Structured Kernel Dictionary Learning With Correlation Constraint for Object Recognition

Zhengjue Wang, Yinghua Wang, Hongwei Liu, *Member, IEEE*, and Hao Zhang

***Abstract*— In this paper, we propose a new discriminative non-linear dictionary learning approach, called correlation constrained structured kernel KSVD, for object recognition. The objective function for dictionary learning contains a reconstructive term and a discriminative term. In the reconstructive term, signals are implicitly non-linearly mapped into a space, where a structured kernel dictionary, each sub-dictionary of which lies in the span of the mapped signals from the corresponding class, is established. In the discriminative term, by analyzing the classification mechanism, the correlation constraint is proposed in kernel form, constraining the correlations between different discriminative codes, and restricting the coefficient vectors to be transformed into a feature space, where the features are highly correlated inner-class and nearly independent between-classes. The objective function is optimized by the proposed structured kernel KSVD. During the classification stage, the specific form of the discriminative feature is needless to be known, while the inner product of the discriminative feature with kernel matrix embedded is available, and is suitable for a linear SVM classifier. Experimental results demonstrate that the proposed approach outperforms many state-of-the-art dictionary learning approaches for face, scene, and synthetic aperture radar vehicle target recognition.**

***Index Terms*— Correlation constraint, discriminative code, kernel method, KSVD, non-linear dictionary learning.**

I. Introduction

SPARSE-representation-based dictionary learning (DL) methods have recently drawn much attention in image classification, such as face recognition [1]–[6], digit classification [5]–[8], and texture classification [6]–[9], etc. This is mainly due to the fact that most natural signals can be sparsely approximated as the combinations of a few items from a certain dictionary.

Linear DL is designed under the frame of linear sparse representation, i.e., a signal $\mathbf{y} \in \mathbb{R}^{\alpha}$ can be sparsely represented by the learned dictionary $\mathbf{D} \in \mathbb{R}^{\alpha \times K}$ as $\mathbf{y} \approx \mathbf{D}\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^{K}$ is the sparse coefficient vector. Linear DL can be well adapted to a certain task, such as reconstruction, classification, and so on.

Manuscript received October 19, 2016; revised April 19, 2017 and June 4, 2017; accepted June 5, 2017. Date of publication June 21, 2017; date of current version July 11, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61671354 and in part by the National Science Fund for Distinguished Young Scholars of China under Grant 61525105. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vishal Monga. *(Corresponding author: Yinghua Wang.)*

The authors are with the National Laboratory of Radar Signal Processing, Xidian University, Xi'an 710071, China, and also with the Collaborative Innovation Center of Information Sensing and Understanding, Xidian University, Xi'an 710071, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2718187

The classical task of linear DL is reconstruction. Several methods have been developed considering the minimization of the reconstruction error only, e.g., the K-means [10], the K-SVD [11], and the method of optimal directions (MOD) [12]. All these methods are unsupervised which do not utilize the class information of the training set. The supervised linear DL is believed to have better reconstructive ability [13]. The basic idea is to learn a class-specific sub-dictionary for each class independently, so that each sub-dictionary can well represent the signals from the corresponding class. For the task of classification, the classical reconstructive linear DL is extended in different ways. Raina *et al.* [14] learn a dictionary from an unlabeled dataset, and then the sparse coefficient vectors of the labeled dataset over the learned dictionary are treated as features to train an SVM classifier. In [15], the classification is completed by finding the class with the minimum of reconstruction errors associated with different sub-dictionaries. In [1], after concatenating all the sub-dictionaries into a universal structured dictionary, the sparse representation-based classification (SRC) [16] is used. Such kind of methods implies that signals from the same class lie in the same subspace. However, signals from different classes often have relevant components, which may cause common components existing in several sub-dictionaries and result in misclassification further. In addition, there are no explicit terms relating to the classification performance in the objective function.

To address the aforementioned problems, discriminative linear DL is designed by adding some discriminative terms into the objective function, in order to directly serve for the classification task. The added discriminative terms can be mainly categorized into two kinds. The first kind directly incorporates a term relating to the classification performance into the objective function. And the other one imposes constraints on signals, coefficient vectors or the dictionary, in order to find more discriminative features, and indirectly improve the classification performance.

In the first class, the representation ability of the dictionary and the performance of the classifier are considered at the same time. Thus the dictionary would match the classifier better. A logistic loss function for 2-class or a softmax loss function for multiclass are introduced in [7], [15], and [17]. A classification error of a linear classifier is presented in [2], [18], and [19].

In the second class, several discriminative terms are developed to extract discriminative features. The strategies are mainly focused on five sub-classes. The first sub-class is to reduce the correlations among different sub-dictionaries, which is expressed by an incoherence promoting term

introduced in [6] and [20]. The second sub-class arises from the point that signals should have nearly zero coefficients over the sub-dictionaries of other classes, which is expressed by a discriminative term presented in [3], [5], and [21]. Using Fisher discrimination criterion, by minimizing the within-class scatter and maximizing the between-class scatter at the same time, is the third sub-class, which has been imposed on the coefficient vectors [5], or the projected input signals [4]. Besides, this criterion is also combined with the local affinity of data to improve the discrimination of the coefficient vectors in [22]. Compared with the original data space, the mapped space of the data are usually believed to reflect the underlying data structure better, such as the data separability of different classes. Based on this, in the fourth sub-class, several kinds of transformations are designed which are jointly optimized with the dictionary. In [7] and [23], a bilinear transformation acting on the input signals and the coefficient vectors is required to match the classifier better. In [4] and [8], a linear projection matrix is proposed to map the input signals into a space with better class separability. In [18], a term called "discriminative sparse-code error" is proposed restricting the coefficient vectors to be linearly projected into a more separable space. Gangeh *et al.* [9], use Hilbert Schmidt independence criterion to project the input signals into a space where the dependency between the signals and the class labels is maximized. In the fifth sub-class, structured dictionaries are learned via incorporating the discriminative structure information provided by different sparsity regularizations, e.g., group sparsity [24] and hierarchical group sparsity [25].

All the DL methods mentioned above are linear ones. However, recent researches have shown that linear representation is inadequate for revealing the non-linear structure of the data [26], [27]. Motivated by the successful application of kernel trick in SVM [28], KPCA [29], and KICA [30], etc., non-linear DL implicitly maps the original data into a high or even infinite dimensional feature space with a Mercer kernel, and establishes linear sparse representation in the mapped space, which gives a non-linear effect with respect to the original space. Non-linear DL approaches have shown better performance than their linear counterparts [26], [31]–[35]. Nguyen *et al.* [26] propose a reconstructive non-linear DL model whose dictionary atoms lie within the column subspace of the mapped signals, which is optimized through the use of Mercer kernels instead of the mapped signals. As revealed by the experimental results in [26], the approach in [26] show better performance than linear DL approaches in [11] and [12], even than discriminative linear DL approaches in [2] and [18]. In [35], virtual samples are obtained via singular value decomposition (SVD) upon the kernel matrix, which are non-linear representations of the original data that can be used as inputs for the existing linear DL model to perform non-linear effects.

With a similar consideration to the discriminative linear DL, discriminative non-linear DL is a new topic aiming at constructing non-linear representation serving for the task of classification directly. Just a few works are presented. In [36], the signals are non-linearly mapped using kernel trick, and then, a projection matrix w.r.t. the mapped signals, and a structured dictionary are optimized such that in the projected low dimensional space the between-class reconstruction residual is maximized and the within-class reconstruction residual is minimized. Shrivastava *et al.* [31] propose a discriminative non-linear DL method by extending the incoherence promoting term and a Fisher discriminative term into non-linear ones, and utilize reconstruction errors for classification. However, each sub-dictionary in [31] and [36] is assumed to lie in the span of the mapped input signals from all classes, which results in weak connections between sub-dictionaries and class labels.

Motivated by the advantages of non-linear dictionary, and considering the problems in the existing approaches, we propose a novel discriminative non-linear DL method, called correlation constrained structured kernel KSVD (CCSK-KSVD), for object recognition. The objective function for DL contains a reconstructive term and a discriminative term. In the reconstructive term, a structured kernel dictionary is designed, each sub-dictionary of which lies in the span of the mapped signals from the corresponding class. Thus the sub-dictionaries have closer links with the class labels compared with [31] and [36]. Besides, the structured kernel dictionary is optimized as a whole, different from [26] and [31] learning sub-dictionaries class by class. In the discriminative term, the coefficient vectors are desired to be transformed into a feature space which is composed of discriminative codes. According to the classification mechanism, the realization of classification owes not to any single feature but to the correlations between different features. So, the correlation constraint is proposed, constraining the correlations between different discriminative codes, and restricting the features to be highly correlated inner-class and nearly independent between-classes, which matches the task of classification better. Thus, the specific forms of the discriminative codes are more flexible, in contrast to the methods in [18] and [19] setting the specific form of every single code. Besides, the transformation matrix is constructed in a way similar to the structured kernel dictionary, which has a non-linear effect and is different from other transformation matrices in [9], [18], and [19]. Given the structure of the dictionary and inspired by the algorithms in [2], [18], [19], and [37], the objective function is optimized by the proposed structured kernel KSVD (SK-KSVD) algorithm which is developed from KSVD [11] and kernel KSVD [26].

The rest of this paper is organized as follows. In Section 2, we formulate CCSK-KSVD. The optimization of the proposed framework is presented in Section 3. The classification scheme is introduced in Section 4. Section 5 presents experimental results on several publicly available databases for object recognition. Finally, Section 6 concludes this paper.

II. Problem Formulation

Given a set of M input signals $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M] = [\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_N] \in \mathbb{R}^{\alpha \times M}$, where $\mathbf{Y}_n \in \mathbb{R}^{\alpha \times m_n}$ is a sub-set containing m_n input signals from class n, $M = \sum_{n=1}^{N} m_n$, α is the dimension of each signal and N is the total number of classes. Denote $\mathbf{h} = [h_1, h_2, \ldots, h_M] \in \mathbb{R}^M$, where $h_i \in \{1, 2, \ldots, N\}$ is the class label of the input signal $\mathbf{y}_i$.

Let $\Phi : \mathbb{R}^{\alpha} \rightarrow \mathcal{F} \subset \mathbb{R}^{\tilde{\alpha}}$ be a non-linear mapping from $\mathbb{R}^{\alpha}$ to a dot product space $\mathcal{F}$ which is a higher dimensional feature space, i.e., $\tilde{\alpha}$ is often much larger than α, and possibly infinite. All the formulations are restricted to Hilbert spaces so that Mercer kernels can be employed to carry out the mapping implicitly. A Mercer kernel $\kappa(\mathbf{y}_1, \mathbf{y}_2) : \mathbb{R}^{\alpha} \times \mathbb{R}^{\alpha} \rightarrow \mathbb{R}$ is a function defined as $\kappa(\mathbf{y}_1, \mathbf{y}_2) = \langle \Phi(\mathbf{y}_1), \Phi(\mathbf{y}_2) \rangle$ that satisfies Mercer's condition [28]: For all the data $\{\mathbf{y}_i\}_{i=1}^{M}$, the function gives rise to a positive semidefinite matrix $\left[\mathcal{K}_{i,j}\right] = \left[\kappa\left(\mathbf{y}_i, \mathbf{y}_j\right)\right]$. Some commonly used kernel functions are the Gaussian kernel $\kappa(\mathbf{y}_1, \mathbf{y}_2) = \exp\left(-\frac{\|\mathbf{y}_1 - \mathbf{y}_2\|^2}{\delta}\right)$ and the polynomial kernel $\kappa(\mathbf{y}_1, \mathbf{y}_2) = (\langle \mathbf{y}_1, \mathbf{y}_2 \rangle + \phi)^{\eta}$, where δ, ϕ and η are parameters.

Our goal is to obtain a more discriminative representation of the input signals, namely obtaining discriminative features with better separability. This is achieved by solving the following optimization problem.

$$\langle \mathbf{D}, \mathbf{B}, \mathbf{X} \rangle = \arg\min_{\mathbf{D},\mathbf{B},\mathbf{X}} \|\Phi(\mathbf{Y}) - \mathbf{D}\mathbf{X}\|_F^2 + \lambda \|\Upsilon(\mathbf{Y}) - \mathbf{B}\mathbf{X}\|_F^2 \quad s.t.\ \forall i, j\ \|\mathbf{x}_i\|_0 \leq T,\ \left\|\mathbf{d}_j\right\|_2 = 1,\ \left\|\mathbf{b}_j\right\|_2 = 1, \tag{1}$$

where $\Phi(\mathbf{Y}) = \left[\Phi(\mathbf{y}_1), \Phi(\mathbf{y}_2), \ldots, \Phi(\mathbf{y}_M)\right] \in \mathbb{R}^{\tilde{\alpha} \times M}$, $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K] \in \mathbb{R}^{\tilde{\alpha} \times K}$ is the sought dictionary, $\mathbf{X} \in \mathbb{R}^{K \times M}$ is the coefficient matrix, $\Upsilon(\mathbf{Y}) = [\Upsilon(\mathbf{y}_1), \Upsilon(\mathbf{y}_2), \ldots, \Upsilon(\mathbf{y}_M)] \in \mathbb{R}^{\beta \times M}$ is a discriminative coding matrix mapped by another non-linear mapping $\Upsilon : \mathbb{R}^{\alpha} \rightarrow \mathcal{G} \subset \mathbb{R}^{\beta}$, whose coding scheme is associated with the labels of the input signals, $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_K] \in \mathbb{R}^{\beta \times K}$ is the sought transformation matrix, $\mathbf{x}_i$ is the i-th column of $\mathbf{X}$, T is the sparsity level, λ is the scalar parameter to balance the two terms, $\|\cdot\|_0$ is the l_0-norm defined as the number of non-zero elements in a vector, $\|\cdot\|_F$ is the Frobenius norm.

The reconstructive term and the discriminative term in model (1), and the discriminative feature will be discussed in more details in the following.

A. Reconstructive Term

In order to promote the reconstructive and discriminative abilities of the dictionary, $\mathbf{D}$ is designed to be a structured dictionary, i.e., $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_N]$, where $\mathbf{D}_n \in \mathbb{R}^{\tilde{\alpha} \times L}$ is the class-specific sub-dictionary associated with class n, and $NL = K$. $\mathbf{D}_n$ is expected to well represent $\Phi(\mathbf{Y}_n)$. As stated in [26], $\mathbf{D}_n$ should lie in the span of signals in $\Phi(\mathbf{Y}_n)$ and there exists a coefficient matrix $\mathbf{F}_n \in \mathbb{R}^{m_n \times L}$ such that

$$\mathbf{D}_n = \Phi(\mathbf{Y}_n)\mathbf{F}_n. \tag{2}$$

Therefore, the reconstructive term can be re-written as follows:

$$\|\Phi(\mathbf{Y}) - \mathbf{D}\mathbf{X}\|_F^2 \tag{3}$$

$$= \|\Phi(\mathbf{Y}) - [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_N]\mathbf{X}\|_F^2 \tag{4}$$

$$= \|\Phi(\mathbf{Y}) - [\Phi(\mathbf{Y}_1)\mathbf{F}_1, \Phi(\mathbf{Y}_2)\mathbf{F}_2, \ldots, \Phi(\mathbf{Y}_N)\mathbf{F}_N]\mathbf{X}\|_F^2 \tag{5}$$

$$= \|\Phi(\mathbf{Y}) - \Phi(\mathbf{Y})\mathbf{F}\mathbf{X}\|_F^2, \tag{6}$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 & & & \mathbf{0} \\ & \mathbf{F}_2 & & \\ & & \ddots & \\ \mathbf{0} & & & \mathbf{F}_N \end{bmatrix}.$$

Hence, the optimal dictionary $\mathbf{D}$ could be found out by optimizing $\mathbf{F}$. Since the size of $\mathbf{F}$ in column equals to that of $\mathbf{D}$, $\mathbf{F}$ can be regarded as a pseudo-dictionary.

There are two issues should be mentioned. First, the sub-dictionaries in our approach are different from those in [31], [32], and [36] which lie in the span of the mapped input signals from all classes. Second, the structured dictionary in (5) is not a simple concatenation of the sub-dictionaries as that in [26]. Because, the sub-dictionaries in (5) are treated as a whole, i.e., a structured dictionary, to represent the training signals from all classes, such that each atom is updated utilizing the entire training set; whereas, the sub-dictionaries in [26] are trained separately, only utilizing the signals from the corresponding class.

B. Discriminative Term

In order to make the dictionary $\mathbf{D}$ more discriminative, a discriminative term with correlation constraint is imposed on the coefficient matrix $\mathbf{X}$ of $\Phi(\mathbf{Y})$ over $\mathbf{D}$, i.e., the second term in (1).

Motivated by the approaches in [7], [9], [18], [19], and [23] transforming the coefficient vectors into a more discriminative space, in our model, a transformation matrix $\mathbf{B}$ is desired to map the coefficient vectors into a feature space which is composed by discriminative codes. So, there are two key issues in the discriminative term: the first one is which kind of properties the feature space as well as the discriminative codes should possess, and the second one is which kind of mapping is to be chosen.

In former studies, the feature space is constrained by elaborately designing the specific form of every discriminative code. Here, we denote this kind of discriminative codes as $\{\tilde{\mathbf{q}}_i\}_{i=1}^{M}$. As stated in [2], [18], and [19], $\tilde{\mathbf{q}}_i$ can be defined as a one-hot vector, or a little bit complicated one such as discriminative sparse code associated with labels or dictionary atoms, or others with discriminability. However, this kind of coding scheme emphasize the specific form of every single code which has relatively weak connections with classification mechanism. Because, the realization of classification owes not to any single code but to the correlations between different codes, distance measure or orthogonality, for instance. In other words, the discriminability of codes will finally and actually be reflected in the correlations between one code to another.

Based on these considerations, we propose a general coding scheme from a novel perspective by directly constraining the correlations between different codes instead of designing specific forms of every single code. During the training stage, the correlations between the training codes are used for the unknown matrices learning. While, during the testing stage, the correlations between testing codes and training codes can serve the task of classification directly. More specifically, one realization of the correlation constraint is as follows: For

signals from the same class, the directions of the corresponding codes should be coincident; however, for signals from different classes, the corresponding codes should be mutually orthogonal. Denoting such discriminative codes as $\{\mathbf{q}_i\}_{i=1}^M$, the correlation constraint can be formally stated as follows:

$$\begin{cases} \langle \mathbf{q}_i, \mathbf{q}_j \rangle = 1, & \text{if } h_i = h_j \quad \forall i, j, \\ \langle \mathbf{q}_i, \mathbf{q}_j \rangle = 0, & \text{if } h_i \neq h_j \quad \forall i, j. \end{cases} \tag{7}$$

Compared with the former discriminative codes $\{\tilde{\mathbf{q}}_i\}_{i=1}^M$, the specific forms of $\{\mathbf{q}_i\}_{i=1}^M$ are more flexible as long as the correlation constraint is satisfied.

In order to better represent the relationships between the observations $\{\mathbf{y}_i\}_{i=1}^M$ and the discriminative codes $\{\mathbf{q}_i\}_{i=1}^M$, $\mathbf{q}_i$ is rewritten as $\Upsilon(\mathbf{y}_i)$, where $\Upsilon : \mathbb{R}^\alpha \to \mathcal{G} \subset \mathbb{R}^\beta$ is a non-linear mapping from the original space to a reproducing kernel Hilbert space. This embedding can preserve all the statistical features of the observations, while allowing one to compare and manipulate the discriminative codes using Hilbert space operations such as inner products, distances, spectral analysis, and so on [38]. Thus we have:

$$\begin{cases} \langle \Upsilon(\mathbf{y}_i), \Upsilon(\mathbf{y}_j) \rangle = 1, & \text{if } h_i = h_j \quad \forall i, j, \\ \langle \Upsilon(\mathbf{y}_i), \Upsilon(\mathbf{y}_j) \rangle = 0, & \text{if } h_i \neq h_j \quad \forall i, j, \end{cases} \tag{8}$$

where all the discriminative codes are normalized by the l_2-norm, such that $\|\Upsilon(\mathbf{y}_i)\|_2^2 = 1, \forall i$. For convenience, the correlation constraint can be represented by a kernel matrix $[\mathcal{S}_{i,j}] = [\langle \Upsilon(\mathbf{y}_i), \Upsilon(\mathbf{y}_j) \rangle]$[1] implicitly. Hence, the computational complexity would not be influenced by the dimensionality of the discriminative code.

Then we have the following discriminative term:

$$\|\Upsilon(\mathbf{Y}) - \mathbf{BX}\|_F^2, \tag{9}$$

where, $\Upsilon(\mathbf{Y}) = [\Upsilon(\mathbf{y}_1), \Upsilon(\mathbf{y}_2), \ldots, \Upsilon(\mathbf{y}_M)]$ is the discriminative coding matrix.

Constrained by the discriminative codes, as the transformation matrix $\mathbf{B}$ is available, $\mathbf{BX}$ can be regarded as training features for the classifier. With such discriminative term, the property of the feature space is evident: The features in the feature space are of high correlation inner-class and nearly independent between-classes.

Furthermore, the discriminative term can be seen as another reconstructive non-linear dictionary learning problem where $\mathbf{B}$ is the non-linear dictionary. Thus, the discriminative term can be re-written as follows similarly to (6):

$$\|\Upsilon(\mathbf{Y}) - \mathbf{BX}\|_F^2 = \|\Upsilon(\mathbf{Y}) - \Upsilon(\mathbf{Y})\mathbf{GX}\|_F^2, \tag{10}$$

where

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_1 & & & \mathbf{0} \\ & \mathbf{G}_2 & & \\ & & \ddots & \\ \mathbf{0} & & & \mathbf{G}_N \end{bmatrix}$$

is another pseudo-dictionary.

[1]In order to make the matrix $\mathcal{S}$ reversible, the diagonal elements of $\mathcal{S}$ are added with a small positive constant. This is reasonable, because the correlation of one feature vector with itself is more likely higher than that with another one.

Thus, as $\mathbf{G}$ is available, the columns of $\Upsilon(\mathbf{Y})\mathbf{GX}$ are the training features and their inner product matrix $\mathbf{X}^T\mathbf{G}^T\mathcal{S}(\mathbf{Y}, \mathbf{Y})\mathbf{GX}$ is used to train the classifier. Obviously, the correlation constraint indeed serve the task of classification more directly.

C. Final Objective Function

By incorporating (6) and (10) into (1), we have the final objective function:

$$\begin{aligned} \langle \mathbf{F}, \mathbf{G}, \mathbf{X} \rangle = \arg\min_{\mathbf{F},\mathbf{G},\mathbf{X}} & \|\Phi(\mathbf{Y}) - \Phi(\mathbf{Y})\mathbf{FX}\|_F^2 \\ & + \lambda \|\Upsilon(\mathbf{Y}) - \Upsilon(\mathbf{Y})\mathbf{GX}\|_F^2 \\ s.t.\ \forall i, j\ & \|\mathbf{x}_i\|_0 \leq T, \quad \|\Phi(\mathbf{Y})\mathbf{f}_j\|_2 = 1, \\ & \|\Upsilon(\mathbf{Y})\mathbf{g}_j\|_2 = 1, \end{aligned} \tag{11}$$

where, $\mathbf{f}_j$ and $\mathbf{g}_j$ are the j-th column of $\mathbf{F}$ and $\mathbf{G}$, respectively.

III. Optimization

In this section, we present our algorithm for optimizing the objective function in (11). First, (11) can be re-written as

$$\begin{aligned} \langle \mathbf{F}, \mathbf{G}, \mathbf{X} \rangle = \arg\min_{\mathbf{F},\mathbf{G},\mathbf{X}} & \left\| \begin{bmatrix} \Phi(\mathbf{Y}) \\ \sqrt{\lambda}\Upsilon(\mathbf{Y}) \end{bmatrix} - \begin{bmatrix} \Phi(\mathbf{Y})\mathbf{F} \\ \sqrt{\lambda}\Upsilon(\mathbf{Y})\mathbf{G} \end{bmatrix} \mathbf{X} \right\|_F^2 \\ s.t.\ \forall i\ & \|\mathbf{x}_i\|_0 \leq T, \end{aligned} \tag{12}$$

where, the matrix $\left[[\Phi(\mathbf{Y})\mathbf{F}]^T, \left[\sqrt{\lambda}\Upsilon(\mathbf{Y})\mathbf{G}\right]^T\right]^T$ is normalized by the l_2-norm columnwise which can be regarded as a new dictionary $\mathbf{D}_{new}$, and the matrix $\left[\Phi(\mathbf{Y})^T, \sqrt{\lambda}\Upsilon(\mathbf{Y})^T\right]^T$ can be regarded as a whole to be the input matrix.

And then, the dictionary learning problem can be divided into two sub-problems: sparse coding and dictionary update.

A. Sparse Coding With KOMP

Suppose that the dictionary is fixed, and the objective function in (12) is reduced to a sparse coding problem, which can be re-written as:

$$\begin{aligned} \langle \mathbf{X} \rangle = \arg\min_{\mathbf{X}} & \left\| \begin{bmatrix} \Phi(\mathbf{Y}) \\ \sqrt{\lambda}\Upsilon(\mathbf{Y}) \end{bmatrix} - \begin{bmatrix} \Phi(\mathbf{Y}) & \mathbf{0} \\ \mathbf{0} & \sqrt{\lambda}\Upsilon(\mathbf{Y}) \end{bmatrix} \begin{bmatrix} \mathbf{F} \\ \mathbf{G} \end{bmatrix} \mathbf{X} \right\|_F^2 \\ s.t.\ \forall i\ & \|\mathbf{x}_i\|_0 \leq T. \end{aligned} \tag{13}$$

Denote

$$\Gamma(\mathbf{Z}_{new}) = \left[\Phi(\mathbf{Y})^T, \sqrt{\lambda}\Upsilon(\mathbf{Y})^T\right]^T, \tag{14}$$

$$\Gamma(\mathbf{Y}_{new}) = diag\left(\Phi(\mathbf{Y}), \sqrt{\lambda}\Upsilon(\mathbf{Y})\right), \tag{15}$$

$$\mathbf{A}_{new} = \left[\mathbf{F}^T, \mathbf{G}^T\right]^T. \tag{16}$$

Eq. (13) is equivalent to the following problem:

$$\begin{aligned} \langle \mathbf{X} \rangle = \arg\min_{\mathbf{X}} & \|\Gamma(\mathbf{Z}_{new}) - \Gamma(\mathbf{Y}_{new})\mathbf{A}_{new}\mathbf{X}\|_F^2 \\ s.t.\ \forall i\ & \|\mathbf{x}_i\|_0 \leq T, \end{aligned} \tag{17}$$

which can be solved by the kernel orthogonal matching pursuit algorithm (KOMP) [26].

It should be noted that KOMP is a method with kernels embedded. As for the above problem in (17), all the

kernel matrices utilized in the sparse coding stage are summarized as follows:

$$\Gamma(\mathbf{Z}_{new})^T\Gamma(\mathbf{Z}_{new}) = \mathcal{K}(\mathbf{Y},\mathbf{Y}) + \lambda\mathcal{S}(\mathbf{Y},\mathbf{Y}), \quad (18)$$

$$\Gamma(\mathbf{Y}_{new})^T\Gamma(\mathbf{Y}_{new}) = diag(\mathcal{K}(\mathbf{Y},\mathbf{Y}), \lambda\mathcal{S}(\mathbf{Y},\mathbf{Y})), \quad (19)$$

$$\Gamma(\mathbf{Z}_{new})^T\Gamma(\mathbf{Y}_{new}) = [\mathcal{K}(\mathbf{Y},\mathbf{Y}), \lambda\mathcal{S}(\mathbf{Y},\mathbf{Y})]. \quad (20)$$

B. Dictionary Update With Structured Kernel K-SVD (SK-KSVD)

When the coefficient matrix $\mathbf{X}$ is fixed, the dictionary $\mathbf{D}_{new}$ is updated by optimizing the following problem:

$$\langle \mathbf{F}, \mathbf{G}\rangle = \arg\min_{\mathbf{F},\mathbf{G}} \left\| \begin{bmatrix} \Phi(\mathbf{Y}) \\ \sqrt{\lambda}\Upsilon(\mathbf{Y}) \end{bmatrix} - \begin{bmatrix} \Phi(\mathbf{Y})\mathbf{F} \\ \sqrt{\lambda}\Upsilon(\mathbf{Y})\mathbf{G} \end{bmatrix} \mathbf{X} \right\|_F^2. \quad (21)$$

where $\left[[\Phi(\mathbf{Y})\mathbf{F}]^T, \left[\sqrt{\lambda}\Upsilon(\mathbf{Y})\mathbf{G}\right]^T\right]^T = \mathbf{D}_{new}$. It is clear that the dictionary $\mathbf{D}_{new}$ is updated by optimizing $\mathbf{F}$ and $\mathbf{G}$ instead.

Note that unlike the traditional kernel KSVD framework, the dictionary $\mathbf{D}_{new} = \left[[\mathbf{D}_1,\ldots,\mathbf{D}_N]^T, \sqrt{\lambda}[\mathbf{B}_1,\ldots,\mathbf{B}_N]^T\right]^T$ with $\mathbf{D}_n = \Phi(\mathbf{Y}_n)\mathbf{F}_n$ and $\mathbf{B}_n = \Upsilon(\mathbf{Y}_n)\mathbf{G}_n$ is a structured dictionary which has some distinct characteristics. The pseudo-dictionaries $\mathbf{F}$ and $\mathbf{G}$ are both block diagonal matrices. That is, the columns of the matrix $\left[\mathbf{F}^T, \mathbf{G}^T\right]^T$ are sparse. Therefore, it is infeasible to treat $\mathbf{F}$ and $\mathbf{G}$ as a whole to be optimized by utilizing kernel KSVD algorithm. In what follows, we propose a structured kernel KSVD (SK-KSVD) algorithm to solve the above problem, whose basic idea is similar to KSVD and kernel KSVD.

The structured dictionary $\mathbf{D}_{new}$ is updated by optimizing $\mathbf{F}$ and $\mathbf{G}$ class by class. And for every specific class $n = 1, 2, \ldots, N$, $\mathbf{F}_n$ and $\mathbf{G}_n$ are updated atom by atom.

When updating the sub-dictionary of class n, namely $\mathbf{F}_n$ and $\mathbf{G}_n$, let the signal matrix $\Phi(\mathbf{Y})$ be decomposed into two sub-matrices, $\Phi(\mathbf{Y}_n)$ and $\Phi(\mathbf{Y}_r)$, containing the signals from class n and the rest, respectively. Then, the corresponding pseudo-dictionary $\mathbf{F}$ can be rearranged into $\mathbf{F}_n$ and $\mathbf{F}_r$. In a similar way, $\Upsilon(\mathbf{Y}_n)$, $\Upsilon(\mathbf{Y}_r)$, $\mathbf{G}_n$ and $\mathbf{G}_r$ are obtained. The objective function in (21) can be re-written as:

$$\left\| \begin{bmatrix} \Phi(\mathbf{Y}) \\ \sqrt{\lambda}\Upsilon(\mathbf{Y}) \end{bmatrix} - \begin{bmatrix} [\Phi(\mathbf{Y}_n)\mathbf{F}_n, \Phi(\mathbf{Y}_r)\mathbf{F}_r] \\ \sqrt{\lambda}[\Upsilon(\mathbf{Y}_n)\mathbf{G}_n, \Upsilon(\mathbf{Y}_r)\mathbf{G}_r] \end{bmatrix} \begin{bmatrix} \mathbf{X}_n \\ \mathbf{X}_r \end{bmatrix} \right\|_F^2 \quad (22)$$

$$= \left\| \begin{bmatrix} \Phi(\mathbf{Y}) - \Phi(\mathbf{Y}_r)\mathbf{F}_r\mathbf{X}_r \\ \sqrt{\lambda}(\Upsilon(\mathbf{Y}) - \Upsilon(\mathbf{Y}_r)\mathbf{G}_r\mathbf{X}_r) \end{bmatrix} - \begin{bmatrix} \Phi(\mathbf{Y}_n)\mathbf{F}_n \\ \sqrt{\lambda}\Upsilon(\mathbf{Y}_n)\mathbf{G}_n \end{bmatrix} \mathbf{X}_n \right\|_F^2 \quad (23)$$

$$= \left\| \begin{bmatrix} \mathbf{E}_1 \\ \mathbf{E}_2 \end{bmatrix} - \begin{bmatrix} \Phi(\mathbf{Y}_n)\mathbf{f}_n^k \\ \sqrt{\lambda}\Upsilon(\mathbf{Y}_n)\mathbf{g}_n^k \end{bmatrix} \mathbf{x}_n^k \right\|_F^2, \quad (24)$$

where,

$$\mathbf{E}_1 = \Phi(\mathbf{Y}) - \Phi(\mathbf{Y}_r)\mathbf{F}_r\mathbf{X}_r - \Phi(\mathbf{Y}_n)\sum_{j\neq k}\mathbf{f}_n^j\mathbf{x}_n^j, \quad (25)$$

$$\mathbf{E}_2 = \sqrt{\lambda}\left(\Upsilon(\mathbf{Y}) - \Upsilon(\mathbf{Y}_r)\mathbf{G}_r\mathbf{X}_r - \Upsilon(\mathbf{Y}_n)\sum_{j\neq k}\mathbf{g}_n^j\mathbf{x}_n^j\right), \quad (26)$$

$\mathbf{X}_n$ and $\mathbf{X}_r$ represent the coefficient matrix of training signals from all classes over the atoms from class n and the rest, respectively, $\mathbf{f}_n^k$ and $\mathbf{g}_n^k$ are the k-th column of $\mathbf{F}_n$ and $\mathbf{G}_n$, respectively, $\mathbf{x}_n^k$ is the k-th row of $\mathbf{X}_n$. Eq. (23) indicates that when updating the n-th sub-dictionary, all the others are fixed. Eq. (24) indicates that when updating the k-th dictionary atom from class n, all the other atoms are fixed. Now the aim is to find the optimal $\mathbf{f}_n^k$, $\mathbf{g}_n^k$ and $\mathbf{x}_n^k$ that minimize (24).

The coefficient vector $\mathbf{x}_n^k$ plays an important role when minimizing the above error function in (24). The non-zero elements of $\mathbf{x}_n^k$ establish bridges between the associated columns in $\left[\mathbf{E}_1^T, \mathbf{E}_2^T\right]^T$ and the k-th dictionary atom from class n: Columns in $\left[\mathbf{E}_1^T, \mathbf{E}_2^T\right]^T$ corresponding to the non-zero elements of $\mathbf{x}_n^k$ are expected to update the k-th dictionary atom from class n; columns in $\left[\mathbf{E}_1^T, \mathbf{E}_2^T\right]^T$ corresponding to the zero-value elements of $\mathbf{x}_n^k$ do not affect the value of (24), therefore these columns can be neglected and discarded when updating $\mathbf{f}_n^k$, $\mathbf{g}_n^k$ and $\mathbf{x}_n^k$.

Define $w_n^k = \left\{i | 1 \le i \le M, \mathbf{x}_n^k(i) \neq 0\right\}$ as the group of indices of the non-zero elements in $\mathbf{x}_n^k$. Define Ω_n^k as a matrix of size $M \times \left|w_n^k\right|$, with ones on the $\left(w_n^k(i), i\right)$-th entries and zeros elsewhere. The multiplications $\mathbf{x}_n^k\Omega_n^k$ and $\left[\mathbf{E}_1^T, \mathbf{E}_2^T\right]^T\Omega_n^k$ discard all zero elements in the vector $\mathbf{x}_n^k$ as well as the corresponding useless columns in $\left[\mathbf{E}_1^T, \mathbf{E}_2^T\right]^T$.

Now, the optimization problem can be re-written as:

$$\left\langle \mathbf{f}_n^k, \mathbf{g}_n^k, \mathbf{x}_n^{kR}\right\rangle = \arg\min_{\mathbf{f}_n^k,\mathbf{g}_n^k,\mathbf{x}_n^{kR}} \left\| \mathbf{E}^R - \begin{bmatrix} \Phi(\mathbf{Y}_n)\mathbf{f}_n^k \\ \sqrt{\lambda}\Upsilon(\mathbf{Y}_n)\mathbf{g}_n^k \end{bmatrix} \mathbf{x}_n^{kR} \right\|_F^2, \quad (27)$$

where, $\mathbf{E}^R = \left[\left(\mathbf{E}_1^R\right)^T, \left(\mathbf{E}_2^R\right)^T\right]^T$, $\mathbf{E}_1^R = \mathbf{E}_1\Omega_n^k$, $\mathbf{E}_2^R = \mathbf{E}_2\Omega_n^k$, $\mathbf{x}_n^{kR} = \mathbf{x}_n^k\Omega_n^k$.

The above optimization problem is equivalent to finding the rank-1 matrix that best approximates the matrix $\mathbf{E}^R$ in terms of the Frobenius norm. It is noticed that the columns of $\Phi(\mathbf{Y}_n)\mathbf{f}_n^k\mathbf{x}_n^{kR}$ lie in the column subspace of $\Phi(\mathbf{Y}_n)$, which means that $\Phi(\mathbf{Y}_n)\mathbf{f}_n^k\mathbf{x}_n^{kR}$ could only approximate $\mathbf{E}_1^R$ in terms of the component lying in the column subspace of $\Phi(\mathbf{Y}_n)$. Similarly, $\sqrt{\lambda}\Upsilon(\mathbf{Y}_n)\mathbf{g}_n^k\mathbf{x}_n^{kR}$ could only approximate $\mathbf{E}_2^R$ in terms of the component lying in the column subspace of $\Upsilon(\mathbf{Y}_n)$. Hence, $\mathbf{E}_1^R$ and $\mathbf{E}_2^R$ should be projected into the column subspaces $\Phi(\mathbf{Y}_n)$ and $\Upsilon(\mathbf{Y}_n)$ respectively, before finding the closest rank-1 matrix approximation. The projections can be realized by left-multiplying $\mathbf{E}_1^R$ and $\mathbf{E}_2^R$ with the projection operators $\mathbf{P}_f$ and $\mathbf{P}_g$, respectively.

$$\mathbf{P}_f = \Phi(\mathbf{Y}_n)\left(\Phi(\mathbf{Y}_n)^T\Phi(\mathbf{Y}_n)\right)^{-1}\Phi(\mathbf{Y}_n)^T, \quad (28)$$

$$\mathbf{P}_g = \Upsilon(\mathbf{Y}_n)\left(\Upsilon(\mathbf{Y}_n)^T\Upsilon(\mathbf{Y}_n)\right)^{-1}\Upsilon(\mathbf{Y}_n)^T. \quad (29)$$

$\mathbf{P}_f$ and $\mathbf{P}_g$ are both idempotent matrices. And then, the optimization problem can be written as:

$$\left\langle \mathbf{f}_n^k, \mathbf{g}_n^k, \mathbf{x}_n^{kR}\right\rangle = \arg\min_{\mathbf{f}_n^k,\mathbf{g}_n^k,\mathbf{x}_n^{kR}} \left\| \mathbf{E} - \begin{bmatrix} \Phi(\mathbf{Y}_n)\mathbf{f}_n^k \\ \sqrt{\lambda}\Upsilon(\mathbf{Y}_n)\mathbf{g}_n^k \end{bmatrix} \mathbf{x}_n^{kR} \right\|_F^2, \quad (30)$$

where,

$$\mathbf{E} = \begin{bmatrix} \mathbf{P}_f\mathbf{E}_1^R \\ \mathbf{P}_g\mathbf{E}_2^R \end{bmatrix} \quad (31)$$

is the total error matrix.

According to the principle of singular value decomposition (SVD), the total error matrix $\mathbf{E}$ can be decomposed as

$$\mathbf{E} = \mathbf{U}\Sigma\mathbf{V}^T. \tag{32}$$

And then the rank-1 matrix $\sigma_1\mathbf{u}_1\mathbf{v}_1^T$ is the best approximation to $\mathbf{E}$, where $\sigma_1 = \Sigma(1,1)$ is the largest singular value, $\mathbf{u}_1$ and $\mathbf{v}_1$ are the corresponding singular vectors belonging to $\mathbf{U}$ and $\mathbf{V}$, respectively. As stated in [11] and [26], the rank-1 matrix $\sigma_1\mathbf{u}_1\mathbf{v}_1^T$ can be broken down into $\mathbf{u}_1$ and $\sigma_1\mathbf{v}_1^T$ to represent the atom and the non-zero coefficient vector, which insures the resulting dictionary atom being normalized to unit-norm. Thus we have

$$\begin{pmatrix} \Phi(\mathbf{Y}_n)\mathbf{f}_n^k \\ \sqrt{\lambda}\Upsilon(\mathbf{Y}_n)\mathbf{g}_n^k \end{pmatrix} = \mathbf{u}_1, \tag{33}$$

$$\mathbf{x}_n^{kR} = \sigma_1\mathbf{v}_1^T. \tag{34}$$

Similar with kernel KSVD [26], eigen decomposition is utilized instead of performing SVD decomposition directly:

$$\mathbf{E}^T\mathbf{E} = \mathbf{V}\Delta\mathbf{V}^T, \tag{35}$$

where, $\Delta = \Sigma^T\Sigma$. This gives us $\mathbf{v}_1$ as the first column of $\mathbf{V}$, and $\sigma_1 = \sqrt{\Delta(1,1)}$. By substituting them into (34), $\mathbf{x}_n^{kR}$ can be found.

As for solving $\mathbf{f}_n^k$ and $\mathbf{g}_n^k$, the following steps are performed in sequence.

Right-multiplying both sides of (32) by $\mathbf{V}$ and pick up the first columns of both sides, which gives us:

$$\mathbf{E}\mathbf{v}_1 = \sigma_1\mathbf{u}_1. \tag{36}$$

By substituting (31) and (33) into (36), the following equations are obtained to seek for $\mathbf{f}_n^k$ and $\mathbf{g}_n^k$.

$$\begin{pmatrix} \mathbf{P}_f\mathbf{E}_1^R \\ \mathbf{P}_g\mathbf{E}_2^R \end{pmatrix}\mathbf{v}_1 = \sigma_1\begin{pmatrix} \Phi(\mathbf{Y}_n)\mathbf{f}_n^k \\ \sqrt{\lambda}\Upsilon(\mathbf{Y}_n)\mathbf{g}_n^k \end{pmatrix} \tag{37}$$

$$\Leftrightarrow$$

$$\begin{cases} \Phi(\mathbf{Y}_n)\big(\Phi(\mathbf{Y}_n)^T\Phi(\mathbf{Y}_n)\big)^{-1}\Phi(\mathbf{Y}_n)^T\mathbf{E}_1^R\mathbf{v}_1 = \sigma_1\Phi(\mathbf{Y}_n)\mathbf{f}_n^k \\ \Upsilon(\mathbf{Y}_n)\big(\Upsilon(\mathbf{Y}_n)^T\Upsilon(\mathbf{Y}_n)\big)^{-1}\Upsilon(\mathbf{Y}_n)^T\mathbf{E}_2^R\mathbf{v}_1 = \sigma_1\sqrt{\lambda}\Upsilon(\mathbf{Y}_n)\mathbf{g}_n^k \end{cases} \tag{38}$$

where the two sub-equations in (38) are similar in form but mutually independent. Thus, we can solve them separately. Then we have:

$$\begin{aligned} \mathbf{f}_n^k &= {\sigma_1}^{-1}\Big(\Phi(\mathbf{Y}_n)^T\Phi(\mathbf{Y}_n)\Big)^{-1}\Phi(\mathbf{Y}_n)^T\mathbf{E}_1^R\mathbf{v}_1 \\ &= {\sigma_1}^{-1}\mathcal{K}(\mathbf{Y}_n,\mathbf{Y}_n)^{-1}\times\ \big(\mathcal{K}(\mathbf{Y}_n,\mathbf{Y}) - \mathcal{K}(\mathbf{Y}_n,\mathbf{Y}_r)\mathbf{F}_r\mathbf{X}_r \\ &\quad - \mathcal{K}(\mathbf{Y}_n,\mathbf{Y}_n)\sum_{j\neq k}\mathbf{f}_n^j\mathbf{x}_n^j)\Omega_n^k\mathbf{v}_1 \end{aligned} \tag{39}$$

$$\begin{aligned} \mathbf{g}_n^k &= \Big(\sigma_1\sqrt{\lambda}\Big)^{-1}\Big(\Upsilon(\mathbf{Y}_n)^T\Upsilon(\mathbf{Y}_n)\Big)^{-1}\Upsilon(\mathbf{Y}_n)^T\mathbf{E}_2^R\mathbf{v}_1 \\ &= {\sigma_1}^{-1}\mathcal{S}(\mathbf{Y}_n,\mathbf{Y}_n)^{-1}\times\ \big(\mathcal{S}(\mathbf{Y}_n,\mathbf{Y}) - \mathcal{S}(\mathbf{Y}_n,\mathbf{Y}_r)\mathbf{G}_r\mathbf{X} \\ &\quad - \mathcal{S}(\mathbf{Y}_n,\mathbf{Y}_n)\sum_{j\neq k}\mathbf{g}_n^j\mathbf{x}_n^j)\Omega_n^k\mathbf{v}_1 \end{aligned} \tag{40}$$

Based on the above procedures, the pseudo-code for the overall SK-KSVD algorithm is given in Algorithm 1.

Algorithm 1 The SK-KSVD Algorithm

Input: A set of signals $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_N]$ from N classes, a kernel function κ, and a discriminative kernel matrix $\mathcal{S}(\mathbf{Y},\mathbf{Y})$ defined in (8).

Task: Find a dictionary $\mathbf{D}_{new}$ via pseudo-dictionaries $\mathbf{F} = diag(\mathbf{F}_1,\ldots,\mathbf{F}_N)$ and $\mathbf{G} = diag(\mathbf{G}_1,\ldots,\mathbf{G}_N)$ such that the signals can be sparsely represented by solving the optimization problem in (21).

Initialization: If $\mathbf{F}_n^{(0)}$ and $\mathbf{G}_n^{(0)}$ are not preset, set a random element of each column in $\mathbf{F}_n^{(0)}$ and $\mathbf{G}_n^{(0)}$ to be 1, $n = 1,\ldots,N$, and normalize each column in the initial dictionary $\mathbf{D}_{new}^{(0)}$ in terms of the l_2-norm. Set iteration $J = 1$.

Repeat until convergence or maximum number of iterations is reached:

- ***Sparse Coding Stage:***
  Fix the pseudo-dictionaries $\mathbf{F}^{(J-1)}$ and $\mathbf{G}^{(J-1)}$, and rewrite the optimization problem as (13). Use the KOMP algorithm described in [26] to obtain the sparse coefficient matrix $\mathbf{X}^{(J)}$.
- ***Dictionary Learning:***
  For $\mathbf{F}_n^{(J-1)}$ and $\mathbf{G}_n^{(J-1)}$ corresponding to a certain class n, where $n = 1,\ldots,N$.
  For the k-th column $\mathbf{f}_n^{k(J-1)}$ in $\mathbf{F}_n^{(J-1)}$ and $\mathbf{g}_n^{k(J-1)}$ in $\mathbf{G}_n^{(J-1)}$, where $k = 1,\ldots,N$, update them by
  - Define the group of indices of the non-zero elements in $\mathbf{x}_n^k$ as $w_n^k = \{i | 1 \le i \le M, \mathbf{x}_n^k(i) \neq 0\}$.
  - Compute the error matrix $\left[\mathbf{E}_1^T, \mathbf{E}_2^T\right]^T$. And then choose only the columns in $\left[\mathbf{E}_1^T, \mathbf{E}_2^T\right]^T$ with indices in w_n^k, i.e., obtaining $\mathbf{E}^R$ as $\mathbf{E}^R = \left[\left(\mathbf{E}_1\Omega_n^k\right)^T, \left(\mathbf{E}_2\Omega_n^k\right)^T\right]^T$.
  - Project $\mathbf{E}^R$ into a certain sub-space, and obtain $\mathbf{E} = \left[\left(\mathbf{P}_f\mathbf{E}_1\Omega_n^k\right)^T, \left(\mathbf{P}_g\mathbf{E}_2\Omega_n^k\right)^T\right]^T$.
  - Apply eigen decomposition to get $\mathbf{E}^T\mathbf{E} = \mathbf{V}\Delta\mathbf{V}^T$.
  - Use the first column of $\mathbf{V}$ and the largest eigen value in Δ to obtain the updated $\mathbf{f}_n^{k(J)}$ and $\mathbf{g}_n^{k(J)}$, by (39) and (40).
- ***Set:*** $J = J + 1$.

Output: $\mathbf{F}$ and $\mathbf{G}$.

IV. Classification Scheme

When the dictionary $\mathbf{D}_{new}$ is available, namely the pseudo-dictionaries $\mathbf{F}$ and $\mathbf{G}$ are available, the columns of $\Upsilon(\mathbf{Y})\mathbf{G}\mathbf{X}$ are the training features. The correlations between different training features is expressed by the inner product matrix $\mathbf{X}^T\mathbf{G}^T\mathcal{S}(\mathbf{Y},\mathbf{Y})\mathbf{G}\mathbf{X}$, which is easy to compute. Besides, such expression matches the classification mechanism of SVM classifier [39]. In order to highlight the discriminability of the extracted features, here we only choose the simplest one, i.e., a linear SVM, for classification. More details about the classification scheme are discussed in what follows.

In the training stage, the pseudo-dictionaries $\mathbf{F}$ and $\mathbf{G}$ are firstly optimized via (12) using SK-KSVD. And then, the coefficient matrix $\mathbf{X}$ is obtained by solving the sparse coding problem in (13) with KOMP. Subsequently, the inner product matrix between the training features, i.e., $\mathbf{X}^T\mathbf{G}^T\mathcal{S}(\mathbf{Y},\mathbf{Y})\mathbf{G}\mathbf{X}$, is used to train the linear SVM classifier.

In the testing stage, given a test signal $\mathbf{z}$, with its label unknown, only the reconstructive term is used to seek its optimal sparse coefficient vector, which is a commonly used strategy in supervised discriminative dictionary learning methods, such as [2], [18], and [19].

We were supposed to solve the sparse coefficient vector $\mathbf{a}$ via:

$$\mathbf{a} = \arg\min_{\mathbf{a}} \|\Phi(\mathbf{z}) - \mathbf{D}\mathbf{a}\|_F^2 \quad s.t.\ \|\mathbf{a}\|_0 \le T. \tag{41}$$

and use $\mathbf{B}\mathbf{a}$ as the testing feature for classification. However, recalling that the dictionary $\mathbf{D}$ and the transformation matrix $\mathbf{B}$ are treated as a whole matrix $\mathbf{D}_{new} = \left[\mathbf{D}^T, \sqrt{\lambda}\mathbf{B}^T\right]^T$ and are normalized jointly during training, i.e.,

$$\left\| \begin{bmatrix} \Phi(\mathbf{Y}_n)\mathbf{f}_n^k \\ \sqrt{\lambda}\Upsilon(\mathbf{Y}_n)\mathbf{g}_n^k \end{bmatrix} \right\|_2^2 = 1, \quad \forall n,k, \tag{42}$$

the problem in (41) can not be solved using KOMP algorithm [26], because the condition that atoms in the dictionary should have unit-norm is not satisfied.

Following the similar ideas in [2], [18], and [19], a desired dictionary $\hat{\mathbf{D}}$ and the corresponding transformation matrix $\hat{\mathbf{B}}$ are derived according to the following formulas:

$$\hat{\mathbf{D}} = \left\{ \frac{\Phi(\mathbf{Y}_n)\mathbf{f}_n^k}{\|\Phi(\mathbf{Y}_n)\mathbf{f}_n^k\|_2} \right\}_{n=1,\dots,N}^{k=1,\dots,K} \tag{43}$$

$$= \left\{ \frac{\Phi(\mathbf{Y}_n)\mathbf{f}_n^k}{\sqrt{\mathbf{f}_n^{kT}\mathcal{K}(\mathbf{Y}_n,\mathbf{Y}_n)\mathbf{f}_n^k}} \right\}_{n=1,\dots,N}^{k=1,\dots,K} \tag{44}$$

$$= \left\{ \Phi(\mathbf{Y}_n)\hat{\mathbf{f}}_n^k \right\}_{n=1,\dots,N}^{k=1,\dots,K} \tag{45}$$

$$= \Phi(\mathbf{Y})\hat{\mathbf{F}} \tag{46}$$

$$\hat{\mathbf{B}} = \left\{ \frac{\Upsilon(\mathbf{Y}_n)\mathbf{g}_n^k}{\|\Phi(\mathbf{Y}_n)\mathbf{f}_n^k\|_2} \right\}_{n=1,\dots,N}^{k=1,\dots,K} \tag{47}$$

$$= \left\{ \frac{\Upsilon(\mathbf{Y}_n)\mathbf{g}_n^k}{\sqrt{\mathbf{f}_n^{kT}\mathcal{K}(\mathbf{Y}_n,\mathbf{Y}_n)\mathbf{f}_n^k}} \right\}_{n=1,\dots,N}^{k=1,\dots,K} \tag{48}$$

$$= \left\{ \Upsilon(\mathbf{Y}_n)\hat{\mathbf{g}}_n^k \right\}_{n=1,\dots,N}^{k=1,\dots,K} \tag{49}$$

$$= \Upsilon(\mathbf{Y})\hat{\mathbf{G}} \tag{50}$$

which are actually realized by modifying the pseudo-dictionaries $\mathbf{F}$ and $\mathbf{G}$ into $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$.

And then, the sparse coefficient vector $\hat{\mathbf{a}}$ which corresponds to the desired dictionary $\hat{\mathbf{D}}$ is obtained by solving the following problem using KOMP.

$$\hat{\mathbf{a}} = \arg\min_{\hat{\mathbf{a}}} \left\| \Phi(\mathbf{z}) - \hat{\mathbf{D}}\hat{\mathbf{a}} \right\|_F^2 \quad s.t.\ \|\hat{\mathbf{a}}\|_0 \le T. \tag{51}$$

Algorithm 2 The CCSK-KSVD Algorithm

Input: A set of signals $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_N]$ from N classes, a kernel function κ, and a discriminative kernel matrix $\mathcal{S}(\mathbf{Y},\mathbf{Y})$ defined in (8).

Initialization: Set a random element of each column in $\mathbf{F}_n^{(0)}$ and $\mathbf{G}_n^{(0)}$ to be 1, where $n = 1,\dots,N$. Solve the following optimization problem to obtain better $\mathbf{F}_n^{(0)}$ and $\mathbf{G}_n^{(0)}$ class by class.

$$\min_{\mathbf{F}_n^{(0)},\mathbf{G}_n^{(0)},\mathbf{X}} \left\| \begin{bmatrix} \Phi(\mathbf{Y}_n) \\ \sqrt{\lambda}\Upsilon(\mathbf{Y}_n) \end{bmatrix} - \begin{bmatrix} \Phi(\mathbf{Y}_n)\mathbf{F}_n^{(0)} \\ \sqrt{\lambda}\Upsilon(\mathbf{Y}_n)\mathbf{G}_n^{(0)} \end{bmatrix} \mathbf{X} \right\|_F^2 \quad s.t. \quad \forall i \quad \|\mathbf{x}_i\|_0 \le T_0 \tag{54}$$

Such problem can be solved in a similar way to SK-KSVD algorithm.

Training stage: Compute $\mathbf{F}$ and $\mathbf{G}$ in (12) with SK-KSVD. Compute $\mathbf{X}$ with KOMP. Use $\mathbf{X}^T\mathbf{G}^T\mathcal{S}(\mathbf{Y},\mathbf{Y})\mathbf{G}\mathbf{X}$, the inner product matrix of the training features to train a linear SVM classifier.

Testing stage: Compute $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$ as stated in (43)-(50). Given a test signal $\mathbf{z}$, use the KOMP algorithm to solve the optimal $\hat{\mathbf{a}}$ in (51). Then, use the inner product matrix between the testing feature and training features, $\hat{\mathbf{a}}^T\hat{\mathbf{G}}^T\mathcal{S}(\mathbf{Y},\mathbf{Y})\mathbf{G}\mathbf{X}$, as the input for the SVM classifier.

It is easy to prove that the desired $\{\hat{\mathbf{D}},\hat{\mathbf{B}}\}$ and the learned $\{\mathbf{D},\mathbf{B}\}$ have the following relations:

$$\Phi(\mathbf{z}) \approx \mathbf{D}\mathbf{a} = \hat{\mathbf{D}}\hat{\mathbf{a}} \tag{52}$$

$$\Upsilon(\mathbf{z}) \approx \mathbf{B}\mathbf{a} = \hat{\mathbf{B}}\hat{\mathbf{a}} \tag{53}$$

Thus, we can use $\hat{\mathbf{B}}\hat{\mathbf{a}}$ to express the testing feature instead of using $\mathbf{B}\mathbf{a}$.

Now, the inner product matrix between the testing feature and the training features is used as the input for the SVM classifier, namely:

$$\langle\hat{\mathbf{B}}\hat{\mathbf{a}}, \mathbf{B}\mathbf{X}\rangle = \langle\Upsilon(\mathbf{Y})\hat{\mathbf{G}}\hat{\mathbf{a}}, \Upsilon(\mathbf{Y})\mathbf{G}\mathbf{X}\rangle \tag{55}$$

$$= \hat{\mathbf{a}}^T\hat{\mathbf{G}}^T\mathcal{S}(\mathbf{Y},\mathbf{Y})\mathbf{G}\mathbf{X} \tag{56}$$

The proposed CCSK-KSVD algorithm is summarized in Algorithm 2.

V. Experiments

In this section, we present experimental results on some publicly available databases to illustrate the effectiveness of the proposed CCSK-KSVD algorithm for classification task. In particular, we present face recognition results on the Extended YaleB database [40] and the AR face database [41] using random-face features [16], object classification results on the Caltech101 dataset [42] using spatial pyramid features [18], and synthetic aperture radar (SAR) target recognition results on the MSTAR database [43] using random-face features.

Apart from the MSTAR database, all the features of the other datasets are coincidence with [18] which can be downloaded from:

http://www.umiacs.umd.edu/~zhuolin/projectlcksvd.html

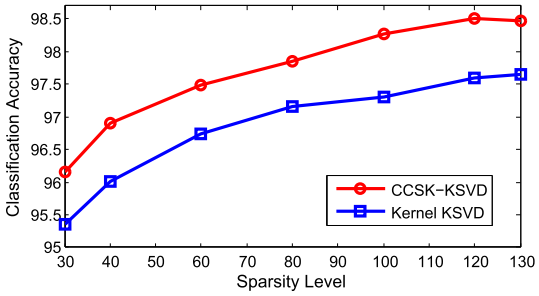


Fig. 1. Performance on the Extended YaleB database with varying sparsity level.

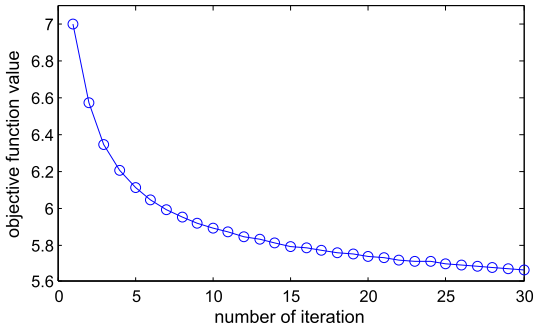


Fig. 2. Example of the convergence of SK-KSVD.

The feature descriptors used in the MSTAR database is captured via the same way as [16] and [18]. Namely, each image is projected onto a α-dimensional feature vector with a randomly generated matrix from the standard normal distribution. Each row of the matrix is normalized in terms of the l_2-norm. The MSTAR database has fixed training set and testing set. For the other databases, we repeat the experiments 10 times on each of these databases with different random splits of the training and testing sets to obtain reliable results. The final recognition rates are reported as the average of each run.

We compare our approach with sparse representation-based classification (SRC) [16], K-SVD[2] [11], D-KSVD [2], LC-KSVD[3] [18], and Kernel KSVD[4] [26]. Among these methods, SRC is the only one that uses the original training signals as the dictionary. For fair comparison, we randomly select part of the training samples from each class to construct the dictionary, which is denoted as SRC*. In all the experiments, we employ only 2 iterations for the dictionary initialization shown in (54). The main dictionary updates with 50 iterations for the Caltech101 dataset and 30 iterations for the others, since features of Caltech101 dataset have higher dimensionality. Other implementation details will be shown in the following subsections.

A. Extended YaleB Face Recognition

The Extended YaleB database consists of 2414 frontal-face images from 38 people captured under various illumination conditions and expressions [40]. The original images were cropped to 192×168 pixels. The dimension of the extracted random-face feature is 504. There are about 64 images for each person. We randomly select half of the images per category as training and the other half for testing.

Except for SRC [16], the size of each sub-dictionary is 15, which means that the total dictionary consists of 570 atoms. A Gaussian kernel is used for our approach. The parameters λ and δ are determined by fivefold cross validation on the training dataset. The best performance is achieved at $\lambda = 0.49$ and $1/\delta = 0.6$.

Fig. 1 compares the performances of Kernel KSVD [26] and CCSK-KSVD when varying the sparsity level T in the range from 30 to 130. It can be easily observed that both of these two approaches have relatively better performance with the increase of the sparsity level from 30 to 120. There is no significant improvement with the sparsity level larger than 120. Based on this analysis, the sparsity levels of the two non-linear methods are set as 120.

The effectiveness of the proposed approach is validated from the following two aspects:

Firstly, we check the objective function value in each iteration to analyze the convergence of the proposed SK-KSVD algorithm. The objective function value is computed as:

$$\begin{aligned}&\|\Phi(\mathbf{Y}) - \mathbf{DX}\|_F^2 + \lambda \|\Upsilon(\mathbf{Y}) - \mathbf{BX}\|_F^2 \\ &= tr\left(\mathcal{K}(\mathbf{Y},\mathbf{Y}) + \mathbf{X}^T\mathbf{F}^T\mathcal{K}(\mathbf{Y},\mathbf{Y})\mathbf{FX} - 2\mathcal{K}(\mathbf{Y},\mathbf{Y})\mathbf{FX}\right) \\ &\quad + \lambda \mathrm{tr}\left(\mathcal{S}(\mathbf{Y},\mathbf{Y}) + \mathbf{X}^T\mathbf{G}^T\mathcal{S}(\mathbf{Y},\mathbf{Y})\mathbf{GX} - 2\mathcal{S}(\mathbf{Y},\mathbf{Y})\mathbf{GX}\right).\end{aligned} \tag{57}$$

Fig. 2 shows the variation of the objective function value versus the numbers of iteration. It is clear that the objective function value decreases during the optimization procedure, which validate the convergence of the proposed algorithm.

Secondly, the effectiveness of the proposed discriminative term and the discriminabilities of the extracted features are examined. Fig. 3 displays the inner product matrices of the extracted features in our approach, which implies the correlations between different features. Owing to the dictionary learning with correlation constraint, the obtained training features are highly correlated inner-class and nearly independent between different classes (see Fig. 3a). Such discriminative dictionary is used for classification, and the test features are discriminative as what we wish (see Fig. 3b). For better demonstration, we make a contrast experiment via keeping the reconstructive term of the proposed CCSK-KSVD but using the discriminative term introduced in LC-KSVD in [18]. The objective function is as follows:

$$\begin{aligned}&\min_{\mathbf{F},\mathbf{A},\mathbf{X}} \|\Phi(\mathbf{Y}) - \Phi(\mathbf{Y})\mathbf{FX}\|_F^2 + \lambda\|\tilde{\mathbf{Q}} - \mathbf{AX}\|_F^2 \\ &s.t.\ \forall i\ \ \|\mathbf{x}_i\|_0 \le T\end{aligned} \tag{58}$$

where, $\tilde{\mathbf{Q}} = \{\tilde{\mathbf{q}}_i\}_{i=1}^M$ are the discriminative codes whose specific forms are explicitly defined according to the labels of the signals and the atoms, columns in $\mathbf{X}$ are the final features for classification, which follow the principle of [18]. We optimize (58) using the basic idea of SK-KSVD.

[2]An algorithm that uses the dictionary learned by the K-SVD algorithm in [11] and employs a linear classifier introduced in [18].

[3]An algorithm called "LC-KSVD2" in [18].

[4]An algorithm called "C-Kernel KSVD" in [26].

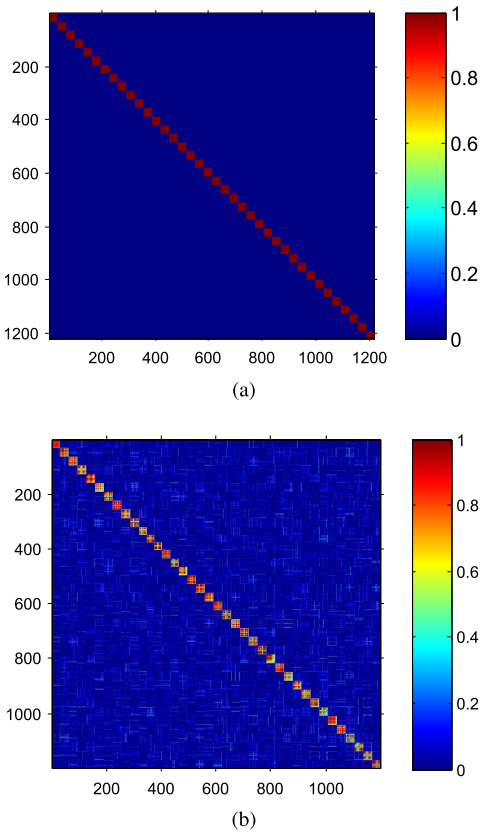


Fig. 3. Inner product matrix of (a) training features and (b) testing features of the Extended YaleB database with CCSK-KSVD. The indexes of the training features and the testing features are permutated class by class, separately.

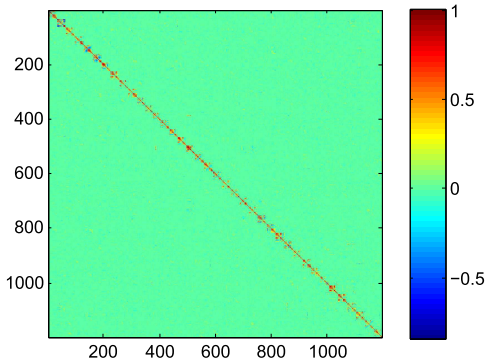


Fig. 4. Inner product matrix of testing features of the Extended YaleB database with the approach by combining the reconstructive term of the CCSK-KSVD and the discriminative term of the LC-KSVD. The indexes of the testing features are permutated class by class. The classification accuracy of such kind of feature is 97.83% with a linear SVM classifier and 97.75% with a linear classifier as in [18].

The classification accuracy is 97.83% with a linear SVM classifier and 97.75% with a linear classifier as in [18], which are relatively higher than that of the Kernel KSVD but still lower than that of the proposed CCSK-KSVD. We also plot the inner product matrix of the testing features with such approach in Fig. 4. Note that, the inner product values between different features from the same class range from negative to positive values. The features from the same class are not as correlative as ours. The comparison results may caused by

TABLE I

RECOGNITION RESULTS USING RANDOM-FACE FEATURES ON THE EXTENDED YALEB DATABASE

Method	Accuracy (%)
SRC(all train. samp.) [16]	97.2
SRC*(15 per class) [16]	80.5
K-SVD(15 per class) [11]	93.1
D-KSVD(15 per class) [2]	94.1
LC-KSVD(15 per class) [18]	95.0
Kernel KSVD(15 per class) [26]	97.56
CCSK-KSVD(15 per class)	98.50

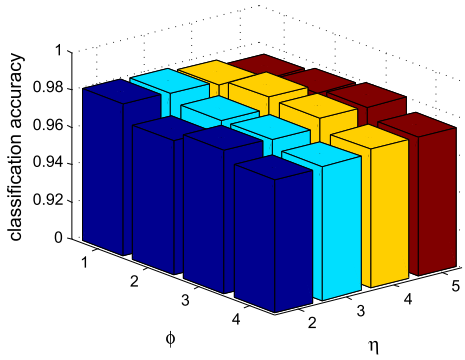


Fig. 5. Effects of different polynomial kernels on the classification accuracy on the Extended YaleB database.

two reasons: 1) the structured and highly dimensional transformation matrix $\Upsilon(\mathbf{Y})\mathbf{G}$ is more powerful than the simple matrix $\mathbf{A}$ in expressing the relationships between the sparse-code-space and the discriminative-code-space; 2) compared with the coding scheme of $\tilde{\mathbf{Q}}$ emphasizing the specific form of every single code, the correlation constraint emphasizes the correlations between different codes, which matches the classification mechanism better.

The recognition results of our proposed CCSK-KSVD and other state-of-the-art approaches are summarized in Table I. Our approach achieves better result than the others. Especially, our approach outperforms SRC [16] when SRC uses all the training samples as the dictionary.

The performances using different polynomial kernels with other parameters fixed are compared. The recognition accuracies are shown in Fig. 5. Clearly, the best performance is achieved at $\phi = 2$, $\eta = 4$, but still a little bit worse than the best performance when using a Gaussian kernel with $1/\delta = 0.6$.

In addition, the robustness of the proposed method is evaluated when the test samples are corrupted by random Gaussian noise with different standard deviations as shown in Fig. 6. As the distortion level increases, the performance differences between the kernel methods and the linear method become more drastic. Our proposed CCSK-KSVD performs the best with moderate noise.

B. AR Face Recognition

The AR face database [41] consists of over 4000 frontal images from 126 people. A subset of the database consisting 2600 images from 50 male subjects and 50 female subjects

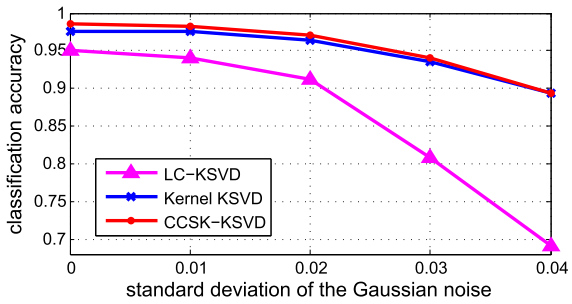


Fig. 6. Comparison of the recognition results on the Extended YaleB database with the presence of Gaussian noise.

TABLE II
RECOGNITION RESULTS USING RANDOM-FACE FEATURES ON THE AR FACE DATABASE

Method	Accuracy (%)
SRC(all train. samp.) [16]	97.5
SRC*(5 per class) [16]	66.5
K-SVD(5 per class) [11]	86.5
D-KSVD(5 per class) [2]	88.8
LC-KSVD(5 per class) [18]	89.60
Kernel KSVD(5 per class) [26]	96.21
CCSK-KSVD(5 per class)	97.77

is used. There are 26 images for each person. For each person, we randomly select 20 images for training and the other 6 for testing. The dimension of the extracted random-face feature is 540. The learned dictionary has 500 atoms, which corresponds to 5 atoms per person.

We choose the following parameters for learning the dictionary in our approach: a Gaussian kernel with $1/\delta = 2$ is used, the scalar parameter is set as $\lambda = 0.09$, and the sparsity level is $T = 90$.

We evaluate our approach using random face features and compare it with the state-of-the-art approaches. The recognition results are summarized in Table II. Our approach outperforms all the linear methods and the non-linear Kernel KSVD [26]. As there is no classification result publicly available of Kernel KSVD on the AR face database in [26], the comparison is based on our implementation, the training and testing data of which are the same as those of our approach. According to the average classification accuracy, CCSK-KSVD improves at least 1.7% over Kernel KSVD and is competitive with other linear methods.

As it is mentioned before, the final classification accuracy is an average value based on different random splits of the training and testing signals. Fig. 7 shows the comparison of the classification results of Kernel KSVD and CCSK-KSVD under different splits of training and testing samples. Note that there exist "worse" splits of samples corresponding to lower classification accuracies and "better" ones corresponding to higher classification accuracies. CCSK-KSVD has better performance than Kernel KSVD in all the 10 times of experiments. In particular, CCSK-KSVD significantly outperforms Kernel KSVD when using the "worst" split of samples. This illustrates that, compared with reconstructive dictionary learning method, discriminative dictionary learning method is more

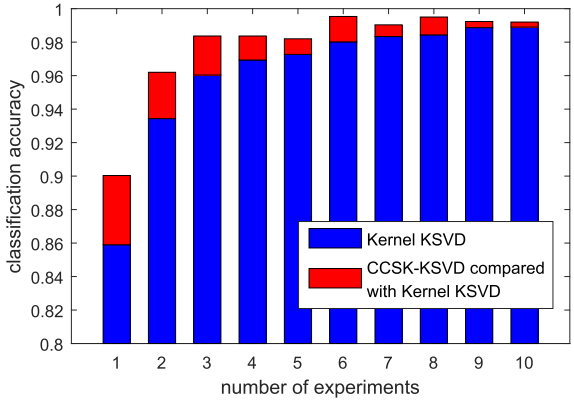


Fig. 7. The classification results of Kernel KSVD [26] and CCSK-KSVD with different random splits of training and testing samples of the AR face database.

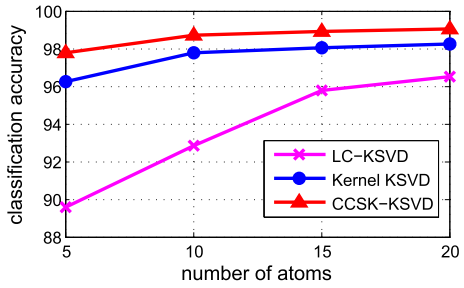


Fig. 8. The recognition performance on the AR face database with varying numbers of atoms per category.

robust when serving the task of classification. This should be likely attributed to the discriminative term.

Besides, we also evaluate the effect of different numbers of atoms in the dictionary. We set the number of atoms per category as 5, 10, 15, and 20, respectively. As shown in Fig. 8, all these three approaches have relatively better performance with the increase of atom numbers. CCSK-KSVD performs better than the others with various atom numbers. Compared with the linear LC-KSVD, the two non-linear method, i.e., kernel KSVD and CCSK-KSVD, are more robust to atom numbers. Especially, CCSK-KSVD shows good performance even with small dictionary size. This is due to the fact that it is not the reconstructive accuracy of the reconstructive term, but the discriminative ability of the discriminative feature, that influence the recognition accuracy directly.

C. Caltech101 Object Classification

The Caltech101 dataset [42] consists of 9144 images from 102 classes (i.e. 101 object classes and 1 background class collected randomly from the Internet) and each category contains from 31 to 800 images. This dataset is very challenging as it has diverse objects like animals, buildings and natural scenes, and the images have significant shape variability.

Following the suggested experiment settings in [18], [26], and [34], we train on 5, 10, 15, 20, 25, and 30 samples per category and test on the rest. The corresponding parameter settings are listed in Table III.

TABLE III

PARAMETER SETTINGS OF THE EXPERIMENTS ON THE CALTECH101 DATASET

Number of train. samp.	5	10	15	20	25	30
Dictionary size	2	2	3	3	3	3
Sparsity level	30	40	80	80	80	80
λ	0.64	0.64	0.64	0.64	0.64	0.64
$1/\delta$	0.3	0.4	0.4	0.4	0.4	0.4

TABLE IV

RECOGNITION RESULTS USING SPATIAL PYRAMID FEATURES ON THE CALTECH101 DATASET

Number of train. samp.	5	10	15	20	25	30
Malik [44]	46.6	55.8	59.1	62	-	66.2
Lazebnik [45]	-	-	56.4	-	-	64.6
Griffin [46]	44.2	54.5	59	63.3	65.8	67.6
Irani [47]	-	-	65	-	-	70.4
Grauman [48]	-	-	61	-	-	69.1
Pham [49]	-	-	42	-	-	-
Gemert [50]	-	-	-	-	-	64.16
Yang [51]	-	-	67	-	-	73.2
Wang [10]	51.15	59.77	65.43	67.74	70.16	73.44
SRC [16]	48.8	60.1	64.9	67.7	69.2	70.7
K-SVD [11]	49.8	59.8	65.2	68.7	71	73.2
D-KSVD [2]	49.6	59.5	65.1	68.6	71.1	73
LC-KSVD [18]	54	63.1	67.7	70.5	72.3	73.6
Kernel KSVD[5] [26]	55.73	63.72	68.46	71.73	74.02	76.12
CCSK-KSVD	57.00	64.61	69.52	72.69	74.99	77.08

The recognition results of our approach compared with some related work on the Caltech101 dataset are summarized in Table IV. CCSK-KSVD outperforms all the competing approaches with small dictionary size. The basic reason for the good recognition performance is that CCSK-KSVD is not a reconstructive approach to classification, in contrast to purely reconstructive ones [11], [16], [26]. In CCSK-KSVD, the reconstructive ability of the sparse codes over the learned dictionary is not related to the recognition performance directly. However, the projected sparse codes, being discriminative features, are directly related to the recognition performance. In other words, the sparse code provides a good representation of the input signal in our approach, which does not depend on its good reconstructive ability, but on the high discriminability of the new feature obtained by projecting it. Therefore, we do not need to learn a dictionary with large size to obtain sparse code with better reconstructive ability as long as the captured feature is discriminative enough for classification task.

D. MSTAR Target Recognition

MSTAR database is a standard dataset for evaluating SAR automatic target recognition (ATR) algorithms [43], [52], containing X-band SAR images with 1 ft $\times$ 1 ft resolution for multiple targets, such as tank, army truck, self-propelled anti-aircraft gun, and etc. For each target, images were captured at different depression angles over full 360° aspect views.

[5]It should be noted that the best recognition performance of Kernel KSVD in [26] used a combination of multiple features. For fair comparison, we execute it using the same feature as our approach.

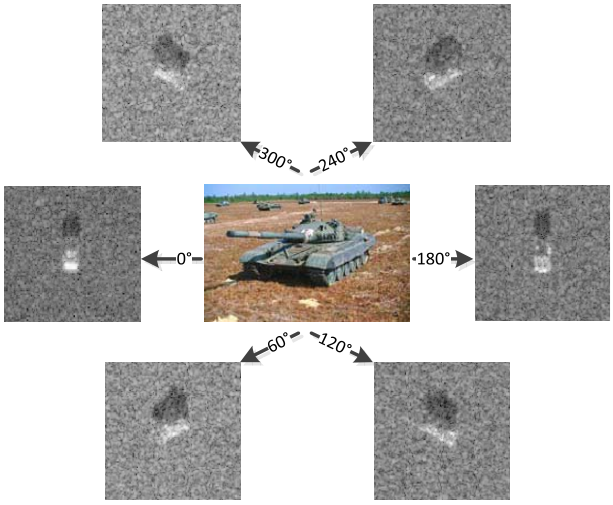


Fig. 9. Illustration of the multi-view log amplitude SAR images of T72 tank at depression angle 17°.

TABLE V

NUMBER OF THE MSTAR IMAGES USED FOR THREE-CLASS TARGET RECOGNITION

Class Label	Target Type		Train (17°)	Test(15°)
1	BMP2	SNC21	233	196
		SN9563	0	195
		SN9566	0	196
2	BTR70	C71	233	196
3	T72	SN132	232	196
		SN812	0	195
		SNS7	0	191
SUM	-	-	698	1365

The optical image of T72 tank as well as the corresponding log amplitude SAR images with different aspect angles is displayed in Fig. 9.

Following the suggested experiment settings in [53], we use a subset of the MSTAR database for three-class target recognition, including BMP2 infantry fighting vehicle, BTR70 armored personnel carrier, and T72 tank. As shown in Table V, 698 images captured at depression angle 17° are used for training and 1365 images captured at depression angle 15° are used for testing. The BMP2 and T72 have variants with different serial numbers. In the training stage, we only use the images from serial number SNC21 for BMP2 and SN132 for T72. In the testing stage, images from all the serial numbers listed in Table V are used. Such experiment setting aims at evaluating the classification system when recognizing objects with a 2° difference in depression angle and also recognizing variants without training samples for these variants. The original image, of size 128×128 pixels, are cropped around the center of the image to 64×64 pixels firstly, and then are projected onto a 512 dimensional vector with a randomly generated matrix from the standard normal distribution. Each row of the random matrix is normalized in terms of the l_2-norm.

The learned dictionary consists of 90 atoms, which corresponds to 30 atoms for each sub-dictionary. The following parameters are used for learning the dictionary in our approach: a Gaussian kernel with $1/\delta = 10$ is used; the scalar

TABLE VI
RECOGNITION RESULTS USING RANDOM-FACE FEATURES ON THE MSTAR DATABASE

Method	Accuracy (%)
SRC(all train. samp.) [16]	88.64
SRC*(30 per class) [16]	77.25
K-SVD(30 per class) [11]	86.81
D-KSVD(30 per class) [2]	86.74
LC-KSVD(30 per class) [18]	80.64
Kernel KSVD(30 per class) [26]	93.47
CCSK-KSVD(30 per class)	94.63

parameter is set as $\lambda = 0.0049$, and the sparsity level is $T = 60$, where the parameters λ and δ are determined by fivefold cross validation on the training dataset. Table VI illustrates the recognition results of our approach compared with some related works on the MSTAR database. For fair comparison, all the approaches except SRC learn dictionaries with the same size. Compared with the linear approaches, CCSK-KSVD has a remarkable improvement. Because, rather than learning linear dictionaries in the original data space, learning non-linear dictionaries in the projected high-dimensional space can better reveal the underlying structure of the data. Besides, compared with the non-linear Kernel KSVD [26] considering the reconstructive error only, CCSK-KSVD learns a non-linear dictionary with a discriminative constraint. And the constraint does make sense as the final recognition results show.

VI. CONCLUSION

We have proposed a novel discriminative non-linear DL approach, correlation constrained structured kernel KSVD, for object recognition. In order to learn a non-linear dictionary simultaneously reconstructive and discriminative, and to obtain discriminative features for classification, several contributions have been made. First, a structured kernel dictionary is designed, each sub-dictionary of which lies in the span of the mapped signals from the corresponding class with closer links with class labels. Second, a correlation constraint is added into the objective function, constraining the correlations between different discriminative codes, and restricting the coefficient vectors to be transformed into a feature space, in which the features are highly correlated inner-class and nearly independent between-classes. Third, a structured kernel KSVD algorithm is proposed to solve the optimization problem. Fourth, a linear SVM is trained for the classification task which matches the form of the extracted features. Experimental results show the effectiveness of the proposed CCSK-KSVD on the discriminability of the extracted features and on the classification performance. Improvements over the recognition accuracies on the four well-known publicly available databases indicate that exploiting discriminative non-linear dictionary can match the task of classification better than the linear counterparts as well as purely reconstructive non-linear ones.

Kernel methods are attractive because they can realize any linear or non-linear mapping implicitly. However, methods using kernel matrix scale poorly with the size of the dataset. Future work will include extending our approach to online dictionary learning method in order to deal with large datasets.

REFERENCES

[1] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Metaface learning for sparse representation based face recognition," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 1601–1604.

[2] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2691–2698.

[3] L. Ma, C. Wang, B. Xiao, and W. Zhou, "Sparse representation for face recognition based on discriminative low-rank dictionary learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2586–2593.

[4] Z. Feng, M. Yang, L. Zhang, Y. Liu, and D. Zhang, "Joint discriminative dimensionality reduction and dictionary learning for face recognition," *Pattern Recognit.*, vol. 46, no. 8, pp. 2134–2143, Aug. 2013.

[5] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 543–550.

[6] Y. Sun, Q. Liu, J. Tang, and D. Tao, "Learning discriminative dictionary for group sparse representation," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 3816–3828, Sep. 2014.

[7] J. Mairal, F. R. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2009, pp. 1033–1040.

[8] H. Zhang, Y. Zhang, and T. S. Huang, "Simultaneous discriminative projection and dictionary learning for sparse representation based classification," *Pattern Recognit.*, vol. 46, no. 1, pp. 346–354, Jan. 2013.

[9] M. J. Gangeh, A. Ghodsi, and M. S. Kamel, "Kernelized supervised dictionary learning," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4753–4767, Oct. 2013.

[10] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 3360–3367.

[11] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[12] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Mar. 1999, pp. 2443–2446.

[13] K. Skretting and J. H. Husoy, "Texture classification using sparse frame-based representations," *EURASIP J. Adv. Signal Process.*, vol. 2006, p. 052561, Dec. 2006.

[14] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 759–766.

[15] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[16] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[17] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.

[18] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013.

[19] W. Liu, Z. Yu, M. Yang, L. Lu, and Y. Zou, "Joint kernel dictionary and classifier learning for sparse coding via locality preserving K-SVD," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jun. 2015, pp. 1–6.

[20] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 3501–3508.

[21] Y. Xu, G. Bao, X. Xu, and Z. Ye, "Single-channel speech separation using sequential discriminative dictionary learning," *Signal Process.*, vol. 106, pp. 134–140, Jan. 2015.

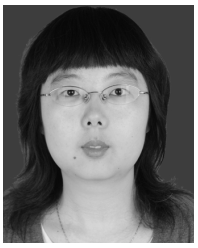
[22] J. Dong, C. Sun, and W. Yang, "A supervised dictionary learning and discriminative weighting model for action recognition," *Neurocomputing*, vol. 158, pp. 246–256, Jun. 2015.

[23] H.-D. Liu, M. Yang, Y. Gao, Y. Yin, and L. Chen, "Bilinear discriminative dictionary learning for face recognition," *Pattern Recognit.*, vol. 47, no. 5, pp. 1835–1845, 2014.

[24] Y. T. Chi, M. Ali, A. Rajwade, and J. Ho, "Block and group regularized sparse modeling for dictionary learning," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 377–382.

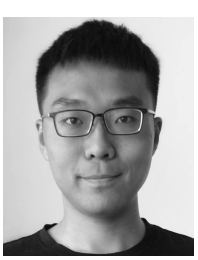
[25] Y. Suo, M. Dao, U. Srinivas, V. Monga, and T. D. Tran. (2014). "Structured dictionary learning for classification." [Online]. Available: https://arxiv.org/abs/1406.1943
[26] H. Van Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Design of non-linear kernel dictionaries for object recognition," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5123–5135, Dec. 2013.
[27] S. Gao, I. W.-H. Tsang, and L. T. Chia, "Kernel sparse representation for image classification and face recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 1–14.
[28] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[29] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, pp. 1299–1319, Jul. 1998.
[30] F. R. Bach and M. I. Jordan, "Kernel independent component analysis," *J. Mach. Learn. Res.*, vol. 3, pp. 1–48, Jan. 2002.
[31] A. Shrivastava, H. V. Nguyen, V. M. Patel, and R. Chellappa, "Design of non-linear discriminative dictionaries for image classification," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, 2012, pp. 660–674.
[32] A. Shrivastava, V. M. Patel, and R. Chellappa, "Non-linear dictionary learning with partially labeled data," *Pattern Recognit.*, vol. 48, no. 11, pp. 3283–3292, Nov. 2015.
[33] S. Bahrampour, N. M. Nasrabadi, A. Ray, and K. W. Jenkins, "Kernel task-driven dictionary learning for hyperspectral image classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 1324–1328.
[34] W. Liu, Z. Yu, L. Lu, Y. Wen, H. Li, and Y. Zou, "KCRC-LCD: Discriminative kernel collaborative representation with locality constrained dictionary for visual categorization," *Pattern Recognit.*, vol. 48, no. 10, pp. 3076–3092, Oct. 2015.
[35] A. Golts and M. Elad, "Linearized kernel dictionary learning," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 726–739, Jun. 2016.
[36] G. Zhang, H. Sun, G. Xia, and Q. Sun, "Kernel collaborative representation based dictionary learning and discriminative projection," *Neurocomputing*, vol. 207, pp. 300–309, Sep. 2016.
[37] Z. Chen, W. Zuo, Q. Hu, and L. Lin, "Kernel sparse representation for time series classification," *Inf. Sci.*, vol. 292, pp. 15–26, Jan. 2015.
[38] L. Song, K. Fukumizu, and A. Gretton, "Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 98–111, Jul. 2013.
[39] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, 2011.
[40] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.
[41] A. M. Martinez and R. Benavente, "The AR face database," CVC, Luxembourg, Tech. Rep. 24, Jun. 1998.
[42] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Comput. Vis. Image Understand.*, vol. 106, no. 1, pp. 59–70, Jan. 2007.
[43] *Moving and Stationary Target Acquisition and Recognition (MSTAR) Public Dataset*, accessed on Mar. 3, 2011. [Online]. Available: https://www.sdms.afrl.af.mil/datasets/mstar/
[44] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. 2006, pp. 2126–2136.
[45] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. 2006, pp. 2169–2178.
[46] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category data set," CIT, Coimbatore, India, Tech. Rep. 7694, 2007.
[47] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, p. 1–8.
[48] P. Jain, B. Kulis, and K. Grauman, "Fast image search for learned metrics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.
[49] D. S. Pham and S. Venkatesh, "Joint learning and dictionary construction for pattern recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.
[50] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders, "Kernel codebooks for scene categorization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 696–709.
[51] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 1794–1801.
[52] H. Zhang, M. Nasser, and Y. Zhang, "Multi-view automatic target recognition using joint sparse representation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 3, pp. 2481–2497, Jul. 2012.
[53] Q. Zhao and J. C. Principe, "Support vector machines for SAR automatic target recognition," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, no. 2, pp. 643–654, Apr. 2001.

Zhengjue Wang received the B.S. and M.S. degrees in electronic engineering from Xidian University, Xi'an, China, in 2013 and 2016, respectively. She is currently pursuing the Ph.D. degree with Xidian University. Her research interests include machine learning and radar automatic target recognition.



Yinghua Wang received the B.S. degree in information engineering and the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2004 and 2010, respectively. In 2007, she was a Visiting Student with the Image and Signal Processing Department, Telecom Paris, Paris, France. She is currently an Associate Professor with the National Laboratory of Radar Signal Processing, Xidian University, Xi'an, China. Her research interests include synthetic aperture radar (SAR) automatic target recognition, polarimetric SAR data analysis and interpretation, and SAR image processing.

Hongwei Liu (M'04) received the M.S. and Ph.D. degrees in electronic engineering from Xidian University, Xi'an, China, in 1995 and 1999, respectively. From 2001 to 2002, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA. He is currently a Professor with the National Laboratory of Radar Signal Processing, Xidian University. His research interests include radar automatic target recognition, radar signal processing, and adaptive signal processing.



Hao Zhang received the B.S. degree in electronic engineering from Xidian University, Xi'an, China, in 2012. He is currently pursuing the Ph.D. degree with Xidian University. His research interests include statistical machine learning and radar automatic target recognition.