Prior Knowledge-Based Probabilistic Collaborative Representation for Visual Recognition

Rushi Lan, Yicong Zhou, *Senior Member, IEEE*, Zhenbing Liu, and Xiaonan Luo

***Abstract*—Collaborative representation is an effective way to design classifiers for many practical applications. In this paper, we propose a novel classifier, called the prior knowledge-based probabilistic collaborative representation-based classifier (PKPCRC), for visual recognition. Compared with existing classifiers which use the collaborative representation strategy, the proposed PKPCRC further includes characteristics of training samples of each class as prior knowledge. Four types of prior knowledge are developed from the perspectives of image distance and representation capacity. They adaptively accommodate the contribution of each class and result in an accurate representation to classify a query sample. Experiments and comparisons on four challenging databases demonstrate that PKPCRC outperforms several state-of-the-art classifiers.**

***Index Terms*—Collaborative representation, prior knowledge, representation-based classifier, visual recognition.**

I. Introduction

THE CLASSIFIER targets determining one (or multiple) class label(s) of a query sample and plays a crucial role in many visual recognition tasks, ranging from face recognition [1], [2]; texture classification [3], [4]; landmark image retrieval [5]; to image categorization [6]; hyperspectral image classification [7], [8]; medical image retrieval [9]; and many others. As a long-standing research topic, a huge effort has been undertaken to develop all kinds of classifiers so far.

To design a classifier, an intuitive strategy lies in that the query sample should have the identical label with its closest one in the database. Following this idea, we can obtain the representative nearest neighbor (NN) classifier [10], which is easy to conduct with low computational cost yet is noise-sensitive. Nearest centroid (NC) and k-NN are two simple variations of NN that are robust to noise. More in-depth improvements of NN have been proposed, such as nearest feature line [11], nearest feature space [12], and neighborhood feature line segment [13]. These methods have shown satisfactory performance in various applications.

Because the nature images can be sparsely represented by structural primitives [14], Wright *et al.* [15] proposed a method named a sparse representation-based classifier (SRC) for face recognition. SRC sparsely codes a query face image over the template ones and classifies it by the least coding error. Apart from face recognition, SRC has been successfully applied to other visual recognition tasks. Many improved versions of SRC have been developed. Representative ones include locality weighted SRCs [16], quaternionic SRCs [17], and manifold-based SRCs [18].

After an extensive study of SRC, Zhang *et al.* [19] further developed a collaborative representation-based classifier (CRC) for face recognition. Unlike SRC, CRC determines the label in terms of collaborative representation with regularized least square, resulting in less complexity but completive performance. SRC and CRC are both regarded as typical classifiers using the representation-based strategy. Numerous efforts have been devoted to improving CRC from diverse aspects, such as the two-phase strategy [20], multiscale adaptive version [21], coarse-to-fine representation [22], quaternionic extension [17], and many others.

Recently, a probabilistic CRC (ProCRC) was developed by Cai *et al.* [23] from the views of representation and probability. By considering the representation coefficients of each class, ProCRC achieves a more accurate representation of the test sample than CRC. The success of ProCRC indicates that it is possible to improve the representation-based classifiers by refining the relationship between the query image and training samples of each class. The problem now turns out to find a useful relationship from the training samples.

In this paper, we propose a novel method, called prior knowledge-based probabilistic CRC (PKPCRC), for visual recognition. PKPCRC extracts the prior knowledge of each class from the training set and then couples the obtained prior knowledge when deriving the probability that the query image belongs to a specific class. In this way, compared with CRC and ProCRC, PKPCRC is more flexible and takes account of more characteristics of the training samples. As a result, it is

Manuscript received May 26, 2018; revised August 19, 2018 and October 23, 2018; accepted November 1, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61702129, Grant 61772149, Grant U1701267, and Grant 61320106008, in part by the Guangxi Colleges and Universities Key Laboratory of Intelligent Processing of Computer Images and Graphics under Grant GIIP201703, in part by the Macau Science and Technology Development Fund under Grant FDCT/189/2017/A3, and in part by the Research Committee at University of Macau under Grant MYRG2016-00123-FST and Grant MYRG2018-00136-FST. This paper was recommended by Associate Editor H. Lu. *(Corresponding author: Yicong Zhou.)*

R. Lan is with the Guangxi Key Laboratory of Intelligent Processing of Computer Image and Graphics, Guilin University of Electronic Technology, Guilin 541004, China, and also with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: rslan2012@gmail.com).

Y. Zhou is with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@umac.mo).

Z. Liu and X. Luo are with the Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China.

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

Digital Object Identifier 10.1109/TCYB.2018.2880290

able to provide an accurate representation of the query sample for classification. Experimental results also demonstrate the effectiveness of PKPCRC. In summary, the main contributions of this paper are listed as follows.

1) PKPCRC is proposed as a novel classifier for visual recognition. It further considers the prior knowledge extracted from the training samples. The closed form of PKPCRC is also provided.
2) We present four ways for PKPCRC to derive the prior knowledge from the training images. They are derived from the perspectives of image distance and representation capacity, respectively.
3) Experiments are conducted to evaluate the effectiveness of PKPCRC using four benchmark databases. The comparison results indicate that the proposed PKPCRC achieves state-of-the-art performance.

This paper is an improved and extended version of our previous work [24]. In this paper, we provide a detailed derivation and in-depth analysis of the proposed method, especially the derivation of prior knowledge. Besides, competing results on more benchmark databases and time complexity comparison are given to comprehensively evaluate PKPCRC.

The remaining portion of this paper is organized as follows. Section II briefly reviews some related work. Section III presents the proposed PKPCRC in detail, and Section IV gives the ways to derive prior knowledge for PKPCRC. Subsequently, Section V provides several experimental results to evaluate PKPCRC. Finally, Section VI concludes this paper.

II. Related Works

In this section, we briefly review some related work as background knowledge. Some commonly used notations are first provided. We represent each sample by a column vector. For a visual recognition task with C categories, all training samples form a matrix $\boldsymbol{A}$, i.e., $\boldsymbol{A} = [\boldsymbol{A}_1, \boldsymbol{A}_2, \dots, \boldsymbol{A}_C]$, where $\boldsymbol{A}_c$ is the data matrix of the cth class. The label set of $\boldsymbol{A}$ is denoted as $l_{\boldsymbol{A}}$. Spanning all elements of $\boldsymbol{A}$, we can achieve a linear subspace $\mathcal{A}$. The corresponding subspace of each class $\mathcal{A}_c$ is obtained similarly. Note that any element $\boldsymbol{a} \in \mathcal{A}$ can be linearly represented as $\boldsymbol{a} = \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{x}$ is the vector containing the representation coefficients. Considering a test sample $\boldsymbol{y}$, our goal is to determine its label $l(\boldsymbol{y})$ from $l_{\boldsymbol{A}}$.

A. NN and NC

As aforementioned, the NN classifier is to find a sample $\boldsymbol{a}'$ from the training set $\boldsymbol{A}$ that has the smallest distance to $\boldsymbol{y}$ as follows:

$$\boldsymbol{a}' = \arg\min_{\boldsymbol{a}\in\boldsymbol{A}} \ \|\boldsymbol{y} - \boldsymbol{a}\|_2 \tag{1}$$

and then to assign the label of $\boldsymbol{y}$ to that of $\boldsymbol{a}'$, i.e., $l(\boldsymbol{y}) = l(\boldsymbol{a}')$. The NN classifier is the most simple one. However, it is sensitive to noise because of the lack of any training procedure.

To improve the performance of the NN classifier, the NC classifier uses the centroid to represent each class, denoting the centroid of $\boldsymbol{A}_c$ by $\bar{\boldsymbol{a}}^c_{\text{tr}}$, and then conducts the NN classifier to $\boldsymbol{y}$ by

$$l(\boldsymbol{y}) = \arg\min_{c=1,\dots,C} \ \left\|\boldsymbol{y} - \bar{\boldsymbol{a}}^c_{\text{tr}}\right\|_2. \tag{2}$$

It can be seen that the derivation of $\bar{\boldsymbol{a}}^c_{\text{tr}}$ is able to remove some noise; hence, the NC classifier is more robust to noise than the NN classifier.

B. SRC

Unlike NN and NC classifiers, SRC determines the label of a test sample by representing it using all training samples with a sparsity constraint. SRC can be mathematically described as follows:

$$\boldsymbol{x}' = \arg\min_{\boldsymbol{x}} \ \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \tag{3}$$

where λ denotes the regularization parameter. We then calculate the residual of $\boldsymbol{y}$ and its sparse representation using $\boldsymbol{A}_c$ by

$$r_c(\boldsymbol{y}) = \left\|\boldsymbol{y} - \boldsymbol{A}_c\boldsymbol{x}'_c\right\|_2 \tag{4}$$

where $\boldsymbol{x}'_c$ is the corresponding coefficient vector associated with the cth class in $\boldsymbol{x}'$. The label of $\boldsymbol{y}$ is finally assigned by

$$l(\boldsymbol{y}) = \arg\min_{c} \ r_c(\boldsymbol{y}). \tag{5}$$

C. LRC

LRC treats the classification of $\boldsymbol{y}$ in terms of linear regression. More specifically, it is a linear model that represents $\boldsymbol{y}$ via a linear combination of the samples of the cth class. That is,

$$\boldsymbol{x}'_c = \arg\min_{\boldsymbol{x}_c} \ \|\boldsymbol{y} - \boldsymbol{A}_c\boldsymbol{x}_c\|_2^2. \tag{6}$$

Based on $\boldsymbol{x}'_c$ in (6), the distance between $\boldsymbol{y}$ and the represented one is computed as

$$d_c(\boldsymbol{y}) = \left\|\boldsymbol{y} - \boldsymbol{A}_c\boldsymbol{x}'_c\right\|_2. \tag{7}$$

Finally, the label of $\boldsymbol{y}$ can be obtained by

$$l(\boldsymbol{y}) = \arg\min_{c} \ d_c(\boldsymbol{y}). \tag{8}$$

D. CRC and ProCRC

CRC is a simple yet effective method that collaboratively represents the test sample $\boldsymbol{y}$ by all training samples in $\boldsymbol{A}$ [19]. Unlike the sparsity constraint used in SRC, the least square constraint is utilized in CRC; hence, we can easily achieve its closed-form solution. The CRC is mathematically modeled as

$$\boldsymbol{x}' = \arg\min_{\boldsymbol{x}} \ \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_2^2. \tag{9}$$

Once the representation coefficient $\boldsymbol{x}'$ is obtained, we can compute the residual and determine the label as (4) and (5) in the same way.

Apart from the view of collaborative representation, CRC is explained from a probabilistic perspective [23]. Rather than directly determining $l(\boldsymbol{y})$, the probability that $l(\boldsymbol{y})$ belongs to $l_{\boldsymbol{A}}$ is considered as

$$p(l(\boldsymbol{y}) \in l_A) \propto \exp\Big(-\Big(\kappa\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \upsilon\|\boldsymbol{x}\|_2^2\Big)\Big) \tag{10}$$

where κ and υ are two constants.

To maximize $p(l(\boldsymbol{y}) \in l_{\boldsymbol{A}})$, we conduct the logarithmic operator to it and yield the following result:

$$\begin{aligned} \max\ p(l(\boldsymbol{y}) \in l_{\boldsymbol{A}}) &= \max\ \ln p(l(\boldsymbol{y}) \in l_{\boldsymbol{A}}) \\ &= \min_{\boldsymbol{x}}\ \kappa \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \upsilon \|\boldsymbol{x}\|_2^2 \\ &= \min_{\boldsymbol{x}}\ \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_2^2 \end{aligned} \tag{11}$$

where $\lambda = \upsilon/\kappa$. As seen, (11) is equivalent to (9), but they are derived from different perspectives, namely, probabilistic interpretation and collaborative representation.

Equation (11) describes the probability that $l(\boldsymbol{y})$ belongs to $l_{\boldsymbol{A}}$. To improve CRC, Cai *et al.* [23] investigated the joint probability of the test sample, i.e., $p(l(\boldsymbol{y}) = 1, \ldots, l(\boldsymbol{y}) = C)$, and they finally derived the following ProCRC model:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left\{ \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_2^2 + \frac{\gamma}{C} \sum_{c=1}^{C} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}_c\boldsymbol{x}_c\|_2^2 \right\} \tag{12}$$

where γ is a constant. The joint probability refines the relation between $\boldsymbol{y}$ and each $\boldsymbol{A}_c$, significantly improving the performance of CRC.

III. Proposed Approach

This section presents the proposed PKPCRC algorithm in detail. We first explain the motivation of PKPCRC and then give its mathematical model. The optimization and classification rule of PKPCRC are finally introduced.

A. Motivation

Observing (12), it can be seen that the success of ProCRC is attributed to the third term $\sum_{c=1}^{C} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}_c\boldsymbol{x}_c\|_2^2$. It describes the distance between two points, i.e., $\boldsymbol{A}\boldsymbol{x} \in \mathcal{A}$ and $\boldsymbol{A}_c\boldsymbol{x}_c \in \mathcal{A}_c$. ProCRC attempts to find $\boldsymbol{x}$ to minimize the total distances between $\boldsymbol{A}\boldsymbol{x}$ and $\{\boldsymbol{A}_1\boldsymbol{x}_1, \ldots, \boldsymbol{A}_C\boldsymbol{x}_C\}$. On the other hand, we can rewrite $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}_c\boldsymbol{x}_c\|_2^2$ as follows:

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}_c\boldsymbol{x}_c\|_2^2 = \|\boldsymbol{A}\boldsymbol{x}\|_2^2 + \|\boldsymbol{A}_c\boldsymbol{x}_c\|_2^2 - 2(\boldsymbol{A}\boldsymbol{x})^T(\boldsymbol{A}_c\boldsymbol{x}_c). \tag{13}$$

From the above formula, it can be seen that ProCRC not only minimizes $\|\boldsymbol{A}\boldsymbol{x}\|_2^2$ and $\|\boldsymbol{A}_c\boldsymbol{x}_c\|_2^2$ but also maximizes their correlations via the inner product. Compared with CRC, ProCRC further includes the intra-actions among $\boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{A}_c\boldsymbol{x}_c$, providing a better representation for classification.

$\boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{A}_c\boldsymbol{x}_c$ denote the representations of the test sample using all training samples and those of the cth class. Note that with a given training set, there must be some inherent relations between $\boldsymbol{A}$ and $\boldsymbol{A}_c$. To verify this observation, we use all 40 male images from the AR face database [25] to form a training set to study this. For simplicity, we set $\boldsymbol{x}$ and $\boldsymbol{x}_c$ to the same values, resulting in the centroids of $\boldsymbol{A}$ and each $\boldsymbol{A}_c$, respectively. Next, we calculate the correlation coefficients between $\boldsymbol{A}\boldsymbol{x}$ and each $\boldsymbol{A}_c\boldsymbol{x}_c$. It is a normalized version of $(\boldsymbol{A}\boldsymbol{x})^T(\boldsymbol{A}_c\boldsymbol{x}_c)$ in (13). The corresponding results are depicted in Fig. 1. The maximum, minimum, and average values of these correlation coefficients are 0.9236, 0.5722, and 0.8477, respectively. However, these values cannot correctly describe the relations between $\boldsymbol{A}$ and $\boldsymbol{A}_c$ because there are large variants within the captured images, such as the wearing of classes, change of poses, and different facial expressions. Due to these effects, it is difficult to find an accurate representation for the test sample if the correlation coefficients are too large or too small. In this paper, the proposed method aims to improve the representation capacity by refining the relations between $\boldsymbol{A}$ and $\boldsymbol{A}_c$.

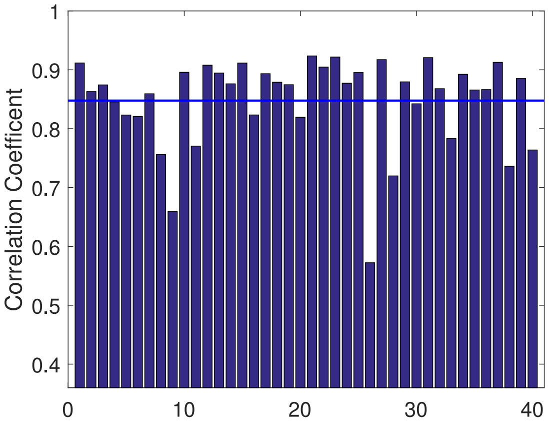


Fig. 1. Correlation coefficients between $\boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{A}_c\boldsymbol{x}_c$ with equal representation coefficients using the 40 male images in the AR face database.

B. PKPCRC Model

Assume that $\boldsymbol{a}_c = \boldsymbol{A}_c\boldsymbol{x}_c$ and $\boldsymbol{a} = \sum_{c=1}^{C} \boldsymbol{A}_c\boldsymbol{x}_c$ are two elements of $\mathcal{A}_c$ and $\mathcal{A}$, respectively. Similar to ProCRC, the proposed PKPCRC here considers the probability that $\boldsymbol{a}$ is with the identical label as $\boldsymbol{a}_c$, and defines this probability as

$$p\Big(l(\boldsymbol{a}) = c \Big| l(\boldsymbol{a}) \in l_{\boldsymbol{A}}\Big) \propto \exp\Big(-\delta\beta_c \|\boldsymbol{a} - \boldsymbol{A}_c\boldsymbol{x}_c\|_2^2\Big) \tag{14}$$

where δ is a constant and β_c is the prior knowledge. It should be noted that (14) will be equivalent to the probability used in ProCRC if we set the prior knowledge β_c to be a uniform distribution.

Equation (14) indicates that PKPCRC further takes account of the prior knowledge β_c in contrast to ProCRC, yielding the following merits.

1) Note that $\boldsymbol{a}$ is the sum of $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_C\}$. ProCRC treats each distance $\|\boldsymbol{a} - \boldsymbol{A}_c\boldsymbol{x}_c\|_2^2$ equally. Considering PKPCRC, β_c can be regarded as a weight to refine the distance between $\boldsymbol{a}$ and its cth component $\boldsymbol{a}_c$. In this way, PKPCRC handles each distance differently via β_c and is more flexible than ProCRC.
2) The prior knowledge β_c is extracted from the training samples; hence, more inherent characteristics of these samples are considered. We need to find a proper way to derive β_c that represents the relation between all training samples and those of the cth class. It is possible to obtain a more accurate representation of $\boldsymbol{y}$ with β_c.

Based on the above advantages, PKPCRC represents the test sample $\boldsymbol{y}$ collaboratively using the prior knowledge β_c and the probability in (14).

Denote the probability that the test sample $\boldsymbol{y}$ belongs to the cth class by $p(l(\boldsymbol{y}) = c)$. Using (14), $p(l(\boldsymbol{y}) = c)$ can be achieved as follows:

$$p(l(\boldsymbol{y})=c)=p(l(\boldsymbol{y})\in l_{\boldsymbol{A}})\cdot p(l(\boldsymbol{a})=c|l(\boldsymbol{a})\in l_{\boldsymbol{A}})$$
$$\propto \exp\Big(-\Big(\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2+\lambda\|\boldsymbol{x}\|_2^2+\gamma\beta_c\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{A}_k\boldsymbol{x}_k\|_2^2\Big)\Big) \tag{15}$$

where $\gamma=\delta/\kappa$. $l(\boldsymbol{y})$ can be determined by directly maximizing $p(l(\boldsymbol{y})=c)$ from all classes, but our experimental results indicate that PKPCRC, similar to ProCRC, cannot obtain a stable and discriminative representation of $\boldsymbol{y}$ in this way.

To further improve the representation capacity, PKPCRC also uses the joint probability rather than the probability that $\boldsymbol{y}$ belongs to the cth class only, yielding the following result:

$$\max\ p(l(\boldsymbol{y})=1,\ldots,l(\boldsymbol{y})=C)=\max\prod_c p(l(\boldsymbol{y})=c)$$
$$\propto \max\exp\Bigg(-\Bigg(\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2+\lambda\|\boldsymbol{x}\|_2^2 + \frac{\gamma}{C}\sum_{c=1}^{C}\beta_c\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{A}_c\boldsymbol{x}_c\|_2^2\Bigg)\Bigg). \tag{16}$$

We conduct the logarithmic operator to the above equation and ignore the constant term. Then, (16) transforms to

$$\hat{\boldsymbol{x}}=\arg\min_{\boldsymbol{x}}\left\{\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2+\lambda\|\boldsymbol{x}\|_2^2 + \frac{\gamma}{C}\sum_{c=1}^{C}\beta_c\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{A}_c\boldsymbol{x}_c\|_2^2\right\}. \tag{17}$$

In this paper, the obtained model in (17) is named PKPCRC. It improves ProCRC by further coupling a prior knowledge to represent the test sample.

C. Optimization and Classification Rule of PKPCRC

To optimize (17), we keep the cth component of $\boldsymbol{A}$ unchanged and set the rest components to the $\boldsymbol{0}$ matrix, yielding the following matrix $\boldsymbol{A}_c'=[\boldsymbol{0},\ldots,\boldsymbol{A}_c,\ldots,\boldsymbol{0}]$. Let $\bar{\boldsymbol{A}}_c'=\boldsymbol{A}-\bar{\boldsymbol{A}}_c'$. To achieve the closed-form solution of PKPCRC, we first derive the following projection matrix:

$$\boldsymbol{M}=\left(\boldsymbol{A}^T\boldsymbol{A}+\frac{\gamma}{C}\sum_{c=1}^{C}\beta_c\left(\bar{\boldsymbol{A}}_c'\right)^T\bar{\boldsymbol{A}}_c'+\lambda\boldsymbol{I}\right)^{-1}\boldsymbol{A}^T \tag{18}$$

where $\boldsymbol{I}$ is the identity matrix. Note that we can calculate the above projection matrix $\boldsymbol{M}$ offline to alleviate the computation cost. Considering the test sample $\boldsymbol{y}$, once the matrix $\boldsymbol{M}$ is available via (18), the solution of PKPCRC in (17) can be achieved by

$$\hat{\boldsymbol{x}}=\boldsymbol{M}\boldsymbol{y}. \tag{19}$$

Now, the label of $\boldsymbol{y}$ can be determined with $\hat{\boldsymbol{x}}$. More specifically, the probability that $\boldsymbol{y}$ belongs to the cth category can be represented as

$$p(l(\boldsymbol{y})=c)\propto\exp\Big(-\Big(\big\|\boldsymbol{y}-\boldsymbol{A}\hat{\boldsymbol{x}}\big\|_2^2+\lambda\big\|\hat{\boldsymbol{x}}\big\|_2^2 + \gamma\beta_c\big\|\boldsymbol{A}\hat{\boldsymbol{x}}-\boldsymbol{A}_k\hat{\boldsymbol{x}}_k\big\|_2^2\Big)\Big). \tag{20}$$

Then $\boldsymbol{y}$ will be classified to the class that accords to the maximal probability, namely,

$$l(\boldsymbol{y})=\arg\max_c\{p(l(\boldsymbol{y})=c)\}. \tag{21}$$

Algorithm 1 PKPCRC

```
Input: A set of training samples $A$, a test sample $y$, the prior information of each class $\{\beta_1, \cdots, \beta_C\}$, and the parameter $\lambda$ and $\gamma$.
Output: The label of $y$.
  (a) Calculate the projection matrix:
        $M = (A^T A + \frac{\gamma}{C}\sum_{c=1}^{C}\beta_c(\bar{A}_c')^T\bar{A}_c' + \lambda I)^{-1}A^T.$
  (b) Obtain the representation of $y$: $\hat{x} = My$.
  (c) Calculate the probability that $y$ belongs to each class:
        $p_c = \exp(-\|A\hat{x} - A_c\hat{x}_c\|_2^2).$
  (d) Predict the label: $l(y) = \arg\max_c\{p_c\}$.
```

Equations (20) and (21) provide one way to determine $l(\boldsymbol{y})$. However, we find that PKPCRC cannot achieve good classification results in this way. The reason is that β_c used in (20) causes a problem of overfitting the test sample. It can be seen from (18) that β_c has been used to derive the representation coefficients of $\boldsymbol{y}$. Based on the derived $\hat{\boldsymbol{x}}$, (20) calculates the probability using the prior knowledge for the second time. This operation will make the obtained probability to follow the distribution of the training samples, making adverse affects to the classification accuracy.

To address the aforementioned problem, β_c will not be considered to derive the probability in (20). For simplicity, we also omit the constant terms. Finally, the probability that $\boldsymbol{y}$ belongs to the cth class, denoted by p_c, is represented as

$$p_c=\exp\Big(-\big\|\boldsymbol{A}\hat{\boldsymbol{x}}-\boldsymbol{A}_c\hat{\boldsymbol{x}}_c\big\|_2^2\Big). \tag{22}$$

Then $l(\boldsymbol{y})$ is determined using the following rule:

$$l(\boldsymbol{y})=\arg\max_c\{p_c\}. \tag{23}$$

Algorithm 1 presents the detailed steps to implement the proposed PKPCRC.

IV. Strategies to Extract Prior Knowledge

The extraction of β_c from the training samples is the vital step for PKPCRC. Only the β_c that correctly reveals the relationship between all training samples and those of the cth class will improve the representation accuracy. In this section, we will provide four different strategies to extract β_c.

A. Distance-Based Prior Knowledge

As mentioned in Section III-A, the representation coefficients are expected to maximize the inner product between $\boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{A}_c\boldsymbol{x}_c$. It can be observed from Fig. 1 that the mean of one subject may have a too small or a too large inner product to the mean of all subjects. It is hard to obtain accurate

```
Algorithm 2 NC-Based Prior Knowledge
Input: A training set $\boldsymbol{A}$ containing $C$ class samples, namely $\boldsymbol{A} = [\boldsymbol{A}_1\ \boldsymbol{A}_c\ \cdots\ \boldsymbol{A}_C]$.
Output: The NC-based prior information of each class $\{\beta_1^1, \cdots, \beta_C^1\}$.
    (a) Calculate the mean vectors of $\boldsymbol{A}$ and $\{\boldsymbol{A}_c | c = 1, \cdots, C\}$ respectively, i.e., $\bar{\boldsymbol{a}}_{\text{tr}}$ and $\bar{\boldsymbol{a}}_{\text{tr}}^c$
    (b) For each $\bar{\boldsymbol{a}}_{\text{tr}}^c$, compute the distance between $\bar{\boldsymbol{a}}_{\text{tr}}$ and $\bar{\boldsymbol{a}}_{\text{tr}}^c$ as Eq. (24), $c = 1, \cdots, C$.
    (c) Derive the NC-based prior information $\{\beta_1^1, \ldots, \beta_C^1\}$ using Eq. (25).
```

representation coefficients for these subjects. To address this problem, we use the prior knowledge to adjust those subjects and propose the following prior knowledge methods.

1) NC-Based Prior Knowledge: Denote the mean vector of all training samples of $\boldsymbol{A}$ by $\bar{\boldsymbol{a}}_{\text{tr}}$, and $\bar{\boldsymbol{a}}_{\text{tr}}^c$ represents the mean vector of those samples of the cth class. We calculate the following distance between $\bar{\boldsymbol{a}}_{\text{tr}}$ and $\bar{\boldsymbol{a}}_{\text{tr}}^c$:

$$d_c^1 = \left\|\bar{\boldsymbol{a}}_{\text{tr}} - \bar{\boldsymbol{a}}_{\text{tr}}^c\right\|_2^2. \tag{24}$$

Then the prior knowledge for $\boldsymbol{A}$ and $\boldsymbol{A}_c$, denoted by β_c^1, is defined as

$$\beta_c^1 = \exp(-d_c) = \exp\left(-\left\|\bar{\boldsymbol{a}}_{\text{tr}} - \bar{\boldsymbol{a}}_{\text{tr}}^c\right\|_2^2\right). \tag{25}$$

The extraction of β_c^1 indicates that two close centroids are set with a larger weight, otherwise the far ones are given a small weight. β_c^1 is denoted as the NC-based prior knowledge because it is derived similarly to the NC classifier.

2) LDA-NC-Based Prior Knowledge: From the NC-based prior knowledge described in Algorithm 2, it can be seen that it is directly derived from the distance between two centroids. Although the centroid is robust to noise, it heavily depends on the data itself (or the used feature representations). This is difficult to explore the essential structure of the data. That is, the data may not be separable in the original space. Inspired by the kernel method, here we first transform the original data into another space in which the data is distributed separably.

There are several possible algorithms to transform the original data according to specific rules. In this paper, we select the commonly used linear discriminant analysis (LDA) because it is able to maximize the between-class scatter while minimizing the within-class scatter. In this situation, with the training set $\boldsymbol{A}$, we remove the mean vector from each sample of $\boldsymbol{A}$ and then compute the total scatter matrix. After that, we find a number of eigenvectors by decreasing the corresponding eigenvalues and finally project the elements of $\boldsymbol{A}$ using the obtained eigenvectors.

After LDA, we represent all the training samples and those for the cth class using $\boldsymbol{B}$ and $\boldsymbol{B}_c$, respectively. Similarly, let $\bar{\boldsymbol{b}}_{\text{tr}}$ be the mean vector of all training samples in $\boldsymbol{B}$, while $\bar{\boldsymbol{b}}_{\text{tr}}^c$ is the mean vector of the cth training samples in $\boldsymbol{B}_c$. Then the LDA-NC-based prior knowledge, denoted by β_c^2, can be achieved as follows:

$$d_c^2 = \left\|\bar{\boldsymbol{b}}_{\text{tr}} - \bar{\boldsymbol{b}}_{\text{tr}}^c\right\|_2^2 \tag{26}$$

```
Algorithm 3 LDA-NC-Based Prior Knowledge
Input: A training set $\boldsymbol{A}$ containing $C$ class samples, namely $\boldsymbol{A} = [\boldsymbol{A}_1\ \boldsymbol{A}_c\ \cdots\ \boldsymbol{A}_C]$.
Output: the LDA-NC-based prior information of each class $\{\beta_1^2, \cdots, \beta_C^2\}$.
    (a) Conduct LDA to $\boldsymbol{A}$, getting the transformed data $\boldsymbol{B} = [\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots, \boldsymbol{B}_C]$;
    (b) Calculate the mean vectors of $\boldsymbol{B}$ and $\{\boldsymbol{B}_c | c = 1, \cdots, C\}$ respectively, i.e., $\bar{\boldsymbol{b}}_{\text{tr}}$ and $\bar{\boldsymbol{b}}_{\text{tr}}^c$
    (c) For each $\bar{\boldsymbol{b}}_{\text{tr}}^c$, compute the distance between $\bar{\boldsymbol{b}}_{\text{tr}}$ and $\bar{\boldsymbol{b}}_{\text{tr}}^c$ as Eq. (24), $c = 1, \cdots, C$.
    (d) Derive the LDA-NC-based prior information $\{\beta_1^2, \ldots, \beta_C^2\}$ using Eq. (25).
```

$$\beta_c^2 = \exp\left(-d_c^2\right) = \exp\left(-\left\|\bar{\boldsymbol{b}}_{\text{tr}} - \bar{\boldsymbol{b}}_{\text{tr}}^c\right\|_2^2\right). \tag{27}$$

B. Representation-Based Prior Knowledge

As previously mentioned, β_c can be regarded as a weight to balance $\boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{A}_c\boldsymbol{x}_c$. It can be seen that NC-based prior knowledge and LDA-NC-based prior knowledge measure the weight via the Euclidean distance. The existing algorithms described in Section II indicate that, apart from the Euclidean distance, the representation residual is an important way for classification. Hence, we also develop the prior knowledge in terms of representation perspectives.

Here, we divide the training samples $\boldsymbol{A}$ into two parts, denoted as $\boldsymbol{A}^d$ and $\boldsymbol{A}^r$. The samples of $\boldsymbol{A}^r$ will be represented by the samples of $\boldsymbol{A}^d$. The representative LRC and CRC methods are considered to derive prior knowledge in this paper as follows.

1) LRC-Based Prior Knowledge: In the LRC model, a test sample is represented by the training ones of the cth class. Following this framework, we make use of $\boldsymbol{A}_c^d$, the samples of $\boldsymbol{A}^d$ belonging to the cth class, to represent $\boldsymbol{A}^r$. Assume that $\boldsymbol{y}^r$ is a sample of $\boldsymbol{A}^r$; hence, it can be described by

$$\tilde{\boldsymbol{y}}^r = \boldsymbol{A}_c\left(\boldsymbol{A}_c^{d^T}\boldsymbol{A}_c^d\right)^{-1}\boldsymbol{A}_c^{d^T}\boldsymbol{y}^r. \tag{28}$$

Then the residual of $\boldsymbol{y}^r$ is $\|\boldsymbol{y}^r - \tilde{\boldsymbol{y}}^r\|$. Considering all elements of $\boldsymbol{A}^r$, the total residual can be achieved by

$$r_c(\boldsymbol{A}^r) = \sum_{\boldsymbol{y}^r \in \boldsymbol{A}^r} \left\|\boldsymbol{y}^r - \tilde{\boldsymbol{y}}^r\right\| \tag{29}$$

where $r_c(\boldsymbol{A}^r)$ is considered as a special distance between $\boldsymbol{A}_c^d$ and $\boldsymbol{A}^r$. Similar to the distance-based prior knowledge, we derive the LRC-based prior knowledge, denoted by β_c^3, in the following way:

$$\beta_c^3 = \exp\left(-r_c(\boldsymbol{A}^r)\right). \tag{30}$$

2) CRC-Based Prior Knowledge: In contrast with LRC, CRC represents the test sample using all training ones. Hence, $\boldsymbol{A}^d$ is used to represent $\boldsymbol{y}^r \in \boldsymbol{A}^r$, resulting in the following representation coefficient:

$$\boldsymbol{x}^r = \left((\boldsymbol{A}^d)^T\boldsymbol{A}^d + \bar{\lambda}\boldsymbol{I}\right)^{-1}\left(\boldsymbol{A}^d\right)^T\boldsymbol{y}^r. \tag{31}$$

```
Algorithm 4 LRC-Based Prior Knowledge
Input: A training set $\boldsymbol{A}$ containing $C$ class samples, namely $\boldsymbol{A} = [\boldsymbol{A}_1\ \boldsymbol{A}_c\ \cdots\ \boldsymbol{A}_C]$.
Output: The LRC-based prior information of each class $\{\beta_1^3, \cdots, \beta_C^3\}$.
    (a) Separate $\boldsymbol{A}_c$, the training samples of the $c$th class, into $\boldsymbol{A}_c^d$ and $\boldsymbol{A}_c^r$, $c = 1, \cdots, C$;
    (b) Represent each sample of $\boldsymbol{A}^r$ by Eq. (28);
    (c) Compute the total residual as Eq. (29);
    (d) Derive the LRC-based prior knowledge $\{\beta_1^3, \ldots, \beta_C^3\}$ using Eq. (30).
```

```
Algorithm 5 CRC-Based Prior Knowledge
Input: A training set $\boldsymbol{A}$ containing $C$ class samples, namely $\boldsymbol{A} = [\boldsymbol{A}_1\ \boldsymbol{A}_c\ \cdots\ \boldsymbol{A}_C]$.
Output: The CRC-based prior information of each class $\{\beta_1^4, \cdots, \beta_C^4\}$.
    (a) Separate $\boldsymbol{A}_c$, the training samples of the $c$th class, into $\boldsymbol{A}_c^d$ and $\boldsymbol{A}_c^r$, $c = 1, \cdots, C$; Denote $\boldsymbol{A}^d = [\boldsymbol{A}_1^d, \cdots, \boldsymbol{A}_C^d]$ and $\boldsymbol{A}^r = [\boldsymbol{A}_1^r, \cdots, \boldsymbol{A}_C^r]$.
    (b) Represent each sample of $\boldsymbol{A}^r$ by Eq. (31);
    (c) Compute the total residual as Eq. (32);
    (d) Derive the CRC-based prior knowledge $\{\beta_1^4, \ldots, \beta_C^4\}$ using Eq. (33).
```

Then we use the elements of $\boldsymbol{x}^r$ that are associated with the cth class, denoted by $\boldsymbol{x}_c^r$, to study the relation between $\boldsymbol{A}$ and $\boldsymbol{A}_c$ as $\|\boldsymbol{A}^d\boldsymbol{x}^r - \boldsymbol{A}_c^d\boldsymbol{x}_c^r\|_2$. Considering all elements of $\boldsymbol{A}^r$, we can derive the CRC-based prior knowledge β_c^4 by

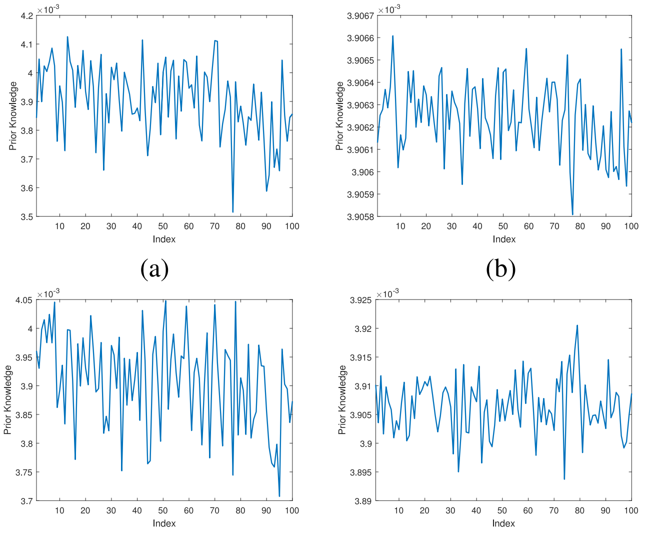
$$r_c(\boldsymbol{A}^r) = \sum_{\boldsymbol{y}^r \in \boldsymbol{A}^r} \left\| \boldsymbol{A}^d\boldsymbol{x}^r - \boldsymbol{A}_c^d\boldsymbol{x}_c^r \right\| \tag{32}$$

$$\beta_c^4 = \exp\left(-r_c(\boldsymbol{A}^r)\right). \tag{33}$$

We summarize the detailed implementations of the above-mentioned extraction methods as Algorithms 2–5. Here, we also take the well-known Caltech-256 database as an example to derive the corresponding prior knowledge. Thirty samples for each class are randomly selected from the whole database to form the training set. Fig. 2 illustrates the first 100 elements of NC, LNC, LRC, and CRC-based prior knowledge extraction methods, respectively. We can observe that these prior knowledge extraction methods differ from each other.

C. Discussion

From the detailed derivation of PKPCRC, we can observe that it can be considered as a two-phase classification model, including prior knowledge extraction and calculation of the representation coefficients, respectively. Considering a visual recognition task, there must be a specific inherent relation among the samples of different classes. The prior knowledge, obtained in the first phase, aims to explore some useful information from the training data. This information will change the inherent relation of the original training data such that it benefits to obtain a more accurate representation of the test sample in the second phase.



(a) (b)

(c) (d)

Fig. 2. Examples of the prior knowledge using 30 training images per class for the Caltech-256 database. (a)–(d) Correspond to the NC, LDA-NC, LRC, and CRC-based prior knowledge. Note that only the first 100 numbers are plotted here.

In this section, we provide four extraction methods of prior knowledge for PKPCRC, namely, NC-based, LDA-NC-based, LRC-based, and CRC-based prior knowledge. They are derived from the perspectives of image distance and representation capacity. The first two methods are based on the distances between the centroids of all training samples and those of the cth class, while the other two methods make use of the representation error. The small distance and representation error give large weights as prior knowledge, resulting in that the training samples of each class have a uniform distribution. This operation can remove the effect of some noise samples such that it is easy to find accurate representation coefficients for the test sample. Because the inherent relations among the samples of different class vary, PKPCRC with different types of prior knowledge achieves different performances on the same database.

V. Experimental Results

In this section, we will carry out several experiments to evaluate the classification performance of PKPCRC by comparing it with some state-of-the-art methods. First, the used databases and experimental setting are introduced. Subsequently, the study of the representation data and results on four different databases is given, respectively. Finally, a comparison of time complexity is provided.

A. Databases and Experimental Settings

In the following experiments, four benchmark databases, namely, Caltech-UCSD Birds (CUB200-2011) [26], Caltech-256 [27], Oxford 102 Flowers [28], and Stanford 40 Actions [29], are utilized to evaluate the performance of all competing algorithms. A brief introduction of these databases are presented as follows.

1) *Caltech-UCSD Birds (CUB200-2011) Database [26]:* This database is composed of 200 different bird

TABLE I
SUMMARY OF THE USED DATABASES IN THE EXPERIMENTS

Database	Num. of class	Num. of each class	Total Num.
Caltech-UCSD Birds	200	41∼60	11788
Caltech-256	256	80∼827	30608
Oxford 102 Flowers	102	40∼258	8198
Stanford 40 Actions	40	180∼300	9352

categories, forming 11 788 images in total. It is a difficult task to recognize these images because of the large similarity among some bird categories. There are about 60 images for each category on average.

2) *Caltech-256 Database [27]:* Consisting of totally 30 608 object images of 256 categories, this database is popularly applied to assess several large-scale image classification methods. The objects in this database include zebra, waterfall, owl, and many others. The image number of each category changes from 80 to 827.
3) *Oxford 102 Flowers Database [28]:* This database is commonly used for the fine-grained image classification, including 8198 flower images from 102 different categories. Apart from the large variations within each category, these images were captured under various scales, pose, and lighting conditions. There are 40–258 flower images for each category.
4) *Stanford 40 Actions Database [29]:* This database contains 40 different human actions, such as applauding, climbing, cooking, and running. The number of images is 9352 in total. It contains 180–300 images per category.

Table I also gives a summary of these databases, including the class numbers, image number in each category, and total image numbers, respectively.

To better classify an unknown image, we also need a discriminative and robust feature representation for images. Many well-known features, such as scale-invariant feature transform [30], local binary pattern [31], and their improvements, have been proposed to describe the image contents. But these features are hand-crafted ones, and they cannot provide high level and abstract representation of images. In this paper, the recent CNN features, extracted via VGG-verydeep-19 [32], are applied here, resulting in a 4096 × 1 vector as the feature representation of images for classification.

Based on the prior knowledge extraction methods in Section IV, we can derive for classifiers, which are denoted PKPCRC-NC, PKPCRC-LNC, PKPCRC-LRC, and PKPCRC-CRC, respectively. The proposed methods involve some parameters that are determined as follows. In (18), λ is empirically set to 0.1, while γ is selected by a fivefold cross validation on the training set. Besides, the parameter $\bar{\lambda}$ in (31) is set to 0.01 to derive the CRC-based prior knowledge.

B. Study of the Data Separation

For the LRC and CRC-based prior knowledge, we have to further separate the training data into two parts. The first part is used as a training set, while the second part is used as a test set. The prior knowledge is obtained by representing the second part by the first part. The separation of training data will affect the obtained prior knowledge. Here, we study the performance of PKPCRC-LRC and PKPCRC-CRC with different separation ratios.

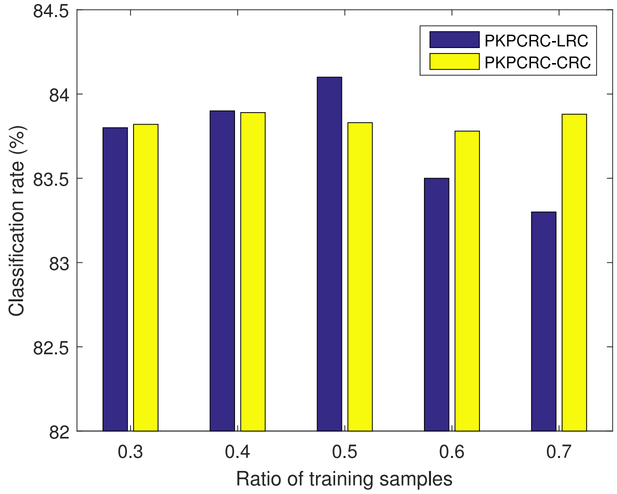


Fig. 3. Classification performance of PKPCRC-LRC and PKPCRC-CRC on the Caltech-256 database using different ratios of training samples. The ratio here is set to 0.3, 0.4, . . . , and 0.7, respectively.

The Caltech-256 database is applied here. We choose 30 images from each category to form the training data, which is divided into two parts. The ratio of the first part is set to 0.3, 0.4, . . . , and 0.7, respectively. The average results of ten repeats for PKPCRC-LRC and PKPCRC-CRC are plotted in Fig. 3. From these results, we can find that PKPCRC-LRC is more sensitive to the partition ratio than PKPCRC-CRC. When the ratio is 0.5, PKPCRC-LRC achieves the best classification performance, while it is 0.4 for PKPCRC-CRC. When the partition ratio is 0.4 or 0.5, PKPCRC-CRC achieves comparable classification performance. Therefore, in the following experiments, we equally separate the training data into two parts to extract prior knowledge.

C. Results on the CUB200-2011 Database

In this experiment, the training and testing sets, given in the CUB200-2011 database, are used for evaluation. Each bird category contains about 30 samples in the training set, while the rest of the images form the testing set. The following competing algorithms are selected for comparison, including NN [10], NC, Softmax [33], SVM [34], Kernel SVM [34], NSC [35], CRC [19], SRC [15], CROC [36], ProCRC [23], PN-CNN [37], FV-CNN [38], and POOF [39], respectively. The classification performance of all these classifiers and the proposed PKPCRC are illustrated in Table II.

From the results in Table II, we can observe that the classification rate of NN is 50.1%, which is the worst performance in this situation because NN, as aforementioned, is a simple and native classifier without the help of training. NC, which applies the centroid representation to replace the nearest sample, surpasses NN by about ten percentages. The well-known representation-based methods, namely, CRC, SRC, and CROC, obtain comparable accuracy for this databases. Their results are all about 76%. The kernel SVM slightly outperforms CRC, SRC, and CROC. The performance of ProCRC,

TABLE II
CLASSIFICATION PERFORMANCE (%) OF DIFFERENT CLASSIFIERS ON CUB200-2011 DATABASE

Classifier	CUB200-2011 Database
NN	50.1
NC	60.3
Softmax	72.1
SVM	75.4
Kernel SVM	76.6
NSC	74.5
CRC	76.2
SRC	76.0
CROC	76.2
ProCRC	78.3
PN-CNN	75.7
FV-CNN	66.7
POOF	56.9
PKPCRC-NC	79.3
PKPCRC-LNC	78.7
PKPCRC-LRC	**79.7**
PKPCRC-CRC	78.7

TABLE III
CLASSIFICATION PERFORMANCE (%) OF DIFFERENT CLASSIFIERS ON THE CALTECH-256 DATABASE WITH 30 TRAINING SAMPLES

Classifier	Caltech-256 Database
NN	65.2
NC	71.9
Softmax	75.3
SVM	80.1
Kernel SVM	81.3
NSC	80.2
CRC	81.1
SRC	81.3
CROC	81.7
ProCRC	83.3
ZF	70.6
M-HMP	50.7
LLC	41.2
ScSPM	34.0
PKPCRC-NC	**84.3**
PKPCRC-LNC	83.9
PKPCRC-LRC	84.1
PKPCRC-CRC	83.8

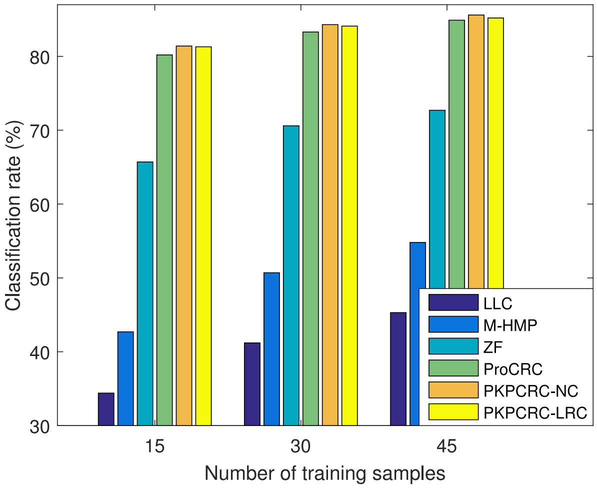


Fig. 4. Classification rates of different classifiers, including LLC, M-HMP, ZF, ProCRC, PKPCRC-NC, and PKPCRC-LRC, on the Caltech-256 database using 15, 30, and 45 training samples.

derived from a probabilistic view, is superior to those of other competing methods. The proposed PKPCRC-based classifiers all work better than other competing algorithms. PKPCRC-LRC and PKPCRC-CRC achieve the same results for this database, while PKPCRC-LRC obtains the most satisfactory performance in contrast to other methods.

The proposed PKPCRC methods are also compared with three state-of-the-art algorithms, namely, POOF, FV-CNN, and PN-CNN. They are derived using a specially developed CNN architecture for bird recognition. PN-CNN works better than NN, NC, Softmax, SVM, and NSC. The classification rates of the proposed methods are all higher than that of PN-CNN at least three percentages.

D. Results on the Caltech-256 Database

In this experiment, we evaluate the performance of PKPCRC using the Caltech-256 database. In this situation, the commonly used experimental setting is considered here. That is, we randomly select N images from each category to form the training set. The images, except for the training set, are regarded as test images. We run this procedure for ten times to achieve an average classification rate for each competing algorithm. In the following experiments, the number of training images for each category N is set to 15, 30, and 45, respectively.

Table III shows the results of different algorithms when N is set to 30. Note that ZF [40], M-HMP [41], LLC [42], and ScSPM [43] are traditional methods that are not based on the CNN features, while the rest classifiers determine the class label of each test image by the aforementioned CNN feature. It can be seen that NC with the CNN feature works better than the traditional four methods. The performance of kernel SVM is superior to those of Softmax, SVM, NSC, and CRC and is the same as that of SRC. ProCRC outperforms kernel SVM and SRC by two percentages. Considering the proposed methods, PKPCRC-NC achieves the best accuracy, which is 1% higher than that of ProCRC.

The classification results of LLC, M-HMP, ZF, ProCRC, PKPCRC-NC, and PKPCRC-LRC are plotted in Fig. 4, and N is set to 15, 30, and 45, respectively. We can observe that ProCRC and the proposed two methods here significantly surpass the competing schemes here. When the training number of each class is set to 15, PKPCRC-NC and PKPCRC-LRC outperform ProCRC by 1.2% and 1.1%. When N increases to 45, PKPCRC-NC works better than PKPCRC-LRC, and it improves ProCRC by 0.7%.

E. Results on the Oxford 102 Flowers Database

The experiment on this database is conducted using the setting reported in [23] and [28]. Table IV shows the performance of different classifiers. For this database, the classification rates of NN, NC, Softmax, OverFeat [44], GMP [45], BiCos seg [46], and DAS [47] are all smaller than 90%, and Softmax achieves the best result among these methods. The kernel SVM outperforms SVM and NSC, but its result is worse than those of CRC, SRC, CROC, ProCRC, and the proposed ones. Considering the proposed classifiers, their classification rates are all higher than 95%. PKPCRC-LNC and PKPCRC-CRC

TABLE IV
CLASSIFICATION PERFORMANCE (%) OF DIFFERENT CLASSIFIERS ON THE OXFORD 102 FLOWERS DATABASE

Classifier	Oxford 102 Flowers Database
NN	79.3
NC	81.6
Softmax	87.3
SVM	90.9
Kernel SVM	92.2
NSC	90.1
CRC	93.0
SRC	93.2
CROC	93.1
ProCRC	94.8
OverFeat	86.8
GMP	84.6
BiCos seg	79.4
DAS	80.7
PKPCRC-NC	95.5
PKPCRC-LNC	95.4
PKPCRC-LRC	**96.4**
PKPCRC-CRC	95.4

obtain the same performance here, whose classification rates are better than that of ProCRC by 0.6%. PKPCRC-NC slightly outperforms the LNC-based and CRC-based prior knowledge for PKPCRC. Among four proposed classifiers, PKPCRC-LRC achieves the highest classification rate for this database, which is 1% and 1.6% higher than those of PKPCRC-LNC and ProCRC, respectively.

F. Results on the Stanford 40 Actions Database

In this experiment, following the experimental setting used in [23] and [29], we applied 100 images from each category, totally 4000 images, for training and the rest 5352 images for test. Apart from the representation-based classifiers, ASPD [48] and SMP [49] are also considered here. The classification performance of all competing algorithms are illustrated in Table V. For this database, it can be observed that the learning-based classifiers all outperform the traditional NN and NC. CRC and SRC achieve 78.2% and 78.7%, while SVM and kernel SVM all works better than CRC and SRC. The ProCRC, probabilistic extension of CRC, surpasses SVM, kernel SVM, CRC, and SRC by 1% at least. Considering the proposed methods, PKPCRC-NC slightly outperforms ProCRC by 0.3%. The rest three classifiers obtain similar results, all about 82%, and work better than other competing classifiers.

G. Time Complexity Analysis

Efficiency of a classifier is also an important aspect in a recognition system apart from the accuracy. In this experiment, we will study the time complexity of the proposed PKPCRC. Cai *et al.* [23] pointed out that ProCRC and CRC take the same running time, and their speeds are faster than those of SRC and CROC. Considering PKPCRC, it further takes account of the prior knowledge, resulting in additional running time compared to ProCRC. As described in Section IV, four types of prior knowledge were introduced, and they will need a different time complexity. Here, we take the Caltech-256 database as an example to quantitatively compare the running times of each prior knowledge. Fifteen training samples for each category are randomly selected from the whole database to derive the corresponding prior knowledge, and this procedure is repeated ten times to obtain an average running time. The NC-based prior knowledge, as the simplest one, takes the shortest running time, which is 0.1141 s. We denote the running times of LNC, LRC, and CRC-based prior knowledge by t_1, t_2, and t_3, respectively. The running time of (18), denoted by t_4, is also considered here because it is a key part to get the closed form. The results are plotted in Fig. 5, where the detailed values of t_1–t_4 are 14.0744, 86.0704, 22.7782, and 102.2426, respectively. It can be seen that the LRC-based prior knowledge takes much more time in contrast with other two types of prior knowledge. However, compared with the derivation of the projection matrix, it takes less time to achieve the prior knowledge.

TABLE V
CLASSIFICATION PERFORMANCE (%) OF DIFFERENT CLASSIFIERS ON THE STANFORD 40 ACTIONS DATABASE

Classifier	Stanford 40 Actions Database
NN	57.9
NC	65.7
Softmax	77.2
SVM	79.0
Kernel SVM	79.8
NSC	74.7
CRC	78.2
SRC	78.7
CROC	79.1
ProCRC	80.9
ASPD	75.4
SMP	53.0
PKPCRC-NC	81.2
PKPCRC-LNC	82.2
PKPCRC-LRC	**82.4**
PKPCRC-CRC	82.3

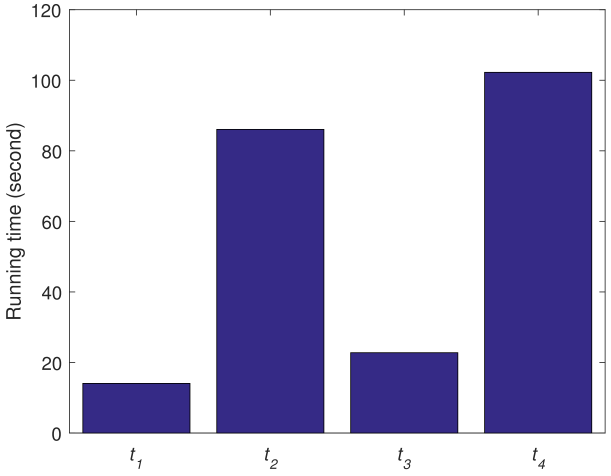


Fig. 5. Running time of related operations for PKPCRC and ProCRC. t_1, t_2, and t_3 are the running times of LNC, LRC, and CRC-based prior knowledge, while t_4 is the running time of the projection matrix of ProCRC in (18).

VI. CONCLUSION

In this paper, we presented a novel classifier called PKPCRC for visual recognition. As a representation-based classifier, in contrast with some existing ones, PKPCRC further takes a prior knowledge into account to achieve a more accurate representation of the query image. The prior knowledge is used to change the contribution of each class in the training set. We also provided four extraction methods of prior knowledge for PKPCRC that were derived from different perspectives. Four visual recognition tasks were used to evaluate the proposed PKPCRC, and the comparison results indicated that the proposed ones obtain better performance in contrast with some state-of-the-art classifiers. In the future, there are some interesting problems deserving further studies based on PKPCRC. For example, we can couple the developed prior knowledge into other classifiers or extract the prior knowledge using the deep learning techniques.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valued comments and constructive suggestions that significantly improved the quality of this paper.

References

[1] Q. Feng *et al.*, "Superimposed sparse parameter classifiers for face recognition," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 378–390, Feb. 2017.

[2] Y. Guo, L. Jiao, S. Wang, S. Wang, and F. Liu, "Fuzzy sparse autoencoder framework for single image per person face recognition," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2402–2415, Aug. 2018.

[3] R. Lan, Y. Zhou, and Y. Y. Tang, "Quaternionic weber local descriptor of color images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 2, pp. 261–274, Feb. 2017.

[4] L. Ji, Y. Ren, G. Liu, and X. Pu, "Training-based gradient LBP feature models for multiresolution texture classification," *IEEE Trans. Cybern.*, vol. 48, no. 9, pp. 2683–2696, Sep. 2018.

[5] X. Zhang, S. Wang, Z. Li, and S. Ma, "Landmark image retrieval by jointing feature refinement and multimodal classifier learning," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1682–1695, Jun. 2018, doi: 10.1109/TCYB.2017.2712798.

[6] J. Gui, T. Liu, D. Tao, Z. Sun, and T. Tan, "Representative vector machines: A unified framework for classical classifiers," *IEEE Trans. Cybern.*, vol. 46, no. 8, pp. 1877–1888, Aug. 2016.

[7] J. Peng, H. Chen, Y. Zhou, and L. Li, "Ideal regularized composite kernel for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 4, pp. 1563–1574, Apr. 2017.

[8] Y. Wei, Y. Zhou, and H. Li, "Spectral-spatial response for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 3, p. 203, 2017.

[9] R. Lan and Y. Zhou, "Medical image retrieval via histogram of compressed scattering coefficients," *IEEE J. Biomed. Health Inf.*, vol. 21, no. 5, pp. 1338–1346, Sep. 2017.

[10] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[11] S. Z. Li and J. Lu, "Face recognition using the nearest feature line method," *IEEE Trans. Neural Netw.*, vol. 10, no. 2, pp. 439–443, Mar. 1999.

[12] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2106–2112, Nov. 2010.

[13] J.-S. Pan, Q. Feng, L. Yan, and J.-F. Yang, "Neighborhood feature line segment for image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 387–398, Mar. 2015.

[14] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.

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

[16] C.-Y. Lu, H. Min, J. Gui, L. Zhu, and Y.-K. Lei, "Face recognition via weighted sparse representation," *J. Vis. Commun. Image Represent.*, vol. 24, no. 2, pp. 111–116, 2013.

[17] C. Zou, K. I. Kou, and Y. Wang, "Quaternion collaborative and sparse representation with application to color face recognition," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3287–3302, Jul. 2016.

[18] Y. Y. Tang and H. Yuan, "Manifold-based sparse representation for hyperspectral image classification," in *Handbook of Pattern Recognition and Computer Vision*. Singapore: World Sci., 2016, pp. 331–350.

[19] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 471–478.

[20] Y. Xu, D. Zhang, J. Yang, and J.-Y. Yang, "A two-phase test sample sparse representation method for use with face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1255–1262, Sep. 2011.

[21] L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification via multiscale adaptive sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 12, pp. 7738–7749, Dec. 2014.

[22] Y. Xu *et al.*, "Using the idea of the sparse representation to perform coarse-to-fine face recognition," *Inf. Sci.*, vol. 238, pp. 138–148, Jul. 2013.

[23] S. Cai, L. Zhang, W. Zuo, and X. Feng, "A probabilistic collaborative representation based approach for pattern classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2950–2959.

[24] R. Lan and Y. Zhou, "An extended probabilistic collaborative representation based classifier for image classification," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, 2017, pp. 1392–1397.

[25] A. M. Martinez, "The AR face database," Centre de Visió per Computador, Universitat Autònoma de Barcelona, Barcelona, Spain, Rep. #24, 1998.

[26] C. Wah, S. Branson, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200–2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.

[27] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2007-001, 2007.

[28] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. IEEE 6th Indian Conf. Comput. Vis. Graph. Image Process. (ICVGIP)*, 2008, pp. 722–729.

[29] B. Yao, *et al.*, "Human action recognition by learning bases of action attributes and parts," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2011, pp. 1331–1338.

[30] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[31] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[33] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[34] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.

[35] J.-T. Chien and C.-C. Wu, "Discriminant waveletfaces and nearest feature classifiers for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1644–1649, Dec. 2002.

[36] Y. Chi and F. Porikli, "Classification and boosting with multiple collaborative representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1519–1531, Aug. 2014.

[37] S. Branson, G. Van Horn, S. Belongie, and P. Perona, "Bird species categorization using pose normalized deep convolutional nets," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–14.

[38] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3828–3836.

[39] T. Berg and P. Belhumeur, "POOF: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 955–962.

[40] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[41] L. Bo, X. Ren, and D. Fox, "Multipath sparse coding using hierarchical matching pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 660–667.

[42] J. Wang *et al.*, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3360–3367.

[43] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1794–1801.

[44] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2014, pp. 512–519.

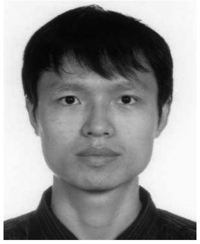
[45] N. Murray and F. Perronnin, "Generalized max pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2473–2480.

[46] Y. Chai, V. Lempitsky, and A. Zisserman, "BiCoS: A bi-level co-segmentation method for image classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2011, pp. 2579–2586.

[47] A. Angelova and S. Zhu, "Efficient object detection and segmentation for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2013, pp. 811–818.

[48] F. S. Khan *et al.*, "Recognizing actions through action-specific person detection," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4422–4432, Nov. 2015.

[49] F. S. Khan, J. van de Weijer, R. M. Anwer, M. Felsberg, and C. Gatta, "Semantic pyramids for gender and action recognition," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3633–3645, Aug. 2014.



Rushi Lan received the B.S. degree in information and computing science and M.S. degree in applied mathematics from the Nanjing University of Information Science and Technology, Nanjing, China, and the Ph.D. degree in software engineering from the University of Macau, Macau, China.

He is currently an Assistant Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. His current research interests include image classification, image denoising, and metric learning.

Yicong Zhou (M'07–SM'14) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA.

He is currently an Associate Professor and the Director of the Vision and Image Processing Laboratory, Department of Computer and Information Science, University of Macau, Macau, China. His current research interests include chaotic systems, multimedia security, computer vision, and machine learning.

Dr. Zhou was a recipient of the Third Price of Macau Natural Science Award in 2014. He serves as an Associate Editor for *Neurocomputing*, the *Journal of Visual Communication and Image Representation*, and *Signal Processing: Image Communication*. He is the Co-Chair of the Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society. He is a Senior Member of the International Society for Optical Engineering.

Zhenbing Liu received the B.S. degree in mathematics and applied mathematics from Qufu Normal University, Qufu, China, and the M.S. degree in probability and statistics and Ph.D degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China.

He was a Visiting Scholar with the Department of Radiology, University of Pennsylvania, Philadelphia, PA, USA, in 2015. He is currently a Professor and a Doctoral supervisor with the School of Computer and Information Security, Guilin University of Electronic Technology, Guilin, China. His current research interests include image processing, machine learning, and pattern recognition.

Xiaonan Luo received the B.S. degree in computational mathematics from Jiangxi University, Nanchang, China, the M.S. degree in applied mathematics from Xidian University, Xi'an, China, and the Ph.D. degree in computational mathematics from Dalian University of Technology, Dalian, China.

Prof. Luo received the National Science Fund for Distinguished Young Scholars granted by the National Natural Science Foundation of China.