Pattern Recognition 48 (2015) 3025–3037

Contents lists available at ScienceDirect

Pattern Recognition

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

# Kernel collaborative face recognition

Dong Wang [a], Huchuan Lu [a,*], Ming-Hsuan Yang [b]

[a] School of Information and Communication Engineering, Dalian University of Technology, Dalian, China
[b] Department of Electrical Engineering and Computer Science, University of California, Merced, CA, USA

ARTICLE INFO

*Article history:*
Received 8 September 2014
Received in revised form
21 November 2014
Accepted 14 January 2015
Available online 22 January 2015

*Keywords:*
Face recognition
Kernel methods
Sparse representation
Collaborative representation

ABSTRACT

Recent research has demonstrated the effectiveness of linear representation (i.e., sparse representation, group sparse representation and collaborative representation) for face recognition and other vision problems. However, this linear representation assumption does not consider the non-linear relationship of samples and limits the usage of different features with non-linear metrics. In this paper, we present some insights of linear and non-linear representation-based classifiers. First, we present a general formulation known as kernel collaborative representation to encompass several effective representation-based classifiers within a unified framework. Based on this framework, different algorithms can be developed by choosing proper kernel functions, regularization terms, and additional constraints. Second, within the proposed framework we develop a simple yet effective algorithm with squared $\ell_2$-regularization and apply it to face recognition with local binary patterns as well as the Hamming kernel. We conduct numerous experiments on the extended Yale B, AR, Multi-PIE, PloyU NIR, PloyU HS, EURECOM Kinect and FERET face databases. Experimental results demonstrate that our algorithm achieves favorable performance in terms of accuracy and speed, especially for the face recognition problems with small training datasets and heavy occlusion. In addition, we attempt to combine different kernel functions by using different weights in an additive manner. The experimental results show that the proposed combination scheme provides some additional improvement in terms of accuracy.



## 1. Introduction

As an interesting and important topic in computer vision, face recognition has many useful applications, such as surveillance, human–computer interface, access control, argument reality, among others. Although numerous algorithms have been proposed for face recognition (e.g., principal component analysis (PCA) [30], linear discriminant analysis (LDA) [7], elastic bunch graph matching (EBGM) [34], local binary patterns (LBP) [1], histogram of Gabor phase patterns (HGPP) [43], rank-one projections (ROP) [37], nearest subspace (NS) [18], locality preserving projection (LPP) [12] and Discriminant Image Filter Learning (DIFL) [19]), it remains as a challenging problem due to intrinsic (e.g., aging and expression variations) and extrinsic appearance change (e.g., occlusion, pose, and illumination variations).

The recent years have witnessed an increasing interest of sparse representation for vision problems, e.g., face recognition [36], super-resolution [38], image inpainting [35], facial expression recognition [47] and visual tracking [23,31]. In [36], Wright et al. propose a method based on sparse representation for face recognition which makes use of the $\ell_1$-minimization techniques. In this method, a query face image is linearly represented (or coded) by all training images with a sparsity constraint imposed on the coding vector, and then classification is performed by identifying the class with minimal reconstruction error. Several approaches based on sparse representation have since been proposed for face recognition. Yang et al. [39] adopt Gabor features rather than raw pixels with sparse representation and learn an occlusion dictionary to handle occluded face images. Zhou et al. [48] exploit the spatial continuity of occluded pixels with a Markov random filed model to handle occlusion problems. Jia et al. [15] and Zhuang et al. [49] focus on designing practical face recognition systems by pursuing structured sparsity or handling image corruption and misalignment. In [46], Zhang et al. attempt to improve the accuracy of the face recognition method by combining low-rank and sparsity constraints to mine discriminative components of facial images. By considering the prototype and variation separately, Deng et al. [5] present a superposed SRC (SSRC) method to deal with face recognition under uncontrolled conditions. However, recent findings [29,45] reveal that sparsity constraint plays a less important role in effective face recognition. In [29], Shi et al. show that a simple $\ell_2$-based approach performs as well as the algorithm with $\ell_1$ constraints [36]. Furthermore, Zhang et al. [45] demonstrate that the role of collaboration between classes in

* Corresponding author.

http://dx.doi.org/10.1016/j.patcog.2015.01.012
0031-3203/

representing a query image is more important than that of the sparsity constraints. In [45], a collaborative representation is presented with a squared $\ell_2$-regularization which achievescompetitive performance in terms of accuracy but with significantly lower complexity than the sparse representation method.

Despite the demonstrated success, there exists a critical issue in both approaches based on sparse representation and collaborative representation that needs to be addressed. Both approaches have the same fundamental assumption that a test sample can be well coded by a set of training samples with a linear representation scheme. However, face recognition is more likely a non-linear problem due to the complexity of facial images and the "small sample size" problem. Therefore the linear assumption may limit face recognition performance as it neither exploits the non-linear relationship of samples nor adopts features (such as local binary patterns [24]) that require non-linear metrics. For example, without effective features and metrics, the sparse representation method [36] requires numerous face samples usage from each individual to construct an overcomplete dictionary such that the approach with sparse linear representation is able to perform well. However, in realistic applications, it may be difficult to maintain a large training set due to both labor cost and computational complexity. Therefore, it is critical to develop algorithms with effective features and non-linear metrics to describe the non-linear relationships among face samples especially when the training sample size is small.

Motivated by the fact that kernel methods [28] are able to capture the non-linear similarity of samples effectively, we present a unified framework and propose a kernel collaborative representation scheme for linear and non-linear representation-based approaches. The contributions of this work are threefold. First, we present a general framework based on kernel collaborative representation, which not only unifies state-of-the-art representation-based classifiers but also facilitates developing new algorithms. Second, within this framework we propose a simple yet effective method by using a squared $\ell_2$ regularization. Third, we apply the proposed algorithm to face recognition with local binary patterns and the Hamming kernel. We conduct numerous experiments on the extended Yale B, AR, large-scale Multi-PIE, PloyU NIR, PloyU HS, EURECOM Kinect FERET face databases to demonstrate the merits of the proposed algorithm.

## 2. Linear representation

In this section, we summarize and discuss the linear representation-based classifiers (i.e., sparse representation (SR) [36], group sparse representation (GSR) [21], and collaborative representation (CR) [45]), and put our work in proper context.

Let $\mathbf{A}_i = [\mathbf{a}_{i,1}, \mathbf{a}_{i,2}, \ldots, \mathbf{a}_{i,n_i}] \in \mathbb{R}^{m \times n_i}$ denote the dataset of the $i$th class, where each column is a sample of class $i$, and $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k]$ represent the whole training set, where $k$ is the number of classes. Given a test sample $\mathbf{y} \in \mathbb{R}^{m \times 1}$, we describe it by a linear representation $\mathbf{y} \approx \mathbf{A}\mathbf{x}$, where $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k]^\top$ ($\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}]^\top$ stands for the coding coefficients associated with the $i$th class). We denote the $\ell_1$- and $\ell_2$-norm of $\mathbf{x}$ as $\|\mathbf{x}\|_1$ and $\|\mathbf{x}\|_2$, respectively. The group norm ($\ell_{2,1}$-norm) of $\mathbf{x}$ is computed by $\|\mathbf{x}\|_{2,1} = \sum_{i=1}^{k} \|\mathbf{x}_i\|_2$, where each class is treated as an individual group.

The main steps of the linear representation framework are listed in Algorithm 1. First, we normalize each column of $\mathbf{A}$ to have a unit $\ell_2$-norm to avoid a degenerate solution (e.g., coding coefficients are too large or too small). In step 2, the first term $\frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ is related to the reconstruction error, and the underlying assumption is that a test sample $\mathbf{y}$ is linearly represented by all training dataset collaboratively rather than by training samples from an individual class. The second term of the optimization problem in step 2 is a regularization term. Different regularization terms lead to different properties of the coding coefficients (i.e., $\ell_1$-regularization (2(a)) encourages sparsity [36], $\ell_{2,1}$-regularization (2(b)) promotes group sparsity, and $\ell_2$-regularization (2(c)) does not lead to sparsity but makes the solution simple and stable [45]). After obtaining the coding vector $\mathbf{x}$, we determine which class a test sample $\mathbf{y}$ belongs to. In [36] and [21], the reconstruction error (or residual) of each class (3(a)) is used for classification, while in [45], the regularized residual of each class (3(b)) is demonstrated to perform better than the residual for classification. Finally, we classify $\mathbf{y}$ by assigning it to the object class that minimizes the residual or regularized residual (Step 4 of Algorithm 1).

**Algorithm 1.** Linear representation framework.

1. Normalize each column of $\mathbf{A}$ to have a unit $\ell_2$-norm.
2. Code a test sample $\mathbf{y}$ by a linear representation (2(a), 2(b) or 2(c)),
   2(a) $\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$ (SR [36])
   2(b) $\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_{2,1}$ (GSR [21])
   2(c) $\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_2^2$ (CR [45])
   to obtain the corresponding coding coefficients.
3. Compute the residuals (3(a)) or regularized residuals (3(b)) of each $i$-class by
   3(a) $r_i = \|\mathbf{y} - \mathbf{A}_i\mathbf{x}_i\|_2$ (SR [36] and GSR [21])
   3(b) $r_i = \|\mathbf{y} - \mathbf{A}_i\mathbf{x}_i\|_2 / \|\mathbf{x}_i\|_2$ (CR [45])
4. Output the identity of $\mathbf{y}$ as
   Identity($\mathbf{y}$) = arg $\min_i\{r_i\}$

For effective face recognition algorithms based on linear models, the following issues need to be discussed and addressed: sparse or collaborative representation, and linear model.

### 2.1. Sparse or collaborative representation

Motivated by the demonstrated success of the sparse representation approach for the face recognition algorithm [36], numerous methods [21,48,39,9] have been proposed which emphasize the importance of sparsity constraints for classification rather than representation schemes and classifiers. However, Shi et al. [29] demonstrate that a simple $\ell_2$-based approach is able to achieve comparable performance with the $\ell_1$-based algorithm. Furthermore, Zhang et al. [45] show that the role of collaborative representation (collaboration between classes in representing the query sample) is more important than that of the sparsity constraints. Since face recognition tasks often suffer from the typical "small sample size" or "lack of samples" problems, it is less effective to represent a test sample simply by training samples from an individual class with the sparse representation scheme. In [36], each test face image is represented by images from all possible classes rather than an individual class, and thus the above-mentioned problem is alleviated. Based on this observation, Zhang et al. [45] propose a collaborative representation method by using a squared $\ell_2$-regularization term. Compared to the method based on sparse representation [36], this collaborative representation algorithm provides a simple yet more efficient classification scheme [45]. It is worth emphasizing that we present a kernel collaborative representation scheme that considers both linear and non-linear approaches in a unified framework.

### 2.2. Linear relationship

Both face recognition algorithms based on sparse representation [36] and collaborative representation [45] assume that a test sample can be well linearly represented by all training samples.

However, this assumption may not hold in real world applications due to the complexity of samples and the "small sample size" problem. In addition, the linear representation assumption limits the usage of certain features and non-linear metrics. Motivated by the fact that kernel methods can capture the non-linear similarity of samples and can exploit different kinds of features, kernel sparse representations [16,9] have been proposed for face recognition and object classification. Although these kernel methods perform favorably against linear sparse representation algorithms, they are nevertheless time-consuming due to the use of complicated $\ell_1$-optimization techniques. In addition, Yang et al. [42,33] present a kernel collaborative representation (KCR) model, which extends the CR [45] model in a non-linear way. However, there is a lack of a general framework to unify linear and non-linear representations. In this work, we present a unified framework and design a simple yet effective algorithm for face recognition. This unified framework not only provides a principled way to understand why a specific method works well or not, but also facilitates developing new algorithms.

## 3. Kernel collaborative representation (KCR)

### 3.1. KCR framework

We present a unified framework based on the kernel collaborative representation (KCR) that accommodates linear and non-linear schemes in this section. Similar to face recognition methods based on linear representation, the KCR framework consists of two main steps: representation and classification (i.e., a test sample is first represented with the coding coefficients by minimizing a specific objective function and then classified based on these coefficients).

Suppose that there exists a function $\phi$, which maps the sample $\mathbf{y}$ and the basis $\mathbf{A}$ to a high dimensional feature space: $\mathbf{y}\rightarrow\phi(\mathbf{y})$ and $[\mathbf{a}_{1,1},\mathbf{a}_{1,2},\ldots,\mathbf{a}_{k,n_k}]\rightarrow[\phi(\mathbf{a}_{1,1}),\phi(\mathbf{a}_{1,2}),\ldots,\phi(\mathbf{a}_{k,n_k})]$. By assuming that the mapped test sample $\phi(\mathbf{y})$ can be linearly represented by the mapped basis $\phi(\mathbf{A})$ in the high dimensional feature space, we define the objective function of kernel collaborative representation (KCR) as

$$\begin{aligned}&\min_{\mathbf{x}}\ \tfrac{1}{2}\|\phi(\mathbf{y})-\phi(\mathbf{A})\mathbf{x}\|_2^2+\Omega(\mathbf{x})\\&\quad\text{s.t.}\quad \boldsymbol{\psi}_1(\mathbf{x})=0\\&\qquad\boldsymbol{\psi}_2(\mathbf{x})\geq 0,\end{aligned}\tag{1}$$

where $\Omega(\mathbf{x})$ is a regularization term, and $\boldsymbol{\psi}_1(\mathbf{x})$ and $\boldsymbol{\psi}_2(\mathbf{x})$ stand for some constraints. By introducing a kernel function $\kappa(\mathbf{x}_i,\mathbf{x}_j)=\langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle$, the objective function can be modified as

$$\begin{aligned}&\min_{\mathbf{x}}\ \ \tfrac{1}{2}\|\phi(\mathbf{y})-\phi(\mathbf{A})\mathbf{x}\|_2^2+\Omega(\mathbf{x})\\&\quad=\min_{\mathbf{x}}\tfrac{1}{2}\kappa(\mathbf{y},\mathbf{y})+\tfrac{1}{2}\mathbf{x}^\top K_{\mathbf{AA}}\mathbf{x}-\mathbf{x}^\top K_{\mathbf{A}}(\mathbf{y})+\Omega(\mathbf{x})\\&\quad=\min_{\mathbf{x}}\tfrac{1}{2}\mathbf{x}^\top K_{\mathbf{AA}}\mathbf{x}-\mathbf{x}^\top K_{\mathbf{A}}(\mathbf{y})+\Omega(\mathbf{x}),\end{aligned}\tag{2}$$

where $K_{\mathbf{AA}}$ is a $n\times n$ matrix with $[K_{\mathbf{AA}}]_{ij}=\kappa(\mathbf{a}_i,\mathbf{a}_j)$, $K_{\mathbf{A}}(\mathbf{y})$ is an $n\times 1$ vector with $[K_{\mathbf{A}}(\mathbf{y})]_i=\kappa(\mathbf{a}_i,\mathbf{y})$, $n$ stands for the number of all training samples, and $\mathbf{a}_i$ denotes the $i$th column of $\mathbf{A}$.

After we obtain the coding coefficients by solving the objective function (Eq. (2)), we classify a test sample $\mathbf{y}$ by assigning it to the object class that minimizes the residuals,

$$\begin{aligned}i^*&=\min_{i=1,2,\ldots,k}\|\phi(\mathbf{y})-\phi(\mathbf{A}_i)\mathbf{x}_i\|_2^2\\&=\min_{i=1,2,\ldots,k}-2\mathbf{x}_i^\top K_{\mathbf{A}_i}(\mathbf{y})+\mathbf{x}_i^\top K_{\mathbf{A}_i\mathbf{A}_i}\mathbf{x}_i,\end{aligned}\tag{3}$$

or the regularized residuals,

$$\begin{aligned}i^*&=\min_{i=1,2,\ldots,k}\|\phi(\mathbf{y})-\phi(\mathbf{A}_i)\mathbf{x}_i\|_2^2/\|\mathbf{x}_i\|_2^2\\&=\min_{i=1,2,\ldots,k}\frac{\kappa(\mathbf{y},\mathbf{y})-2\mathbf{x}_i^\top K_{\mathbf{A}_i}(\mathbf{y})+\mathbf{x}_i^\top K_{\mathbf{A}_i\mathbf{A}_i}\mathbf{x}_i}{\|\mathbf{x}_i\|_2^2},\end{aligned}\tag{4}$$

where $\mathbf{A}_i=[\mathbf{a}_{i,1},\mathbf{a}_{i,2},\ldots,\mathbf{a}_{i,n_i}]$ denotes the dataset of the $i$th class and $\mathbf{x}_i$ is its corresponding representation coefficients. The notions $K_{\mathbf{AA}}$ and $K_{\mathbf{A}}(\mathbf{y})$ can be easily represented by

$$\begin{aligned}K_{\mathbf{AA}}&=\begin{bmatrix}K_{\mathbf{A}_1\mathbf{A}_1}&K_{\mathbf{A}_1\mathbf{A}_2}&\cdots&K_{\mathbf{A}_1\mathbf{A}_k}\\K_{\mathbf{A}_2\mathbf{A}_1}&K_{\mathbf{A}_2\mathbf{A}_2}&\cdots&\cdots\\\cdots&\cdots&\cdots&\cdots\\K_{\mathbf{A}_k\mathbf{A}_1}&\cdots&\cdots&K_{\mathbf{A}_k\mathbf{A}_k}\end{bmatrix}\\K_{\mathbf{A}}(\mathbf{y})&=[K_{\mathbf{A}_1}(\mathbf{y}),K_{\mathbf{A}_2}(\mathbf{y}),\ldots,K_{\mathbf{A}_k}(\mathbf{y})]^\top.\end{aligned}\tag{5}$$

We note that the regularized residual term (4) is better than the residual term (3) for the classification task. For the non-linear representation model, the mapped test sample $\phi(\mathbf{y})$ can be linearly represented by the mapped basis as $\phi(\mathbf{y})=\phi(\mathbf{A}_1)\mathbf{x}_1+\phi(\mathbf{A}_2)\mathbf{x}_2+\cdots+\phi(\mathbf{A}_k)\mathbf{x}_k$. The residual for $i$th class (i.e., $\|\phi(\mathbf{y})-\phi(\mathbf{A}_i)\mathbf{x}_i\|_2^2$) merely considers the reconstruction error with respect to the mapped sub-basis $\phi(\mathbf{A}_i)$ but completely ignores the information of the representation coefficient vector. Intuitively, if the mapped test sample $\phi(\mathbf{y})$ belongs to the $i$th class, it means the sample $\phi(\mathbf{y})$ can be represented by the $i$th sub-basis $\phi(\mathbf{A}_i)$ rather than other basis vectors. For one thing, the reconstruction error $\|\phi(\mathbf{y})-\phi(\mathbf{A}_i)\mathbf{x}_i\|_2^2$ should be as small as possible. For another, the squared norm $\|\mathbf{x}_i\|_2^2$ should be as large as possible to capture more energy. Thus, the regularized residual scheme is more effective by considering both the reconstruction error with respect to the mapped sub-basis and the energy of the coefficient on the mapped sub-basis.

### 3.2. Generalization of KCR framework

We note that the proposed kernel collaborative representation (KCR) facilitates to exploit non-linear relationship of samples by using effective features and non-linear metrics via kernel functions. By choosing different types of kernel functions, regularization terms and additional constraints, many effective algorithms including the nearest neighbor (NN), nearest subspace (NS) [18], sparse representation (SR) [36], collaborative representation (CR) [45], group sparse representation (GSR) [21], and kernel sparse

**Table 1**
Kernel collaborative representation for effective face recognition.

| Algorithm | Kernel function | Regularization | Constraint or assumption |
|---|---|---|---|
| SR [36] | Linear | $\ell_1$-Norm ($\lambda\|\mathbf{x}\|_1$) | – |
| CR [45] | Linear | Squared $\ell_2$-norm ($\lambda\|\mathbf{x}\|_2^2$) | – |
| GSR [21] | Linear | $\ell_{2,1}$-Norm ($\lambda\|\mathbf{x}\|_{2,1}$) | – |
| KSR [16,9] | Gaussian [9] Hamming, Chi-square [16] | $\ell_1$-Norm ($\lambda\|\mathbf{x}\|_1$) | – |
| KCR [42,33] | Gaussian, polynomial | Squared $\ell_2$-norm ($\lambda\|\mathbf{x}\|_2^2$) | – |
| NN | Linear | – | $\|\mathbf{x}\|_0=1,\|\mathbf{x}\|_2^2=1$ |
| NS [18] | Linear | Squared $\ell_2$-norm ($\lambda\|\mathbf{x}\|_2^2$) | Samples from different classes are orthogonal. |

representation (KSR) [16,9]), can be viewed as special cases of the unified KCR framework as summarized in Table 1. We show that these methods operate as representation-based classifiers. In addition, we explain the underlying reasons why they do not perform well in face recognition as a result of enforced strong constraints that affect their generalization abilities.

Within the proposed framework, numerous methods can be developed using different regularizations and constraints to exploit desired properties. A simple idea is that we can choose different types of $\Omega(\mathbf{x})$ to design different algorithms for exploiting different properties of the representation coefficients. For instance, the $\ell_1$-regularization ($\lambda\|\mathbf{x}\|_1$) encourages sparsity and the $\ell_{2,1}$-regularization ($\lambda\|\mathbf{x}\|_{2,1}$) increases group sparsity. In addition, we can use some mixed regularizations (e.g., elastic net regularization ($\lambda_1\|\mathbf{x}\|_1+\lambda_2\|\mathbf{x}\|_2^2$) [50] and sparse group sparsity regularization ($\lambda_1\|\mathbf{x}\|_1+\lambda_2\|\mathbf{x}\|_{2,1}$) [8]) or some additional constraints (e.g., non-negative constraint ($\mathbf{x}\geq 0$) and shift-invariant constraint ($\mathbf{1}^\top\mathbf{x}=1$) [32]).

#### 3.2.1. Relationship to the nearest subspace (NS) classifier

The NS method [18] aims to seek the best representation in terms of all the training samples of each class. For a given sample $\mathbf{y}$, the NS method first $k$ (the number of class) individual regression problem,

$$\min_{\mathbf{x}_i}\frac{1}{2}\|\mathbf{y}-\mathbf{A}_i\mathbf{x}_i\|_2^2+\frac{\lambda}{2}\|\mathbf{x}_i\|_2^2, \tag{6}$$

where $\mathbf{A}_i$ denotes the dataset of the $i$th class, and $\mathbf{x}_i$ describes its regression coefficient. The classification is conducted by using a series of residuals,

$$i^*=\arg\min_i\|\mathbf{y}-\mathbf{A}_i\mathbf{x}_i\|_2^2. \tag{7}$$

Here we present a kernel version of the objective function for the NS method as

$$\begin{aligned}\min_{\mathbf{x}_i}&\frac{1}{2}\|\phi(\mathbf{y})-\phi(\mathbf{A}_i)\mathbf{x}_i\|_2^2+\frac{\lambda}{2}\|\mathbf{x}_i\|_2^2\\ &=\min_{\mathbf{x}_i}\frac{1}{2}\mathbf{x}_i^\top K_{\mathbf{A}_i\mathbf{A}_i}\mathbf{x}_i-\mathbf{x}_i^\top K_{\mathbf{A}_i}(\mathbf{y})+\frac{\lambda}{2}\|\mathbf{x}_i\|_2^2,\end{aligned} \tag{8}$$

where the squared $\ell_2$-regularization term aims to make the solution simple and stable. This objective function needs to be solved $k$ times in order to obtain all representation coefficients $\mathbf{x}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_k]$ where $\mathbf{x}_i$ stands for the coding coefficient of the $i$th class. We note that it is equal to solving the following objective function

$$\begin{aligned}\min_{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_k}&\sum_{i=1}^{k}\frac{1}{2}\mathbf{x}_i^\top K_{\mathbf{A}_i\mathbf{A}_i}\mathbf{x}_i-\mathbf{x}_i^\top K_{\mathbf{A}_i}(\mathbf{y})+\frac{\lambda}{2}\|\mathbf{x}_i\|_2^2\\ &=\min_{\mathbf{x}}\frac{1}{2}\mathbf{x}^\top\overline{K_{\mathbf{AA}}}\mathbf{x}-\mathbf{x}^\top K_{\mathbf{A}}(\mathbf{y})+\frac{\lambda}{2}\|\mathbf{x}\|_2^2,\end{aligned} \tag{9}$$

where $\overline{K_{\mathbf{AA}}}=\mathrm{diag}\{K_{\mathbf{A}_1\mathbf{A}_1},K_{\mathbf{A}_2\mathbf{A}_2},\ldots,K_{\mathbf{A}_k\mathbf{A}_k}\}$.

Therefore we show that the NS method is a special case of our KCR framework under the assumption that the kernel matrix of all training samples is block diagonalized (i.e., samples from different classes are orthogonal), which is a strong assumption for real datasets to hold.

#### 3.2.2. Relationship to the nearest neighbor (NN) classifier

Given a test sample $\mathbf{y}$, the NN method assigns it to class $i$ if the smallest distance from $\mathbf{y}$ to the nearest training sample of class $i$ is the smallest among all classes. It can be viewed as solving the following objective function:

$$\begin{aligned}\min_{\mathbf{x}}\quad&\tfrac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2\\ \text{s.t.}\quad&\|\mathbf{x}\|_0=1,\|\mathbf{x}\|_2^2=1,\end{aligned} \tag{10}$$

which is equivalent to adding the constraint that only one element of $\mathbf{x}$ will be set to 1 and the remaining ones will be set to 0. We present a kernel version of the NN method within our KCR framework as

$$\begin{aligned}\min_{\mathbf{x}}\quad&\tfrac{1}{2}\mathbf{x}^\top K_{\mathbf{AA}}\mathbf{x}-\mathbf{x}^\top K_{\mathbf{A}}(\mathbf{y})\\ \text{s.t.}\quad&\|\mathbf{x}\|_0=1,\|\mathbf{x}\|_2^2=1.\end{aligned} \tag{11}$$

Generally, the diagonal elements of the kernel matrix $K_{\mathbf{AA}}$ are all 1, and thus the first term of Eq. (11) has no effect on the solution. Therefore we conclude that the NN method is also a special case of our KCR framework under a strong constraint on the coding coefficients ($\|\mathbf{x}\|_0=1,\|\mathbf{x}\|_2^2=1$).

After the coding coefficients are obtained by Eqs. (9) and (11), the NS and NN classifications can be carried out by minimizing the residuals (Eq. (3)).

### 3.3. Proposed KCR-$\ell_2$ algorithm

A natural question ensues: which regularization or constraint yields the best performance for a specific problem? It is concerned with the regularizations or constraints which match the problem settings, and whether we can develop an algorithm to obtain an accurate solution. Furthermore, the time complexity should also be considered especially for real-time applications.

Based on the proposed KCR framework, we present a simple classifier for face recognition by using a squared $\ell_2$-regularization (which is denoted as KCR-$\ell_2$). The objective function can be defined as

$$\min_{\mathbf{x}}\frac{1}{2}\mathbf{x}^\top K_{\mathbf{AA}}\mathbf{x}-\mathbf{x}^\top K_{\mathbf{A}}(\mathbf{y})+\frac{\lambda}{2}\|\mathbf{x}\|_2^2, \tag{12}$$

where the parameter $\lambda$ is a regularization constant to make the solution stable. The solution of KCR-$\ell_2$ can be easily derived by setting the derivation of the objective function $J(\mathbf{x})$ to zero,

$$\frac{\partial J(\mathbf{x})}{\partial\mathbf{x}}=K_{\mathbf{AA}}\mathbf{x}-K_{\mathbf{A}}(\mathbf{y})+\lambda\mathbf{x}=\mathbf{0}, \tag{13}$$

where $J(\mathbf{x})=(1/2)\mathbf{x}^\top K_{\mathbf{AA}}\mathbf{x}-\mathbf{x}^\top K_{\mathbf{A}}(\mathbf{y})+\lambda/2\|\mathbf{x}\|_2^2$. Then the analytical solution can be obtained as

$$\mathbf{x}=(K_{\mathbf{AA}}+\lambda\mathbf{I})^{-1}K_{\mathbf{A}}(\mathbf{y}). \tag{14}$$

Let $\mathbf{Q}$ denote the first term, $\mathbf{Q}=(K_{\mathbf{AA}}+\lambda\mathbf{I})^{-1}$. This term can be easily pre-computed merely with a training dataset $\mathbf{A}$. Once a query sample $\mathbf{y}$ arrives, we only need to compute $K_{\mathbf{A}}(\mathbf{y})$, which makes our algorithm efficient.

**Algorithm 2.** KCR-$\ell_2$ for face recognition.

**Input:** a training dataset $\mathbf{A}$, kernel function $\kappa(\cdot,\cdot)$, pre-computed matrix $\mathbf{Q}(=(K_{\mathbf{AA}}+\lambda\mathbf{I})^{-1})$, $K_{\mathbf{AA}}$, and regularization constant $\lambda$.

1. Compute $K_{\mathbf{A}}(\mathbf{y})$ by Eq. (5)
2. Code a test sample $\mathbf{y}$ by KCR-$\ell_2$: $\mathbf{x}=\mathbf{Q}K_{\mathbf{A}}(\mathbf{y})$
3. Compute the regularized residuals of each $i$-class by $r_i=\frac{\kappa(\mathbf{y},\mathbf{y})-2\mathbf{x}_i^\top K_{\mathbf{A}_i}(\mathbf{y})+\mathbf{x}_i^\top K_{\mathbf{A}_i\mathbf{A}_i}\mathbf{x}_i}{\|\mathbf{x}_i\|_2^2}$

**Output:** the identity of $\mathbf{y}$ as $\mathrm{Identity}(\mathbf{y})=\arg\min_i\{r_i\}$

For classification, we use the regularized residuals (Eq. (4)) as suggested in [45]. We summarize the procedures of our KCR-$\ell_2$ method in Algorithm 2. We note that the method based on the collaborative representation [45] can be viewed as a special case of our KCR algorithm (with linear kernel $\kappa(\mathbf{a}_i,\mathbf{a}_j)=\mathbf{a}_i^\top\mathbf{a}_j$ under the assumption that the columns of $\mathbf{A}$ are normalized).

It is also worth mentioning that the proposed KCR-$\ell_2$ method is different from the kernel ridge regression (i.e., least square kernel regression) algorithm [2] for face recognition. Kernel ridge

regression (KRR) is one of the non-linear dimensionality reduction algorithms similar to kernel principal component analysis (KPCA) [28], kernel discriminant analysis (KDA) [3] and kernel locality preserving projection (KLPP) [12]. The objective function of KRR aims to obtain effective low-dimensional features with labels of examples. Thus, it requires to use additional classification methods (such as nearest neighbor classifiers) for classification results. In contrast, the proposed KCR-$\ell_2$ algorithm is a classification method, which is a simple yet effective case within the proposed KCR framework.

## 4. Experimental results

We evaluate the performance of the proposed KCR-$\ell_2$ algorithm for face recognition using several large-scale datasets with different image modalities. Six databases, including the extended Yale B [10], AR [22], Multi-PIE [11], PloyU NIR [44], PloyU HS [6] and EURECOM Kinect [14] datasets, are used to evaluate the proposed KCR-$\ell_2$ algorithm and related methods based on nearest neighbor (NN), nearest subspace (NS) [18], sparse representation (SR) [36], collaborative representation (CR) [45], and kernel sparse representation (KSR) [16]. For the SR method [36], considering the accuracy and computational efficiency, we choose the $\ell_1$-regularized logistic regression algorithm [17] to solve the $\ell_1$ minimization. For the CR algorithm, we use the codes provided by [45]. We implement the KSR approach with the kernel coordinate descent (KCD) algorithm proposed in [16]. To demonstrate the superiority of the proposed algorithm, we also compare it with other two state-of-the-art methods, relaxed collaborative representation (RCR) [41] and regularized robust sparse coding (RRSC) [40]. In this work, the parameter $\lambda$ is set to 0.005 for all evaluated methods and the effect of $\lambda$ is shown in Section 4.2. All experiments are conducted using MATLAB implementations on a standard **i**5-580 2.67 GHz machine with 2.0 GB RAM. The source codes will be made available to the public for research purposes.

For the KCR-$\ell_2$ and KSR algorithms, the LBP features [24] and the Hamming kernel [16] are adopted. We denote these methods as KCR-$\ell_2$(LBP+HK) and KSR(LBP+HK) respectively. The $\mathrm{LBP}_{8,1}$ operator [24] is performed on each facial image for feature extraction, and then the Hamming kernel is applied to the LBP encoded images. This representation has been shown to perform better than the LBP histogram with $\chi^2$ kernel, especially when faces are occluded [16]. The Hamming kernel we use is defined as

$$\kappa(\mathbf{x},\mathbf{y}) = 1 - \frac{1}{mN}\sum_{i=1}^{m} D(\mathbf{x}_i,\mathbf{y}_i), \tag{15}$$

where $\mathbf{x}$ and $\mathbf{y}$ stand for LBP codes of two facial images, $m$ is the number of image pixels, $N$ is the length of the coding sequence, $\mathbf{x}_i$ and $\mathbf{y}_i$ denote LBP codes at the $i$th pixel, and $D(\cdot,\cdot)$ denotes the Hamming distance of two binary sequences. We also implement a method using the collaborative representation with the LBP operator and linear kernel (denoted as CR(LBP)), to demonstrate that the improvement of recognition rates results from both effective features and proper metrics (e.g., kernel function).

### 4.1. Face recognition results

#### 4.1.1. Extended Yale B database

The extended Yale B database [10] contains 2414 frontal face images of 38 subjects taken under 64 illumination conditions. We collect the cropped and normalized face images of $32 \times 32$ pixels from [4]. A random subset with $l$ images per individual is collected with their labels to form a training set ($l$=5,10,20,30,40,50), and the rest is considered as the corresponding test set. Fig. 1(a) shows, under different number of training samples, the KSR(LBP+HK) [16] and the proposed KCR-$\ell_2$(LBP+HK) methods perform better than the other algorithms. Although all methods (except NN) perform relatively well when sufficient training samples ($l > 20$) are used, the KSR [16] and proposed KCR-$\ell_2$ methods are more robust even when the training sample size is small ($l$=5). This can be attributed to that the LBP operator captures intrinsic structure of individual face images and the kernel based framework successfully exploits this property with the proper Hamming kernel. The CR(LBP) method also performs well because the main challenge of the extended Yale B database is illumination variation and the LBP operator captures sufficient image structures under various illumination conditions. However, the combination of LBP features and linear kernel do not always work well, which will be demonstrated in this section.

#### 4.1.2. AR database

The AR database [22] consists of over 4000 frontal images of 126 individual persons. As in [36], we choose a subset (with only illumination and expression change) that contains 50 male subjects and 50 female subjects. For each individual, 14 unoccluded images, 6 occluded images with sunglasses, and 6 occluded images

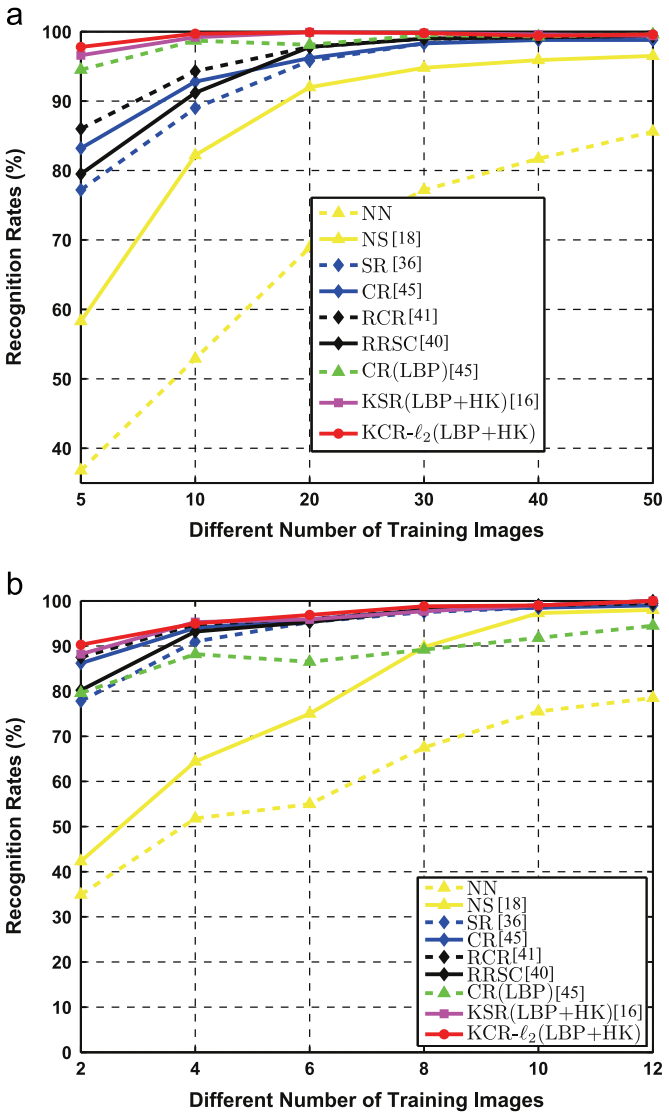


**Fig. 1.** Recognition rates (%) of different algorithms on Extended Yale B [10] and AR [22] (unoccluded) databases. This figure reports the results of our algorithm and its competing methods with different number of training images. (a) Extended Yale B database, (b) AR database (unoccluded).

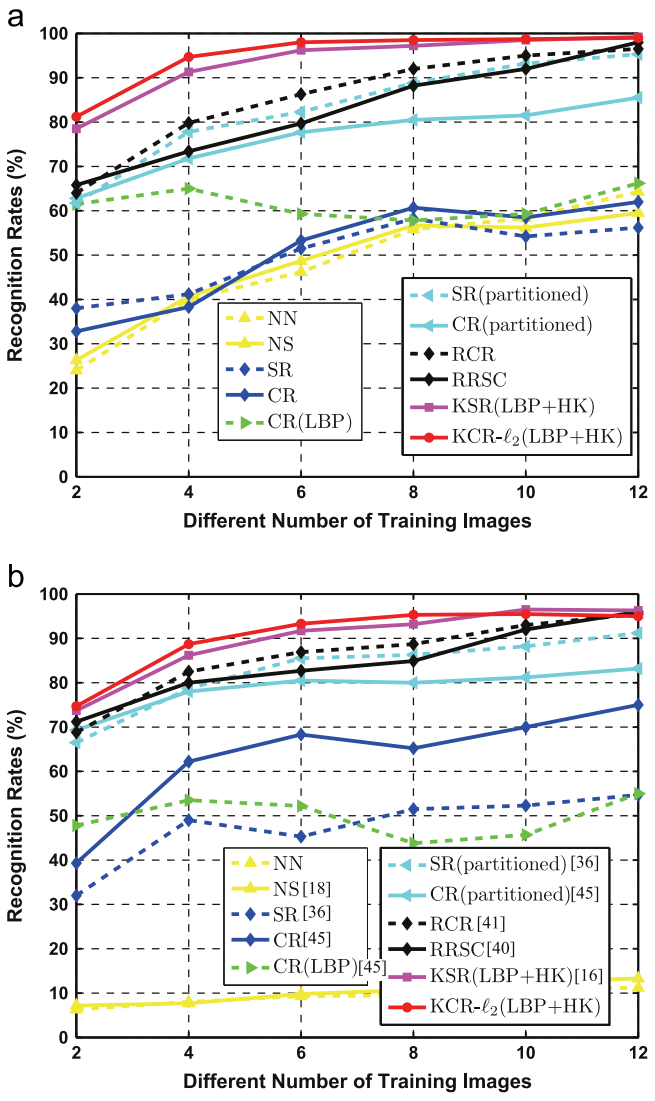


**Fig. 2.** Recognition rates (%) of different algorithms used from the AR database [22]. The results show that the evaluated methods perform with different number of training images. (a) AR database (occlusions with sunglasses), (b) AR database (occlusions with scarves).

with scarves are chosen. Each image is normalized to $32 \times 32$ pixels. For performance evaluation of different algorithms under the un-occlusion condition, a random subset with $l$ images per individual is taken with their labels to form the unoccluded set ($l$=2,4,6,8,10,12), and the rest of the unoccluded set is considered as the corresponding test set. Fig. 1(b) shows that the KSR and the proposed KCR-$\ell_2$ methods perform favorably against the SR and CR algorithms, especially when the training sample size is small ($l$=2,4). The mediocre performance of the SR and CR algorithms can be explained by that the linear representation assumption does not hold under the "small training size" condition. The LBP operator could encode intrinsic structure of individual face images with even small training sample size, and the Hamming kernel makes full use of this property. Therefore the proposed KCR-$\ell_2$ method performs well when only 2 or 4 training samples per individual are used. When 2 training samples per individual are used, the KCR-$\ell_2$ method also achieves 2% improvement over the KSR method. This is because the KSR method encourages sparsity by using $\ell_1$-regularization, which increases the risk of incorrect recognition when only very few training samples are available.

#### 4.1.3. *Face recognition with disguise*

We also evaluate the ability of face recognition methods to deal with real occlusions and disguises (See Fig. 3 for some examples) using the AR database [22]. A random subset with $l$ images per individual is selected with their labels to form the unoccluded set ($l$=2, 4, 6, 8, 10, 12), the occluded sets with sunglasses and scarves are used for tests. Fig. 2 shows the results with different settings. The KSR(LBP+HK) and our KCR-$\ell_2$(LBP+HK) methods perform better than the other methods. On the other hand, the SR and CR methods are less effective in handling occlusions in real-world images. We note that parts-based representations are used to cope with this situation [36,45]. Each face image is partitioned into 8 blocks where each block is processed (recognized) independently, and the results are aggregated by voting for face recognition. Thus, the SR(partitioned) and CR(partitioned) methods achieve more better performance than the traditional SR and CR ones due to the parts-based representation. The RCR method also adopts the parts-based representation and learns discriminative weights on different parts, thus, it performs well in handling this case. In addition, the RRSC algorithm also achieves better performance since the robust sparse coding process can handle

**Fig. 3.** Samples with sunglasses (first row) and scarves (second row) in the AR database [22].

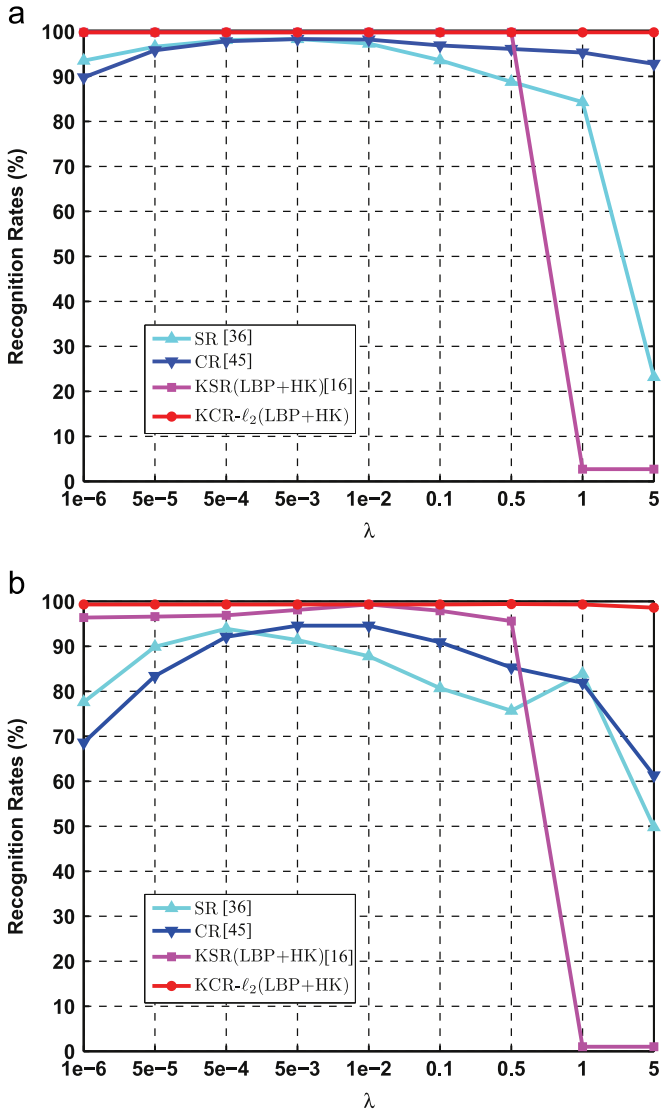


**Fig. 4.** Recognition rates of the SR, CR, KSR(LBP+HK), and KCR-$\ell_2$(LBP+HK) methods versus different values of $\lambda$ on the (a) Extended Yale B and (b) AR (unoccluded) databases.

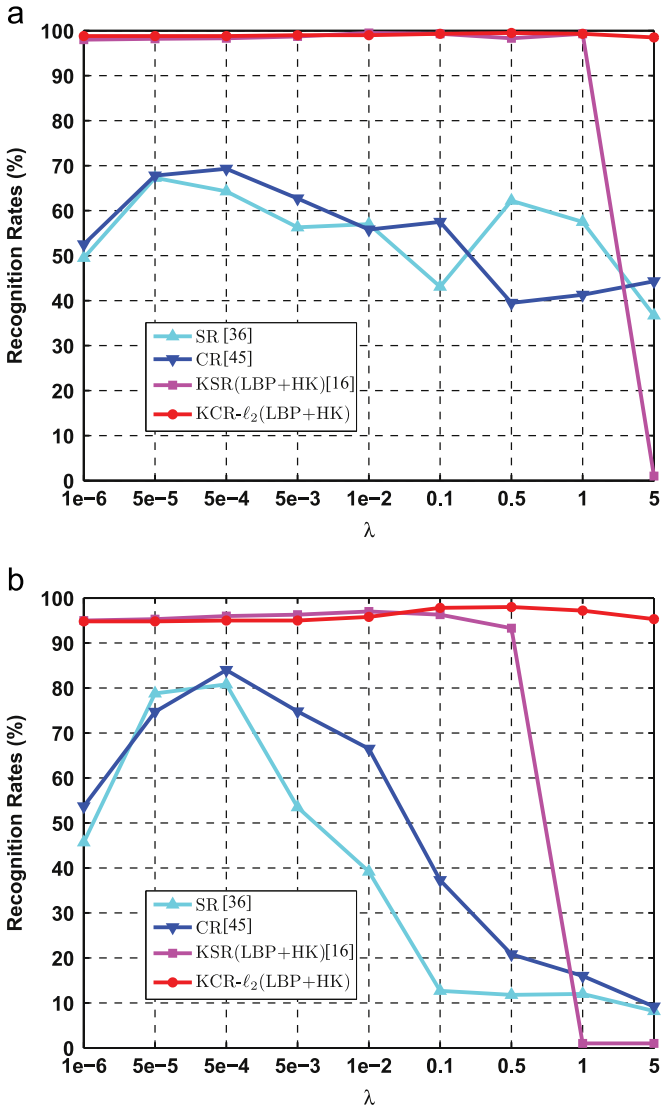


**Fig. 5.** Recognition rates of the SR, CR, KSR(LBP+HK), and KCR-$\ell_2$(LBP+HK) methods versus different values of $\lambda$ on the (a) AR (occluded with sunglasses) and (b)AR (occluded with scarves) databases.

occlusion to some extent. Fig. 2 demonstrates that although the use of parts-based representation (CR(partitioned) and SR(partitioned)) facilitates improving recognition results, the KSR (LBP+HK) and KCR-$\ell_2$(LBP+HK) methods still perform better especially when the training size is small. For instance, the proposed KCR-$\ell_2$(LBP+HK) algorithm outperforms the parts-based methods by almost 20% when $l=2$ in Fig. 2(a). We also note that the method based on LBP features with linear kernel (CR (LBP)) does not perform well, which demonstrates the importance of a proper metric (kernel function).

### 4.2. *Effect of $\lambda$*

In this section, we study the effect of $\lambda$ on the extended Yale B [10], and AR [22] databases. For the extended Yale B database and the unoccluded images of the AR database, we randomly split each of them into two halves. One half is selected as a training set, and the other half is used for tests. The results of our KCR-$\ell_2$(LBP+HK) and related methods on these databases are shown in Fig. 4(a) and (b). For experiments on occluded faces with the AR database, we use 1400 unoccluded face images as a training set, 600 images of subjects wearing sunglasses as a test set, and 600 images of subjects wearing scarves as another test set. Fig. 5(c) and (d) shows the results of all the evaluated methods.

From Figs. 4 and 5, we have two interesting and important observations. First, the kernel-based methods perform more stably with respect to the value of $\lambda$ than the other methods. We note that the most important role of $\lambda$ is to make the coding coefficients stable. Since the combination of the LBP features with the Hamming kernel makes full use of intrinsic structure of face samples, the effect of $\lambda$ in kernel-based methods is less important than that of the other methods. Second, the role of sparsity constraint is less important. The results show that the KCR-$\ell_2$(LBP+HK) method performs as well as the KSR(LBP+HK) algorithm when $\lambda$ is smaller than 0.5. However, the KSR(LBP+HK) method simply fails when $\lambda$ is larger than or equal to 1. For methods based on $\ell_1$-minimization, the importance of sparsity constraint is overemphasized if the value of $\lambda$ is too large and thereby drastically affects the recognition rates. Since there is only one parameter ($\lambda$) in the proposed KCR-$\ell_2$(LBP+HK) algorithm, the results in Figs. 4 and 5 demonstrate that the proposed KCR-$\ell_2$(LBP+HK) method achieves favorable performance in terms of accuracy and stability.

**Table 2**
Recognition rates and speeds on different face databases.

| Methods | Recognition rate (%) | Speed (s) |
|---|---|---|
| (a) Extended Yale B database | | |
| SR | 98.3 | 7.5319 |
| CR | 98.3 | 0.0017 |
| KSR(LBP+HK) | 99.8 | 1.8560 |
| KCR-$\ell_2$(LBP+HK) | 99.8 | 0.1353 |
| (b) AR database (unoccluded) | | |
| SR | 91.4 | 2.9366 |
| CR | 94.6 | 0.0026 |
| KSR(LBP+HK) | 98.1 | 0.8052 |
| KCR-$\ell_2$(LBP+HK) | 99.3 | 0.0805 |
| (c) AR database (occluded with sunglasses) | | |
| SR | 56.3 | 3.2750 |
| CR | 62.7 | 0.0034 |
| SR(partitioned) | 94.8 | 13.212 |
| CR(partitioned) | 83.5 | 0.0731 |
| KSR(LBP+HK) | 98.7 | 2.8387 |
| KCR-$\ell_2$(LBP+HK) | 99.0 | 0.1823 |
| (d) AR database (occluded with scarves) | | |
| SR | 53.5 | 3.2775 |
| CR | 74.8 | 0.0033 |
| SR(partitioned) | 90.3 | 11.688 |
| CR(partitioned) | 82.0 | 0.0658 |
| KSR(LBP+HK) | 96.3 | 2.8485 |
| KCR-$\ell_2$(LBP+HK) | 95.0 | 0.1767 |
| (e) FERET database | | |
| SR | 84.0 | 30.1121 |
| CR | 78.0 | 0.0915 |
| KSR(LBP+HK) | 83.0 | 5.8930 |
| KCR-$\ell_2$(LBP+HK) | 88.3 | 0.6812 |

We compare the run time performance of the SR, CR, KSR (LBP+HK) methods and the proposed KCR-$\ell_2$(LBP+HK) algorithm using the same databases and data collection schemes in Section 4.2. The recognition rates and average speed are reported in Table 2. The CR method [45] performs best in terms of execution time. Compared to the CR method [45], the proposed KCR-$\ell_2$(LBP+HK) algorithm requires some additional overhead, including feature extraction and kernel computation for a test sample. However, the KCR-$\ell_2$(LBP+HK) method outperforms the CR algorithm in terms of accuracy, especially when faces are occluded. We note that although the KCR-$\ell_2$(LBP+HK) method performs as well as the KSR(LBP+HK) algorithm in terms of accuracy, the proposed collaborative representation algorithm runs at least 10 times faster than the kernel sparse representation method.

### 4.3. Face databases of different modalities

In order to show the generalization ability and effectiveness of the proposed method, we conduct several experiments on other face databases captured by different lighting sources and modalities.

#### 4.3.1. Large-scale Multi-PIE database

We assess the scalability of the proposed algorithm and state-of-the-art methods on the large-scale Multi-PIE [11] database. Although there exist other large-scale datasets (e.g., LFW [13]), the CMU Multi-PIE [11] database has more images per person which is necessary for fair evaluations with different number of training images. The CMU Multi-PIE database [11] contains face images of 337 subjects captured in four sessions with different pose, expression, and illumination. For comparisons with the CR method [45], we use the same subset that contains 8916 frontal images of 249 subjects where each face image is cropped and normalized to $32\times32$ pixels. A subset with $l$ ($l$=2,4,6,8,10,12) images per individual from session 1 is randomly selected to form a training set, and the rest is used for tests. Fig. 6(a) shows that two methods of the proposed unified framework, i.e., the KSR and KCR-$\ell_2$ algorithms, consistently outperform other algorithms especially when the training sample size is small ($l$=2,4).

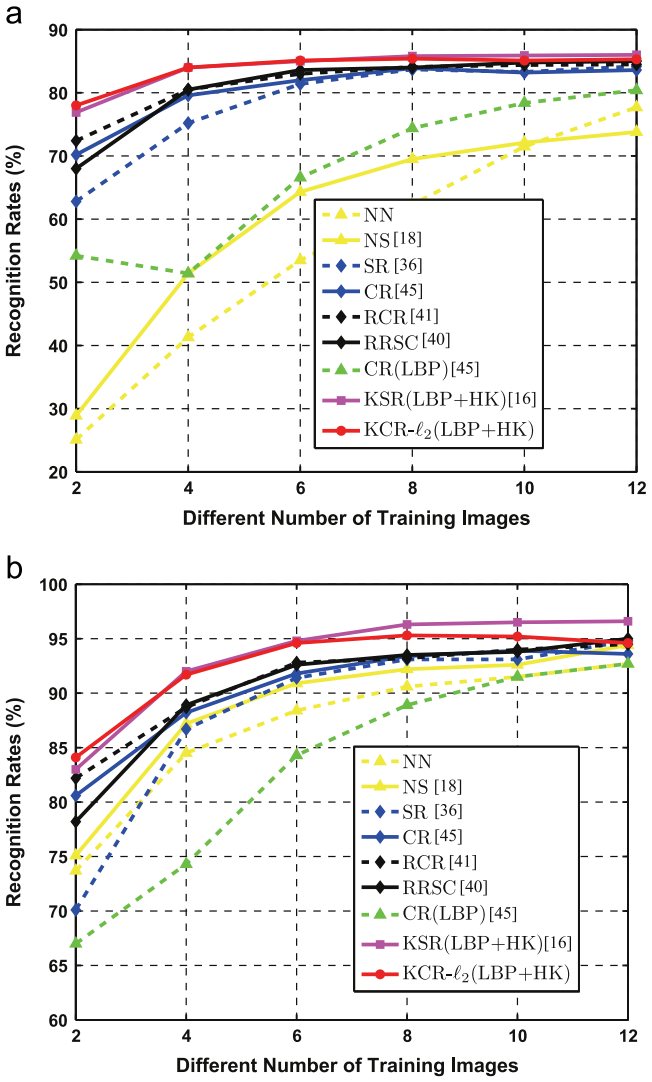


**Fig. 6.** Recognition rates (%) of different algorithms on (a) large-scale Multi-PIE [11] and (b) PolyU Near-Infrared (NIR) [44] databases. This figure reports the results of our algorithm and its competing methods with different number of training images.

#### 4.3.2. PolyU Near-Infrared (NIR) database

In recent years, active NIR-based face recognition methods have demonstrated promising results in several applications [51,20,44]. We evaluate the proposed KCR-$\ell_2$ method and state-of-the-art algorithms using near-infrared face images from the PloyU NIR database [44]. The PloyU NIR database [44] is a large near-infrared face database consisting of 35,000 images from 350 subjects under variation of pose, expression, illumination, scale and blur (see Fig. 8 for some sample images). Similar to [44], we use the same subset in our experiments, which contains 3675 frontal images of 245 subjects. Each face image is cropped and normalized to $32\times32$ pixels. A subset with $l$ ($l$=2,4,6,8,10,12) images per individual is randomly selected to form a training set, and the rest is considered as the corresponding test set. Fig. 6(b) shows that the KSR and KCR-$\ell_2$ methods consistently outperform other algorithms especially when the training sample size is small ($l$=2,4).

#### 4.3.3. PolyU Hyperspectral (HS) database

The PolyU Hyperspectral (HS) database [6] consists of 47 individuals. Fig. 9 shows some sample images where the spectral range is between 400 nm and 720 nm with an increment of 10 nm (i.e., 33 bands in total). The frontal HS images of the 47 individuals (from the first cube) are used in the experiments. Similar to [6], the first six and last three bands are removed due to high noise level, thereby leaving 24 spectral bands for experiments. The face images are cropped (the eye coordinates are located manually for image registration) and normalized to $32 \times 32$ pixels. A subset with $l$ ($l$=2, 4, 6, 8, 10, 12) images per individual is randomly selected to form a training set, and the rest is used for tests. Fig. 7(a) shows that the KSR and KCR-$\ell_2$ algorithms perform less effectively than the other methods. We note that the HS face images contain significantly more noise than face images from other lighting sources. As the original LBP features are known to be sensitive to noise [1], the kernel-based methods (with the LBP features) do not perform well for hyperspectral images.

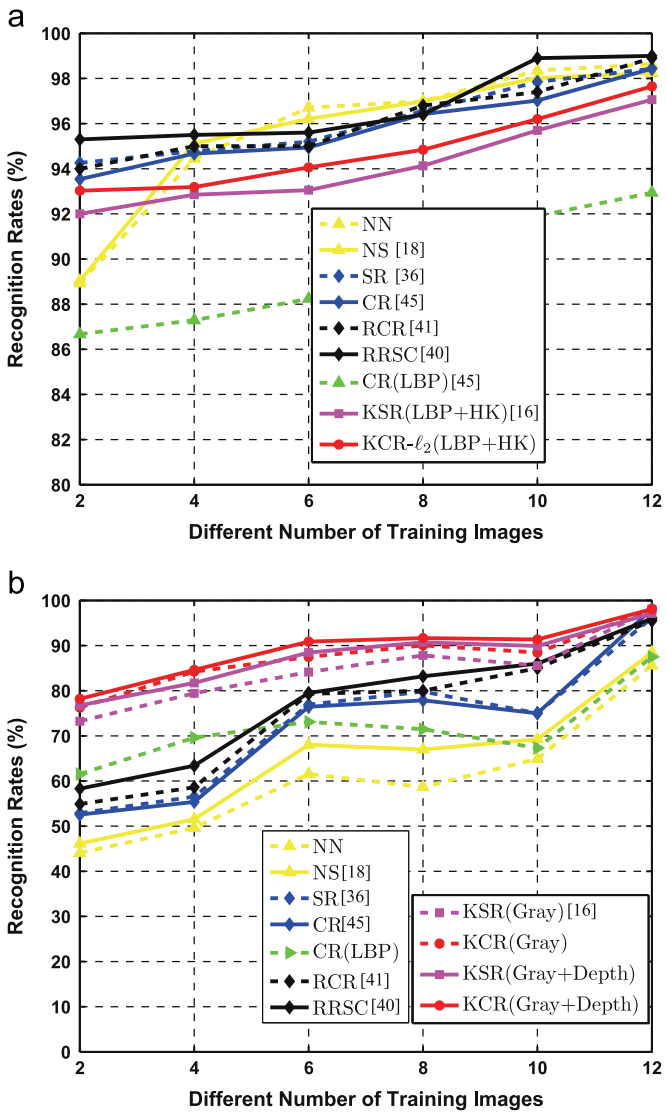


**Fig. 7.** Recognition rates (%) of different algorithms on (a) PolyU Hyperspectral (HS) [1] and (b) EURECOM Kinect [6] databases. This figure reports the results of our algorithm and its competing methods with different number of training images.

#### 4.3.4. EURECOM Kinect database

The EURECOM Kinect [14] dataset consists of multi-modal facial images of 52 people obtained by Kinect sensors (samples are shown in Fig. 10). The images are captured in two sessions at different time periods (about half a month apart). In each session, face images of each person are collected with 9 different combinations of facial expressions, lighting and occlusion conditions. The frontal face images are cropped (where the eye coordinates are located manually for image registration) and normalized to $32 \times 32$ pixels. A subset with $l$ ($l$=2, 4, 6, 8, 10, 12) images per individual is randomly selected to form a training set, and the rest is used for tests. Fig. 7(b) shows the recognition results of different algorithms where the NN, NS, SR, CR and CR(LBP) methods only use the intensity values with the LBP features and Hamming kernel. For the kernel-based methods, we evaluate the recognition results using the intensity values (denoted as KSR(Gray) and KCR(Gray)) and also the combination of gray and depth information (denoted as KSR(Gray+Depth) and KCR(Gray+Depth)), in which the LBP codes on Gray and depth images are concatenated into a single feature vector. The results from Fig. 7(b) show that the kernel-based algorithms perform better than the other methods as the LBP operator with Hamming kernel captures sufficient image structures and is robust to different occlusion conditions. In addition, the depth information is able to provide additional improvements especially when the training sample size is small ($l$=2, 4).

#### 4.3.5. FERET database

The FERET face image database is a result of the FERET program, which was sponsored by the US Department of Defense through the DARPA Program [25]. It has become a standard database for testing and evaluating state-of-the-art face recognition algorithms, as it has variations of facial expression, illumination, and pose. The FERET dataset totally includes 2413 still facial images, representing 856

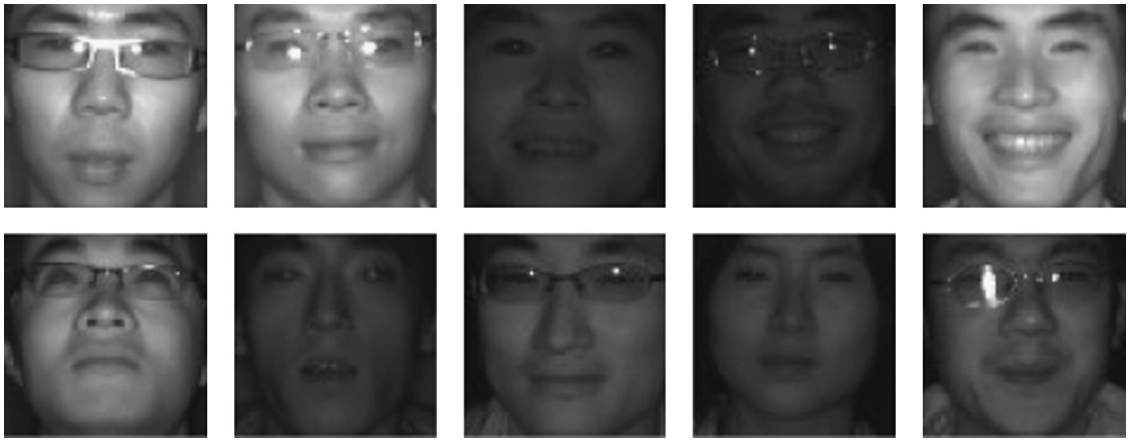

**Fig. 8.** Samples in the PolyU Near-Infrared (NIR) database [44].

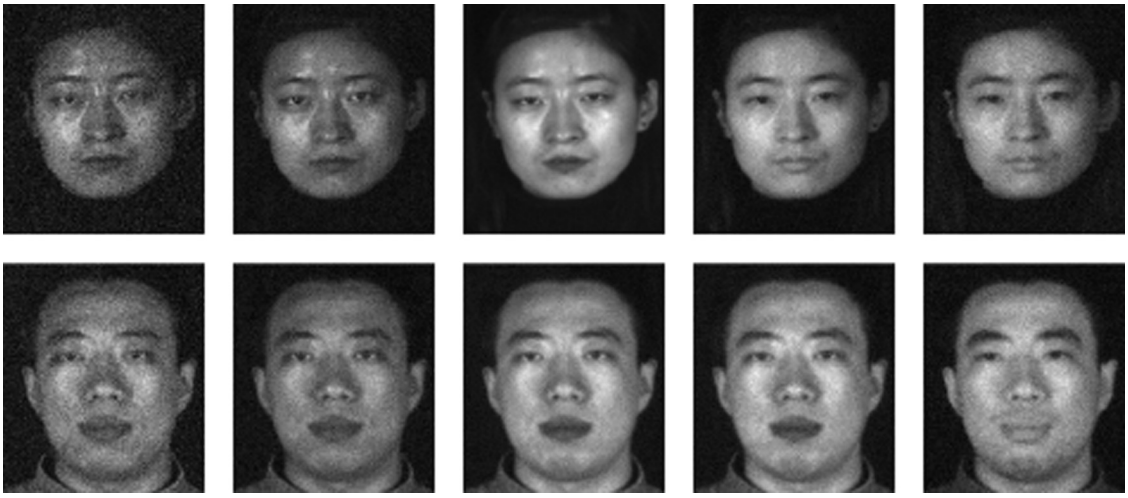

**Fig. 9.** Samples in the PolyU Hyperspectral (HS) database [6].

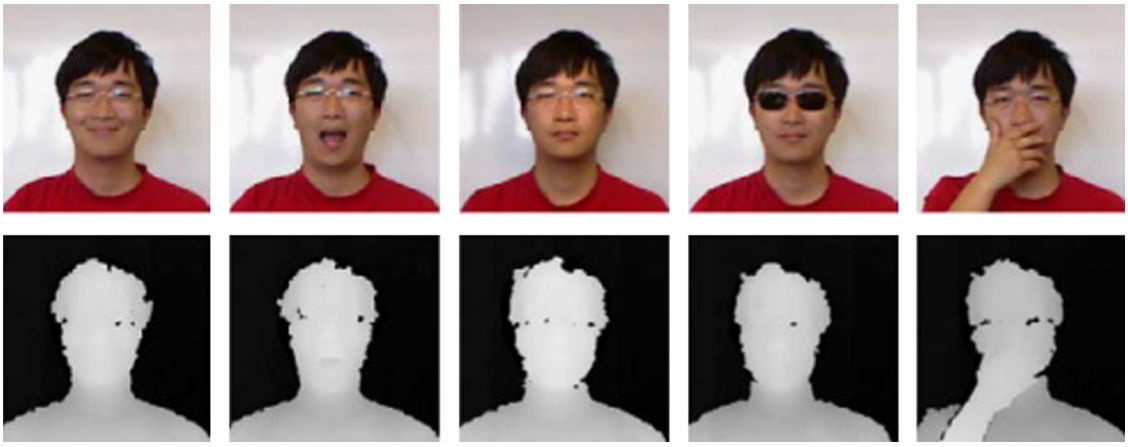

**Fig. 10.** Samples in the EURECOM Kinect database [14]. The first row shows the RGB images. The second row demonstrates the depth maps aligned with above RGB images.

individuals. In our experiment, the facial portion of each original image is automatically cropped according to the eyes' location and the cropped images are resized to 32 by 32 pixels (some sampled examples are illustrated in Fig. 11). To compare different algorithms, a subset with 5 images per individual is randomly selected to form a training set, and the rest is used for tests. Table 3 demonstrates the recognition rates of the proposed method and other competing algorithms, including NN, NS [18], SR [36], CR [45], CR(LBP), RCR [41], RRSC [40], and KSR(LBP+HK) [16]. It can be seen from this table that the proposed KCR-$\ell_2$(LBP+HK) method achieves the best performance in terms of accuracy. In addition, we use Table 4 to validate the residue (Eq. (3)) and regularized residue (Eq. (4)) manners in the KCR framework, in which the notions with "-0" and without "-0" denote the methods with residues and with regularized residues respectively. The results in Table 4 demonstrate the effectiveness of the regularized residue manner. By comparing the 4th and last columns, it also can be seen that the proposed KCR-$\ell_2$(LBP+HK) method outperforms the method presented in [42].

### 4.4. Kernel choice and combination

It can be from Fig. 1(a) that the KCR methods with the LBP features and Hamming kernel achieve more accurate results than the traditional methods with linear kernel functions on the extend Yale B dataset. The underlying reason is that the LBP codes with Hamming kernel could capture sufficient image structures under various illumination. However, the LBP codes with Hamming kernel do not work well on the PolyU Hyperspectral (HS) database (shown in Fig. 7 (a)), the possible reason is that the image noise limits the performance of the LBP features in this dataset. Thus, a natural question ensues: can we choose various kernels for different databases or combine different kernels to achieve a good performance? Motivated by the multiple kernel learning (MKL) frameworks [27], we attempt to combine (or fuse) different kernel functions in an additive manner,

$$K = \sum_{i=1}^{N} w^i K^i$$
$$\text{s.t.} \quad w^i > 0,\ i = 1, 2, \ldots, N$$
$$\sum_{i=1}^{N} w^i = 1 \tag{16}$$

where $K$ denotes the combined kernel function (for both $K_{\mathbf{AA}}$ and $K_{\mathbf{A}}(\mathbf{y})$), and $K^i$ is the $i$th individual kernel function. In this work, it is difficult to learn the weights of different kernel functions within an optimization framework (like [27]) because the discussed representation-based methods are typically non-parametric models. Thus, we adopt the cross validation (CV) technique to determine the optimal weights for classification, i.e., to choose weights that achieve the best CV accuracy (the leave-one-out CV method is adopted in this work).

Here, we adopt the KCR-$\ell_2$ algorithm to investigate the kernel combination problem for its effectiveness and efficiency. To be

**Fig. 11.** Samples in the FERET database [26].

**Table 3**
The comparisons of different algorithms on the FERET database.

| Algorithm | NN | NS [18] | SR [36] | CR [45] | RCR [41] | RRSC [40] | CR [45] (LBP) | KSR [16] (LBP+HK) | KCR-$\ell_2$ (LBP+HK) |
|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 69.5 | 75.0 | 84.0 | 78.0 | 82.5 | 81.3 | 36.3 | 83.0 | 88.3 |

**Table 4**
The comparisons of the residue and regularized residue for different methods.

| Algorithm | CR-0 [45] | CR [45] | KCR-0 [42] (Guassian) | KCR [42] (Guassian) | KCR-$\ell_2$-0 (LBP+HK) | KCR-$\ell_2$ (LBP+HK) |
|---|---|---|---|---|---|---|
| Accuracy | 68.3 | 78.0 | 77.0 | 84.5 | 83.0 | 88.3 |

**Table 5**
Cross Validation (CV) for additive kernel combination.

| $w$ | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| (a) Extended Yale B database (5 training samples per class) | | | | | | | | | | | |
| CV accuracy (%) | **96.3** | 95.8 | 95.3 | 91.6 | 90.0 | 86.3 | 83.2 | 80.5 | 77.9 | 74.7 | 72.6 |
| CV score | **0.1734** | 0.1925 | 0.2190 | 0.2410 | 0.2588 | 0.2757 | 0.2835 | 0.2872 | 0.2927 | 0.2894 | 0.3101 |
| Test accuracy (%) | **97.8** | 97.2 | 96.6 | 95.7 | 94.6 | 92.6 | 91.2 | 89.2 | 87.1 | 85.0 | 83.2 |
| (b) Extended Yale B database (40 training samples per class) | | | | | | | | | | | |
| CV accuracy (%) | 99.8 | 99.8 | 99.8 | 99.8 | **99.9** | **99.9** | **99.9** | 99.8 | 99.7 | 99.7 | 98.8 |
| CV score | 0.0479 | 0.0443 | 0.0423 | 0.0412 | 0.0402 | 0.0388 | 0.0378 | 0.0372 | **0.0368** | 0.0382 | 0.0543 |
| Test accuracy (%) | 99.4 | 99.4 | 99.4 | 99.6 | 99.8 | **99.9** | 99.8 | **99.9** | **99.9** | **99.9** | 98.8 |
| (c) PolyU Hyperspectral (HS) database (2 training samples per class) | | | | | | | | | | | |
| CV accuracy (%) | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| CV score | 0.0036 | 0.0034 | 0.0032 | 0.0031 | 0.0029 | 0.0027 | 0.0025 | 0.0023 | 0.0021 | **0.0019** | 0.0036 |
| Test accuracy (%) | 93.0 | 93.5 | 93.6 | 93.9 | 94.2 | 94.5 | 94.8 | 94.9 | 95.1 | **95.3** | 93.5 |
| (d) PolyU Hyperspectral (HS) database (10 training samples per class) | | | | | | | | | | | |
| CV accuracy (%) | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| CV score | 0.0051 | 0.0047 | 0.0042 | 0.0038 | 0.0033 | 0.0029 | 0.0024 | 0.0020 | 0.0015 | 0.0010 | **0.0009** |
| Test accuracy (%) | 96.2 | 96.4 | 96.4 | 96.4 | 96.4 | 96.7 | 96.7 | 96.5 | **97.2** | **97.2** | 97.0 |

specific, the linear kernel with raw pixels and the hamming kernel with LBP codes are combined in this work, i.e., $K = wK^{Linear} + (1-w)K^{LBPHK}$, $0 \le w \le 1$ ($w^1 = w, w^2 = 1-w$ for considering the constraint $w^1 + w^2 = 1$). Table 5 report the CV accuracy and the test accuracy with different weights and varied training samples on the extended Yale B and PloyU HS datasets. However, from this table, we can see that the CV accuracy rule cannot guarantee to select a satisfying weight (e.g., Table 5(b)–(d)). The underlying reason is that the representation-based methods intend to represent the training sample very well and lead to a very high CV accuracy.

Therefore, to conduct an effective CV in this work, we introduce a classification score, which is defined as

$$s(l) = r^l_{i*} \Big/ \left[ \min_{j=1,2,..,k \text{ and } j \neq i^*} (r^l_i) \right], \tag{17}$$

where $s(l)$ denotes the score of $l$th sample, $r^l_i$ is the regularized residue of the $i$th class for sample $l$[1], $i^*$ stands for the optimal label (in the CV process, the optimal label of the test sample is known in advance). The initiative explanation of Eq. (17) is the regularized residue of the optimal label should be as small as possible and the second smallest regularized residue should be as large as possible.

Thus, we can adopt the average score $s = (1/L)\sum_{l=1}^{L} s(l)$ to evaluate different models (or parameters), which $L$ stands for the test number in the CV phase. The CV scores are also reported in Table 5, in which a smaller score means a better parameter. Compared with the test accuracies, we can find that the CV score

[1] $r_i = \frac{\kappa(\mathbf{y},\mathbf{y}) - 2\mathbf{x}_i^T K_{\mathbf{A}_i}(\mathbf{y}) + \mathbf{x}_i^T K_{\mathbf{A}_i\mathbf{A}_i}\mathbf{x}_i}{\|\mathbf{x}_i\|_2^2}$.

**Table 6**
Kernel combination on Extended Yale B database. EWC: Equal Weight Combination; MKL [27]: Multiple Kernel Learning; AWC: Adaptive Weight Combination.

| Training number | Linear | LBP+HK | EWC | MKL | AWC |
|---|---|---|---|---|---|
| 5 | 83.2 | **97.8** | 94.6 | 85.3 | **97.8** ($w=0$) |
| 10 | 92.8 | **99.7** | 98.5 | 95.1 | **99.7** ($w=0$) |
| 20 | 96.2 | **99.9** | 99.8 | 98.9 | **99.9** ($w=0$) |
| 30 | 98.3 | **99.8** | **99.8** | 99.7 | **99.8** ($w=0$) |
| 40 | 98.8 | 99.4 | **99.9** | 99.7 | **99.9** ($w=0.8$) |
| 50 | 98.8 | 99.8 | 99.6 | 99.8 | **100** ($w=0.8$) |

**Table 7**
Kernel combination on PolyU Hyperspectral (HS) database. EWC: Equal Weight Combination; MKL [27]: Multiple Kernel Learning; AWC: Adaptive Weight Combination.

| Training number | Linear | LBP+HK | EWC | MKL | AWC |
|---|---|---|---|---|---|
| 2 | 93.5 | 93.3 | 94.5 | 94.4 | **95.3** ($w=0.9$) |
| 4 | **94.7** | 93.2 | 94.0 | 94.2 | **94.7** ($w=1.0$) |
| 6 | 94.9 | 94.1 | 94.6 | 95.0 | **95.6** ($w=0.9$) |
| 8 | **96.4** | 94.8 | 95.0 | 95.1 | **96.4** ($w=1.0$) |
| 10 | 97.0 | 96.2 | 97.0 | 96.5 | **97.0** ($w=1.0$) |
| 12 | 97.7 | **98.4** | 98.0 | 98.0 | **98.4** ($w=1.0$) |

rule is able to select a good parameter under different conditions (although the smallest CV score is not corresponding to the best test accuracy in Table 5(d), it also achieves a not bad choice).

Finally, we compare the proposed kernel combination methods with individual kernel methods on the extended Yale B and PolyU HS databases and report the results in Tables 6 and 7. It can be seen from these tables that the adaptive weight combination scheme achieves better (or not worse) performance than individual kernel methods and the equal weight combination scheme.

## 5. Conclusion and further work

In this paper, we present a unified framework based on kernel collaborative representation for linear and non-linear schemes. The framework provides insights of the relationships among several effective representation schemes, and facilitates the designing of new algorithms by choosing kernel functions, regularizations, and/or additional constraints. Within the proposed framework, we design a simple yet effective algorithm by using a squared $\ell_2$-regularization and apply it to face recognition with the LBP features and Hamming kernel. Numerous experiments on the extended Yale B, AR, large-scale Multi-PIE, PloyU NIR, PloyU HS, EURECOM Kinect and FERET face databases show that our algorithm performs favorably against state-of-the-art methods in terms of accuracy and speed, especially when the training set is small and faces are occluded. In addition, an adaptive weight combination scheme is proposed to combine different kernel functions. The experimental results demonstrate that the proposed combination method achieve better (or not worse) performance than individual methods. Our future work will focus on the optimal combination of features, kernel functions, regularizations, and constraints within the proposed framework for face recognition and other recognition problems.

## Conflict of interest

None declared.

## Acknowledgement

This work was supported by the China Postdoctoral Science Foundation under Grant 2014M551085, the Fundamental Research Funds for the Central Universities under Grant DUT14YQ101 and DUT13RC(3)105, the Natural Science Foundation of China (NSFC) under Grants 61472060, the joint Foundation of China Education Ministry, and by the China Mobile Communication Corporation under Grant MCM20122071.

## References

[1] T. Ahonen, A. Hadid, M. Pietikäinen, Face recognition with local binary patterns, in: Proceedings of European Conference on Computer Vision, Prague, Czech Republic, 2004, pp. 469–481.
[2] S. An, W. Liu, S. Venkatesh, Face recognition using kernel ridge regression, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Rio de Janeiro, Brazil, 2007, pp. 1–7.
[3] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Comput. 12 (10) (2000) 2385–2404.
[4] D. Cai, X. He, J. Han, Spectral regression for efficient regularized subspace learning, in: Proceedings of the IEEE International Conference on Computer Vision, Minneapolis, Minnesota, USA, 2007.
[5] W. Deng, J. hu, J. Guo, In defense of sparsity based face recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 2013, pp. 399–406.
[6] W. Di, L. Zhang, D. Zhang, Q. Pan, Studies on hyperspectral face recognition in visible spectrum with feature band selection, IEEE Trans. Syst. Man Cybern. Part A 40 (6) (2010) 1354–1361.
[7] K. Etemad, R. Chellappa, Discriminant analysis for recognition of human face images, J. Opt. Soc. Am. 14 (1) (1997) 1724–1733.
[8] J. Friedman, T. Hastie, R. Tibshirani, A note on the group lasso and a sparse group lasso, http://arXiv:1001.0736 [math.ST], 2010.
[9] S. Gao, I.W.-H. Tsang, L.-T. Chia, Sparse representation with kernels, IEEE Trans. Image Process. 22 (2) (2013) 423–434.
[10] A.S. Georghiades, P.N. Belhumeur, D.J. 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.
[11] R. Gross, I. Matthews, J.F. Cohn, T. Kanade, S. Baker, Multi-pie, Image Vis. Comput. 28 (5) (2010) 807–813.
[12] X. He, S. Yan, Y. Hu, P. Niyogi, H. Zhang, Face recognition using laplacian faces, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 328–340.
[13] G.B. Huang, M. Ramesh, T. Berg, E. Learned-miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Technical Report 07-49, 2007.
[14] T. Huynh, R. Min, J.-L. Dugelay, An efficient LBP-based descriptor for facial depth images applied to gender recognition using rgb-d face data, in: ACCV Workshop on Computer Vision with Local Binary Pattern Variants, Daejeon, Korea, 2012.
[15] K. Jia, T.-H. Chan, Y. Ma, Robust and practical face recognition via structured sparsity, in: Proceedings of European Conference on Computer Vision, Florence, Italy, 2012, pp. 331–344.
[16] C. Kang, S. Liao, S. Xiang, C. Pan, Kernel sparse representation with local patterns for face recognition, in: Proceedings of IEEE International Conference on Image Processing, Brussels, Belgium, 2011, pp. 3009–3012.
[17] S.J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, A interior-point method for large-scale $\ell_1$-regularized least squares, IEEE J. Sel. Top. Signal Process. 1 (4) (2007) 606–617.
[18] K.-C. Lee, J. Ho, D.J. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 684–698.
[19] Z. Lei, D. Yi, S.Z. Li, Robust and practical face recognition via structured sparsity, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Florence, Italy, 2012, pp. 2512–2517.
[20] S.Z. Li, R. Chu, S. Liao, L. Zhang, Illumination invariant face recognition using near-infrared images, IEEE Trans. Pattern Anal. Mach. Intell. 29 (4) (2007) 627–639.
[21] A. Majumdar, R.K. Ward, Classification via group sparsity promoting regularization, in: Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing, Taipei, Taiwan, 2009, pp. 861–864.
[22] A. Martinez, R. Benavente, The ar Face Database, CVC Technical Report, 24 June 1998.
[23] X. Mei, H. Ling, Robust visual tracking using $\ell_1$ minimization, in: Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 2009, pp. 1436–1443.
[24] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[25] P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss, The FERET evaluation methodology for face-recognition algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 22 (10) (2000) 1090–1104.

[26] P.J. Phillips, H. Wechsler, J. Huang, P.J. Rauss, The FERET database and evaluation procedure for face-recognition algorithms, Image Vis. Comput. 16 (5) (1998) 295–306.
[27] A. Rakotomamonjy, F. Bach, S. Canu, Y. Grandvalet, Simplemkl, J. Mach. Learn. Res. 9 (1) (2008) 2491–2521.
[28] B. Schölkopf, A.J. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.
[29] Q. Shi, A. Eriksson, A. van den Hengel, C. Shen, Is face recognition really a compressive sensing problem? in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 2011, pp. 553–560.
[30] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cognit. Neurosci. 3 (1) (1991) 71–86.
[31] D. Wang, H.L.M.-H. Yang, Online object tracking with sparse prototypes, IEEE Trans. Image Process. 22 (1) (2013) 314–325.
[32] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010, pp. 3360–3367.
[33] Z. Wang, W. Yang, J. Yin, C. Sun, Kernel collaborative representation with regularized least square for face recognition, in: Chinese Conference on Biometric Recognition, Jinan, China, 2013, pp. 130–137.
[34] L. Wiskott, J.-M. Fellous, N. Krüger, C. von der Malsburg, Face recognition by elastic bunch graph matching, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 775–779.
[35] J. Wright, Y. Ma, J. Maral, G. Sapiro, T. Huang, S. Yan, Sparse representation for computer vision and pattern recognition, Proc. IEEE 98 (6) (2010) 1031–1044.
[36] 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.
[37] D. Xu, S. Lin, S. Yan, X. Tang, Rank-one projections with adaptive margins for face recognition, IEEE Trans. Syst. Man Cybern. Part B 37 (5) (2007) 1226–1236.
[38] J. Yang, J. Wright, T.S. Huang, Y. Ma, Image super-resolution via sparse representation, IEEE Trans. Image Process. 19 (11) (2010) 2861–2873.
[39] M. Yang, L. Zhang, Gabor feature based sparse representation for face recognition with Gabor occlusion dictionary, in: Proceedings of European Conference on Computer Vision, Heraklion, Crete, Greece, 2010, pp. 448–461.
[40] M. Yang, L. Zhang, J. Yang, D. Zhang, Regularized robust coding for face recognition, IEEE Trans. Image Process. 22 (5) (2013) 1753–1766.
[41] M. Yang, L. Zhang, D. Zhang, S. Wang, Relaxed collaborative representation for pattern classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 2224–2231.
[42] W. Yang, Z. Wang, J. Yin, C. Sun, K. Ricanek, Image classification using kernel collaborative representation with regularized least square, Appl. Math. Comput. 222 (2013) 13–28.
[43] B. Zhang, S. Shan, X. Chen, W. Gao, Histogram of Gabor phase patterns (hgpp): a novel object representation approach for face recognition, IEEE Trans. Image Process. 16 (1) (2007) 57–68.
[44] B. Zhang, L. Zhang, D. Zhang, L. Shen, Directional binary code with application to PolyU near-infrared face database, Pattern Recognit. Lett. 31 (14) (2010) 2337–2344.
[45] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: which helps face recognition? in: Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 471–478.
[46] Q. Zhang, B. Li, Mining discriminative components with low-rank and sparsity constraints for face recognition, in: International Conference on Knowledge Discovery and Data Mining, Beijing, China, 2012, pp. 1469–1477.
[47] R. Zhi, M. Flierl, Q. Ruan, W.B. Kleijn, Graph-preserving sparse nonnegative matrix factorization with application to facial expression recognition, IEEE Trans. Syst. Man Cybern. Part B 41 (1) (2011) 38–52.
[48] Z. Zhou, A. Wagner, H. Mobahi, J. Wright, Y. Ma, Face recognition with contiguous occlusion using Markov random fields, in: Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan, 2009, pp. 1050–1057.
[49] L. Zhuang, A.Y. Yang, Z. Zhou, S.S. Sastry, Y. Ma, Single-sample face recognition with image corruption and misalignment, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 2013, pp. 3546–3553.
[50] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. Ser. B 67 (2005) 301–320.
[51] X. Zou, J. Kittler, K. Messer, Face recognition using active near-ir illumination, in: Proceedings of British Machine Vision Conference, Oxford, UK, 2005.

**Dong Wang** received the B.E. degree in electronic information engineering and the Ph.D. degree in signal and information processing from the Dalian University of Technology (DUT), Dalian, China, in 2008 and 2013 respectively, where he is currently a faculty with the School of Information and Communication Engineering. His research interests include face recognition, interactive image segmentation, and object tracking.

**Huchuan Lu** received the Ph.D. degree in system engineering and the M.Sc. degree in signal and information processing from the Dalian University of Technology (DUT), Dalian, China, in 2008 and 1998, respectively, where he joined the faculty in 1998 and is currently a Full Professor with the School of Information and Communication Engineering. His current research interests include the areas of computer vision and pattern recognition with focus on visual tracking, saliency detection, and segmentation. He is a member of the ACM and an associate editor of the IEEE T-SMC PART:B.

**Ming-Hsuan Yang** is an assistant professor in Electrical Engineering and Computer Science at University of California, Merced. He received the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2000. He was a senior research scientist at the Honda Research Institute working on vision problems related to humanoid robots. In 1999, he received the Ray Ozzie fellowship for his research work. He coauthored the book Face Detection and Gesture Recognition for Human–Computer Interaction (Kluwer Academic 2001) and edited special issue on face recognition for Computer Vision and Image Understanding in 2003. He serves as an area chair for the IEEE Conference on Computer Vision and Pattern Recognition in 2008 and 2009, as a publication chair in 2010, and an area chair for the Asian Conference on Computer in 2009 and 2010. He is an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence, and Image and Vision Computing. He is a senior member of the IEEE and the ACM.