Kernel Regularized Data Uncertainty for Action Recognition

Qingxiang Feng and Yicong Zhou, *Senior Member, IEEE*

***Abstract*— The traditional data uncertainty (DU) classifier fails to encode the importance of each sample for solving the minimum problem. Moreover, it considers only linear information for classification. To overcome these, we propose four classifiers for action recognition. They are called regularized DU (RDU) classifier, RDU coefficient (RDUC) classifier, kernel RDU (KRDU) classifier, and kernel RDUC (KRDUC) classifier, respectively. Extensive experiments on four benchmark action databases demonstrate that the proposed four classifiers achieve better recognition rates than the traditional DU classifier and several state-of-the-art methods. Moreover, the computation costs of the KRDU and KRDUC classifiers are much less than that of the DU classifier.**

***Index Terms*— Action recognition, data uncertainty (DU), kernel sparse representation classification, sparse-representation-based classification (SRC).**

I. Introduction

ACTION recognition systems are extremely complex because automatic inference of action activities from videos requires a large amount of rich activity information. Previous research developed a lot of features selection methods [1] for the action video sequences. Some methods are based on low-level and midlevel features, such as dense point trajectories [2]–[4], local space–time features [5], [6], and dense 3D gradient histograms [7]. However, these methods have been limited in the amount of motion semantics. They capture the low-level feature and often generate representations with inadequate discriminative information for large and complex data. In addition, some approaches focus on obtaining a more semantically rich and discriminative representation. They aim at finding the information of object and scene semantics [8] or human pose [9], [10], but the objective is challenging and unsolved. The method in [11] tries to use the neural network to learn the depth feature for classification. Inspired by the object bank method [12], Sadanand and Corso [13] proposed the action bank feature to obtain better high-level representation of human action in video. The action bank [13] explores a large amount of action detectors that ultimately act like the bases of a high-dimensional action space. The action bank feature is combined with a simple support vector machine (SVM) [14], [15] classifier and obtains the good performance for activity recognition.

In the above methods, they pay attention only to learning the action features and using the out-of-date classifiers, such as SVM. However, action recognition systems are also critically dependent on classifiers. The good classifiers are quite useful for improving the action recognition performance. Therefore, we aim at proposing the better classifier, instead of learning the better feature, to improve the recognition performance for action recognition in this paper. The related classifiers are reviewed as follows.

The nearest neighbor (NN) [16] and nearest subspace (NS) [17] methods are two well-known classifiers. As the extension of NN and NS, linear regression classification (LRC) [18], [19] develops class-specific models of the registered users and redefines the recognition task as a linear regression problem. LRC is based on the concept that samples from a specific object class lie in a linear subspace [20]–[22]. Afterward, several LRC-associated approaches [23] have been proposed, such as the kernel LRC [24] that uses the nonlinear information instead of the linear information in LRC; nearest regularized subspace [25] that uses Tikhonov matrix to encode the importance of each sample for classification; neighborhood feature line segment (NFLS) [26] that combines the feature line and LRC for classification; double LRC [27] that utilizes twice regression projection operations; and linear discrimination regression classification [28] and unitary regression classification [29] that both try to let the samples of the same class be very close such that the performance may be improved. All the above classifiers can be treated as the class-subspace classifiers.

Different from the class-subspace classifiers, sparse-representation-based classification (SRC) [30], [31] adopts all-class-space to classify the test sample. Later, several improvements [32]–[35] have been proposed. For example, Zhang *et al.* [32] proposed the collaborative-representation-based classification (CRC), and Xu *et al.* [33] proposed two-phase sparse representation (TPSR). They both solve the L_2-norm minimum problem instead of the L_1-norm minimum in SRC. Therefore, the computation costs of CRC and TPSR are much lower than that of SRC. Fang *et al.* [36] proposed the nonnegative sparse graph that solves the nonnegative sparse

Manuscript received December 14, 2015; revised March 31, 2016 and July 19, 2016; accepted September 3, 2016. Date of publication October 5, 2016; date of current version March 3, 2017. This work was supported in part by the Macau Science and Technology Development Fund under Grant FDCT/016/2015/A1 and in part by the Research Committee within University of Macau under Grant MYRG2014-00003-FST and Grant MYRG2016-00123-FST. This paper was recommended by Associate Editor T. Bouwmans. *(Corresponding author: Yicong Zhou.)*

The authors are with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: fengqx1988@gmail.com; yicongzhou@umac.mo).

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/TCSVT.2016.2615459

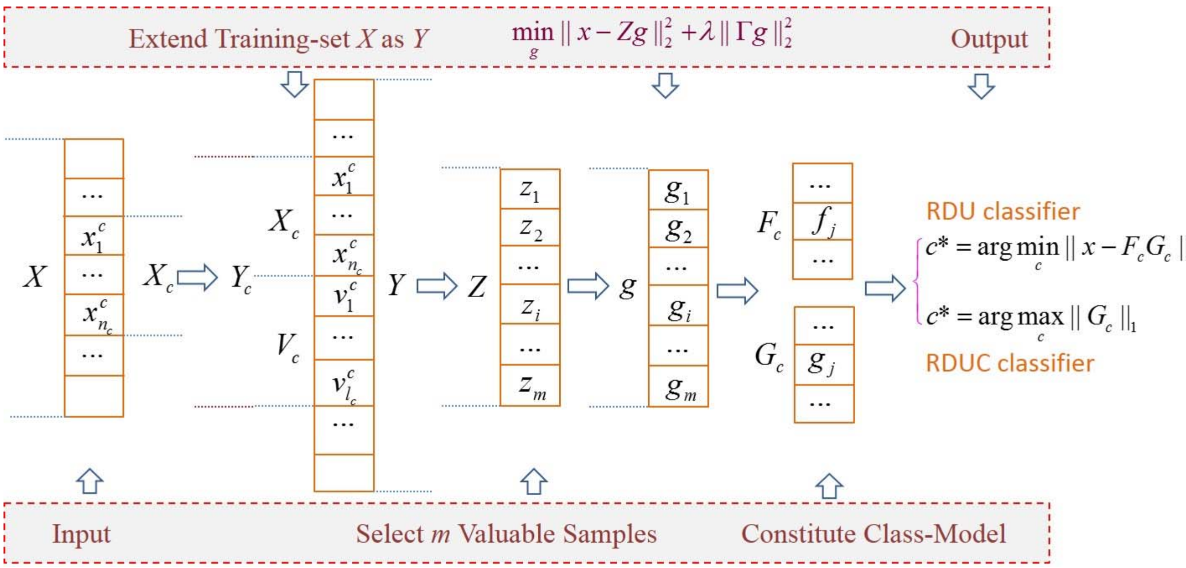

Fig. 1. Flowchart of the proposed RDU and RDUC classifiers.

coefficient such that the better sparsely coefficient is obtained. Lai *et al.* [37], [38] and Yang *et al.* [39] all use the sparse idea to perform the dimensionality reduction such that the better projection space is obtained. Deng *et al.* [34] proposed supposition sparse representation classification (SSRC) that divides the training set into P (prototype) and V (variations). The idea of SSRC is quite simple, but its performance is very well. Xu *et al.* [40] proposed data uncertainty (DU) classifier to generate the virtual samples for classification. The DU classifier obtains the good performance for image recognition.

However, DU does not encode the importance of each sample for solving the L_2-norm minimum problem. Moreover, DU uses only the linear information for classification, while the kernel-based nonlinear information is also useful for classification. To overcome the first weakness of DU classifier, this paper proposes the regularized DU (RDU) classifier and RDU coefficient (RDUC) classifier for action recognition. These two classifiers both use the Tikhonov matrix to encode the importance of each sample for classification. Motivated by the methods in [41]–[45], this paper further proposes the kernel RDU (KRDU) classifier and kernel RDUC (KRDUC) classifier for action recognition. The KRDU and KRDUC classifiers not only include the kernel-based nonlinear information, but also use the kernel-based Tikhonov matrix to encode the importance of each sample for classification. In order to compare the proposed four classifiers with DU classifier and several state-of-the-art classifiers, we carry out extensive experiments to evaluate their recognition performance on the KTH action database [14], UCF50 action database [46], UCF sports action database [47], and HMDB51 action database [48]. The experimental results show that the proposed four classifiers obtain the better recognition performance for action recognition. Moreover, the KRDU and KRDUC classifiers have significantly less computation cost than DU.

The rest of this paper is organized as follows. First, we propose the RDU and RDUC classifiers in Section II. The proposed KRDU and KRDUC classifiers are introduced in Section III. In Section IV, a number of experiments are presented to show the effectiveness of the proposed classifiers. The conclusion is finally addressed in Section V.

II. Proposed RDU and RDUC Classifiers

In this section, we propose two classifiers called RDU classifier and RDUC classifier. The flowchart of the two classifiers is shown in Fig. 1. We describe the method of extending the training set in Section II-A, selecting the valuable samples in Section II-B, and obtaining regularized solution for minimum problem in Section II-C. Afterward, the classification rules of RDU and RDUC are described in Sections II-D and II-E, respectively.

Definition: Let $X = \{x_i^c\}, c = 1, 2, \ldots, M, i = 1, 2, \ldots, n_c$ denote the prototype set, where x_i^c is the ith prototype belonging to the cth class, M is the number of classes, and n_c is the number of prototype samples in the cth class.

A. Extend Training Set

Form a class model X_c by stacking the q-dimensional image vectors that belong to the same class

$$X_c = \left[\, x_1^c \; x_2^c \; \cdots \; x_{N_c}^c \,\right] \in R^{q \times n_c}. \tag{1}$$

The limited training samples of each class cannot comprehensively reflect different variations of the class. It is a typical small sample problem, and can be explained using an example of image recognition. Suppose that the image database contains the pose variations with 30° and 60°. A test image with a pose variation of 45° may be classified into the incorrect class because this database does not contain the pose variation of 45°. Similarity, we may face the same problem for other variations, such as the illumination variation. To solve this problem, we need to produce more training samples from the original class subspace such that the optimized subspace may have the capacity of better representation. One solution is to use virtual samples of each class to increase the number of samples. These virtual samples can be constituted as

$$v_i^c = \frac{x_j^c + x_k^c}{2} \tag{2}$$

where $j, k \in \{1, 2, \ldots, n_c\}$, $i \in \{1, 2, \ldots, l_c\}$, and $l_c = C_2^{n_c}$ denotes the number of combinations. The virtual class model V_c is represented as

$$V_c = \left[v_1^c \cdots v_{l_c}^c \right] \in R^{q \times l_c}. \tag{3}$$

The updated class model $Y_c = [y_1^c \cdots y_{b_c}^c]$ can be constituted as

$$Y_c = \left[X_c \; V_c \right] \in R^{q \times b_c} \tag{4}$$

where $b_c = n_c + l_c$. The entire training set can be constituted by stacking the M class models as

$$Y = \left[Y_1 \cdots Y_c \cdots Y_M \right] \in R^{q \times L} \tag{5}$$

where $L = \sum_c^M b_c$.

B. Select Valuable Samples

The extended training set has different abilities in representing the test sample. However, some samples in the extended training set may have negative effects in representing the test sample. These can be treated as improper training samples. Therefore, the valuable training samples with positive effects need to be selected from the extended training set. From [22], [29], and [35], we know that the training samples closing to the test sample are quite helpful for classifying the test sample correctly. In order to select the valuable samples from the entire training set, the Euclidean distance between the test sample and each training sample will be computed by

$$d_i^c = ||x - y_i^c|| \tag{6}$$

where $c \in \{1, 2, \ldots, M\}$ and $i \in \{1, 2, \ldots, n_c\}$. The Euclidean distance in (6) can be treated as similarity between the test sample and the corresponding training sample. The smallest distance represents the most similarity. Therefore, m useful training samples with the smallest distance will be selected from the entire training set. The remaining training samples will be treated as the negative samples in representing the test sample and will be discarded. The chosen m training samples are used to constitute the novel training space, which can be described as

$$Z = [z_1 \cdots z_i \cdots z_m] \in R^{q \times m}. \tag{7}$$

C. Regularized Solution for Minimum Problem

The L_1-norm-based sparse representation is time consuming and might not satisfy the efficiency requirement of the real-world applications. On the other hand, the L_2-minimum-based representation method is a quite simple and computation efficient algorithm. According to [25], using the regularized L_2-minimum representation, the objective function of the RDU and RDUC classifiers is as follows:

$$\min ||x - Zg||_2^2 + \lambda ||\Gamma g||_2 \tag{8}$$

Algorithm 1 RDU Classifier

Require: The test sample x and the original prototype set $X = \left\{x_i^c, c = 1, 2, \ldots, M, i = 1, 2, \ldots, n_c\right\}$

Ensure: The class index of x.

1: Extend the training set X as the Y by (5).
2: Select m training samples as Z by (7).
3: Calculate the Tikhonov matrix by (9).
4: The regularized coefficient g is computed by (10).
5: Compute the regularized residual of each class according to (12).
6: The RDU classifier will classify the test sample x into the class with the minimum regularized residual by (13).

where λ is a global regularization parameter. Γ is a biased Tikhonov matrix computed by the chosen training space and test sample x. Γ can be computed as

$$\Gamma = \begin{bmatrix} ||x - z_1|| & 0 & 0 \\ \cdots & \cdots & \cdots \\ 0 & 0 & ||x - z_m|| \end{bmatrix}. \tag{9}$$

According to the minimization problem in (8) and the structure of Γ in (9), we know that the greater Euclidean distance should be much less contribution toward the L_2-minimum problem. Therefore, the Tikhonov matrix in (8) can encode the importance of each sample. The least square error is usually utilized due to its mathematical tractability and low computational cost [18], [25], [32]. Therefore, we solve (8) by the least square error as

$$g = (Z^T Z + \lambda \Gamma^T \Gamma)^{-1} Z^T x. \tag{10}$$

D. Classification Rule of RDU

Suppose that the valuable sample z_i belongs to the cth class, i will belong to Θ_c, that is, $i \in \Theta_c$. The valuable samples have different contributions to represent the test sample. The number of valuable samples in each class is different. Therefore, the sum of the contributions of the valuable samples from each class is used to classify the test sample. The sum of the contribution of the cth class is described as

$$s^c = \sum_{i \in \Theta_c} z_i g_i. \tag{11}$$

Next, the residual of the test sample x and the cth class can be calculated as

$$d_c = ||x - s^c||. \tag{12}$$

The smaller residual means the greater contribution to represent the test sample. Thus, the RDU classifier selects the class with the minimum one as given by

$$\min_{c*} d_c, c = 1, 2, \ldots, M. \tag{13}$$

The detailed classification procedures of the RDU classifier are described as in Algorithm 1.

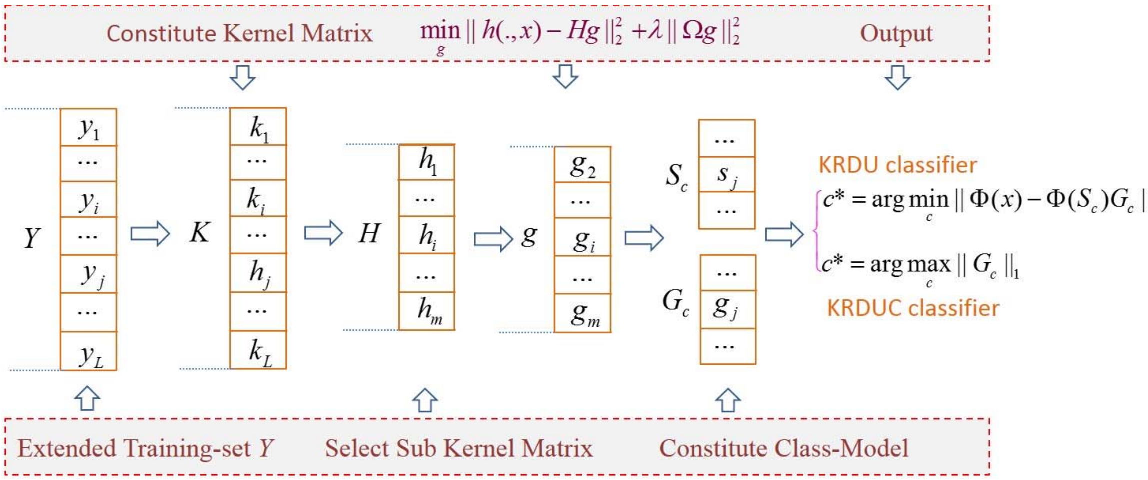

Fig. 2. Flowchart of the proposed KRDU and KRDUC classifiers.

1) Advantages of RDU: RDU uses the Tikhonov matrix to encode the importance of each sample for solving the L_2-minimum problem. The training sample with a greater Euclidean distance should have a much less contribution toward the L_2-minimum problem. This is helpful for classification. However, the DU classifier fails to consider this.

E. Classification Rule of RDUC

Similar to RDU classifier, the RDUC classifier, described in Algorithm 2, also computes the sum of the contribution of the cth class for classification. However, RDUC does not calculate the regularized residual of each class while it computes the sum of regularized coefficients of each for classification. The sum of the regularized coefficients of the cth class is described as

$$d_c = \sum_{i \in \Theta_c} g_i \tag{14}$$

where the definition of Θ_c is the same as that in Section II-D. The larger sum of the regularized coefficients means the greater contribution to represent the test sample. Therefore, the RDUC classifier selects the class with the max sum of the regularized coefficients by

$$\max_{c*} d_c, \quad c = 1, 2, \ldots, M. \tag{15}$$

The detailed classification procedures of the RDUC classifier are shown as in Algorithm 2.

1) Advantages of RDUC: Similar to RDU, RDUC also uses the Tikhonov matrix to encode the importance of each sample for solving the L_2-minimum problem. Thus, the Tikhonov matrix is used to solve the L_2-minimum problem that is helpful for classification.

Moreover, RDUC utilizes the sum of regularized coefficients of each class for classification, which is better than the residual in DU and RDU for action recognition. We will prove this in our experiments.

Algorithm 2 RDUC Classifier

Require: The test sample x and the original prototype set $X = \{x_i^c, c = 1, 2, \ldots, M, i = 1, 2, \ldots, n_c\}$

Ensure: The class index of x.

1: Extend the training set X as the Y by (5).
2: Select m training samples as Z by (7).
3: Calculate the Tikhonov matrix by (9).
4: The regularized coefficient g is computed by (10).
5: Compute the regularized residual of each class according to (14).
6: The RDUC classifier will classify the test sample x into the class with the max sum of regularized coefficients by (15).

III. Proposed KRDU and KRDUC Classifiers

In this section, we propose the KRDU classifier and KRDUC classifier. The flowchart of two classifiers is shown in Fig. 2. We first describe the kernel function in Section III-A, the method of constituting the kernel matrix and test vector in Section III-B, the method of selecting the valuable subkernel matrix in Section III-C, regularized solution for minimum problem in Section III-D, and the method of constituting the model for each class in Section III-E. The classification rules of the KRDU and KRDUC classifiers are then described in Sections III-F and III-G, respectively.

A. Kernel Function

The kernel method is a well-known mapping technique that maps a linear method to a high-dimensional nonlinear counterpart. This paper uses the most popular Gaussian radial basis function (RBF) kernel for classification. The RBF kernel can be represented as

$$k(x, y) = \phi(x)^T \phi(y) = \exp\left(-\frac{||x - y||^2}{\sigma}\right) \tag{16}$$

where x and y are any two original samples and σ is the parameter. In kernel methods, $\phi(*)$ is unknown. The only way to access the feature space is via $k(*, *)$.

B. Constitute Kernel Matrix and Test Vector

Suppose that there exists a nonlinear feature mapping function $\Phi(.) : \mathrm{R}^q \rightarrow \mathrm{R}^Q (q << Q)$. It maps the test sample x and extended training set Y in (5) into a high-dimensional feature space as

$$\begin{aligned} x &\rightarrow \Phi(x) \\ Y &\rightarrow \Phi(Y) = [\,\Phi(y_1)\ \cdots\ \Phi(y_i)\ \cdots\ \Phi(y_L)\,]. \end{aligned} \tag{17}$$

Applying the kernel trick in (16), we have a symmetric matrix $K = \Phi(Y)^T \Phi(Y)$, which can be computed as

$$K = \begin{bmatrix} k(y_1, y_1) & k(y_1, y_2) & \cdots & k(y_1, y_L) \\ k(y_2, y_1^1) & k(y_2, y_1) & \cdots & k(y_2, y_1) \\ \cdots & \cdots & \cdots & \cdots \\ k(y_L, y_1) & k(y_L, y_1) & \cdots & k(y_L, y_L) \end{bmatrix}. \tag{18}$$

The test vector $k(., x) = \Phi(Y)^T \Phi(x)$ can be computed as

$$\begin{aligned} k(., x) &= \phi(Y)^T \phi(x) \\ &= [\,k(y_1, x)\ k(y_2, x)\ \cdots\ k(y_L, x)\,]^T. \end{aligned} \tag{19}$$

Note that the kernel matrix K can be computed in advance without the test sample x. Therefore, the computation cost of constituting K may be ignored in the test phase.

C. Select Valuable Subkernel Matrix and Test Vector

The elements in the kernel matrix of the extended training set have different representation abilities for the test vector. Some elements in the kernel matrix are treated as improper elements because they have negative effects in representing the test vector. Therefore, the valuable elements with positive effects need to be selected from the kernel matrix. Similar to the RDU and RDUC classifiers, the valuable training samples close to the test sample are quite helpful for classifying the test sample. That is, the corresponding the kernel matrix of the chosen training samples is also helpful for classification. In order to select the valuable subkernel matrix H from the entire kernel matrix K, the kernel-based Euclidean distance between the high-dimensional test sample and each high-dimensional training sample will be computed as

$$\begin{aligned} \phi(d_i^c) &= ||\phi(x) - \phi(y_i^c)|| \\ &= \sqrt{\left(\phi(x) - \phi(y_i^c)\right)^T \left(\phi(x) - \phi(y_i^c)\right)} \\ &= \sqrt{k(x, x) - 2k\left(x, y_i^c\right) + k\left(y_i^c, y_i^c\right)} \end{aligned} \tag{20}$$

where $c \in \{1, 2, \ldots, M\}$ and $i \in \{1, 2, \ldots, n_c\}$. The kernel-based Euclidean distance in (20) can be treated as similarity between the high-dimensional test sample and the corresponding high-dimensional training sample. The smallest kernel-based distance represents the most similarity. The chosen m training samples with the smallest distance are used to constitute the new training space $F = [\,f_1\ \cdots\ f_i\ \cdots\ f_m\,] \in R^{q \times m}$. Suppose that the index of f_i in the entire training set Y is b_i. For example, f_i is the 60th elements of Y, then $b_i = 60$. The index of the chosen subtraining set F in the entire training set Y is represented by

$$B = [\,b_1\ \cdots\ b_i\ \cdots\ b_m\,] \in R^{1 \times m}. \tag{21}$$

Next, the valuable subkernel matrix H with a size of $m \times m$ will be chosen from the entire kernel matrix K. The element of valuable subkernel matrix H can be described as

$$H_{i,j} = K_{b_i, b_j} \quad i, j = 1, 2, \ldots, m. \tag{22}$$

Note that the valuable subkernel matrix H is the corresponding kernel matrix of F. However, H is not computed by F because the computation cost of selecting the elements from K is much less than that of being computed by F.

The subtest vector $h(., x)$ with a size of m will be chosen from the entire test vector $k(., x)$. The element of $h(., x)$ is

$$h(., x)_i = k(., x)_{b_i} \quad i = 1, 2, \ldots, m. \tag{23}$$

The remaining elements of the kernel matrix and test vector have negative effects for classification and will be discarded.

D. Regularized Solution for Minimum Problem

Similar to RDU and RDUC classifiers, KRDU and KRDUC classifiers also use the L_2-minimum problem for classification. These two classifiers have the same objective function defined as follows:

$$\min ||h(., x) - Hg||_2^2 + \lambda ||\Omega g||_2 \tag{24}$$

where λ is a global regularization parameter. Ω is a kernel-based Tikhonov matrix defined by the chosen training space and the test sample x. Ω can be computed by

$$\Omega = \begin{bmatrix} ||\phi(x) - \phi(f_1)|| & 0 & 0 \\ \cdots & \cdots & \cdots \\ 0 & 0 & ||\phi(x) - \phi(f_m)|| \end{bmatrix} \tag{25}$$

where

$$\begin{aligned} ||\phi(x) - \phi(f_i)|| &= \sqrt{(\phi(x) - \phi(f_i))^T (\phi(x) - \phi(f_i))} \\ &= \sqrt{k(x, x) - 2k(x, f_i) + k(f_i, f_i)}. \end{aligned} \tag{26}$$

According to the minimization problem in (24) and the structure of Ω in (25), we know that the greater kernel-based Euclidean distance should be much less contribution toward the L_2-minimum problem. Therefore, the kernel-based Tikhonov matrix in (25) can encode the importance of each training sample. The least square error is also used to solve (24) as

$$g = (H^T H + \lambda \Omega^T \Omega)^{-1} H^T h(., x). \tag{27}$$

E. Constitute Model for Each Class

Suppose that the chosen sample f_i belongs to the cth class, i will belong to Θ_c, and marked as $i \in \Theta_c$. The sample set and the corresponding parameter from the cth class are described as

$$\begin{aligned} S_c &= [\cdots\ f_i\ \cdots], \quad i \in \Theta_c \\ G_c &= [\cdots\ g_i\ \cdots]^T, \quad i \in \Theta_c. \end{aligned} \tag{28}$$

The cth class-based kernel matrix E_c and test vector $e(., x)$ are described as

$$\begin{aligned} E_c &= \phi(S_c)^T \phi(S_c) \\ e(., x) &= \phi(S_c)^T \phi(x). \end{aligned} \tag{29}$$

Algorithm 3 KRDU Classifier

Require: The test sample x and the original prototype set $X = \{x_i^c, c = 1, 2, \ldots, M, i = 1, 2, \ldots, n_c\}$

Ensure: The class index of x.

1: Constitute the kernel matrix K by (18) and the test vector $k(., x)$ by (19).
2: Select valuable submatrix H by (22) and subtest vector $h(., x)$ by (23).
3: Calculate the kernel-based Tikhonov matrix Γ by (25).
4: The regularized coefficient g is computed by (27).
5: Constitute the class model of each by (29).
6: Compute the regularized residual of each class by (30) and classify the test sample x into the class with the minimum value by (31).

Algorithm 4 KRDUC Classifier

Require: The test sample x and the original prototype set $X = \left\{x_i^c, c = 1, 2, \ldots, M, i = 1, 2, \ldots, n_c\right\}$

Ensure: The class index of x.

1: Constitute the kernel matrix K by (18) and the test vector $k(., x)$ by (19).
2: Select valuable submatrix H by (22) and subtest vector $h(., x)$ by (23).
3: Calculate the kernel-based Tikhonov matrix Γ by (25).
4: The regularized coefficient g is computed by (27).
5: Constitute the class model of each by (29).
6: Compute the sum of kernel regularized coefficients of each class by (32) and classify the test sample x into the class with the max value by (33).

Note that the class-based kernel matrix E_c and test vector $e(., x)$ may be directly chosen from H and $h(., x)$, and do not need to be computed. Therefore, the computation cost of constituting E_c and $e(., x)$ is quite low.

F. Classification Rule of KRDU

For the KRDU classifier presented in Algorithm 3, the residual of the test sample x and the cth class can be calculated by

$$\begin{aligned} d_c &= ||\phi(x) - \phi(S_c)G_c|| \\ &= \sqrt{(\phi(x) - \phi(S_c)G_c)^T(\phi(x) - \phi(S_c)G_c)} \\ &= \sqrt{k(x, x) - 2{G_c}^T k(x, S_c) + {G_c}^T k(S_c, S_c)G_c} \\ &= \sqrt{1 - 2{G_c}^T e(., x) + {G_c}^T E_c G_c}. \end{aligned} \tag{30}$$

The smaller residual means the greater contribution to represent the test sample. Thus, the KRDU classifier selects the class with the minimum residual given by

$$\min_{c*} d_c, \quad c = 1, 2, \ldots, M. \tag{31}$$

The detailed classification procedures of the KRDU classifier are described as in Algorithm 3.

1) Advantages of KRDU: KRDU uses the kernel-based Tikhonov matrix to encode the importance of each sample for solving the L_2-minimum problem. The training samples with a greater kernel-based Euclidean distance have much less contribution toward the L_2-minimum problem. Therefore, the regularized solution of the L_2-minimum problem with Tikhonov matrix is helpful for classification.

Moreover, KRDU also utilizes the nonlinear information for classification. It has much less computation cost than the linear information in DU and RDU because the number of samples is much less than the dimension of samples and the entire kernel matrix can be computed in advance without the test sample x. Our experiments will prove this later.

G. Classification Rule of KRDUC

For the KRDUC classifier presented in Algorithm 4, the sum of kernel-based regularized coefficients of the cth class is defined by

$$d_c = ||G_c||_1. \tag{32}$$

The greater sum of coefficients means the greater contribution to represent the test sample. Thus, the KRDUC classifier selects the class with the max sum given by

$$\max_{c*} d_c, \quad c = 1, 2, \ldots, M. \tag{33}$$

The detailed classification procedures of the KRDUC classifier are described as in Algorithm 4.

1) Advantages of KRDUC: Similar to KRDU, KRDUC also uses the kernel-based Tikhonov matrix to encode the importance of each sample for solving the L_2-minimum problem. The regularized solution of the L_2-minimum problem with Tikhonov matrix is helpful for classification.

KRDUC utilizes the nonlinear information for classification as well. It has much less computation cost than the linear information in DU and RDU, which is also similar to KRDU.

Furthermore, KRDUC applies the sum of kernel-based regularized coefficients of each class for classification. This is better than the residual in KRDU for action recognition. The experiments show this later.

IV. Experimental Results

To verify the performance of the four proposed classifiers for action recognition, they are compared with several existing methods on four well-known databases. These methods include the SVM [15], SRC [30], SSRC [34], NFLS-I [26], NFLS-II [26], KSR [41], [42], KSRC [43], fisher discrimination dictionary learning (FDDL) [39], dictionary pair learning (DPL) [49], and DU [40]. For SVM, we use the function provided by MATLAB. For other comparison methods, we use the source codes provided by the corresponding authors. The optimization parameter is set to 0.001 in our experiments. Parameter for RBF kernel function is set as follows:

$$\sigma = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{||y_i - y_{mean}||^2 \times 9} \tag{34}$$

where y_{mean} is the mean sample of all training samples in Y.

Fig. 3. Some action images from (a) KTH action database, (b) UCF50 action database, (c) UCF sport action database, and (d) HMDB51 action database, respectively.

Note that this paper pays attention to proposing the better classifiers for action recognition. The four proposed classifiers and the DU classifier can be treated as the improvements of the sparse-representation-based methods. Thus, we choose several well-known classifiers and improved sparse-representation-based methods for fair comparison.

A. Action Recognition on KTH Database

The KTH action database [14] contains six types of human actions (walking, jogging, running, boxing, hand waving, and hand clapping). They are performed several times by 25 persons in four different scenarios: outdoors s1, outdoors with scale variation s2, outdoors with different clothes s3, and indoors s4. In the experiments, each kind of human action contains 100 videos chosen from the database. Several action images of the KTH action database are shown in Fig. 3(a).

In the experiment, the high-dimensional representation of each video provided by [13] is used for the classification task. The experiment settings are as follows: choose 5, 10, and 15 video features of each person from KTH action database as the prototype set, and the rest videos are used as the test set. We evaluate the performance of several classifiers for action recognition. The experimental results are shown in Table I and Fig. 4(a)–(c). It can be seen that the four proposed classifiers obtain the better performance than other classifiers for action recognition. The DU classifier obtains the competitive performance compared with other SRC-based methods. The RDUC classifier outperforms the DU classifier by 2.66%.

TABLE I
RECOGNITION RATES OF SEVERAL CLASSIFIERS ON KTH ACTION DATABASE

Classifier	5	10	15	mean
SVM [15]	81.40	88.89	93.83	88.04
SRC [31]	91.40	90.56	94.90	92.29
SSRC [34]	92.11	90.00	96.28	92.80
NFLS-I [26]	90.53	89.63	93.14	91.10
NFLS-II [26]	89.47	89.26	92.55	90.43
KSR [42]	91.75	92.04	93.92	92.57
KSRC [43]	89.83	91.67	92.75	91.42
FDDL [39]	91.18	91.30	96.08	92.85
DPL [49]	91.08	91.30	96.00	92.79
DU [40]	90.70	90.37	95.29	92.12
RDU	92.63	91.48	96.28	93.46
RDUC	92.81	**94.26**	**97.26**	**94.78**
KRDU	92.28	92.04	95.49	93.27
KRDUC	**93.16**	92.59	96.08	93.94

B. Action Recognition on UCF50 Database

The UCF50 action database [46] has 50 action categories, containing real-world videos taken from the YouTube Web site. This database can be considered as an extended version of the YouTube database. The 50 action categories range from common sports to daily life exercises. These videos are split into 25 groups, and each group contains at least four action clips. The video clips of the same group may have common features, such as the same person, similar background, or sim-

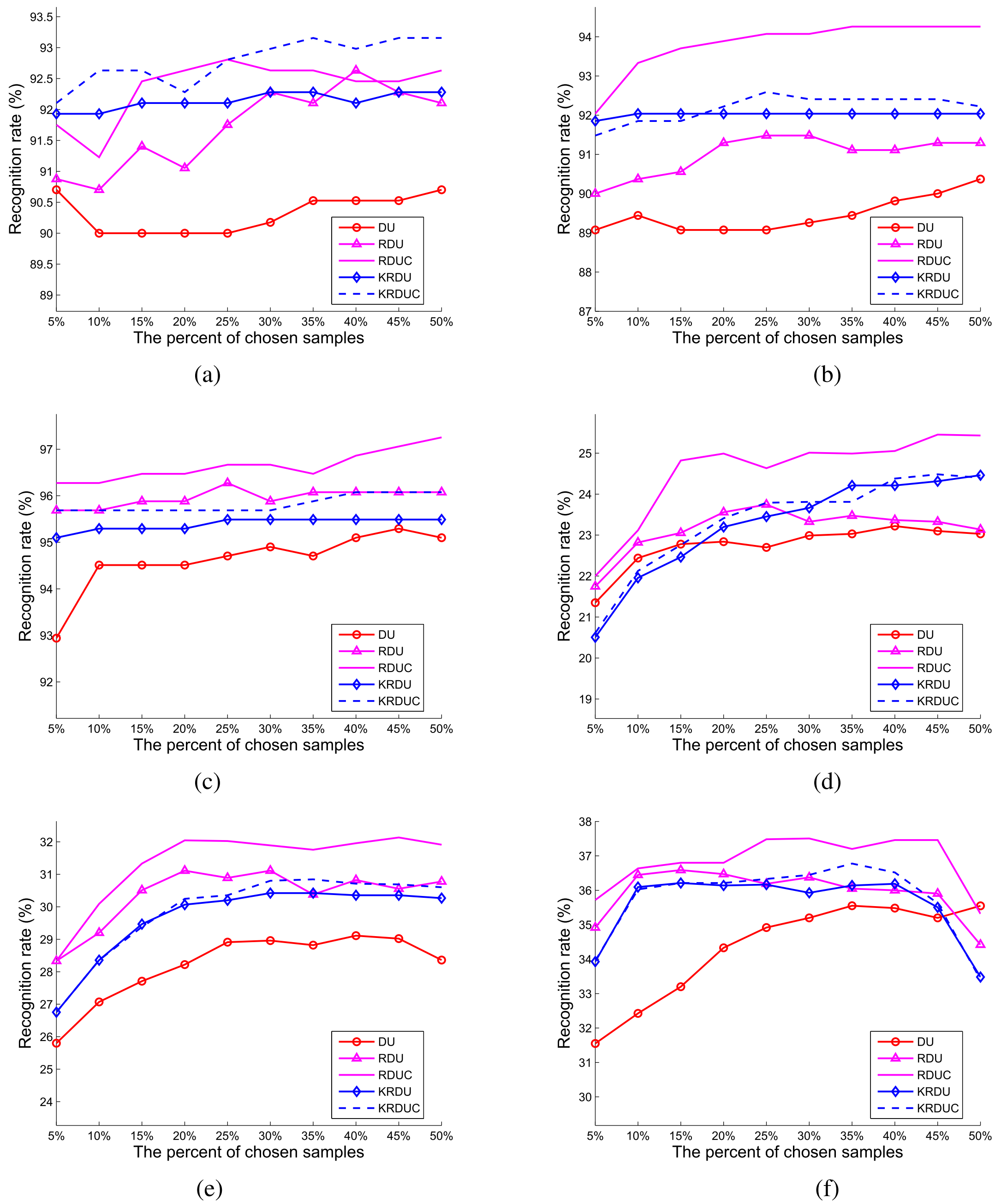

Fig. 4. Recognition rates of several classifiers. (a) 5-randomly, (b) 10-randomly, and (c) 15-randomly scheme on KTH action database. (d) 5-randomly, (e) 10-randomly, and (f) 15-randomly scheme on UCF50 action database. The horizontal axis denotes the percent of chosen samples. The vertical axis denotes the recognition rates.

ilar viewpoint. In the experiments, each kind of human action category contains 100 videos chosen from the database. There are 5000 videos in total for the experiments. Some action images of the UCF50 action database are shown in Fig. 3(b).

In the experiment, each video is disposed as high-dimensional representation according to [13] for the classification task. The experiment settings are as follows: 5, 10, and 15 video features of each action category are selected from the UCF50 action database as the training set, and the rest videos are used as the test set. We evaluate the performance of several classifiers for action recognition. The experimental results are shown in Table II and Fig. 4(d)–(f). We can observe that the proposed four classifiers have the better performance than the existing classifiers for action recognition. The RDUC classifier obtains the best performance among the four proposed classifiers and outperforms the DU classifier by 2.41%.

C. Action Recognition on HMDB51 Database

The HMDB51 action database [48] was collected from a variety of sources ranging from digitized movies to YouTube videos. This action database includes simple facial actions, general body movements, and human interactions. In the experiments, each kind of human action contains 100 videos chosen from the database. Several action images of the HMDB51 action database are shown in Fig. 3(d).

TABLE II
RECOGNITION RATES OF SEVERAL CLASSIFIERS ON UCF50 ACTION DATABASE

Classifier	5	10	15	mean
SVM [15]	13.16	18.84	25.53	19.18
SRC [31]	23.03	27.02	31.65	27.23
SSRC [34]	23.03	26.40	29.25	26.23
NFLS-I [26]	17.93	22.76	28.19	22.96
NFLS-II [26]	19.43	22.93	28.09	23.48
KSR [42]	17.52	19.33	22.28	19.71
KSRC [43]	16.61	18.36	21.08	18.68
FDDL [39]	23.16	29.33	34.65	29.05
DPL [49]	21.90	29.00	34.20	28.37
DU [40]	23.22	29.11	35.55	29.29
RDU	23.75	31.11	36.59	30.48
RDUC	**25.45**	**32.13**	**37.51**	**31.70**
KRDU	24.46	30.42	36.21	30.36
KRDUC	24.48	30.84	36.78	30.70

TABLE III
RECOGNITION RATES OF SEVERAL CLASSIFIERS ON HMDB51 ACTION DATABASE

Classifier	5	10	15	mean
SVM [15]	4.87	6.36	8.88	6.70
SRC [31]	9.72	12.64	14.97	12.44
SSRC [34]	9.45	12.18	14.90	12.18
NFLS-I [26]	8.90	10.68	12.78	10.79
NFLS-II [26]	8.79	10.85	12.78	10.81
KSR [42]	6.91	9.19	10.43	8.84
KSRC [43]	6.89	8.67	9.53	8.36
FDDL [39]	9.79	14.19	17.43	13.80
DPL [49]	9.40	13.70	17.40	13.50
DU [40]	9.91	14.21	16.80	13.64
RDU	9.85	14.44	17.07	13.79
RDUC	9.87	**14.77**	17.14	13.93
KRDU	9.78	14.68	17.60	14.02
KRDUC	**9.93**	14.51	**17.86**	**14.10**

In the experiment, each video is disposed as high-dimensional representation according to [13] for classification task. The experiment settings are as follows: 5, 10, and 15 video features of each action are chosen from the HMDB51 action database as the prototype set, and the rest videos are used as the test set. We evaluate the performance of several classifiers for action recognition. The experimental results are shown in Table III and Fig. 5(a)–(c). From the results, we can conclude that the four proposed classifiers all obtain the better performance compared with existing classifiers for action recognition. Because the HMDB51 action database is quite complex, the recognition rates of several classifiers are quite low.

D. Action Recognition on UCF Sport Database

The UCF sports action database [47] includes ten human actions, which contain swinging (on the pommel horse and on the floor), diving, kicking (a ball), weight lifting, horse riding, running, skateboarding, swinging (at the high bar), golf swinging, and walking. This database consists of 150 video samples to show a large intra-class variability. Some action images of the UCF50 action database are shown in Fig. 3(c).

In the experiment, the high-dimensional representation of each video provided by [13] is used for the classification task. The experiment settings are as follows: choose five, six, and seven video features of each action from UCF sports database as the prototype set, and the rest videos are used as the test set. We evaluate the performance of several classifiers for action recognition. The experimental results are shown in Table IV and Fig. 5(d)–(f). We can observe that the proposed RDUC classifier obtains the best performance among all compared classifiers for action recognition. The recognition rate of the RDUC classifier outperforms that of the DU classifier by 3.65%.

TABLE IV
RECOGNITION RATES OF SEVERAL CLASSIFIERS ON UCF SPORT ACTION DATABASE

Classifier	5	10	15	mean
SVM [15]	87.50	84.38	91.67	87.85
SRC [31]	87.50	87.50	95.83	90.28
SSRC [34]	87.50	87.50	95.83	90.28
NFLS-I [26]	85.00	81.25	87.50	84.58
NFLS-II [26]	82.50	78.13	87.50	82.71
KSR [42]	85.00	81.25	91.67	85.97
KSRC [43]	77.50	81.25	83.33	80.69
FDDL [39]	87.50	87.50	**95.83**	90.28
DPL [49]	85.00	87.50	**95.83**	89.44
DU [40]	85.00	84.38	**95.83**	88.40
RDU	85.00	87.50	**95.83**	89.44
RDUC	**90.00**	**90.63**	**95.83**	**92.15**
KRDU	87.50	87.50	**95.83**	90.28
KRDUC	87.50	87.50	**95.83**	90.28

E. Computation Cost

Here, the computation costs of the proposed four classifiers and the DU classifier are discussed. We show the running time of the classifiers on the UCF50 action databases described in Section IV-B. This experiment is executed on MATLAB 2013a in Windows 8 on PC with 3.50 GHz and 16-GB RAM.

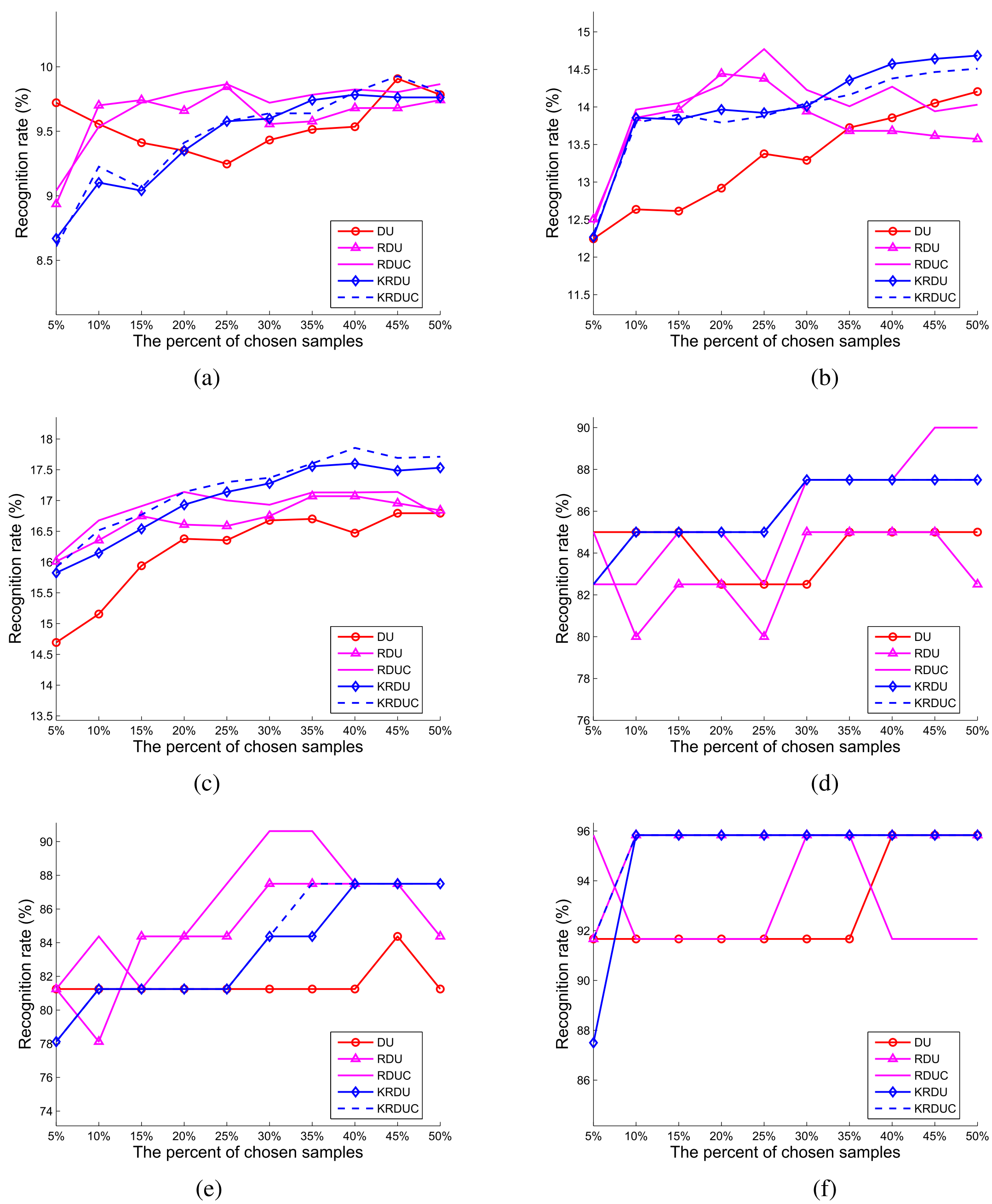

Fig. 5. Recognition rates of several classifiers. (a) 5-randomly, (b) 10-randomly, and (c) 15-randomly scheme on HMDB51 action database. (d) 5-randomly, (e) 6-randomly, and (f) 7-randomly scheme on UCF sports database. The horizontal axis denotes the percent of chosen samples. The vertical axis denotes the recognition rates.

In this experiment, we test the computation costs of these classifiers on the UCF50 action databases. The five video features are chosen as the training set, and the rest video features are used as the test set. Because the recognition rates are shown in Table II, we provide only the running time in this section. The results are exhibited in Table V. We observe that the DU, RDU, and RDUC classifiers obtain the competitive computation costs. The KRDU and KRDUC classifiers have the less computation costs than DU, RDU, and RDUC. When the number of chosen samples becomes larger, the computation costs of DU, RDU, and RDUC increase significantly, while the computation costs of KRDU and KRDUC increase slightly.

F. Evaluation With Various Kernel Parameters

In (16), there is a kernel parameter in the kernel function. The appropriate value of the kernel parameter may be helpful for the proposed classifiers. Therefore, we further show the performance of the proposed KRDU and KRDUC classifiers with various kernel parameters in this section. The KTH action database is used in this experiment. Similar to Section IV-A, the high-dimensional representation of each video provided by [13] is used for the classification task. Select five video features of each person from the KTH action database as the prototype set, and the rest videos are used as the test set. In the experiment, the percentage of chosen samples is 30%. The value of the kernel parameter is set as 0.000001,

TABLE V
COMPUTATION COSTS OF SEVERAL CLASSIFIERS ON THE UCF50 ACTION DATABASES (UNIT: SECONDS)

Classifier	DU	RDU	RDUC	KRDU	KRDUC
5%	302.93	303.57	302.01	122.88	122.16
10%	329.66	329.47	328.06	126.26	124.92
15%	350.64	350.64	349.80	125.33	125.24
20%	376.41	376.86	374.27	128.26	126.33
25%	410.40	410.40	408.87	128.35	127.41
30%	459.35	460.26	457.64	130.29	128.34
35%	502.92	503.35	501.30	131.13	129.54
40%	556.16	556.67	554.86	132.23	130.71
45%	605.24	605.46	603.61	132.66	131.41
50%	653.45	653.94	652.15	133.97	132.59

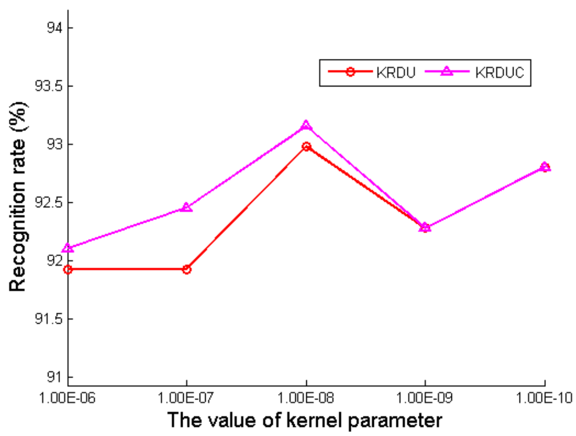

Fig. 6. Performance of the proposed KRDU and KRDUC classifiers with various kernel parameter settings. The horizontal axis denotes the value of the kernel parameter. The percentage of chosen samples is 30%.

0.0000001, 0.00000001, 0.000000001, and 0.0000000001. Fig. 6 shows the experimental results. We can observe that the kernel parameter indeed has the significant impact for the proposed KRDU and KRDUC classifiers. It is noted that we use only the parameter value computed by (34) for the proposed classifiers in the above experiments (Sections IV-A–IV-D). If we try to find the most appropriately value of the kernel parameter, the proposed classifiers can obtain the better performance.

G. Observations and Discussion

According to the above experiments and analysis, we have several main observations and discussions as follows.

1) This paper pays attention to proposing the better classifier (rather than the better action features) to improve the performance of action recognition. Moreover, the proposed four classifiers and the DU classifier may be treated as the extensions of the sparse representation. Therefore, all comparison methods can be treated as classifiers or the extension of the sparse representation such that the comparison is fair. For the methods based on feature selection [4], [11], we think that the feature could be used in our classifiers such that the better performance could be obtained. However, the comparison is not necessary.
2) The proposed classifiers (RDU, RDUC, KRDU, and KRDUC) outperform DU and several state-of-the-art methods on four action databases. Meanwhile, the computation costs of KRDU and KRDUC are lower than that of DU because the number of samples is much less than the dimension of samples and the entire kernel matrix can be computed in advance without the test sample. This advantage becomes significant when the number of chosen samples increases.
3) Action recognition is critically dependent on features and classifiers. After obtaining the feature, action recognition system is similar to other recognition tasks, such as face recognition [35], [40] and object recognition. This observation can be proved in [39] and [23]. They both treat action recognition as the general classification task with the specific feature. In this paper, the high-dimensional representation of each video provided by [13] is used for the classification task. Therefore, the experimental results prove that the proposed methods have better performance than several state-of-the-art classifiers.

V. CONCLUSION

This paper has proposed four novel classifiers for action recognition, including the RDU, RDUC, KRDU, and KRDUC classifiers. To improve the classification performance, the proposed classifiers not only generate the virtual sample but also utilize the Tikhonov matrix to encode the importance of each sample. Moreover, the KRDU and KRDUC classifiers further include the kernel-based nonlinear information for classification. Thus, the proposed classifiers have achieved the better recognition rates than the DU classifier and several state-of-the-art methods. The experimental results have confirmed the recognition performance of the four proposed classifiers, and also prove that the computation costs of the KRDU and KRDUC classifier are much less than that of the DU classifier.

REFERENCES

[1] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, May 2013.
[2] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3169–3176.
[3] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 650–663.
[4] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 3551–3558.
[5] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, nos. 2–3, pp. 107–123, 2005.
[6] M. Bregonzio, S. Gong, and T. Xiang, "Recognising action as clouds of space-time interest points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 1948–1955.
[7] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2008, pp. 275:1–275:10.

[8] A. Hauptmann, R. Yan, W.-H. Lin, M. Christel, and H. Wactlar, "Can high-level concepts fill the semantic gap in video retrieval? A case study with broadcast news," *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 958–966, Aug. 2007.

[9] S. Ali, A. Basharat, and M. Shah, "Chaotic invariants for human action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.

[10] D. Ramanan and D. A. Forsyth, "Automatic annotation of everyday movements," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 16. 2003, pp. 1547–1554.

[11] A. Iosifidis, A. Tefas, and I. Pitas, "Semi-supervised classification of human actions based on neural networks," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, 2014, pp. 1336–1341.

[12] L.-J. Li, H. Su, L. Fei-Fei, and E. P. Xing, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2010, pp. 1378–1386.

[13] S. Sadanand and J. J. Corso, "Action bank: A high-level representation of activity in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1234–1241.

[14] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th Int. Conf. Pattern Recognit.*, vol. 3. Aug. 2004, pp. 32–36.

[15] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

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

[17] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. Jun. 2003, pp. I-11–I-18.

[18] 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.

[19] Q. Feng and Y. Zhou, "Iterative linear regression classification for image recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 1566–1570.

[20] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[21] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 218–233, Feb. 2003.

[22] X. Chai, S. Shan, X. Chen, and W. Gao, "Locally linear regression for pose-invariant face recognition," *IEEE Trans. Image Process.*, vol. 16, no. 7, pp. 1716–1725, Jul. 2007.

[23] Q. Feng, Y. Zhou, and R. Lan, "Pairwise linear regression classification for image set retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4865–4872.

[24] S.-M. Huang and J.-F. Yang, "Kernel linear regression for low resolution face recognition under variable illumination," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 1945–1948.

[25] W. Li, E. W. Tramel, S. Prasad, and J. E. Fowler, "Nearest regularized subspace for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 477–489, Jan. 2014.

[26] 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.

[27] Q. Feng, Q. Zhu, L.-L. Tang, and J.-S. Pan, "Double linear regression classification for face recognition," *J. Mod. Opt.*, vol. 62, no. 4, pp. 288–295, 2015.

[28] S. M. Huang and J. F. Yang, "Linear discriminant regression classification for face recognition," *IEEE Signal Process. Lett.*, vol. 20, no. 1, pp. 91–94, Jan. 2013.

[29] S.-M. Huang and J.-F. Yang, "Unitary regression classification with total minimum projection error for face recognition," *IEEE Signal Process. Lett.*, vol. 20, no. 5, pp. 443–446, May 2013.

[30] 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.

[31] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.

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

[33] 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.

[34] W. Deng, J. Hu, and J. Guo, "In defense of sparsity based face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 399–406.

[35] Q. Feng *et al.*, "Superimposed sparse parameter classifiers for face recognition," *IEEE Trans. Cybern.*, [Online]. Available: http://ieeexplore.ieee.org/document/7390229/

[36] X. Fang, Y. Xu, X. Li, Z. Lai, and W. K. Wong, "Learning a nonnegative sparse graph for linear regression," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2760–2771, Sep. 2015.

[37] Z. Lai, Y. Xu, Q. Chen, J. Yang, and D. Zhang, "Multilinear sparse principal component analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1942–1950, Oct. 2014.

[38] Z. Lai, W. K. Wong, Y. Xu, J. Yang, and D. Zhang, "Approximate orthogonal sparse embedding for dimensionality reduction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 723–735, Apr. 2016.

[39] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based Fisher discrimination dictionary learning for image classification," *Int. J. Comput. Vis.*, vol. 109, no. 3, pp. 209–232, Sep. 2014.

[40] Y. Xu *et al.*, "Data uncertainty in face recognition," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1950–1961, Oct. 2014.

[41] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Kernel sparse representation for image classification and face recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 1–14.

[42] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Sparse representation with kernels," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 423–434, Feb. 2013.

[43] L. Zhang *et al.*, "Kernel sparse representation-based classifier," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1684–1695, Apr. 2012.

[44] Q. Feng, C. Yuan, J. Huang, and W. Li, "Center-based weighted kernel linear regression for image classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2015, pp. 3630–3634.

[45] Q. Feng and Y. Zhou, "Kernel combined sparse representation for disease recognition," *IEEE Trans. Multimedia*, vol. 18, no. 10, pp. 1956–1968, Oct. 2016.

[46] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of Web videos," *Mach. Vis. Appl.*, vol. 24, no. 5, pp. 971–981, 2013.

[47] M. Rodriguez, "Spatio-temporal maximum average correlation height templates in action recognition and video summarization," Ph.D. dissertation, Univ. Central Florida, Orlando, FL, USA, 2010.

[48] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2556–2563.

[49] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Projective dictionary pair learning for pattern classification," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 793–801.

Qingxiang Feng received the B.S. degree in computer science and technology from Henan University, Kaifeng, China, and the M.S. degree in computer science and technology from Harbin Institute of Technology, Harbin, China. He is currently working toward the Ph.D. degree with the University of Macau, Macau, China.

He has authored over 20 scientific papers, of which 10 are indexed by science citation index. His research interests include pattern recognition and machine 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 an Associate Professor and the Director of the Vision and Image Processing Laboratory with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include chaotic systems, multimedia security, image processing and understanding, and machine learning.

Dr. Zhou was a recipient of the third place of the Macau Natural Science Award in 2014.