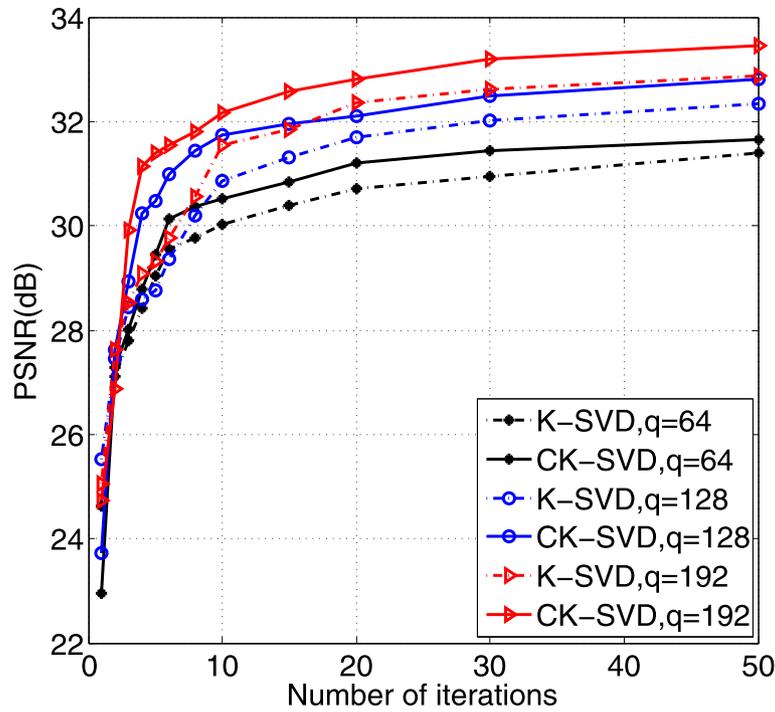


Fig. 7 PSNR of the sparse representations of test images versus the number of iterations, $k = 4$

where $\mathbf{y} \in \mathbb{R}^m$ is the measurements and $\Phi \in \mathbb{R}^{m \times n}$ with $m < n$. When \mathbf{x} denotes the vectorization of an image patch, its sparse representation in terms of the dictionary $\mathbf{D} \in \mathbb{R}^{n \times q}$ can be written as:

$$\mathbf{x} = \mathbf{D}\mathbf{s}, \quad (9)$$

where $\mathbf{s} \in \mathbb{R}^q$ is the sparse coefficients vector. The operation that recovers the sparse coefficients \mathbf{s} from measurements \mathbf{y} , sensing matrix Φ , and dictionary \mathbf{D} is referred to as reconstruction, which is expressed by:

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s}} \|\mathbf{s}\|_0 \quad s.t. \quad \mathbf{y} = \Phi \mathbf{D}\mathbf{s}. \quad (10)$$

This problem can be directly solved by greedy algorithms [17, 24–26], and then the original signal \mathbf{x} can be obtained by (9).

In this part of experiments, we considered the applications of the proposed methods on compressive sensing. We trained the dictionaries from the samples introduced in Section 4.1 using the compared methods. For the training process, we divided the training images into 36,000 patches with the sizes of 6×6 and 8×8 . For the size of 6×6 , i.e., $n = 36$, the maximum number of coefficients for all methods was set to 6 and the total number of atoms was set to 324; for the size of 8×8 , i.e., $n = 64$, the maximum number of coefficients for all methods was set to 9 and the total number of atoms was set to 576. The number of atoms for all compared method is the same. Hence, for the standard CK-SVD, we set the numbers of clusters to 6 and 9 as selected only 1 atom for each cluster. For the SwCK-SVD, we set the number of clusters to 3, and therefore, the maximum numbers of used atoms for

Table 2 Comparison of PSNR (dB) of reconstructed image “Elephant” with different number of measurements and different patch sizes

Methods	Patch size 6×6				Patch size 8×8			
	$m = 9$	$m = 12$	$m = 15$	$m = 18$	$m = 14$	$m = 20$	$m = 26$	$m = 32$
K-SVD	20.04	21.97	23.16	23.70	19.85	21.83	23.14	23.92
Method in [28]	19.89	21.90	22.82	23.53	19.78	21.66	22.91	23.77
Method in [29]	26.06	26.83	27.24	27.69	25.73	26.80	27.32	27.75
CK-SVD	27.23	27.89	28.38	28.60	26.95	27.74	28.40	28.81
SwCK-SVD	27.65	28.82	28.79	28.97	27.21	28.24	28.70	29.03
DCK-SVD	27.44	27.92	28.51	28.78	27.07	27.93	28.45	28.90

Table 3 Comparison of PSNR (dB) of reconstructed image “Horse” with different number of measurements

Methods	Patch size 6×6				Patch size 8×8			
	$m = 9$	$m = 12$	$m = 15$	$m = 18$	$m = 14$	$m = 20$	$m = 26$	$m = 32$
K-SVD	17.90	19.54	20.51	21.22	17.32	19.27	20.60	21.39
Method in [28]	17.95	19.53	20.44	21.08	17.40	19.43	20.55	21.36
Method in [29]	23.77	24.69	25.12	25.58	23.41	24.56	25.20	25.63
CK-SVD	24.79	25.26	26.01	26.34	24.33	25.02	25.94	26.30
SwCK-SVD	25.21	25.64	26.37	26.58	24.74	25.30	26.32	26.55
DCK-SVD	24.93	25.37	26.18	26.39	24.51	25.20	26.17	26.42

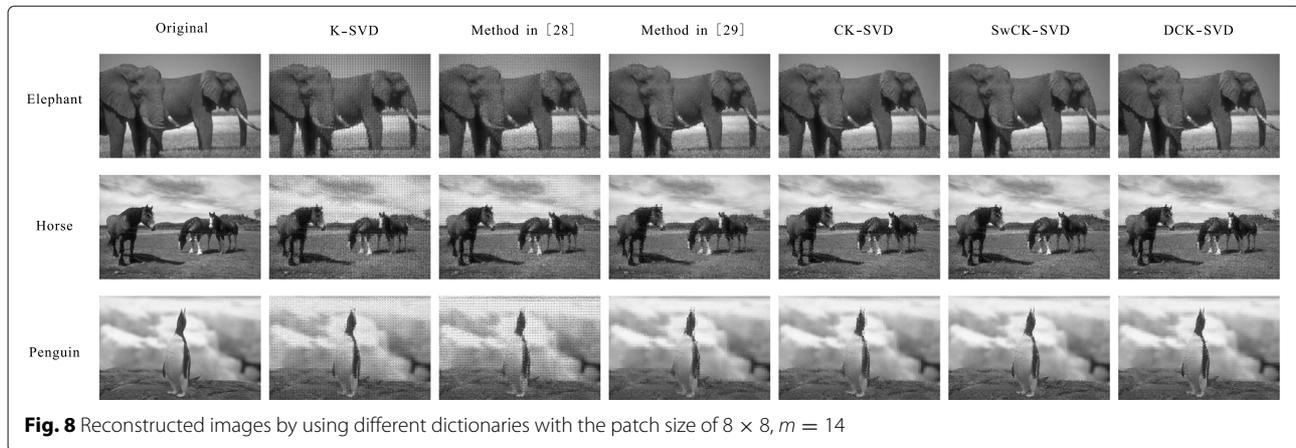
each cluster were 2 and 3 for different patch sizes. The maximum number of iterative cycles was set to 20 for all compared methods. We select three typical images, “Elephant,” “Horse,” and “Penguin” from the testing dataset as original images and divided them into a set of patches, the size of which was the same as training samples. Each patch was compressed by using the Gaussian random matrices with numbers of measurements, denoted as m . This technique is also referred as block-CS [27]. The measurements were then reconstructed by using the OMP algorithm, and their accuracy was measured by the PSNR. To reduce the influence of stochastic factors that were brought by the Gaussian random sensing matrices, we repeated each trial for 50 times. The average results are presented in Tables 2, 3, and 4.

The results demonstrate that the PSNR of the reconstructed images based on the dictionaries trained by the CK-SVD is much higher than those based on the dictionary trained by the conventional K-SVD, regardless of the original image, patch size, and the number of measurements. The reason is the CS greedy reconstruction requires that the number of measurements should be at least twice the number of sparse coefficients; if not, the primary sparse coefficients may not be completely reconstructed, and the reconstruction accuracy would be significantly influenced. On the other hand, the required number of sparse coefficients for accurate representa-

tion using the proposed method is much smaller than that using the conventional K-SVD, as the main features can be presented by only the coefficients and atoms of the first cluster. Taking the example of the experiments with the patch size of 8×8 , for the conventional K-SVD, the maximum number of sparse coefficients is 9. For the proposed methods, the number of cluster is 3, i.e., each cluster contains at most three coefficients. Then, when employed for CS reconstruction, for the dictionaries trained by the conventional K-SVD, maybe only when all nine sparse coefficients are found out would the image can be reconstructed accurately. But for the dictionaries trained by the proposed method, finding out only the coefficients of the first cluster, i.e., 3 coefficients, could lead to a satisfied reconstruction. Therefore, the proposed method has an advantage on CS reconstruction, especially for the low sampling ratio. As the extensions of CK-SVD, both the SwCK-SVD and the DCK-SVD show the improvement over the standard CK-SVD, benefiting from the adaptive sparsity and the dynamic refinement of atoms, respectively. The CK-SVD, the SwCK-SVD, and the DCK-SVD also outperform the methods proposed in [28] and [29]. Besides the PSNR comparison, we also provide the visual comparison of the results. The reconstructed images based on the patch size of 8×8 with $m = 14$ and $m = 32$ are presented in Figs. 8 and 9. It can be noted that the quality of reconstructed images based on the proposed

Table 4 Comparison of PSNR (dB) of reconstructed image “Penguin” with different number of measurements

Methods	Patch size 6×6				Patch size 8×8			
	$m = 9$	$m = 12$	$m = 15$	$m = 18$	$m = 14$	$m = 20$	$m = 26$	$m = 32$
K-SVD	17.85	19.42	20.40	21.16	17.02	19.21	20.27	21.08
Method in [28]	17.53	19.37	20.34	21.12	17.06	19.20	20.21	21.10
Method in [29]	25.45	26.69	27.40	28.02	25.13	26.50	27.22	27.97
CK-SVD	27.48	28.79	29.61	30.34	27.19	28.60	29.65	30.53
SwCK-SVD	27.87	29.22	29.94	30.63	27.63	29.00	29.98	30.86
DCK-SVD	27.60	29.01	29.75	30.42	27.44	28.80	29.72	30.55



methods is obviously higher than those based on the conventional K-SVD and the methods proposed in [28] and [29].

Besides the Gaussian-initialized dictionary, we also used the discrete cosine transform (DCT) basis and pre-learned dictionary as the initialized dictionary for experiment. For the DCT case, we first generated an over-complete DCT basis containing the same amount of atoms as the Gaussian-initialized dictionary. Then, we employed all the compared methods to conduct the DL processes. The learned dictionaries were utilized for CS reconstruction of the “Horse” image. Other experimental settings remained unchanged. The results are provided in Table 5.

For the experiment of pre-learned dictionary, we firstly chose 10,000 patches for per-training using the conventional K-SVD. Then, the pre-learned dictionary was trained again by another 10,000 patches using all compared methods. The initialized dictionaries were the over-complete DCT basis. Other experimental setting were the same as those used in the previ-

ous CS reconstruction experiments. The PSNR of the reconstructed “Horse” image using different dictionaries (including the pre-learned dictionaries) was summarized in Table 6.

The results in Table 5 demonstrate the over-complete DCT basis can also be utilized as the initialized dictionary for the proposed DL methods. Comparing to the Gaussian random-initialized dictionary, the DCT-initialized dictionary provide better performance of accuracy, regardless of the DL methods. In Table 6, it can be noted that the pre-learned dictionaries can be still trained by new samples with or without using the online DL methods, and the performance is improved obviously after the new training process. Similarly, the proposed methods outperforms the conventional methods.

4.3 Applied to image denoising

Besides the CS, we also performed the experiments to apply the proposed method to image denoising. Two

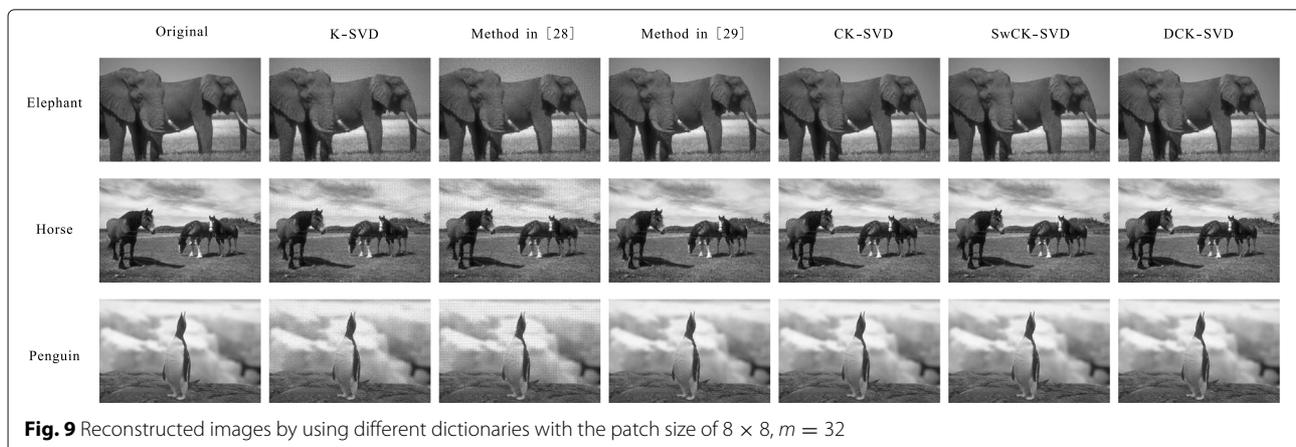


Table 5 Comparison of PSNR (dB) of reconstructed image “Horse” using the DCT basis for initialization

Methods	Patch size 6×6				Patch size 8×8			
	$m = 9$	$m = 12$	$m = 15$	$m = 18$	$m = 14$	$m = 20$	$m = 26$	$m = 32$
K-SVD	19.31	20.98	22.25	23.76	19.24	21.16	22.50	24.07
Method in [28]	19.42	21.13	22.30	23.82	19.49	21.32	22.67	24.28
Method in [29]	24.67	25.83	26.34	26.60	24.22	25.37	26.18	26.49
CK-SVD	25.55	26.16	26.79	27.20	25.41	25.98	26.62	27.03
SwCK-SVD	25.87	26.49	27.07	27.36	25.70	26.28	26.91	27.34
DCK-SVD	25.69	26.27	26.95	27.12	25.60	26.04	26.75	27.11

images were selected for this part of experiments. The first is the “Koala” image from the Berkeley image dataset with the size of 320×480 , and it has been introduced in Section 2.1. The other is the standard test image “Pepper” with the size of 512×512 . The experimental setup is described as follows.

- We first added the Gaussian white noise to the original images with zero mean and the variance of σ .
- We selected 20,000 patches and 40,000 patches of the noised “Koala” image and the noised “Pepper” image, respectively. The size of all patches was 8×8 .
- We used the vectorization of these patches to conduct the DL process for two test images.
- We employed the learned dictionary for denoising by using the strategy given in [32].

We initialized all dictionaries as the over-complete DCT basis with the size of 64×256 and set the maximum of sparse coefficients for each training sample to 6. The maximum number of iterative cycles was set to 10. We set the noise level to be $\sigma = 15$ and $\sigma = 20$. For each noise level, we repeated the above process. More detailed description of the denoising experiments can be found in [32]. The conventional K-SVD, the methods introduced in [28] and [29], and the proposed CK-SVD were employed for com-

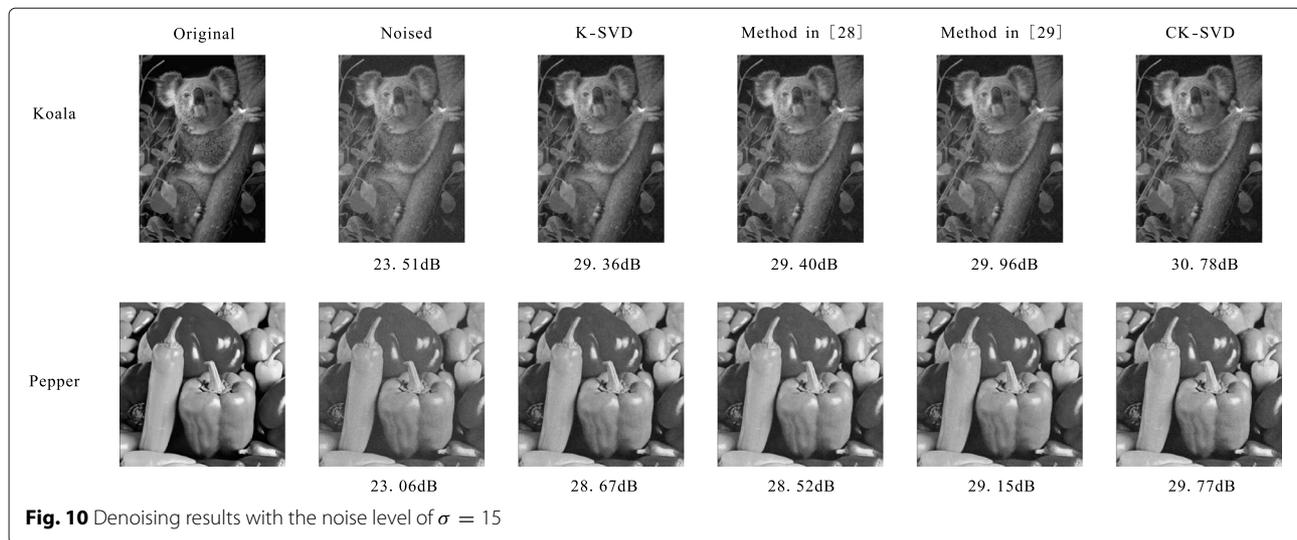
parison. The results are presented in Figs. 10 and 11. It can be noted from the results that the dictionaries trained by the proposed CK-SVD provide to the most accurate denoised images, regardless of the original images and the noise levels.

5 Discussion

As mentioned in Section 3, this study not only develops a K-SVD-based method but also provides the clustering DL model. The potential and advantage of the clustering model mainly come from two aspects. First, different cluster of dictionaries is isolated from each other. Thus, an atom of learned dictionary could concentrate on a specific type of feature, leading greater utilization of atoms. In other words, a common phenomenon in the conventional DL model can be avoid, that is, a part of atoms is widely employed by training samples whereas others are seldom used. Second, the clustering DL model makes it possible to adjust the sparsity based on different training samples and therefore to reduce the underfitting of overfitting of sparse representation. We provide the SwCK-SVD by adaptively selected the number of used atoms for each cluster. It is believed that the adaptive strategy can also be implemented by adjusting the number of clusters. This potential is verified by the fact the SwCK-SVD performs obviously better than the standard CK-SVD.

Table 6 Comparison of PSNR (dB) of reconstructed image “Horse” using the pre-learned dictionaries for initialization

Methods	Patch size 6×6				Patch size 8×8			
	$m = 9$	$m = 12$	$m = 15$	$m = 18$	$m = 14$	$m = 20$	$m = 26$	$m = 32$
Pre-learned	18.72	19.89	21.36	22.53	18.66	19.90	21.51	22.74
K-SVD	19.23	20.79	22.07	23.15	19.10	20.99	22.28	23.87
Method in [28]	19.26	20.94	22.01	23.04	19.20	21.18	22.49	23.35
Method in [29]	22.45	23.62	24.17	24.43	22.02	23.08	23.96	24.30
CK-SVD	23.32	23.90	24.48	24.85	23.07	23.69	24.31	24.78
SwCK-SVD	23.61	24.14	24.73	25.12	23.39	23.98	24.65	25.03
DCK-SVD	23.47	24.02	24.60	24.99	23.22	23.86	24.54	24.95

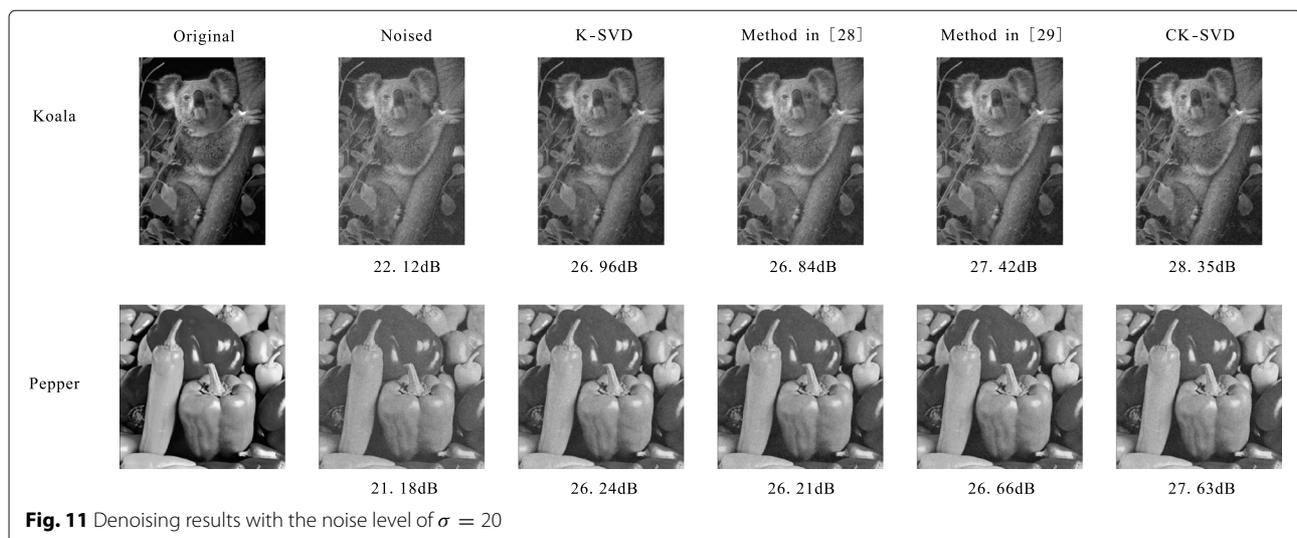


Future work could consider extending the clustering DL model to online learning. In this study, we focus on the batch DL and the dictionary updating strategy is based on the SVD. We believe the proposed clustering DL model is not limited to batch DL but can be extended to online DL problem. In [7], the standard ODL method is proposed, for which the information-storing variables are updated when a new training sample inputs. The variables are then used to update the learned dictionary through an optimization approach. For clustering DL, we may utilize a set of information-storing variables for different cluster of dictionaries. When a new sample inputs, we could employ Algorithm 1 or Algorithm 3 to solve the sparse coefficients with respect to different cluster of sub-dictionaries. Then, the sparse coefficients are used to update different cluster of information-storing variables. Finally, we could update the sub-dictionaries based on

the information-storing variables, such that the clustering ODL is achieved. We believe the clustering ODL has the potential to be applied in the cases where training samples cannot be obtained simultaneously.

6 Conclusions

We proposed a DL method named CK-SVD for sparse representation of images. For CK-SVD, the atoms of dictionary are divided into a set of groups, and each group of atoms serve for image features of a specific cluster. Hence, the features of all clusters can be utilized and the redundant atoms are avoided. Based on this strategy, we introduced the CK-SVD and two practical extensions. Experimental results demonstrated that the proposed methods could provide more accurate sparse representation of images, compared to the conventional K-SVD algorithm and its extended methods.



Ethical approval and consent to participate

Not applicable.

Abbreviations

CK-SVD: Clustering K-singular value decomposition; CS: Compressive sensing; DCK-SVD: Dynamic CK-SVD; DL: Dictionary learning; K-SVD: K-singular value decomposition; LS: Least square; MOD: Method of optimal directions; OMP: Orthogonal matching pursuit; PSNR: Peak signal-to-noise ratio; SwCK-SVD: Sparsity-wise CK-SVD

Acknowledgements

The authors would like to thank the National Key R&D Program of China and the National Natural Science Foundation of China for the financial support.

Authors' contributions

JF provided the methodology. RZ wrote the original manuscript. HY reviewed and edited the manuscript. JF and LR funded this study. All authors read and approved the final manuscript.

Funding

This work was supported by the National Key R&D Program of China under Grant 2017YFD0700302 and by the National Natural Science Foundation of China under Grant 51705193.

Availability of data and materials

Please contact the authors for data requests.

Consent for publication

This manuscript does not contain any individual person's data in any form.

Competing interests

The authors declare that they have no competing interests.

Received: 7 May 2019 Accepted: 2 October 2019

Published online: 25 October 2019

References

- Rubinstein, A. M., Bruckstein, M., Elad, M.: Dictionaries for sparse representation modeling. *Proc. IEEE*. **98**(6), 1045–1057 (2010)
- Xu, L., Wang, D., Shi, W., Deng, D.: Group-based single image super-resolution with online dictionary learning. *EURASIP J. Adv. Signal Process.* **2016**(84), 1–12 (2016)
- Naumova, V., Schnass, K.: Fast dictionary learning from incomplete data. *EURASIP J. Adv. Signal Process.* **2018**(12), 1–21 (2018)
- Zhang, L., Zuo, W., Zhang, D.: LSDT: latent sparse domain transfer learning for visual adaptation. *IEEE Trans. on Image Process.* **25**(3), 1177–1191 (2016)
- Engan, S. O., Aase, J. H., Husøy, J.: Multi-frame compression: theory and design. *EURASIP Signal Process.* **90**(2), 2121–2140 (2000)
- Aharon, M., Elad, A., Bruckstein, M.: The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.* **54**(11), 4311–4322 (2006)
- Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.* **11**, 19–60 (2010)
- Dumitrescu, B., Irofti, P.: Regularized K-SVD. *IEEE Signal Process. Lett.* **24**(3), 309–313 (2017)
- Nazzari, M., Yeganli, H., Ozkaramanli, A.: A strategy for residual component-based multiple structured dictionary learning. *IEEE Signal Process. Lett.* **22**(11), 2059–2063 (2015)
- Pant, J. K., Krishnan, S.: Compressive sensing of electrocardiogram signals by promoting sparsity on the second-order difference and by using dictionary learning. *IEEE Trans. Biomed. Circuits Syst.* **8**(2), 293–302 (2014)
- Smith, L. N., Elad, M.: Improving dictionary learning: multiple dictionary updates and coefficient reuse. *IEEE Signal Process. Lett.* **20**(1), 79–82 (2013)
- Zhao, R., Wang, Q., Shen, Y., Li, J.: Multidimensional dictionary learning algorithm for compressive sensing-based hyperspectral imaging. *J. Electron. Imaging.* **25**(6), 063013 (2016)
- Skretting, K., Engang, R.: Recursive least squares dictionary learning algorithm. *IEEE Trans. Signal Process.* **58**(4), 2121–2130 (2010)
- Tropp, J. A.: Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Inf. Theory.* **50**(10), 2231–2242 (2004)
- Candès, E. J., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory.* **52**(2), 489–509 (2006)
- Candès, E. J., Tao, T.: Decoding by linear programming. *IEEE Trans. Inf. Theory.* **51**(12), 4203–4215 (2005)
- Tropp, J. A., Gilbert, A. C.: Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theory.* **53**(12), 4655–4666 (2007)
- Martin, D., Fowlkes, C., Tal, J., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics (IEEE, Vancouver, 2001), pp. 416–423
- Donoho, D. L.: Compressed sensing. *IEEE Trans. Inf. Theory.* **52**(4), 1289–1306 (2006)
- Candès, E. J.: Compressive sampling. *Int. Congress of Mathematicians, Madrid, Spain.* **3**, 1433–1452 (2006)
- Massa, A., Rocca, P., Oliveri, G.: Compressive sensing in electromagnetics - a review. *IEEE Antenn. Propag. Mag.* **57**(1), 224–238 (2015)
- Craven, D., McGinley, B., Kilmartin, L., Glavin, E., Jones, E.: Compressed sensing for bioelectric signals: a review. *IEEE J. Biomed. Health Inf.* **19**(2), 539–540 (2015)
- Zhang, Y., Zhang, L. Y., et al.: A review of compressive sensing in information security field. *IEEE Access.* **4**, 2507–2519 (2016)
- Nion, D., Sidiropoulos, N. D.: Tensor algebra and multidimensional harmonic retrieval in signal processing for MIMO radar. *IEEE Trans. Signal Process.* **58**(11), 5693–4705 (2010)
- Dai, W., Milenkovic, O.: Subspace pursuit for compressive sensing signal reconstruction. *IEEE Trans. Inf. Theory.* **55**(5), 2230–2249 (2009)
- Donoho, D. L., Tsaig, Y., Drori, I., Starck, J. L.: Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *IEEE Trans. Inf. Theory.* **58**(2), 1094–1121 (2012)
- Gan, L.: *Proc. IEEE Int. Conf. Digit. Signal Process.* Block compressed sensing of natural images (IEEE, Wales, 2007), pp. 403–406
- Rubinstein, R., Zibulevsky, M., Elad, M.: Efficient Implementation of the K-SVD Algorithm Using Batch Orthogonal Matching Pursuit. Technical Report CS-2008-08 (Technion University, Haifa, 2008)
- Mairal, J., Sapiro, G., Elad, M.: Learning multi-scale sparse representations for image restoration. *Multiscale Model. Simul.* **7**(1), 214–241 (2008)
- Yi, Y., Cheng, C., Xu, C.: Visual tracking based on hierarchical framework and sparse representation. *Multimed. Tools Appl.* **77**(13), 16267–16289 (2018)
- Tariyal, S., Majumdar, R., Singh, M., Vatsa, D.: Deep dictionary learning. *Multimed. Tools Appl.* **4**, 10096–10109 (2016)
- Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.* **15**(12), 3736–3745 (2006)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)