Pattern Recognition 48 (2015) 3076–3092

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

KCRC-LCD: Discriminative kernel collaborative representation with locality constrained dictionary for visual categorization

Weiyang Liu [a], Zhiding Yu [b,*], Lijia Lu [a], Yandong Wen [c], Hui Li [a], Yuexian Zou [a]

[a] School of Electronic and Computer Engineering, Peking University, China
[b] Department of Electrical and Computer Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA
[c] School of Electronic and Information Engineering, South China University of Technology, China

ARTICLE INFO

Article history:
Received 5 October 2014
Received in revised form
3 April 2015
Accepted 10 April 2015
Available online 22 April 2015

Keywords:
Kernel collaborative representation
Regularized least square algorithm
Nearest neighbor
Locality constrained dictionary

ABSTRACT

We consider the image classification problem via kernel collaborative representation classification with locality constrained dictionary (KCRC-LCD). Specifically, we propose a kernel collaborative representation classification (KCRC) approach in which kernel method is used to improve the discrimination ability of collaborative representation classification (CRC). We then measure the similarities between the query and atoms in the global dictionary in order to construct a locality constrained dictionary (LCD) for KCRC. In addition, we discuss several similarity measure approaches in LCD and further present a simple yet effective unified similarity measure whose superiority is validated in experiments. There are several appealing aspects associated with LCD. First, LCD can be nicely incorporated under the framework of KCRC. The LCD similarity measure can be kernelized under KCRC, which theoretically links CRC and LCD under the kernel method. Second, KCRC-LCD becomes more scalable to both the training set size and the feature dimension. Example shows that KCRC is able to perfectly classify data with certain distribution, while conventional CRC fails completely. Comprehensive experiments on widely used public datasets also show that KCRC-LCD is a robust discriminative classifier with both excellent performance and good scalability, being comparable or outperforming many other state-of-the-art approaches.

1. Introduction

Recent years have witnessed the great success of the sparse representation techniques in a variety of problems in computer vision, including image restoration [1], image denoising [2] as well as image classification [3–5]. Sparse representation is widely believed to bring many benefits to classification problems in terms of robustness and discriminativeness. Specifically, sparsity regularization term can reduce the solution space under ill-conditioned problems, by seeking to represent a signal as a linear combination of only a few bases. These bases are called the "atoms" and the whole overcomplete collection of atoms together form what one call a "dictionary". Many natural signals such as image and audio indeed have sparsity priors [6]. Imposing sparsity not only returns a unique solution, but also helps us to recover the true signal structure, giving more robust estimation against noise. In addition, the sparse representation of a signal often leads to better separation and decorrelation which benefits subsequent classification problems. Despite the fact that sparse optimization is a nonconvex problem, the l_1-norm convex relaxation and its optimization techniques have been thoroughly studied [7–9].

Wright et al. [3] employed the entire set of the training samples as the dictionary and reported a discriminative sparse representation-based classification (SRC) with promising performance on face recognition. SRC approximates an input signal $\boldsymbol{y}$ with a linear combination of the atoms from an overcomplete dictionary $\boldsymbol{D}$ under the sparsity constraint and gives the predicted label by selecting the minimum reconstruction residuals. Despite the fact that SRC was widely used in various applications [10–12], some work [13–15] still questioned the role that sparse representation plays in the image classification tasks.

Zhang et al. [15] further commented that it is unnecessary to enforce the sparse constraint with computationally expensive l_1-norm if the feature dimension is high enough. Their work

Abbreviation: CRC, Collaborative representation classification; KCRC, Kernel collaborative representation classification; LCD, Locality constrained dictionary; GD, Global dictionary; KCRC-LCD, CRC with locality constrained dictionary; KCRC-GD, CRC with global dictionary; KCRC-LCD, KCRC with locality constrained dictionary; KCRC-GD, KCRC with global dictionary; SRC, Sparse representation-based classification; KSRC, Kernel sparse representation-based classification; SVM, Support vector machine; KPCA, Kernel principle component analysis; KFDA, Kernel fisher discriminant analysis; KCRC-Identity, KCRC with no dimensionality reduction; KCRC-KPCA, KCRC with KPCA dimensionality reduction; KCRC-RP, KCRC with random projection; KCRC-Graph, KCRC with graph projection; CRC-RLS, CRC with regularized least square; KCRC-RLS, Kernel CRC with regularized least square; RCRC, Robust CRC; LC-KSVD, Locality constrained K-SVD; D-KSVD, Discriminative K-SVD

[*] Corresponding author. Tel.: +1 850 567 2583.
E-mail address: yzhiding@andrew.cmu.edu (Z. Yu).

http://dx.doi.org/10.1016/j.patcog.2015.04.014
0031-3203/© 2015 Elsevier Ltd. All rights reserved.

emphasized the importance of collaborative representation (CR) rather than the sparse representation, arguing that CR is the key to the improvement of classification accuracy, which was validated by their comparison experiments. They unified both l_1-norm and l_2-norm into a generic framework, and used the l_2-norm regularization instead of l_1-norm in classification tasks, further improving the classification accuracy while significantly reducing much computational cost. The corresponding proposed framework is called collaborative representation classification (CRC).

Despite their robust performance, the linear nature of both SRC and CRC makes them perform poorly when the training data are distributed linearly in one direction.[1] Kernel function, proven useful in kernel principal component analysis (KPCA) [16] and support vector machine (SVM) [17], was introduced to overcome such shortcoming for both SRC and CRC, leading to the kernel sparse representation-based classification (KSRC) [18] and the kernel collaborative representation classification (KCRC) [19]. In particular, a Mercer kernel implicitly defines a nonlinear mapping to map the data from the input space into a high or even infinite dimensional kernel space where different classes become more separable.

Besides summarizing the KCRC [19], our major contribution in this paper lies in proposing a generalized framework for KCRC with locality constrained dictionary and unified similarity measure, giving both performance gain and significant reduction of computational cost. Due to the poor scalability of the global dictionary (GD) used in CR-based methods, classification becomes intractable in large dataset for KCRC with GD (KCRC-GD). To enable the scalability to large dataset, we prune the dictionary via k-nearest neighbor (K-NN) classifier to enforce locality. Specifically, the nearest neighbors of a query serve to construct a locality constrained dictionary (LCD) for KCRC. Such strategy is both intuitively reasonable and mathematically appealing. Intuitively, LCD is well motivated by the psychological findings about human perception that visual categories are not defined by lists of features, but rather by similarity to prototypes [20]. In other words, coarse level matching, for which K-NN is used, plays an important role in human perception. Mathematically, LCD can be nicely incorporated under the framework of KCRC. First, the LCD similarity measure can also be kernelized under KCRC, which theoretically links CRC and LCD under the kernel method. Second, KCRC-LCD becomes more scalable to both the training set size and the feature dimension. The kernelized query sample is directly obtained from the similarity measure that is used to construct LCD, while KCRC operates on the reduced kernel matrix without referring to original features. Moreover, the kernel Gram matrix is now obtained from a subset of the global dictionary. (The kernel Gram matrix of the global dictionary is computed in advance.) Therefore, the combination of KCRC and LCD makes classification highly efficient and scalable. In fact, the advantages of KCRC-LCD become more obvious with extremely large training set.

The high level intuition of LCD is that local dictionary atoms are typically the most important and informative samples. Looking into these representative exemplars often brings even more gains than globally considering all samples together. It is not hard to see similar concepts and link the connections. For example, if the query is located near decision boundary, then these local atoms play the role similar to support vectors in an SVM, or in an extreme case, exemplars in an exemplar SVM [21]. In a model recommendation system, selecting the most responsive models instead of all models has been reported to give gains [15]. In fact, SRC also seeks to use only few exemplar atoms in the dictionary. Yet the proposed method is able to run much faster with even better performance. In the extreme scenario, if K equals the number of atoms, then the proposed KCRC-LCD degenerates to regular KCRC with global dictionary. If K equals 1, KCRC-LCD degenerates to the simplest nearest neighbor classifier.

In this paper, we specifically focus on the application of our proposed framework to the image classification/visual categorization problem, demonstrating its robust performance with a comprehensive series of image classification tasks. Image classification is among the most fundamental computer vision problems where each image is labeled with a certain or multiple categories/tags. Though great advance has been achieved, much is pending to be done since the current state-of-the-art approaches are far from being able to achieve human-level performance, particularly in handling cluttered, complicated scenarios and inferring abstract concepts. Such gap between the machine and human remains an open challenge, motivating us to exploit more discriminative and efficient image classifiers.

The outline of the paper is as follows. Section 2 discusses related work of KCRC-LCD and presents our main contributions. In Section 3, necessary preliminaries are briefly introduced. Section 4 elaborates the formulation of KCRC-LCD and discusses some important details. The locality constrained dictionary is proposed and discussed in Section 5. Experimental results are discussed in Section 6, followed by concluding remarks in Section 7.

2. Related work

Pioneering works on kernelizing SRC were proposed in [22,18,23]. Gao et al. first proposed the idea of kernel-based SRC with promising experimental results. Zhang et al. [18] further unified the mathematical model [22] to a generic kernel framework and conducted more comprehensive experiments to evaluate the performance. To overcome the shortcoming of handling data with certain distributions (e.g. the same direction distribution), our previous work [19] addressed the problem of kernel collaborative representation. The authors presented a smooth formulation to incorporate kernel function into the CRC model. A practical application of kernel CRC in vehicle logo recognition was further discussed in [24]. We happened to notice that a very recent work [25] proposed a similar idea by combining the column-generation kernel to CRC for hyperspectral image classification. It should be pointed out, however, that both the formulations and the applications are significantly different when dealing with the high dimension in kernel space. In terms of application, [25] formulated the kernel collaborative representation on pixel-level tasks for hyperspectral images while our method focuses on image-level classification. Significant differences also exist in the formulation: [25] incorporated the kernel function with column generation without considering dimensionality reduction in kernel space (possibly due to the characteristics of the hyperspectral classification task). On the contrary, our method combines the CRC with kernel function in a strategy similar to KPCA and kernel Fisher discriminant analysis (KFDA). Moreover, a series of dimensionality reduction approaches have been taken into account in our generalized formulation. In general, we aim at extending the idea of KCRC by further improving formulation details of KCRC and presenting specific methods to perform dimensionality reduction in kernel space.

We also noticed that [26] presented a similar idea of constructing locally adaptive dictionary, but such dictionary pruning strategy was only applied in the standard CRC framework instead of the kernel CRC framework. As one shall see, the proposed KCRC-LCD is not a trivial extension of CRC-LCD by combining KCRC with locality

[1] To be clear, the data that are distributed linearly in one direction, or in other words, a special linear space spanned by base vectors of (approximately) the same direction.

constrained dictionary, but a well-motivated and appealing framework in which LCD and KCRC are theoretically linked by kernelizing the distance used in LCD. In addition, the kernelization in conjunction with LCD not only brings scalability in terms of dataset size, but also further extends its scalability to feature dimensionality. We will show that the locally adaptive dictionary in [26] is a special case of our proposed LCD.

Compared to existing kernel-based sparse representation approaches [5,27–29], our approach attaches more importance to the scalability issue while achieving competitive classification performance. Refs. [27,28] propose a non-linear kernel dictionary learning method. Specifically, it replaces the original dictionary with the multiplication of the non-linear mapping of input samples and a transformation matrix, and learns the transformation matrix instead of the original dictionary. Different from [27,28], our approach directly kernelizes the dictionary without learning one, and utilizes a coarse-to-fine framework to generate the LCD for classification. Without the matrix learning process introduced in [27,28], KCRC-LCD is a more scalable and efficient classification framework and is capable of handling large dictionary. Ref. [5] presents a multi-task joint sparse representation model to combine the strength of multiple features and instances for classification, and further extend such model to the kernelized features setup. KCRC-LCD considers a different task from [5] and does not explicitly aim at multiple features and instances setup. Moreover, KCRC-LCD is proposed under the collaborative representation framework, while [5,27,28] are under sparse representation framework. The underlying basic classification frameworks therefore are quite different.

3. Preliminaries

3.1. Collaborative representation classification

The principle of CRC [15] is briefly presented in this section. In CRC, the dictionary $\boldsymbol{D}$ is constructed by all training samples and a test sample $\boldsymbol{y}$ is encoded collaboratively over the whole dictionary.

Let $\boldsymbol{D}$ be the dictionary which is a set of k-class training samples (with the ith class having n_i samples), i.e., $\boldsymbol{D} = \{\boldsymbol{D}_1, \boldsymbol{D}_2, \dots, \boldsymbol{D}_k\} \in \mathbb{R}^{m\times n}$ where $n = \sum_{i=1}^{k} n_i$ and m is the feature dimension. The dictionary associated with the ith class is denoted by $\boldsymbol{D}_i = \{\boldsymbol{d}_1^{[i]}, \boldsymbol{d}_2^{[i]}, \dots, \boldsymbol{d}_{n_i}^{[i]}\} \in \mathbb{R}^{m\times n_i}$, where $\boldsymbol{d}_j^{[i]}$, also called atom, stands for the jth training image in the ith class. Note that, $\boldsymbol{d}_j^{[i]}$ and $\boldsymbol{y}$ should be normalized to have unit l_2-norm. We also denote the label of a query sample $\boldsymbol{y}$ as $id(\boldsymbol{y})$. To encode each query sample, one solves the following optimization problem:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_p$$
$$\text{subj. to} \quad \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|_q \le \varepsilon \tag{1}$$

where $p, q \in \{1, 2\}$, $\varepsilon > 0$ is a small error constant. Using Lagrangian multiplier, CRC can be reformulated as

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left(\|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|_q^q + \mu\|\boldsymbol{x}\|_p^p \right) \tag{2}$$

where μ is the regularization parameter. The combinations of p, q lead to different instantiations of CRC model. For instance, having p=1 leads to the SRC model, and different settings of q can be used to handle classification with or without occlusion. Similar to SRC, CRC predicts the class label via reconstruction residuals:

$$id(\boldsymbol{y}) = \arg\min_i \left(\|\boldsymbol{y} - \boldsymbol{D}_i\hat{\boldsymbol{x}}_i\|_2 / \|\hat{\boldsymbol{x}}_i\|_2 \right). \tag{3}$$

Setting p to 2 instead of 1 can reduce the computational complexity. Based on different combinations of p, q, two CRC algorithms were proposed [15,30]. One is the CRC regularized least square (CRC-RLS) algorithm with $p = 2, q = 2$. The other is the robust CRC (RCRC) algorithm with $p = 2, q = 1$. The authors of [15] argued that the sparsity of a signal can be useful but not crucial for face recognition. What really plays an important role is the mechanism of collaborative representation..

3.2. Kernel technique

Kernel methods refer to a class of algorithms for pattern analysis, whose best known members are the SVM [31,17], KPCA [16] and KFDA [32]. The general task of pattern analysis is to find and study general types of relations (for example clusters, rankings, principal components, correlations, classifications) in datasets. For many algorithms that solve these tasks, the data in raw representation have to be explicitly transformed into feature vector representations via a user-specified feature map: in contrast, kernel methods require only a user-specified kernel, i.e., a similarity function over pairs of data points in raw representation. Via kernels, one can easily generalize a linear classifier to a nonlinear one, generating a reasonable decision boundary and consequently enhancing the discrimination power.

Kernel methods owe their name to the use of kernel functions, which enable them to operate in a high-dimensional, implicit feature space without computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs of data in the feature space. This operation is often computationally cheaper than the explicit computation of the coordinates. Kernel functions have been introduced for sequence data, graphs, text, images, as well as vectors. The amazing part of kernel function is that it surpasses the direct calculation in the feature space and performs the classification in the reproducing kernel Hilbert space (RKHS), boosting the classification performance.

Algorithms that are capable of operating with kernels include the kernel perceptron [33–35], SVM [31,17], Gaussian processes [36], principal components analysis (PCA) [16], canonical correlation analysis [37], and spectral clustering [38]. In general, any linear model can be transformed into a non-linear model by applying the kernel trick: replacing its features by a kernel function.

4. Proposed KCRC approach

4.1. Formulation of KCRC

To overcome the shortcoming of CRC in handling data with certain distributions (e.g. the same direction distribution), kernel technique is smoothly combined with CRC. Kernel function is used to create a nonlinear mapping mechanism $\boldsymbol{v} \in \mathbb{R}^m \mapsto \boldsymbol{\phi}(\boldsymbol{v}) \in \mathbb{H}$ in which $\mathbb{H}$ is a unique associated RKHS. If every sample is mapped into higher dimensional space via transformation $\boldsymbol{\phi}$, the kernel function computes a dot-product in the higher dimensional space, written as

$$K(\boldsymbol{v}', \boldsymbol{v}'') = \langle \boldsymbol{\phi}(\boldsymbol{v}'), \boldsymbol{\phi}(\boldsymbol{v}'') \rangle \tag{4}$$

where $\boldsymbol{v}'$ and $\boldsymbol{v}''$ are any two samples, and $\boldsymbol{\phi}$ denotes the implicit nonlinear mapping associated with the kernel function $K(\boldsymbol{v}', \boldsymbol{v}'')$. There are some empirical kernel functions satisfying the Mercer condition such as the linear kernel $K(\boldsymbol{v}', \boldsymbol{v}'') = \boldsymbol{v}'^T\boldsymbol{v}''$ and Gaussian radial basis function (RBF) kernel $K(\boldsymbol{v}', \boldsymbol{v}'') = \exp(-\beta\|\boldsymbol{v}' - \boldsymbol{v}''\|_2^2)$. According to [39], the distance function for similarity measurement, designed to construct the LCD, can be transformed in a straightforward way to the kernel for KCRC, via the linear kernel function:

$$K(\boldsymbol{v}', \boldsymbol{v}'') = \langle \boldsymbol{\phi}(\boldsymbol{v}), \boldsymbol{\phi}(\boldsymbol{v}') \rangle = \langle \boldsymbol{v}, \boldsymbol{v}' \rangle$$

$$\frac{1}{2}(\langle \boldsymbol{v}', \boldsymbol{v}' \rangle + \langle \boldsymbol{v}'', \boldsymbol{v}'' \rangle - \langle \boldsymbol{v}' - \boldsymbol{v}'', \boldsymbol{v}' - \boldsymbol{v}'' \rangle)$$
$$\frac{1}{2}(Dist(\boldsymbol{v}', 0) + Dist(\boldsymbol{v}'', 0) - Dist(\boldsymbol{v}', \boldsymbol{v}'')) \tag{5}$$

where *Dist* is the carefully designed distance function, and the location of the origin(0) does not affect the result [39]. Various ways of transforming a distance function into a kernel are possible [40], e.g. $K(\boldsymbol{v}', \boldsymbol{v}'')$ can be $\exp(-\beta Dist(\boldsymbol{v}', \boldsymbol{v}''))$.[2] Such reformulated kernels can make best use of the distance metrics that are introduced in the following sections, because we just need to do a simple exponential operation on the distance matrix to obtain kernel matrix, reducing the computational cost.

It is learned in [17] that the sample feature nonlinearly transformed to high dimensional space becomes more separable. Most importantly, the same direction distribution of data can be avoided in kernel space. However, mapping to high dimensional space makes CRC model harder to solve, so we need to perform dimensionality reduction in the kernel feature space. The nonlinear mapping mechanism is

$$\boldsymbol{y} \in \mathbb{R}^m \mapsto \boldsymbol{\phi}(\boldsymbol{y}) = [\phi_1(\boldsymbol{y}), \phi_2(\boldsymbol{y}), \ldots, \phi_s(\boldsymbol{y})] \in \mathbb{F} \tag{6}$$

where $\boldsymbol{\phi}(\boldsymbol{y}) \in \mathbb{R}^s$ is the high dimensional feature (possibly of infinite dimensions, namely s could be infinite) corresponding to the sample $\boldsymbol{y}$ in the feature space $\mathbb{F}$, and s is much larger than m. We then define a universal label $[k]$ for $\boldsymbol{d}_j^{[i]}$ that denotes its position in the global dictionary, satisfying $k = j + \sum_{l=1}^{i-1} n_l$ (When i=1, k simply equals to j). For conciseness, we only preserve the universal label, representing atom as $\boldsymbol{d}_{[k]}$. According to the nonlinear mapping mechanism, the original dictionary $\boldsymbol{D}$ becomes a much higher dimensional one: $\boldsymbol{\Phi} = \{\boldsymbol{\phi}(\boldsymbol{d}_{[1]}), \boldsymbol{\phi}(\boldsymbol{d}_{[2]}), \ldots, \boldsymbol{\phi}(\boldsymbol{d}_{[n]})\} \in \mathbb{R}^{s \times n}$, and the test sample becomes $\boldsymbol{\phi}(\boldsymbol{y}) = \boldsymbol{\Phi x}$. The KCRC model is formulated as

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_p$$
$$\text{subj. to} \quad \|\boldsymbol{\phi}(\boldsymbol{y}) - \boldsymbol{\Phi x}\|_q \le \varepsilon. \tag{7}$$

However, Eq. (7) is even harder to solve than Eq. (1) because the high dimensionality results in high complexity. A dimensionality reduction matrix $\boldsymbol{R}$, namely a projection matrix, can be constructed by utilizing the methodology in KPCA [16] and KFDA [32]. With the matrix $\boldsymbol{R} \in \mathbb{R}^{s \times c}$, we derive

$$\boldsymbol{R}^T \boldsymbol{\phi}(\boldsymbol{y}) = \boldsymbol{R}^T \boldsymbol{\Phi x} \tag{8}$$

where $\boldsymbol{R}$ is related to kernelized samples. According to KPCA and KFDA, each column vector in $\boldsymbol{R}$ should be a linear combination of kernelized samples in KCRC. Namely

$$\boldsymbol{R} = \boldsymbol{\Phi\Psi} = \{\boldsymbol{\phi}(\boldsymbol{d}_{[1]}), \ldots, \boldsymbol{\phi}(\boldsymbol{d}_{[n]})\} \cdot \{\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_c\} \tag{9}$$

where $\boldsymbol{R} = \{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_s\}$ and $\boldsymbol{\psi}_i$ is the n-dimensional linear projection coefficients vector corresponding to the $\boldsymbol{R}_i = \sum_{j=1}^{n} \psi_{i,j} \boldsymbol{\phi}(\boldsymbol{d}_{[j]}) = \boldsymbol{\Phi\psi}_i$. Moreover, $\boldsymbol{\Psi} \in \mathbb{R}^{n \times c}$ is also called pseudo-transformation matrix [18]. Then we put Eq. (9) into Eq. (8) and obtain

$$(\boldsymbol{\Phi\Psi})^T \boldsymbol{\phi}(\boldsymbol{y}) = (\boldsymbol{\Phi\Psi})^T \boldsymbol{\Phi x} \tag{10}$$

from which we get $\boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}) = \boldsymbol{\Psi}^T \boldsymbol{G x}$, where $\boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}) = [K(\boldsymbol{d}_{[1]}, \boldsymbol{y}), \ldots, K(\boldsymbol{d}_{[n]}, \boldsymbol{y})^T$. $\boldsymbol{G}$ ($G_{ij} = K(\boldsymbol{d}_{[i]}, \boldsymbol{d}_{[j]})$), also equal to $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$, is defined as the kernel Gram matrix that is symmetric and positive semi-definite according to Mercer's theorem. Since $\boldsymbol{G}$ and $\boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y})$ are given a priori, dimensionality reduction requires to find $\boldsymbol{\Psi}$ instead of $\boldsymbol{R}$. Several methods were introduced in [18,32,16] to determine the pseudo-transformation matrix $\boldsymbol{\Psi}$. We will also further introduce the selection of matrix $\boldsymbol{\Psi}$ in the next subsection. Note that, if $\boldsymbol{\Psi}$ is an identity matrix, no dimensionality reduction is applied. Particularly, $\boldsymbol{\Psi}$ can also be a random projection matrix to achieve dimensionality reduction. After substituting the equivalent kernel function constraint, we can derive

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_p$$
$$\text{subj. to} \quad \|\boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}) - \boldsymbol{\Psi}^T \boldsymbol{G x}\|_q < \varepsilon \tag{11}$$

which is the model of the KCRC approach. $\boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y})$ and $\boldsymbol{\Psi}^T \boldsymbol{G}$ should be normalized to have unit l_2-norm. The normalization maps both test and training data onto a hypersphere, so that the representation coefficients are no longer affected by unbalanced feature values. Additionally, a small perturbation would be added to $\boldsymbol{\Psi}^T \boldsymbol{G}$ if the norm of its column is close to 0. Another form of KCRC model is expressed as

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left(\|\boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}) - \boldsymbol{\Psi}^T \boldsymbol{G x}\|_q^q + \mu \|\boldsymbol{x}\|_p^p \right) \tag{12}$$

from which we could derive two specific algorithms. It can be learned from standard optimization theory that Eqs. (11) and (12) are equivalent if ϵ and μ obey some special relationship [41]. With $p = 2, q = 2$, $\boldsymbol{x}$ can be solved at the cost of low computational complexity. The regularized least square algorithm is used to solve the optimization problem (Algorithm 1). The corresponding KCRC model is

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left(\|\boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}) - \boldsymbol{\Psi}^T \boldsymbol{G x}\|_2^2 + \mu \|\boldsymbol{x}\|_2^2 \right). \tag{13}$$

For handling images with occlusion and corruption, we can set $p = 2, q = 1$ for robustness, making the first term a l_1 regularized one. Let $\boldsymbol{e} = \boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}) - \boldsymbol{\Psi}^T \boldsymbol{G x}$ and $p = 2, q = 1$. Eq. (11) is rewritten as

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left(\|\boldsymbol{e}\|_1 + \mu \|\boldsymbol{x}\|_2^2 \right)$$
$$\text{subj. to} \quad \boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y}) = \boldsymbol{\Psi}^T \boldsymbol{G x} + \boldsymbol{e} \tag{14}$$

which is a constrained convex optimization problem that can be solved by the augmented Lagrange multiplier (ALM) method [42,43] as shown in Algorithm 2.

4.2. *Determining the pseudo-transformation matrix for dimensionality reduction*

This subsection reviews several typical methods that are proposed in [18] to determine the pseudo-transformation matrix $\boldsymbol{\Psi}$ for dimensionality reduction. Moreover, we also present a graph preserving method that has not been utilized to construct pseudo-transformation matrix in previous work.

4.2.1. *KPCA*

Following the methodology in KPCA, the pseudo-transformation vectors $\boldsymbol{\psi}_i \in \mathbb{R}^n$ refer to normalized eigenvectors corresponding to nonzero eigenvalues (or greater than a threshold) which can be obtained from the following eigenvalue problem [16]:

$$n\lambda \boldsymbol{\psi}_i = \boldsymbol{G} \boldsymbol{\psi}_i \tag{15}$$

where $\boldsymbol{\psi}_i$ is normalized to satisfy $\lambda_i \boldsymbol{\psi}_i^T \boldsymbol{\psi}_i = 1$. Eq. (15) can be easily solved by singular value decomposition (SVD) method. $\boldsymbol{\Psi}$ is equal to $\{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_c\}$.

4.2.2. *KFDA*

For KFDA [32], $\boldsymbol{\Psi} \in \mathbb{R}^{n \times c}$ is the solution of the optimization problem shown as follows:

$$\hat{\boldsymbol{\Psi}} = \arg\max_{\boldsymbol{\Psi}} \frac{\operatorname{tr}\left(\boldsymbol{\Psi}^T \boldsymbol{S}_b^{\boldsymbol{G}} \boldsymbol{\Psi}\right)}{\operatorname{tr}\left(\boldsymbol{\Psi}^T \boldsymbol{S}_\omega^{\boldsymbol{G}} \boldsymbol{\Psi}\right)} \tag{16}$$

where $\operatorname{tr}(\cdot)$ denotes the trace of a matrix, and $\boldsymbol{S}_b^{\boldsymbol{G}}$ and $\boldsymbol{S}_\omega^{\boldsymbol{G}}$ stand for quasi-between-class and quasi-within-class scatter matrices

[2] For intuitive interpretation, we stick to this simple kernel function throughout this paper as well as experiments.

respectively. Specifically, $\boldsymbol{S}_b^{\boldsymbol{G}}$ and $\boldsymbol{S}_\omega^{\boldsymbol{G}}$ are defined respectively as

$$\boldsymbol{S}_b^G = \frac{1}{k(k-1)} \sum_{i=1}^{k} \sum_{j=1}^{k} (\boldsymbol{u}_i - \boldsymbol{u}_j)(\boldsymbol{u}_i - \boldsymbol{u}_j)^T$$

$$\boldsymbol{S}_\omega^G = \frac{1}{k} \sum_{i=1}^{k} \frac{1}{n_i} \sum_{j=1}^{n_i} (\boldsymbol{\phi}(\boldsymbol{d}_j^{[i]}) - \boldsymbol{u}_i)(\boldsymbol{\phi}(\boldsymbol{d}_j^{[i]}) - \boldsymbol{u}_i)^T \tag{17}$$

where $\boldsymbol{u}_i = 1/n_i \cdot \sum_{j=1}^{n_i} \boldsymbol{\phi}(\boldsymbol{d}_j^{[i]})$ denotes the sample mean of class i. Note that, k denotes the class number and n_i is the sample number of class i.

4.2.3. Random projection

Ref. [18] also proposed a simple and practical random dimensionality reduction method. Since random projection cannot be performed in the RHKS, we can make $\boldsymbol{\Psi}$ a Gaussian random matrix to reduce the dimensionality. Random projection can be viewed as a less-structured counterpart to classic dimensionality reduction methods like PCA and FDA. In other words, the critical information will be preserved in a less-structured way.

4.2.4. Identity

Particularly, the pseudo-transformation matrix $\boldsymbol{\Psi}$ can be defined as an identical matrix with ones on the main diagonal and zeros elsewhere, which indicates no dimensionality reduction is performed in the RHKS. The method is the most simple way for dimensionality reduction in KCRC, but it is effective at most time, especially in KCRC-LCD. LCD is usually constructed in relatively small size compared to the training sets, so we do not always need to perform dimensionality reduction in kernel space whose dimension is equal to the dictionary size. Thus, in the classification experiments on public datasets, we simply use identity matrix as $\boldsymbol{\Psi}$.

4.2.5. Graph

Further, we propose a graph preserving dimensionality reduction method for the pseudo-transformation matrix $\boldsymbol{\Psi}$. In the light of [44], we first construct a weighted graph with n nodes (n is the dictionary size, one node represents one atom in the dictionary). Then we put an edge between nodes i and j if they are close enough. There are two popular methods to find the nodes that we use to construct the graph. The first is ϵ-neighborhoods in which node i and node j are connected by an edge if $\|\boldsymbol{d}_{[i]} - \boldsymbol{d}_{[j]}\|^2 < \epsilon$.[3] The second is n-nearest neighbors in which nodes i and j connected by an edge if $\boldsymbol{d}_{[i]}$ is among n nearest neighbors of $\boldsymbol{d}_{[j]}$ or $\boldsymbol{d}_{[j]}$ is among n nearest neighbors of $\boldsymbol{d}_{[i]}$. After establishing graphical connection between nodes, we choose a measure to determine the weights. In [44], the following weight measure ($\mathbf{R}$ is a parameter) between two connected nodes is formulated as

$$W_{ij} = \exp\left(\frac{\|\boldsymbol{d}_{[i]} - \boldsymbol{d}_{[j]}\|^2}{t}\right). \tag{18}$$

There is another simple weighting method that $W_{ij} = 1$ if and only if vertices i and j are connected by an edge. In order to group the connected nodes and separate the distant nodes as much as possible, the object function is defined as

$$\sum_{ij} (\boldsymbol{g}_i - \boldsymbol{g}_j)^2 W_{ij} = 2\boldsymbol{g}^T \boldsymbol{L}\boldsymbol{g} \tag{19}$$

where $\boldsymbol{g}_i$ $(0 \le i \le n)$ is the map from the graph to the real sample and $\boldsymbol{L}$ is the Laplacian matrix satisfying $\boldsymbol{L} = \boldsymbol{B} - \boldsymbol{W}$ in which $\mathbf{B}$ is a diagonal weight matrix and $\boldsymbol{B}_{ii} = \sum_j W_{ij}$. Laplacian matrix is a symmetric, positive semi-definite matrix which can be thought of as an operator on functions defined on vertices of the graph. Then we can formulate the minimization problem as

$$\arg\min_{\boldsymbol{g}} \boldsymbol{g}^T \boldsymbol{L}\boldsymbol{g}$$

$$\text{subj. to } \boldsymbol{g}^T \boldsymbol{B}\boldsymbol{g} = 1 \tag{20}$$

which is equivalent to the solution of the following generalized eigenvalue decomposition problem:

$$\boldsymbol{L}\boldsymbol{g} = \lambda \boldsymbol{B}\boldsymbol{g} \tag{21}$$

which is similar to the optimization problem in PCA (or KPCA). We let $\boldsymbol{g}_0, \boldsymbol{g}_1, \ldots, \boldsymbol{g}_{n-1}$ be the solutions of Eq. (21), sorted according to their eigenvalues with $\boldsymbol{g}_0$ having the smallest eigenvalue (actually it is zero). After normalizing the columns of $\{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_c\}$ to have unit l_2-norm, we take them as the pseudo-transformation matrix $\boldsymbol{\Psi}$, namely

$$\boldsymbol{\Psi} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_c\}. \tag{22}$$

The motivation of this dimensionality reduction approach is quite intuitive. We construct $\boldsymbol{\Psi}$ with the graph constraint in order to combine the graph information, or more accurately, the neighborhood information among atoms in the kernel space.

4.2.6. Further discussion

We present four methods to perform the dimensionality reduction in the kernel space. Reducing the dimensionality in the kernel space brings several gains such as lowering the computational cost and enhancing the discrimination power. There also exist a number of other ways to perform the dimensionality reduction in the kernel space, namely construct the matrix $\boldsymbol{\Psi}$. Empirically, if the rank of matrix $\boldsymbol{\Psi}$ stays unchanged, then different construction of $\boldsymbol{\Psi}$ will not lead to dramatical difference in classification accuracy. Thus, the matrix $\boldsymbol{\Psi}$ is not very crucial to the classifier, which is supported by the experiments conducted in [18]. Instead, the rank of the matrix $\boldsymbol{\Psi}$ plays a crucial part in classification accuracy. This is why even using random matrix as $\boldsymbol{\Psi}$ still serves our classifier well. In Section 4, we conduct relevant experiments to study what the selection of the matrix $\boldsymbol{\Psi}$ will do to the classification accuracy.

4.3. Practical KCRC algorithms

There are two algorithms designed for KCRC. For normal situations, both p and q are set as 2. The regularized least square algorithm is adopted to solve the model with $p, q = 2$. Specifically, we derive the new dictionary $\boldsymbol{D}' = \boldsymbol{\Psi}^T \boldsymbol{G}$ and define $\boldsymbol{P}'$ as the coding basis in kernel CRC-RLS (KCRC-RLS). Namely

$$\boldsymbol{P}' = \left((\boldsymbol{\Psi}^T \boldsymbol{G})^T (\boldsymbol{\Psi}^T \boldsymbol{G}) + \mu \cdot \boldsymbol{I}\right)^{-1} \left(\boldsymbol{\Psi}^T \boldsymbol{G}\right)^T \tag{23}$$

where μ is a small constant. The query sample is transformed to $\boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y})$. Apparently, $\boldsymbol{P}'$ is independent of $\boldsymbol{y}'$ so it can be pre-calculated. When a query $\boldsymbol{y}$ comes, the query is first transformed to the kernel space via $\boldsymbol{y}' = \boldsymbol{\Psi}^T \boldsymbol{K}(\boldsymbol{D}, \boldsymbol{y})$ and then can be simply projected onto the coding basis $\boldsymbol{P}'$ via $\boldsymbol{P}'\boldsymbol{y}'$. In the decision making stage, class-specified representation residual $\|\boldsymbol{y}' - \boldsymbol{D}_i' \hat{\boldsymbol{x}}_i\|_2$ is used for classification. Further, a l_2 norm term $\|\hat{\boldsymbol{x}}_i\|_2$ is added for more discriminative classification. The specific algorithm of KCRC-RLS is shown in Algorithm 1. Under normal situations, we use the faster KCRC-RLS as our priority, so the default KCRC-LCD algorithm (in following content and experiments) refers to KCRC-RLS with LCD.

For high level corruption and occlusion, kernel robust CRC (KRCRC) algorithm $(p = 2, q = 1)$ can be applied. Note that, $\boldsymbol{D}'' = \boldsymbol{\Psi}^T \boldsymbol{G}$ and $\boldsymbol{P}_k''$ are designed as the new dictionary and coding

[3] All atoms including $\boldsymbol{d}_{[i]}, \boldsymbol{d}_{[j]}$ should be normalized to have unit l_2-norm before any operation.

basis in kernel space respectively:

$$\boldsymbol{P}''_k = \left((\boldsymbol{\Psi}^T\boldsymbol{G})^T(\boldsymbol{\Psi}^T\boldsymbol{G}) + 2\mu/\sigma_k \cdot \boldsymbol{I}\right)^{-1}\left(\boldsymbol{\Psi}^T\boldsymbol{G}\right)^T \tag{24}$$

where μ and σ_k are small positive constants. The augmented Lagrangian function used for the optimization in Eq. (14) is formulated as

$$L_\sigma(\boldsymbol{e},\boldsymbol{x},\boldsymbol{z}) = \|\boldsymbol{e}\|_1 + \mu\|\boldsymbol{x}\|_2^2 + \langle \boldsymbol{z}, \boldsymbol{y}'' - \boldsymbol{D}''\boldsymbol{x} - \boldsymbol{e}\rangle + \frac{\sigma}{2}\|\boldsymbol{y}'' - \boldsymbol{D}''\boldsymbol{x} - \boldsymbol{e}\|_2^2 \tag{25}$$

where σ is a positive constant that is the penalty for large representation error, and $\boldsymbol{z}$ is a vector of Lagrange multiplier. The ALM method iteratively estimates $\boldsymbol{e},\boldsymbol{x}$ for the Lagrange multiplier $\boldsymbol{z}$ via the following minimization:

$$(\boldsymbol{e}_{k+1},\boldsymbol{x}_{k+1}) = \arg\min_{\boldsymbol{e},\boldsymbol{x}} L_{\sigma_k}(\boldsymbol{e},\boldsymbol{x},\boldsymbol{z}_k) \tag{26}$$

where $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \sigma_k(\boldsymbol{y}'' - \boldsymbol{D}''\boldsymbol{x} - \boldsymbol{e})$. According to [30,42], this iteration will converge to an optimal solution for Eq. (14) if $\{\sigma_k\}$ is a monotonically increasing sequence.

The minimization process in Eq. (26) can be implemented by optimizing $\boldsymbol{e},\boldsymbol{x}$ alternatively and iteratively:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \arg\min_{\boldsymbol{x}} L_{\sigma_k}(\boldsymbol{x},\boldsymbol{e}_k,\boldsymbol{z}_k),\\ \boldsymbol{e}_{k+1} &= \arg\min_{\boldsymbol{e}} L_{\sigma_k}(\boldsymbol{x}_{k+1},\boldsymbol{e},\boldsymbol{z}_k), \end{aligned} \tag{27}$$

which has the closed-form solution as follows:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= (\boldsymbol{D}''^T\boldsymbol{D}'' + 2\mu/\sigma_k\boldsymbol{I})^{-1}\boldsymbol{D}''^T(\boldsymbol{y}'' - \boldsymbol{e}_k + \boldsymbol{z}_k/\sigma_k)\\ &= \boldsymbol{P}''_k(\boldsymbol{y}'' - \boldsymbol{e}_k + \boldsymbol{z}_k/\sigma_k),\\ \boldsymbol{e}_{k+1} &= S_{1/\sigma_k}(\boldsymbol{y}'' - \boldsymbol{D}''\boldsymbol{x}_{k+1} + \boldsymbol{z}_k/\sigma_k), \end{aligned} \tag{28}$$

where the function $S_\alpha, \alpha \ge 0$ is the soft-thresholding (shrinkage) operator given by

$$S_\alpha(h) = \begin{cases} h-\alpha & \text{if } h \ge \alpha\\ h+\alpha & \text{if } h \le -\alpha\\ 0 & \text{otherwise.} \end{cases} \tag{29}$$

If $\boldsymbol{h}$ represents a n-dimensional vector, then $S_\alpha(\boldsymbol{h})$ is given by $\{S_\alpha(\boldsymbol{h}_1), S_\alpha(\boldsymbol{h}_2), \ldots, S_\alpha(\boldsymbol{h}_n)\}$. Similar to the KCRC-RLS, the coding basis $\boldsymbol{P}''_k$ is independent of $\boldsymbol{y}''$ for the given σ_k, so the set of projection matrices $\{\boldsymbol{P}_k\}$ can also be pre-calculated. Once a query sample $\boldsymbol{y}$ comes, it is first transformed in the kernel space via $\boldsymbol{\Psi}^T\boldsymbol{K}(\boldsymbol{D},\boldsymbol{y})$ and then projected onto $\boldsymbol{P}''_k$ via $\boldsymbol{P}''_k\boldsymbol{y}''$. After performing the iterative minimization above, a classification strategy similar to KCRC-RLS is applied in KRCRC. The robustness test of KRCRC is provided in our previous work [19]. Details of KRCRC are given in Algorithm 2.

Algorithm 1. KCRC-RLS.

```
1.  Normalize the columns of $\boldsymbol{D}' = \boldsymbol{\Psi}^T\boldsymbol{G}$ to unit $l_2$-norm.
2.  Represent $\boldsymbol{y}' = \boldsymbol{\Psi}^T\boldsymbol{K}(\boldsymbol{D},\boldsymbol{y})$ over dictionary $\boldsymbol{D}'$ by
        $\hat{\boldsymbol{x}} = \boldsymbol{P}'\boldsymbol{y}'$
    where $\boldsymbol{P}' = (\boldsymbol{D}'^T\boldsymbol{D}' + \mu\boldsymbol{I})^{-1}\boldsymbol{D}'^T$.
3.  Obtain the regularized residuals
        $r_i = \|\boldsymbol{y}' - \boldsymbol{D}'_i\hat{\boldsymbol{x}}_i\|_2 / \|\hat{\boldsymbol{x}}_i\|_2$
    where $\hat{\boldsymbol{x}}_i$ is the coding coefficients associated with class
    $i$ over $\boldsymbol{P}'$.
4.  Output the identity of $\boldsymbol{y}'$ (class label) as
        $id(\boldsymbol{y}') = \arg\min_i (r_i)$.
```

Algorithm 2. KRCRC.

```
1. Normalize the columns of $\boldsymbol{D}'' = \boldsymbol{\Psi}^T\boldsymbol{G}$ to unit $l_2$-norm.
2. Input $\boldsymbol{y}'' = \boldsymbol{\Psi}^T\boldsymbol{K}(\boldsymbol{D},\boldsymbol{y})$, $\boldsymbol{x}_0$, $\boldsymbol{e}_0$, $k=1$ and $\tau > 0$.
3. Proceed if $|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k|_2 > \tau$ is true. If not, output $\hat{\boldsymbol{e}}, \hat{\boldsymbol{x}}$ and go to
   step 5.
4. Do the following iteration:
       $\boldsymbol{x}_{k+1} = \boldsymbol{P}''_k(\boldsymbol{y}'' - \boldsymbol{e}_k + \boldsymbol{z}_k/\sigma_k)$
       $\boldsymbol{e}_{k+1} = S_{1/\sigma_k}(\boldsymbol{y}'' - \boldsymbol{D}''\boldsymbol{x}_{k+1} + \boldsymbol{z}_k/\sigma_k)$
       $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \sigma_k(\boldsymbol{y}'' - \boldsymbol{D}''\boldsymbol{x}_{k+1} - \boldsymbol{e}_{k+1})$
   where $\boldsymbol{P}''_k = (\boldsymbol{D}''^T\boldsymbol{D}'' + 2\mu/\sigma_k\boldsymbol{I})^{-1}\boldsymbol{D}''^T$ and $S_\alpha, \alpha \ge 0$
   is the shrinkage coefficient. $k \leftarrow k+1$ and go to step 3.
5. Represent $\boldsymbol{y}''$ over dictionary $\boldsymbol{D}''$ by the converged $\boldsymbol{x}$.
6. Obtain the regularized residuals
       $r_i = \|\boldsymbol{y}'' - \boldsymbol{D}''_i\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{e}}\|_2 / \|\hat{\boldsymbol{x}}_i\|_2$
   where $\hat{\boldsymbol{x}}_i$ is the coding coefficients related to class $i$.
7. Output the identity of $\boldsymbol{y}''$ (class label) as
       $id(\boldsymbol{y}'') = \arg\min_i(r_i)$.
```

5. On the locality constrained dictionary

This section elaborates the locality constrained dictionary. Additionally, we present some typical distances used in LCD for similarity measurement and further introduce a distance fusion model, followed by the introduction of the KCRC method combined with LCD, termed as KCRC-LCD.

5.1. Locality constrained dictionary

Most collaborative representation based methods [30,45,46] employ all the high-dimensional training samples as the global dictionary. They may work fine when the global dictionary is small, but the classification becomes intractable with increasingly more training samples. To address this problem, we propose the LCD that utilizes the K-NN classifier to measure the similarities between the query sample and all atoms in the global dictionary, and then selects K nearest atoms as the local dictionary. The locality in LCD ensures discrimination, efficiency and robustness of KCRC. Compared to the locality constrained dictionary proposed in [47], we adopt a more straightforward way to constrain the locality, which needs no learning and training process, greatly reducing the computational cost in training. Under such locality constrained dictionary, scaling to a large number of categories does not require adding new features, because the discriminative distance function needs to only be defined for similar enough samples. From biological and psychological perspective, similarity between samples is the most important criteria to recognize and classify objects for human brains. So intuitively speaking, the proposed locality constrained dictionary with various optional discriminative distances makes our KCRC approach more scalable, discriminative, efficient, and most importantly, free from the curse of high-dimensional feature space. Moreover, the kernel idea within KCRC well suits the idea of locality both mathematically and experimentally.

We define $Dist(\boldsymbol{v}', \boldsymbol{v}'')$ as the distance metric between any two vectors. For example, we can adopt the l_2 distance: $Dist(\boldsymbol{v}', \boldsymbol{v}'') = \|\boldsymbol{v}' - \boldsymbol{v}''\|_2$. We need to calculate the distance between every atom $\boldsymbol{d}_{[k]}$ and the query sample $\boldsymbol{y}$ first. Given K, then the LCD can be obtained via the following optimization:

$$\begin{aligned} &\arg\min_{\{t_1,t_2,\ldots,t_K\}} \sum_{m=1}^{K} Dist(\boldsymbol{d}_{[t_m]}, \boldsymbol{y})\\ &\text{subj. to}\quad 1 \le t_i \ne t_j \le n,\ for\ \forall i \ne j \end{aligned} \tag{30}$$

where $\boldsymbol{d}_{[t_1]}, \boldsymbol{d}_{[t_2]}, \ldots, \boldsymbol{d}_{[t_K]}$ denote different atoms in the global dictionary. In fact, to solve Eq. (30) is to find the K atoms that are located nearest to the query sample. As a result, the LCD is obtained as $\boldsymbol{D}_{lc} = \{\boldsymbol{d}_{[t_1]}, \boldsymbol{d}_{[t_2]}, \ldots, \boldsymbol{d}_{[t_K]}\}$. Moreover, the computational complexity of solving Eq. (30) is $O(n \log n)$, which is efficient enough to perform in large-scale image datasets. Note that, when $K = n$, KCRC-GD becomes a special case of KCRC-LCD.

One of the most significant features of KCRC-LCD is the close connection between KCRC and LCD. When we compute the LCD, we need to compute the distance (similarity) measure between the test sample and all training samples first. Therefore, $\boldsymbol{K}(\boldsymbol{D}_{lc}, \boldsymbol{y})$ which is needed in KCRC, can be directly obtained using the information from the LCD construction. Since the kernel Gram matrix of LCD can also be obtained directly from the kernel Gram matrix of the global dictionary which can be computed in advance. LCD makes KCRC-LCD scalable to training set size, while KCRC operates on the kernel matrix without reference to the underlying feature space, bypassing the feature space operation and reducing the computational cost. Therefore, the entire KCRC-LCD performs very fast and is able to handle high dimensional features and extremely large datasets without complex training process.

To summarize, KCRC-LCD include three appealing aspects. First, the well motivated coarse-to-fine classification framework, inspired from human perception, plays a core role in KCRC-LCD. Second, the carefully designed similarity measure, used in LCD, can be transformed in a straightforward way to the kernel for KCRC via numerous kernel functions. Third, KCRC-LCD is scalable in terms of the training set size and feature dimension.

5.2. Discriminative distances for similarity measure

In the previous subsection, we simply use the l_2 distance as an example. In fact, there are many discriminative distances for similarity measurement. Several well-performing distances are introduced in [40], i.e., Mahalanobis distance, χ^2 distance [48], marginal distance [49], tangent distance [50], shape context based distance [51] and geometric blur based distance [52]. Each can be used to measure the similarity in order to construct a well-performing LCD. These distances can either be used alone or used in conjunction with each other, making the LCD flexible and adaptive. In general, the similarity measure can be understood as the combination of feature extraction and sample distance. In other words, features and distances construct the similarity measure. We briefly review some discriminative distances.

5.2.1. General pixel similarity measure

We consider several classical pixel distance metrics below. Euclidean distance (l_2 distance) is the most popular similarity measure. It is simple yet effective in certain situations and is defined as

$$Dist(\boldsymbol{v}', \boldsymbol{v}'') = \|\boldsymbol{v}' - \boldsymbol{v}''\|_2. \tag{31}$$

City block distance, also known as Manhattan distance, assumes that it is only possible to travel along pixel grid lines from one pixel to another. This distance metric is defined as

$$Dist(\boldsymbol{v}', \boldsymbol{v}'') = \|\boldsymbol{v}' - \boldsymbol{v}''\|_1. \tag{32}$$

Chessboard distance metric assumes that you can make moves on the pixel grid as if you were a King making moves in chess, i.e., a diagonal move counts the same as a horizontal move. The metric is given by

$$Dist(\boldsymbol{v}', \boldsymbol{v}'') = \|\boldsymbol{v}' - \boldsymbol{v}''\|_\infty. \tag{33}$$

There are a lot of other general pixel distances that can be utilized in our framework, such as correlation distance and Mahalanobis distance.

5.2.2. Texture similarity measure

We present some texture similarity measure as examples below. The χ^2 distance is proposed in [48] for texture similarity measure. The main idea of the χ^2 distance is to construct a vocabulary of 3D textons by clustering a set of samples. Associated with each texton is an appearance vector which characterizes the local irradiance distribution. The similarity can be measured by characterizing samples with these 3D textons. In the view of statistics, marginal distance [49] is another version of the χ^2 distance. They both measure the difference between two joint distribution of texture response. The difference is that marginal distance metric simply sums up the distance between response histograms from each filter while the χ^2 distance metric measures the similarity of the two joint distribution by comparing the histogram of textons. As another texture similarity measure initially used for hand-written digits recognition, Tangent distance [50] is defined to compute the minimum distance between the linear surfaces that best approximate the non-linear manifolds of different sample categories. These linear surfaces, which are crucial to Tangent distance, are derived from the images by including perturbations from small affine transformation of the spatial domain and change in the thickness of pen-stroke.

5.2.3. Shape similarity measure

The shape in an image can be represented by a set of points, with a descriptor at a fixed point to measure the relative position to that point. These descriptors are iteratively matched using a deformation model. Shape context based distance [51] is derived from the discrepancy left in the final matched shapes and a score that denotes how far the deformation is from an affine transformation. Various shape descriptors can be defined on a gray scale image, for example, the shape context descriptor on the edge map [53], the SIFT descriptor [54], the geometric blur descriptor [52], and optimized local shape descriptor [55].

5.2.4. Unified similarity measure

We use a simple and intuitive method to combine general pixel similarity, texture similarity and shape similarity into a unified locality constrained dictionary. Assume that we need to construct a LCD with size K out of total N training samples. First, we enforce locality to dictionary via general pixel similarity, texture similarity and shape similarity, obtaining three LCDs with size $K : \boldsymbol{D}_{lc}^{[pixel]}$, $\boldsymbol{D}_{lc}^{[texture]}$ and $\boldsymbol{D}_{lc}^{[shape]}$. From the sets perspective, these three dictionaries constructed via different similarity measures can be viewed in Venn diagram as shown in Fig. 1. Specifically, an atom in dictionary represents an element in a set, so a LCD can be regarded as a set with K elements. According to the demand of the given task, we can use different combinations of similarity measures to construct the LCD. Mathematically, we achieve the combination of similarity measures by getting the union of the corresponding LCDs, i.e., $\boldsymbol{D}_{lc} = \boldsymbol{D}_{lc}^{[texture]} \cup \boldsymbol{D}_{lc}^{[shape]}$ for the combination of texture and shape similarity measure. With unified similarity measure, the distance metric used in the kernel function can be a linear combination of the distances that are utilized to construct the new LCD. Normally, we suppose the weight of each distance metric in the unified distance is equal and the distance metrics are suggested to be normalized. To avoid normalization, we can either multiply all distance metrics to obtain the unified distance metric, or alternatively just use single distance metric since the unified LCD already contains information of the other distance metrics. Applying a more delicately designed metric fusion approach, such as [56,57], can greatly improve the performance of KCRC-LCD, but it is out of the scope of this paper. In this paper, we simply multiply the distance metrics to unify the distance metrics. However, to use unified similarity measure could add to computational cost, so we do not recommend to use it under normal circumstance. Note that, the same type similarity measures can also be unified by the proposed method with similar procedure.

5.3. KCRC-LCD algorithm

Intuitively, we use the locality constrained dictionary $\boldsymbol{D}_{lc}$ in place of global dictionary $\boldsymbol{D}$ and then perform the KCRC algorithm on $\boldsymbol{D}_{lc}$. Following such idea, the naive version of the KCRC-LCD algorithm is as follows:

Algorithm 3. Naive KCRC-LCD.

```
1. Compute the distances between the query sample and all training samples, and pick the nearest $K$ neighbors.
2. If the $K$ neighbors have the same labels, the query is labeled and exit; Else, construct the LCD $\boldsymbol{D}_{lc}$ with the $K$ labeled neighbors and goto Step 3.
3. Convert the pairwise distance into kernel matrix via kernel trick and utilize KCRC approach with dictionary $\boldsymbol{D}_{lc}$ instead of the global dictionary $\boldsymbol{D}$.
4. Output the label of the query sample.
```

The naive version of KCRC-LCD performs slowly because it has to compute the distances of the query to all training samples. Inspired by [40], we consider to boost the efficiency in the coarse-to-fine framework which is similar to the human perception procedure that human first perform a fast coarse pruning and then recognize the object by details. The practical version of the KCRC-LCD algorithm is as follows:

Algorithm 4. Practical KCRC-LCD.

```
1. Compute the coarse distances (i.e., Euclidean distance) between the query sample and all training samples, and pick the nearest $K_c$ neighbors. ($K_c \geq K_f$)
2. Compute the fine distances between the query sample and pick the nearest $K_f$ neighbors.
3. If the $K_f$ neighbors have the same labels, the query is labeled and exit; Else, construct the LCD $\boldsymbol{D}_{lc}$ with the $K_f$ labeled neighbors and goto Step 3.
4. Convert the pairwise distance into kernel matrix via kernel trick and utilize KCRC approach with dictionary $\boldsymbol{D}_{lc}$ instead of the global dictionary $\boldsymbol{D}$.
5. Output the label of the query sample.
```

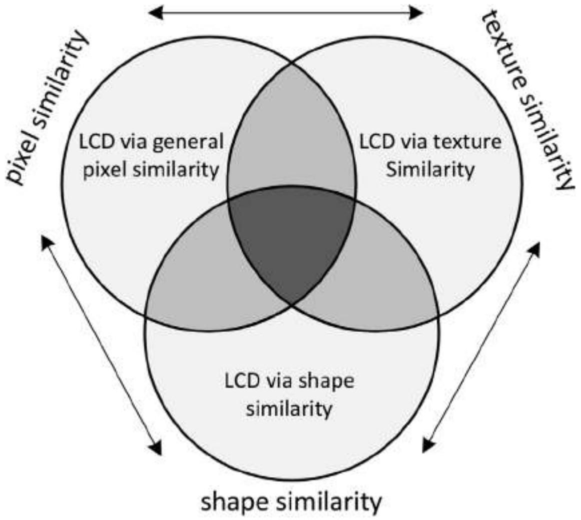


Fig. 1. Venn diagram of LCDs constructed via general pixel similarity, texture similarity and shape similarity.

6. Experiments and results

6.1. Implementation details

In this subsection, we provide implementation details for KCRC-LCD. We use five-fold cross validation to obtain parameters for our model. To start with, we usually follow the parameter settings in other existing papers [18,30,40,58] and gradually adjust them until our model has best average performance in cross validation. Once the parameters are set, we fix all parameters of KCRC-LCD in each dataset for fair comparison. In all experiments, we use $\mu = 0.001 \times n/700$ (n is the size of the dictionary) for KCRC model,[4] and set the default kernel function in KCRC-LCD as $K(\boldsymbol{v}', \boldsymbol{v}'') = \exp(-\beta Dist(\boldsymbol{v}', \boldsymbol{v}''))$ with $\beta = 0.5$. Normally, we use the Euclidean distance as default distance metric except for the distance evaluation part. Moreover, we use the identity matrix for the dimensionality reduction of KCRC, namely $\boldsymbol{\Psi} = \boldsymbol{I}$, except in the evaluation part of dimensionality reduction in kernel space. In fact, we optimize the parameters in KCRC following [30]. For distance metric evaluation part, we mainly follow the parameter selection in [40] and slightly adjust these parameters to obtain the best performance. All experiments are implemented on Matlab 2013b and run on a PC with 3.4 GHz Intel dual-core CPU and 32G RAM.

For MNIST and USPS datasets, we use the raw pixel (downsample to 28 × 28, totally 784 dimensions) as features for classification. For the Extended Yale Face Database B (Extended Yale B), we use the 504-dimension randomface features, following the same experimental settings as [3]. For Caltech 101 and 15 scene category datasets, we use the spatial pyramid feature [59] and apply PCA to reduce its dimension to 3000, using the same as in [58] for better comparison. For Caltech 256 dataset, we first compute the HOG descriptors from each patch at three scales, 16 × 16, 25 × 25, and 32 × 31. Then we compute the spatial pyramid features via 1 × 1, 2 × 2 and 4 × 4 subregions. The feature dimensions are reduced to 305 by PCA, same as [58]. Note that, LC-KSVD we use in experiments is LC-KSVD2 [58] which jointly learns an optimal linear classifier in optimization. More specific experimental settings and parameter selection are elaborated in the relevant sections.

6.2. Evaluation of dimensionality reduction in kernel space

We evaluate the dimensionality reduction from two aspects. First we compare the representation coefficients and reconstruction residuals that are obtained by SRC, CRC, KCRC with no dimensionality reduction (KCRC-Identity), KCRC with KPCA dimensionality reduction (KCRC-KPCA), KCRC with random projection (KCRC-RP) and KCRC with graph projection (KCRC-Graph). Second, we compare the recognition accuracy of these methods in Extended Yale B. Note that, for KCRC-KPCA, KCRC-Graph and KCRC-RP, we reduce the dimension to 80. For KCRC-Graph, we use the n-neighbors to construct the graph and n is set as 35 in Extended Yale B.

We randomly select 38 images per person (32 person, totally 1216 images) in Extended Yale B [60] as training samples. For the computation of representation coefficients and reconstruction

[4] For KCRC, we mainly follow the parameters selection in the CRC model [30].

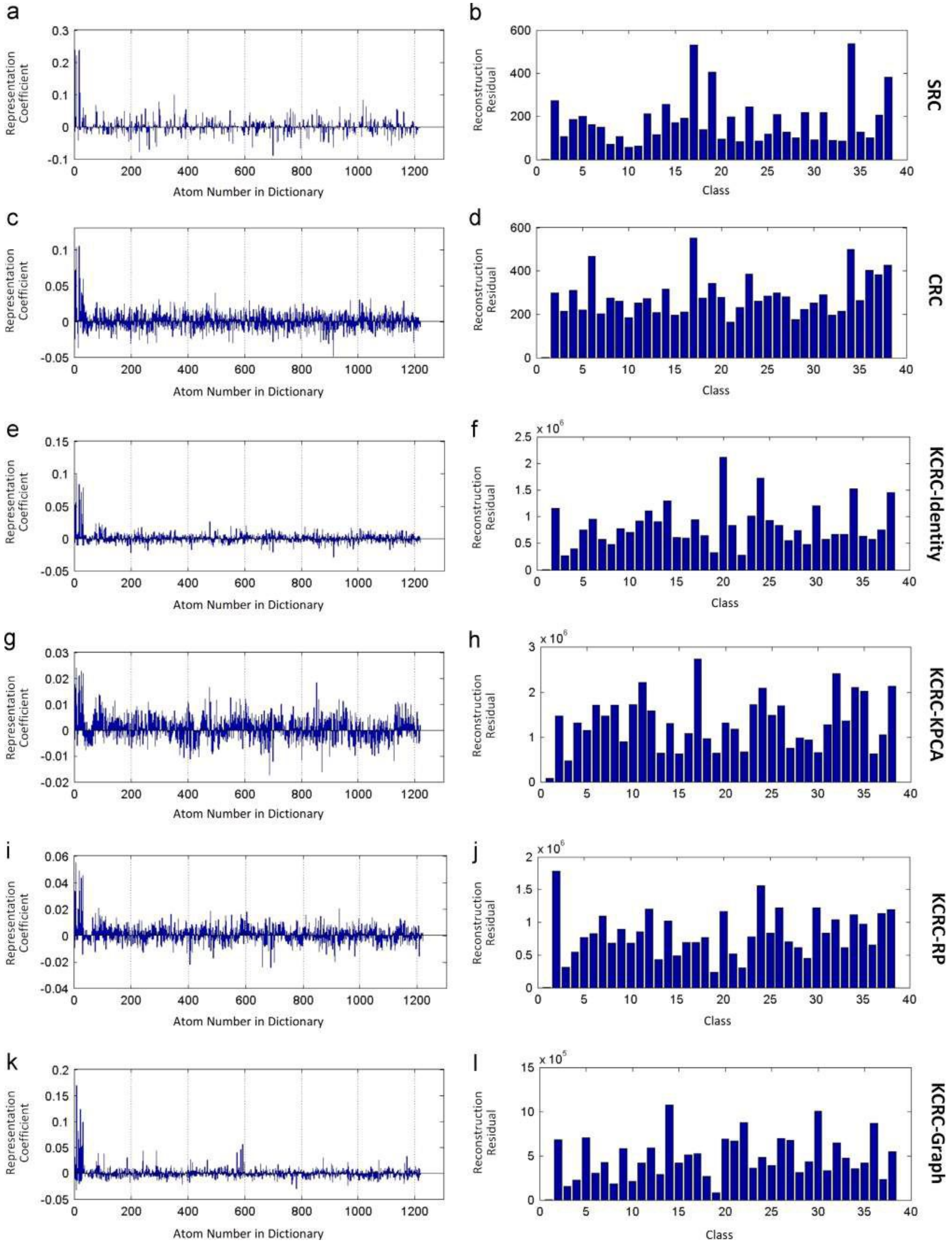


Fig. 2. (a) Representation coefficients obtained by SRC. (b) Reconstruction residuals obtained by SRC. (c) Representation coefficients obtained by CRC. (d) Reconstruction residuals obtained by CRC. (e) Representation coefficients obtained by KCRC-Identity. (f) Reconstruction residuals obtained by KCRC-Identity. (g) Representation coefficients obtained by KCRC-KPCA. (h) Reconstruction residuals obtained by KCRC-KPCA. (i) Representation coefficients obtained by KCRC-RP. (j) Reconstruction residuals obtained by KCRC-RP. (k) Representation coefficients obtained by KCRC-Graph. (l) Reconstruction residuals obtained by KCRC-Graph.

Table 1
Recognition results on Extended Yale Face Dataset B. 504 random projection features and the global dictionary (1216 atoms) are adopted. The results below are averaged over 10 times experiments.

Method	Accuracy (%)
SRC	97.15
CRC	97.67
KCRC-Identity	97.23
KCRC-KPCA	97.07
KCRC-RP	96.61
KCRC-Graph	97.35

residuals, we use a single fixed test sample for better comparison. To simplify the experiment, we only use the global dictionary since the experiment focuses on the dimensionality reduction methods in kernel space. In the recognition accuracy test, we follow the same experiment settings as [3] by randomly selecting 38 images per person (32 person, totally 1216 images) and using the remaining images as test samples. The results are averaged over 20 times experiments.

Fig. 2 gives the representation coefficients and reconstruction residuals of SRC, CRC, KCRC-Identity, KCRC-KPCA, KCRC-RP and KCRC-Graph. We can see that all of these five approaches tell the correct label (the first class) of the test sample. It can be obtained from Fig. 2 that KCRC-Identity and KCRC-Graph achieve better sparsity, similar to the representation coefficients of SRC. Moreover, the reconstruction residuals of all these approaches indicate that the first class has the fewest reconstruction residual. Table 1 shows the recognition accuracy of SRC, CRC, KCRC-Identity, KCRC-KPCA, KCRC-RP and KCRC-Graph on Extended Yale B. We can see that CRC has the best recognition accuracy of these five methods. However, all these approaches are of the same level discrimination ability since the difference between the highest recognition rate and the lowest is only about 1%.

6.3. Experiments on data with the same direction distribution

We evaluate the performance on data with the same direction distribution. In Fig. 3, we compare 3 classifiers: CRC-GD, KCRC-GD and KCRC-LCD. Two-class training data $\boldsymbol{Q}$ and $\boldsymbol{W}$ with m-dimension are generated for the experiment. Each feature of $\boldsymbol{Q}$ and $\boldsymbol{W}$ uniformly takes value from the interval $[1,3]$ and $[-3,-1]$ respectively, corrupted by Gaussian noise with zero mean and 0.15 variance. We test on all the coordinates (f_1, f_2) in the space $\{f_1 \in [-4, 4], f_2 \in [-4, 4]\}$ with step 0.1. The gray point indicates its predicted label is the same as $\mathbf{W}$ while the white point indicates its predicted label is the same as $\mathbf{Q}$. Then, we let m vary from 2 to 256 and perform the experiment. The results show that both KCRC-GD and KCRC-LCD can perfectly classify data with the same direction distribution while CRC performs poorly. The main reason why the original CRC fails so dramatically is that the data in the same direction would overlap each other after the normalization process. This experiment shows that both KCRC-GD and KCRC-LCD could handle data with special distribution, i.e., the same direction distribution in this case. Thus, kernel function makes our proposed approach more prepared for unknown data distribution than conventional CRC.

6.4. Experiments on public datasets

The subsection evaluates KCRC-LCD on several public datasets. We use the RBF-like kernel function $K(\boldsymbol{v}', \boldsymbol{v}'') = \exp(-\beta Dist(\boldsymbol{v}', \boldsymbol{v}''))$ with $\beta = 0.5$ and the default distance metric, namely Euclidean distance. All the parameters in KCRC model are fixed in all experiments. Since we simply use the Euclidean distance, the practical KCRC-LCD is the same version as the naive KCRC-LCD. Reliable results are obtained by 20 times repeated experiments with different random splits of the training and test images.

6.4.1. MNIST

The MNIST dataset [61] of handwritten digits contains 60,000 samples (10 digits) for training and 10,000 for testing. For the experimental settings, Euclidean distance and 28 × 28 raw pixel features are used for similarity measure and classification. We evaluate our approach via different dictionary sizes 100, 200, 500, 700, 1000 and 1500, namely 10, 20, 50, 70, 100 and 150 samples for training per category. For settings of KCRC-LCD and CRC-LCD, we use the global dictionary of size 2000 (200 training samples per category) to generate the LCD and set K for LCD as 100, 200, 500, 700, 1000 and 1500 for comparison. Experimental results in Fig. 4(a) show that KCRC-LCD has the best performance compared to CRC-LCD, CRC-GD and KCRC-GD in the MNIST dataset. From Fig. 4(b), it can be learned that K has slight impact on classification accuracy if the global dictionary is fixed and K is large enough (atom number of GD stays unchanged).

6.4.2. Extended Yale Face Dataset B

The Extended Yale Face Dataset B contains 2414 frontal face images of 38 individuals [60]. The cropped 192 × 168 face images are taken under various lighting conditions [60]. For each person, we randomly select 32 images for training and the remaining for testing. Therefore, there will be 1216 training images and 1198 testing images. For the experimental settings, Euclidean distance and 504-dimension random projection features [3,62] are used for similarity measure and classification. We evaluate our approach via different dictionary sizes 380, 570, 760, 950 and 1216, namely 10, 15, 20, 25 and 32 samples for training per category. For settings of KCRC-LCD, we use the global dictionary of size 1216 (32 training samples per person) to generate the LCD and set K for LCD as 380, 570, 760, 950 and 1216 for comparison. For LC-KSVD and D-KSVD, we use 32 training samples per category to learn the dictionary of sizes 380, 570, 760, 950 and 1216. For MTJSTC [5], we use 10, 15, 20, 25 and 32 samples per category to construct the dictionary of sizes 380, 570, 760, 950 and 1216, and utilize raw pixel features and local binary patterns features, same as settings in [5]. The other approaches use random projection features. Experimental results are given in Fig. 5. In Fig. 5(a), KCRC-LCD has better classification accuracy than the other approaches when dictionary size is small. It is mostly because the global dictionary we use to generate LCD is more informative than the small size dictionary, and LCD itself is designed to be adaptive to use the important information for classification. We can also learn that the classification accuracy of KCRC-LCD no longer stands out when dictionary size becomes 1216. When dictionary size comes to 1216, K for LCD is also equal to 1216, making KCRC-LCD degenerate to KCRC-GD. That is to say, we do not have enough training samples to construct a discriminative LCD. Moreover, when K becomes larger, the locality of LCD becomes weaker as well, leading to less discrimination power. It is obtained from Fig. 5(b) that LCD already has enough critical information to proceed the classification when K reaches 380. Therefore, we can conclude that the performance ceiling for LCD has been reached when $K=380$ in this case. Compared to the global dictionary with 1214 training samples, LCD with only 380 atoms can achieve similar or even better classification accuracy.

6.4.3. Caltech 101

The Caltech 101 dataset [64] contains 9144 images from 102 classes (101 objects and a background class). We train on 5, 10, 15, 20, 25, and 30 samples per category (dictionary sizes are 510, 1020, 1530, 2040, 2550, and 3060 respectively) and test on the rest. Note

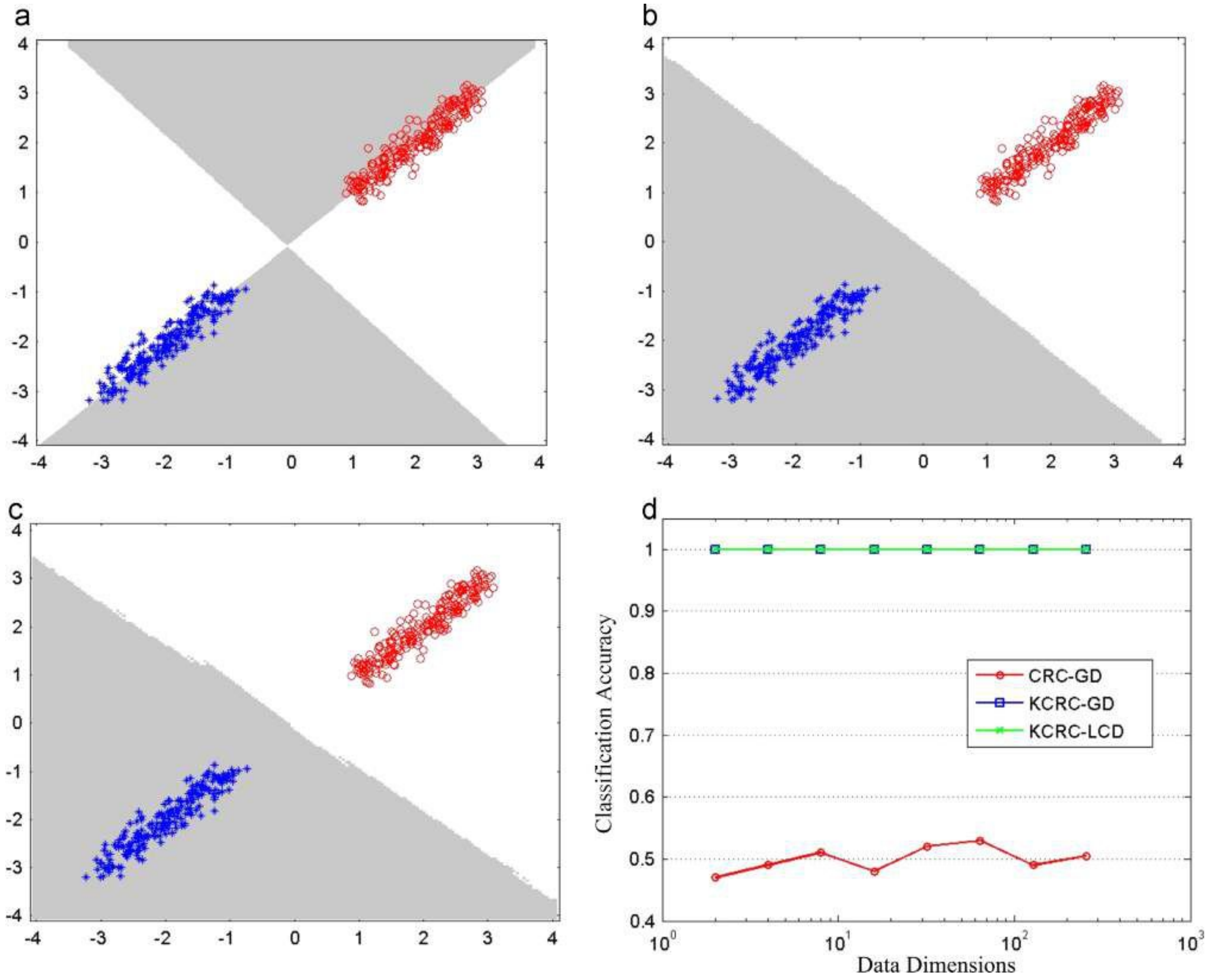


Fig. 3. Performance comparison on data with the same direction distribution. Test samples are from the entire surface whose predicted labels are indicated by gray or white. 2-D decision boundaries obtained by (a) CRC with global dictionary (CRC-GD), (b) KCRC with global dictionary (KCRC-GD), (c) KCRC with locality constrained dictionary (KCRC-LCD), and (d) classification accuracy vs. dimensionality.

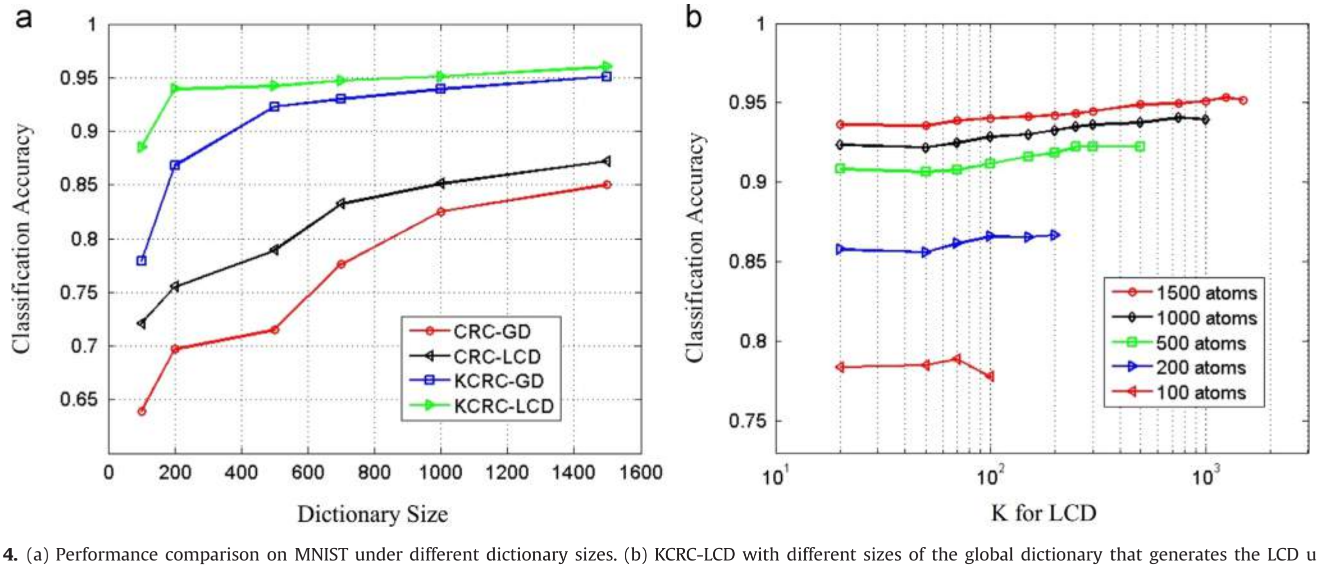


Fig. 4. (a) Performance comparison on MNIST under different dictionary sizes. (b) KCRC-LCD with different sizes of the global dictionary that generates the LCD under different K settings. Note that, Euclidean distance and 28×28 raw pixel features are used for similarity measure and classification.

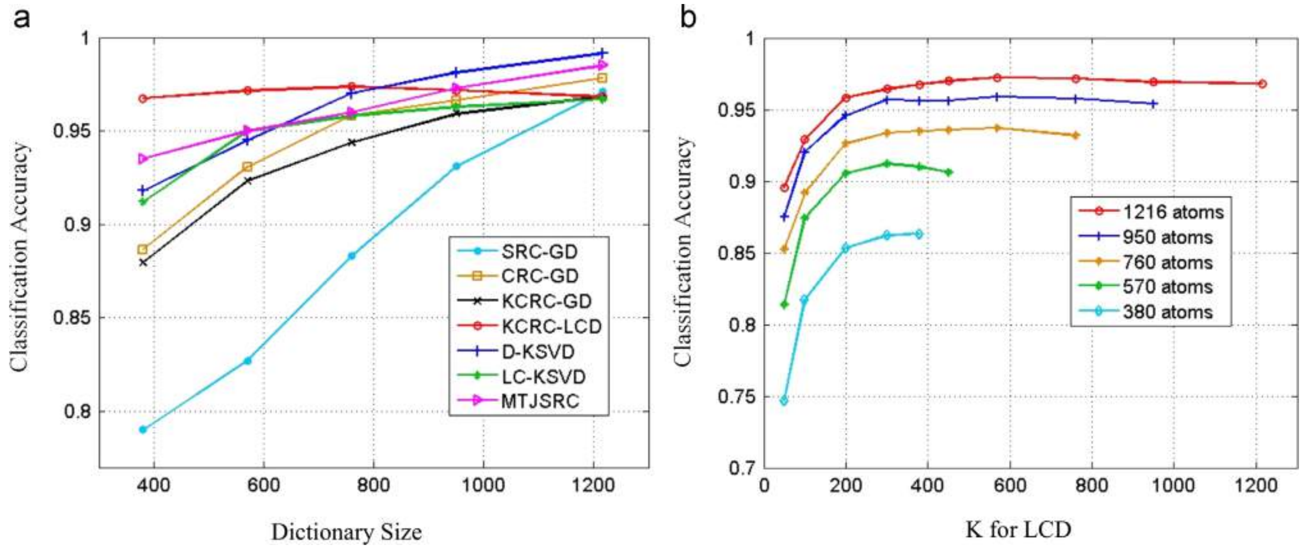


Fig. 5. (a) Performance comparison with SRC-GD, CRC-GD, KCRC-GD, KCRC-LCD, LC-KSVD [58,62], discriminative K-SVD (D-KSVD) [63] and MTJSRC [5] on Extended Yale B under different dictionary sizes. (b) KCRC-LCD with different sizes of the global dictionary that generates the LCD under different K settings. Note that, Euclidean distance and the random projection features are used for similarity measure and classification.

that, we use the global dictionary of size 3060 (30 training samples per category) to generate the LCD and set K for LCD as 510, 1020, 1530, 2040, 2550 and 3060 for comparison. For fairness, we use two setups for LC-KSVD and D-KSVD. First, we use 5, 10, 15, 20, 25, and 30 training samples per category to learn the dictionary of sizes 510, 1020, 1530, 2040, 2550 and 3060 respectively. Second, we only use 30 training samples per category to learn the dictionary of size 510, 1020, 1530, 2040, 2550 and 3060, and term such setup for LC-KSVD and D-KSVD as LC-KSVD* and D-KSVD* respectively. Euclidean distance and the spatial pyramid features are used for similarity measure and classification. For kernel KSVD [27], we use 5, 10, 15, 20, 25, and 30 samples per category to construct a dictionary of sizes 1020, 1530, 2040, 2550, 3060 with collective approach [27]. For KMTJSRC [5], we only use 15 samples per category and test on the rest, and adopt the same multiple features in [5]. Results in Fig. 6(a) show that KCRC-LCD outperforms many other competitive approaches in Caltech 101 dataset, especially when dictionary size is small. It is worth noting that kernel KSVD [27,28] is a very competitive method with the highest performance when the dictionary number becomes large. We believe that one major reason of the performance gain lies in the benefit from learning dictionaries. While this helps us to boost the classification performance, the computational complexity also increases significantly. On the other hand, our scheme tends to emphasize more on better scalability while maintaining a good performance, especially with less dictionaries.

Fig. 6(a) shows that the classification accuracy of CRC-LCD is improved little when dictionary size is high. The reason is similar to the previous experiment on Extended Yale B: as K approaches 3060, the dataset lacks extra training samples, or extra discriminative information, to construct LCD and KCRC-LCD degenerates to KCRC-GD. Fig. 6(b) shows that when LCD has obtained the most discriminative and crucial atoms in the dictionary. Keep increasing K will not help the classification much. In fact, if we conduct the experiment in Fig. 6 (b) with smaller K values, it will end up similar to the curves in Fig. 5 (b) where the classification accuracy goes up quickly before saturation.

6.4.4. Caltech 256

The Caltech 256 dataset [65] contains 30,607 images of 256 categories, each category with more than 80 images. It is a very difficult visual categorization dataset due to the large variations in object background, pose and size. For fair comparison, We randomly select 30 training samples per category to learn the dictionary for LC-KSVD and D-KSVD (the dictionary sizes are set as 1280, 3840 and 7680). For KCRC-LCD, we use the global dictionary of size 7680 (30 training samples per category, same settings as LC-KSVD and D-KSVD) to generate the LCD and set K for LCD as 1280, 3840 and 7680 for comparison. Euclidean distance is used for similarity measure. Note that, we use the same features as in [58]. The detailed feature extraction is introduced in Section 6.1. Results in Fig. 7 show that when dictionary size is 7680, KCRC-GD, KCRC-LCD and LC-KSVD have similar classification accuracy. It is observed that KCRC-LCD performs much better than the other competitive approaches, especially with small dictionary size.

6.4.5. 15 Scene categories

This dataset contains 15 natural scene categories such as office, kitchen and bedroom, introduced in [59]. Following the same experimental settings as [58], we randomly select 30 images per category for training and the rest for testing. Note that, we generate the LCD with $K=450$ from a global dictionary of size 1500, similar to the training settings of LC-KSVD. For approaches using global dictionary, i.e., KSRC-GD, CRC-GD and KCRC-GD, we use a global dictionary of size 450. Testing samples that we use here are the same for all compared approaches. Results in Table 2 and Fig. 8 validate the superiority of KCRC-LCD in scenes. From Fig. 8, we observe that KCRC-LCD can well classify the challenging scene category in which CRC-GD has poor accuracy. To summarize, we can see that KCRC-LCD outperforms most state-of-the-art approaches in scenes.

6.5. Evaluation of running time and scalability

We conduct experiments on running time to evaluate the computational cost of KCRC-LCD. We use public datasets including MNIST, Extended Yale B, Caltech 101 and 15 scene categories to perform our experiments. The detailed experimental settings are given in Table 3. Note that, Euclidean distance is used for similarity measure. For SRC, we use the basis pursuit (BP) algorithm to solve the l_1 minimization problem. Experimental results are shown in Table 4. Compared to SRC approach, KCRC performs much faster due to its l_2 regularization. Constrained with locality, KCRC-LCD performs faster than KCRC-GD and CRC-GD under most circumstances.

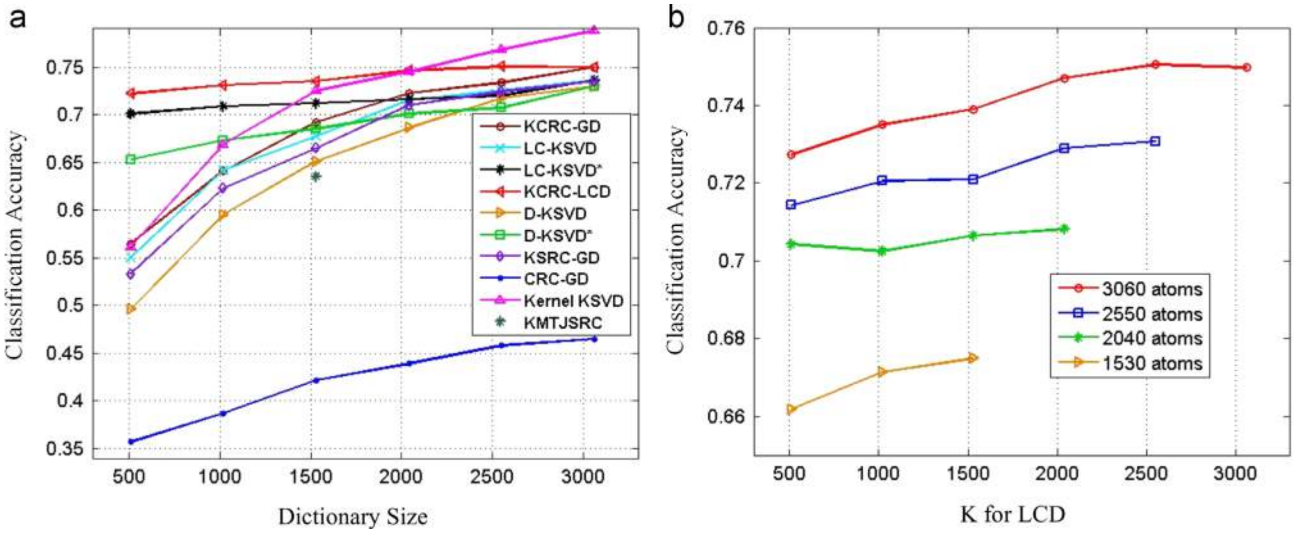


Fig. 6. (a) Performance comparison with KCRC-GD, KCRC-LCD, CRC-GD, LC-KSVD [58,62], D-KSVD [63], KSRC-GD [18], kernel KSVD [27,28] and KMTJSRC [5] on Caltech 101 under different dictionary sizes. (b) KCRC-LCD with different sizes of the global dictionary that generates the LCD under different *K* settings. Note that, Euclidean distance and the spatial pyramid features are used for similarity measure and classification.

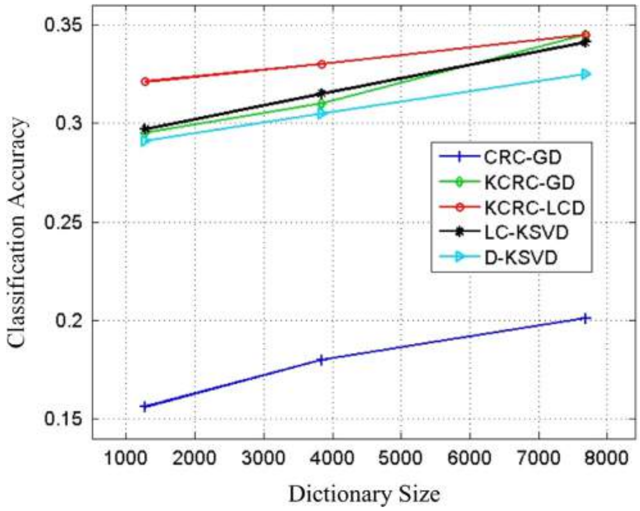


Fig. 7. Performance comparison with CRC-GD, KCRC-GD KCRC-LCD, and LC-KSVD [58,62] on Caltech 256 under different dictionary sizes. Note that, l_2 distance and the spatial pyramid features are used for similarity measure and classification.

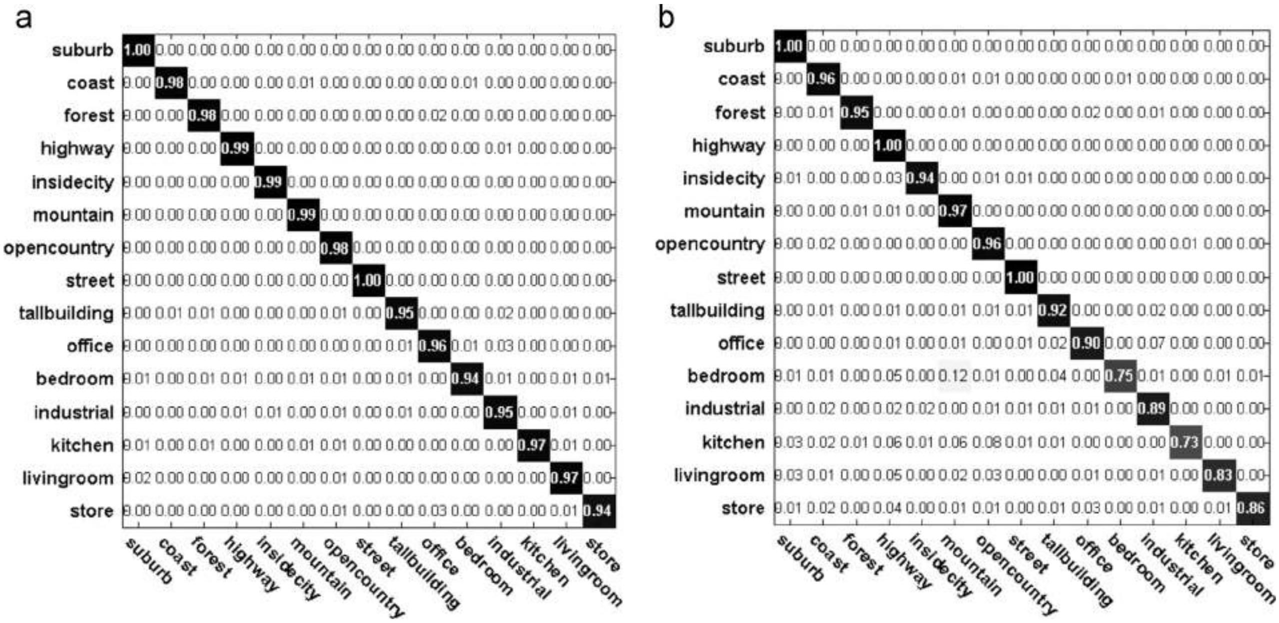
Table 2
Classification results using spatial pyramid features on 15 scene categories dataset. Both the dictionary size and *K* for LCD are set as 450.

Method	Accuracy (%)	Method	Accuracy (%)
KCRC-LCD	98.05	LC-KSVD [58,62]	92.94
KCRC-GD [19]	97.21	D-KSVD [63]	89.16
CRC [30]-LCD	92.13	KSRC [18]-LCD	96.97
CRC [30]-GD	90.92	KSRC [18]-GD	95.21
SRC [3]-GD	91.80	LLC [66] (30 local bases)	89.20

We evaluate the scalability of KCRC-LCD in terms of dictionary size and feature dimension on Caltech 101 dataset. First, we compare the running time among CRC-GD, KCRC-GD, and KCRC-LCD (*K* is set as 510). The experimental results in Fig. 9(a) show that the proposed KCRC-LCD runs with a relatively constant time. In contrast, the running time of KCRC-GD grows with nearly a exponential speed, and the running time of CRC-GD also increases faster than KCRC-LCD. We can infer that KCRC-GD and CRC-GD may become inefficient when the dictionary size is extremely large. However, KCRC-LCD remain in a reasonable running time when handling large-scale training samples. The biggest advantage of KCRC-LCD is that its running time increases very slowly with larger dictionary size. Second, we study how feature dimensionality can affect the running time of CRC-GD, KCRC-GD and KCRC-LCD (*K* is set as 510). Results in Fig. 9(b) show that KCRC-LCD is much more efficient than KCRC-GD in terms of feature dimension and the running time of KCRC-LCD increases with a much slower rate compared to KCRC-GD. In addition, we also compare the occupied memory of KCRC-GD and KCRC-LCD (*K* is set as 510) when running them on Matlab. Results in Fig. 10 show that KCRC-LCD occupies much less memory than KCRC-GD. Moreover, with larger dictionary size or larger feature dimension, the memory that KCRC-LCD occupies increases much slower than KCRC-GD.

6.6. *Evaluation of the unified distance measurement*

Distance metrics are of great importance in the KCRC-LCD, since they grant KCRC-LCD the scalability and discrimination power. Selecting the proper distance metric for the objects can greatly enhance the classification accuracy. Therefore, we validate the superiority of discriminative distance metrics by comparing different distance metrics in USPS, extended Yale B, MNIST and Caltech 101 dataset. For KCRC-LCD, we use the identity matrix as $\mathbf{\Psi}$. We use Euclidean distance as baseline for comparison. The USPS dataset contains 9288 handwritten digits collected from mail envelops [67]. There are 7291 images for training and 2007 images for testing. This dataset is fairly difficult since its human error rate is 2.5% [40]. We apply the tangent distance to construct the LCD and the corresponding kernel. *K* is set to 40 for USPS dataset. Specifically, tangents are attained by smoothing each image with a Gaussian kernel of width $\sigma = 0.5$. Results are shown in Table 5. For Extended Yale B, we use the same experimental settings as the previous subsection (*K* for LCD is set as 200.). We apply the distance metric that is proposed in [68]. In detail, local binary pattern (LBP) histograms are extracted from divided face area and concatenated into a single feature histogram. Then χ^2 distance is used to measure the similarity of different face histograms. The neighborhood for LBP operator is set as (8,2) and the window size is 11 × 13. We term the distance as LBP-χ^2 distance. For correlation distance, we use $Dist(\boldsymbol{v}_i, \boldsymbol{v}_j) = 1 - \boldsymbol{v}'^T \boldsymbol{v}''$. Results are shown in Table 5. The MNIST dataset [61] of handwritten digits contains 60,000 samples (10 digits) for training and 10,000 for testing. We



a

	suburb	coast	forest	highway	insidecity	mountain	opencountry	street	tallbuilding	office	bedroom	industrial	kitchen	livingroom	store
suburb	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
coast	0.00	0.98	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
forest	0.00	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00
highway	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
insidecity	0.00	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mountain	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
opencountry	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
street	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
tallbuilding	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.00	0.95	0.00	0.00	0.02	0.00	0.00	0.00
office	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.96	0.01	0.03	0.00	0.00	0.00
bedroom	0.01	0.00	0.01	0.01	0.00	0.01	0.01	0.00	0.01	0.00	0.94	0.01	0.00	0.01	0.01
industrial	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.95	0.00	0.01	0.00
kitchen	0.01	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.97	0.01	0.00
livingroom	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.97	0.00
store	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.03	0.00	0.00	0.00	0.01	0.94

b

	suburb	coast	forest	highway	insidecity	mountain	opencountry	street	tallbuilding	office	bedroom	industrial	kitchen	livingroom	store
suburb	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
coast	0.00	0.96	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
forest	0.00	0.01	0.95	0.00	0.00	0.01	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.00
highway	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
insidecity	0.01	0.00	0.00	0.03	0.94	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mountain	0.00	0.00	0.01	0.01	0.00	0.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
opencountry	0.00	0.02	0.00	0.00	0.00	0.00	0.96	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00
street	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
tallbuilding	0.00	0.01	0.00	0.01	0.00	0.01	0.01	0.01	0.92	0.00	0.00	0.02	0.00	0.00	0.00
office	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.01	0.02	0.90	0.00	0.07	0.00	0.00	0.00
bedroom	0.01	0.01	0.00	0.05	0.00	0.12	0.01	0.00	0.04	0.00	0.75	0.01	0.00	0.01	0.01
industrial	0.00	0.02	0.00	0.02	0.02	0.00	0.01	0.01	0.01	0.01	0.00	0.89	0.00	0.00	0.00
kitchen	0.03	0.02	0.01	0.06	0.01	0.06	0.08	0.01	0.01	0.00	0.00	0.00	0.73	0.00	0.00
livingroom	0.03	0.01	0.00	0.05	0.00	0.02	0.03	0.00	0.00	0.01	0.00	0.01	0.00	0.83	0.00
store	0.01	0.02	0.00	0.04	0.00	0.01	0.01	0.00	0.01	0.03	0.00	0.01	0.00	0.01	0.86

Fig. 8. Confusion matrices of (a) KCRC-LCD and (b) CRC-GD on the 15 scene categories dataset with dictionary size 450. Note that, Euclidean distance and the spatial pyramid features are used for similarity measure and classification.

Table 3
Experimental settings for running time test.

Dataset	Feature dimension	Category	Training size	Testing size	K for LCD
MNIST	784	10	500	10,000	50
Extended Yale B	504	38	1216	1198	200
Caltech 101	3000	102	3060	6084	510
15 Scene categories	3000	15	1500	2985	200

Table 4
Comparison results of average running time (ms) per classification.

Dataset	KCRC-LCD	KCRC-GD	CRC-GD	SRC-GD
MNIST	2.2618	3.4723	2.5451	338.69
Extended Yale B	90.434	367.31	200.48	587.31
Caltech 101	3016.1	12,752.4	2255.8	27,119
15 Scene categories	246.64	2516.7	430.48	5943.8

randomly select 20 samples per digit and construct a global dictionary of size 200 and test on the given 10,000 samples. K for LCD is set as 50 and raw pixel features are used. Results are given in Table 5. For Caltech 101 dataset [64], we randomly select 30 samples per class and test on the rest (global dictionary size is 3060). K for LCD is set as 500 and spatial pyramid features (they are the same features we use in previous subsections) are used. Results are given in Table 5. We perform these experiments not to obtain the state-of-the-art performance, but rather to compare the performance gain via different similarity measures (or distance metrics). Therefore, the experimental settings such as feature selection, dictionary size and K for LCD may not necessarily be the optimal choice.

Table 5 shows that properly selecting a good distance metric can enhance the discrimination power, and that the performance of different distance metrics can vary significantly as the distance changes (e.g. Euclidean distance performs worse than Correlation distance on both USPS and extended Yale B, but better on MNIST and Caltech 101). Such variation is ubiquitous since the statistical characteristics among different datasets are different. Fortunately, the proposed KCRC-LCD renders a unified distance framework in which advantages from different distance metrics can be combined. The unified distance measurement framework allows us to bypass the troublesome process of traversing every single metric to examine its performance. In Table 5, we can see that the unified distance mostly achieves the best performance or is very close to the optimal one. Basically, one no longer needs to consider the metric selection problem. The framework not only remediates the selection bias, but is also able to bring performance gain. The main reason why the unified framework is not the best on Yale B is because the distance metrics are not complementary on this dataset (There are very few cases where other distances can complement the Tangent distance. Including other distances is essentially just poisoning the good results). In addition, the classification performance is pretty saturated (close to 100%) and the dataset itself is not diverse enough. If another distance complementary to LBP-χ^2 distance is used, we believe that the classification accuracy can be further improved.

7. Conclusions

We elaborated the KCRC approach in which kernel technique is smoothly combined with CRC. KCRC enhances the discrimination ability of CRC, making the decision boundary more reasonable. Additionally, we present a locality constrained dictionary of which the locality is exploited to further enhance the classification performance. KCRC and LCD are mathematically linked via distance kernelization. On one hand, LCD not only helps the classifier adaptive and scalable to large datasets via pruning the dictionary, but also reduces the dimensionality in kernel space, enhancing both discrimination ability and efficiency. On the other hand, kernel function makes our approach discriminative and robust to more data distribution, i.e., the same direction distribution. Furthermore, the coarse-to-fine classification strategy of KCRC-LCD is similar to the human perception process, which makes the intuition of KCRC-LCD even more appealing.

We conduct comprehensive experiments to show the superiority of KCRC-LCD. Our approach yields very good classification

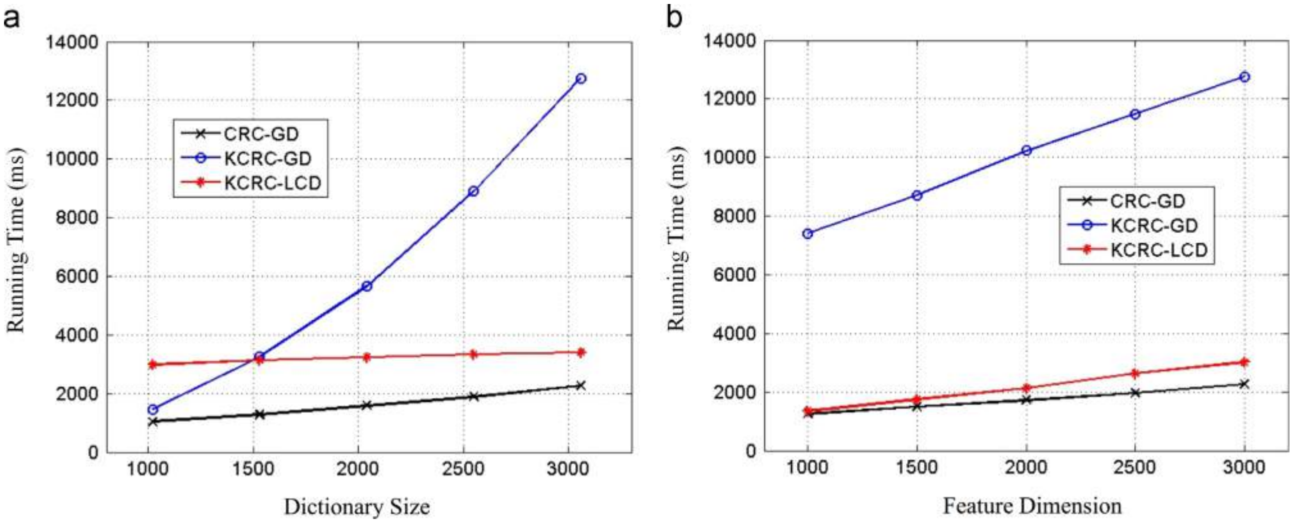


Fig. 9. (a) Running time vs. dictionary size per classification and (b) running time vs. feature dimension per classification.

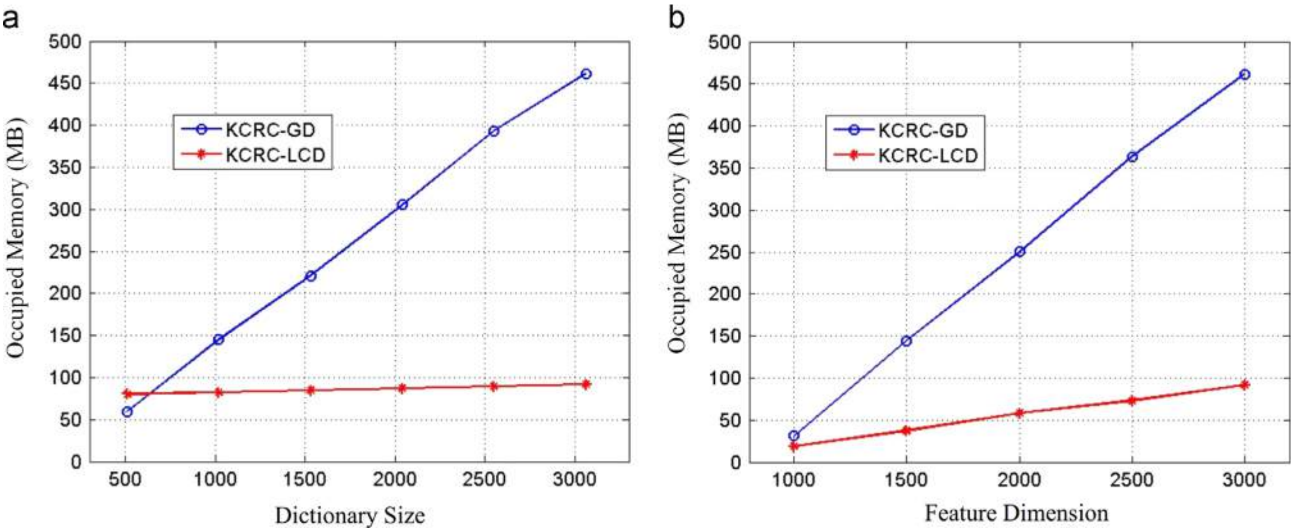


Fig. 10. (a) Occupied memory vs. dictionary size and (b) occupied memory vs. feature dimension.

Table 5
Recognition results (%) of different distance metrics on USPS, MNIST, Extended Yale B and Caltech 101 datasets.

Distance metric	USPS	Extended Yale B	MNIST	Caltech 101
Euclidean distance	95.49	96.93	85.31	72.83
Correlation distance	95.57	97.01	85.17	72.55
Tangent distance	97.37	N/A	N/A	N/A
LBP-χ^2 distance	N/A	98.53	N/A	N/A
Unified distance	97.67	98.22	87.52	73.05

results on various well-known public datasets. While achieving high level discrimination ability, efficiency is one of the biggest merits of KCRC-LCD, which is validated in running time test. Moreover, we simulate the representation and construction of KCRC with different dimensionality reduction for kernel space, and further experiment these methods on public datasets. The simulation results show the discrimination ability of KCRC. Different distance metrics used in LCD are also compared to support the idea that discriminative distance metric can greatly improve the classification accuracy. We also create a toy data sets to show CRC suffers from data with the same direction distribution while KCRC perfectly overcomes such shortcoming. To sum up, tested by various experiments, KCRC is proven discriminative and efficient when combined with LCD.

Possible future work includes improving the unified similarity measure model and learning the most effective kernel for KCRC-LCD instead of selecting a fixed kernel. It can be imagined that KCRC-LCD will become more powerful when combined with kernel learning, or even multiple kernel learning.

Conflict of interest

None declared.

Acknowledgments

This work was partially supported by the Shenzhen Science & Technology Fundamental Research Program (No. JCYJ20130329175141512).

References

[1] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding, J. Mach. Learn. Res. 11 (2010) 19–60.

[2] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. Image Process. 15 (12) (2006) 3736–3745.

[3] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.

[4] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Miami, FL, USA, 2009, pp. 1794–1801.

[5] X.-T. Yuan, X. Liu, S. Yan, Visual classification with multitask joint sparse representation, IEEE Trans. Image Process. 21 (10) (2012) 4349–4360.

[6] C. Rother, M. Kiefel, L. Zhang, B. Schölkopf, P.V. Gehler, Recovering intrinsic images with a global sparsity prior on reflectance, in: Advances in Neural Information Processing Systems, 2011, pp. 765–773.

[7] D.L. Donoho, For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution, Commun. Pure Appl. Math. 59 (6) (2006) 797–829.

[8] E.J. Candes, J.K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Commun. Pure Appl. Math. 59 (8) (2006) 1207–1223.

[9] E.J. Candes, T. Tao, Near-optimal signal recovery from random projections: universal encoding strategies? IEEE Trans. Inf. Theory 52 (12) (2006) 5406–5425.

[10] C. Lang, G. Liu, J. Yu, S. Yan, Saliency detection by multitask sparsity pursuit, IEEE Trans. Image Process. 21 (3) (2012) 1327–1338.

[11] T. Guha, R.K. Ward, Learning sparse representations for human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 34 (8) (2012) 1576–1588.

[12] X. Mei, H. Ling, Robust visual tracking and vehicle classification via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (11) (2011) 2259–2272.

[13] R. Rigamonti, M.A. Brown, V. Lepetit, Are sparse representations really relevant for image classification? in: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 1545–1552.

[14] Q. Shi, A. Eriksson, A. van den Hengel, C. Shen, Is face recognition really a compressive sensing problem? in: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Colorado Springs, CO, USA, 2011, pp. 553–560.

[15] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: Which helps face recognition? in: 2011 IEEE International Conference on Computer Vision (ICCV), IEEE, Barcelona, Spain, 2011, pp. 471–478.

[16] B. Schölkopf, A. Smola, K.-R. Müller, Kernel principal component analysis, in: Artificial Neural Networks—ICANN'97, Springer Berlin Heidelberg, 1997, pp. 583–588.

[17] C.J. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discov. 2 (2) (1998) 121–167.

[18] L. Zhang, W.-D. Zhou, P.-C. Chang, J. Liu, Z. Yan, T. Wang, F.-Z. Li, Kernel sparse representation-based classifier, IEEE Trans. Signal Process. 60 (4) (2012) 1684–1695.

[19] W. Liu, L. Lu, H. Li, W. Wang, Y. Zou, A novel kernel collaborative representation approach for image classification, in: 2014 IEEE International Conference on Image Processing (ICIP), IEEE, Paris, France, 2014, pp. 4241–4245.

[20] E. Rosch, C.B. Mervis, W.D. Gray, D.M. Johnson, P. Boyes-Braem, Basic objects in natural categories, Cogn. Psychol. 8 (3) (1976) 382–439.

[21] T. Malisiewicz, A. Gupta, A.A. Efros, Ensemble of exemplar-SVMs for object detection and beyond, in: 2011 IEEE International Conference on Computer Vision (ICCV), IEEE, Barcelona, Spain, 2011, pp. 89–96.

[22] S. Gao, I. W.-H. Tsang, L.-T. Chia, Kernel sparse representation for image classification and face recognition, in: Computer Vision—ECCV 2010, Springer Berlin Heidelberg, 2010, pp. 1–14.

[23] J. Yin, Z. Liu, Z. Jin, W. Yang, Kernel sparse representation based classification, Neurocomputing 77 (1) (2012) 120–128.

[24] W. Liu, Y. Wen, K. Pan, H. Li, Y. Zou, A kernel-based l 2 norm regularized least square algorithm for vehicle logo recognition, in: 2014 19th International Conference on Digital Signal Processing (DSP), IEEE, Hong Kong, China, 2014, pp. 631–635.

[25] J. Li, H. Zhang, L. Zhang, Column-generation kernel nonlocal joint collaborative representation for hyperspectral image classification, ISPRS J. Photogramm. Remote Sens. 94 (2014) 25–36.

[26] J. Li, H. Zhang, Y. Huang, L. Zhang, Hyperspectral image classification by nonlocal joint collaborative representation with a locally adaptive dictionary, IEEE Trans. Geosci. Remote Sens. 52 (6) (2014) 3707–3719.

[27] H. Van Nguyen, V.M. Patel, N.M. Nasrabadi, R. Chellappa, Design of non-linear kernel dictionaries for object recognition, IEEE Trans. Image Process. 22 (12) (2013) 5123–5135.

[28] A. Shrivastava, H.V. Nguyen, V.M. Patel, R. Chellappa, Design of non-linear discriminative dictionaries for image classification, in: Computer Vision—ACCV 2012, Springer Berlin Heidelberg, 2013, pp. 660–674.

[29] H. Nguyen, V.M. Patel, N.M. Nasrabadi, R. Chellappa, Kernel dictionary learning, in: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Kyoto, Japan, 2012, pp. 2021–2024.

[30] L. Zhang, M. Yang, X. Feng, Y. Ma, D. Zhang, Collaborative representation based classification for face recognition, arXiv preprint arXiv:1204.2358.

[31] V.N. Vapnik, V. Vapnik, Statistical Learning Theory, New York: Wiley, 1998.

[32] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, K. Mullers, Fisher discriminant analysis with kernels, in: Proceedings of the 1999 IEEE Signal Processing Society Workshop Neural Networks for Signal Processing IX, Madison, WI, USA, 1999, pp. 41–48.

[33] F. Orabona, J. Keshet, B. Caputo, The projectron: a bounded kernel-based perceptron, in: Proceedings of the 25th International Conference on Machine Learning, ACM, Helsinki, Finland, 2008, pp. 720–727.

[34] W. He, S. Wu, A kernel-based perceptron with dynamic memory, Neural Netw. 25 (2012) 106–113.

[35] O. Dekel, S. Shalev-Shwartz, Y. Singer, The forgetron: a kernel-based perceptron on a budget, SIAM J. Comput. 37 (5) (2008) 1342–1372.

[36] C.E. Rasmussen, Gaussian Processes for Machine Learning.

[37] D. Hardoon, S. Szedmak, J. Shawe-Taylor, Canonical correlation analysis: an overview with application to learning methods, Neural Comput. 16 (12) (2004) 2639–2664.

[38] A.Y. Ng, M.I. Jordan, Y. Weiss, et al., On spectral clustering: analysis and an algorithm, in: Advances in Neural Information Processing Systems vol. 2, 2002, pp. 849–856.

[39] B. Scholkopf, The kernel trick for distances, Advances in Neural Information Processing Systems, 2001, pp. 301–307.

[40] H. Zhang, A.C. Berg, M. Maire, J. Malik, SVM-KNN: discriminative nearest neighbor classification for visual category recognition, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE, New York, NY, USA, 2006, pp. 2126–2136.

[41] S. Becker, J. Bobin, E.J. Candès, Nesta: a fast and accurate first-order method for sparse recovery, SIAM J. Imaging Sci. 4 (1) (2011) 1–39.

[42] D.P. Bertsekas, Nonlinear Programming, Belmont: Athena Scientific, 1999.

[43] Z. Lin, M. Chen, Y. Ma, The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices, arXiv preprint arXiv:1009.5055.

[44] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.

[45] P. Zhu, L. Zhang, Q. Hu, S. C. Shiu, Multi-scale patch based collaborative representation for face recognition with margin distribution optimization, in: Computer Vision—ECCV 2012, Springer Berlin Heidelberg, 2012, pp. 822–835.

[46] J. Waqas, Z. Yi, L. Zhang, Collaborative neighbor representation based classification using l2-minimization approach, Pattern Recognit. Lett. 34 (2) (2013) 201–208.

[47] Y. Zhou, K.E. Barner, Locality constrained dictionary learning for nonlinear dimensionality reduction, IEEE Signal Process. Lett. 20 (4) (2013) 335–338.

[48] T. Leung, J. Malik, Representing and recognizing the visual appearance of materials using three-dimensional textons, Int. J. Comput. Vis. 43 (1) (2001) 29–44.

[49] E. Levina, Statistical issues in texture analysis (Ph.D. thesis), University of California, Berkeley, 2002.

[50] P.Y. Simard, Y.A. LeCun, J.S. Denker, B. Victorri, Transformation invariance in pattern recognition—tangent distance and tangent propagation, in: Neural Networks: Tricks of the Trade, Springer Berlin Heidelberg, 2012, pp. 235–269.

[51] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 509–522.

[52] A.C. Berg, J. Malik, Geometric blur for template matching, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001, vol. 1, IEEE, Kauai, HI, USA, 2001, p. I-607.

[53] G. Mori, J. Malik, Estimating human body configurations using shape context matching, in: Computer Vision—ECCV 2002, Springer Berlin Heidelberg, 2002, pp. 666–680.

[54] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.

[55] B. Taati, M. Greenspan, Local shape descriptor selection for object recognition in range data, Comput. Vis. Image Underst. 115 (5) (2011) 681–694.

[56] X. Bai, B. Wang, C. Yao, W. Liu, Z. Tu, Co-transduction for shape retrieval, IEEE Trans. Image Process. 21 (5) (2012) 2747–2757.

[57] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, Z. Tu, Unsupervised metric fusion by cross diffusion, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Providence, Rhode Island, USA, 2012, pp. 2997–3004.

[58] Z. Jiang, Z. Lin, L.S. Davis, Label consistent k-svd: learning a discriminative dictionary for recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2651–2664.

[59] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: 2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, IEEE, New York, NY, USA, 2006, pp. 2169–2178.

[60] A.S. Georghiades, P.N. Belhumeur, D. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 643–660.

[61] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[62] Z. Jiang, Z. Lin, L.S. Davis, Learning a discriminative dictionary for sparse coding via label consistent K-SVD, in: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Colorado Springs, CO, USA, 2011, pp. 1697–1704.

[63] Q. Zhang, B. Li, Discriminative K-SVD for dictionary learning in face recognition, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, San Francisco, CA, USA, 2010, pp. 2691–2698.

[64] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, Comput. Vis. Image Underst. 106 (1) (2007) 59–70.

[65] G. Griffin, A. Holub, P. Perona, Caltech-256 Object Category Dataset.

[66] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, San Francisco, CA, USA, 2010, pp. 3360–3367.

[67] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551.

[68] T. Ahonen, A. Hadid, M. Pietikäinen, Face recognition with local binary patterns, in: Computer vision—ECCV 2004, Springer Berlin Heidelberg, 2004, pp. 469–481.

Weiyang Liu received the B.Eng. degree from the School of Electronic and Information Engineering, South China University of Technology, in 2013. He was the recipient of the National Scholarship of PR China (2012, 2014). He is currently a second-year master student with the School of Electronic and Computer Engineering, Peking University. His research interests mainly focus on sparse signal processing and sparse representation, with applications to computer vision.

Zhiding Yu received the B.Eng. degree (Talented Student Program) from the School of Electronic and Information Engineering, South China University of Technology, in 2008, and the M.Phil. degree from Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology in 2012. Between 2010 and 2012, he was a research intern student with the Multimedia Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, and the Robotics Institute, Carnegie Mellon University. He is currently a third-year Ph.D. candidate with the Department of Electrical and Computer Engineering, Carnegie Mellon University. His main research interests include computer vision and applied machine learning.

He was twice the recipient of the HKTIIT Post-Graduate Excellence Scholarships (2010/2011). He is a co-author of the best student paper in International Symposium on Chinese Spoken Language Processing (ISCSLP) 2014 and the winner of best paper award in IEEE Winter Conference on Applications of Computer Vision (WACV) 2015.

Lijia Lu received the B.Eng. degree from the School of Computer Science and Technology, Xidian University, China, in 2013. She is currently a second-year master student with the School of Electronic and Computer Engineering, Peking University. Her research interests include graphics, applied pattern recognition and computer vision.

Yandong Wen received the B.Eng. degree from School of Electronic and Information Engineering, South China University of Technology, in 2013. He is currently a second-year master student with School of Electronic and Information Engineering, South China University of Technology. His research interests include applied pattern recognition and computer vision.

Hui Li received both the B.S. and M.S. degrees from Tsinghua University, in 1986 and 1989 respectively, and the Ph.D. degree from the Chinese University of Hong Kong, in 2000. He is currently a professor with the School of Electronic and Computer Engineering, Peking University, China. He has been or is currently in charge of several important projects, including projects funded and supported by the 863 program (State High-Tech Development Plan), the 973 Program (National Basic Research Program), the National Natural Science Foundation of China and Ministry of Science and Technology of China. He had published over 50 papers in journals and conferences.

His research interests include cloud computing and multimedia signal processing.

Yuexian Zou received both the B.S. and M.S. degrees from the University of Electronic Science and Technology of China, in 1985 and 1990 respectively, and the Ph.D. degree from the University of Hong Kong, in 2000. She is currently a professor with the School of Electronic and Computer Engineering Peking University, China, and is the Director of the Advanced Digital Signal Processing Laboratory.

Since 2005, she has been actively involved in the national and international academic activities. She serves as the Evaluation/Peer Review Expert for National Natural Science Foundation of China, Shenzhen Bureau of Science Technology and Information, and the Paper Reviewer for several IEEE journals and international conferences. She is currently working on a high-speed and high-resolution analog-to-digital converter research project. Her research interests include digital filter design, adaptive filtering, array signal processing, video signal processing and pattern recognition.