

# Collaborative-Representation-Based Nearest Neighbor Classifier for Hyperspectral Imagery

Wei Li, *Member, IEEE*, Qian Du, *Senior Member, IEEE*, Fan Zhang, and Wei Hu

**Abstract**—Novel collaborative representation (CR)-based nearest neighbor (NN) algorithms are proposed for hyperspectral image classification. The proposed methods are based on a CR computed by an  $\ell_2$ -norm minimization with a Tikhonov regularization matrix. More specific, a testing sample is represented as a linear combination of all the training samples, and the weights for representation are estimated by an  $\ell_2$ -norm minimization-derived closed-form solution. In the first strategy, the label of a testing sample is determined by majority voting of those with  $k$  largest representation weights. In the second strategy, local within-class CR is considered as an alternative, and the testing sample is assigned to the class producing the minimum representation residual. The experimental results show that the proposed algorithms achieve better performance than several previous algorithms, such as the original  $k$ -NN classifier and the local mean-based NN classifier.

**Index Terms**—Collaborative representation (CR), hyperspectral data, nearest neighbors (NNs), pattern classification.

## I. INTRODUCTION

IN STATISTICAL classification tasks, it is usual to make a simplified assumption that data abide by a normal or multimodal distribution. Thus, popular choices of statistical classifiers are the maximum-likelihood estimation classifier and the Gaussian mixture model classifier [1]. However, a single Gaussian or mixture Gaussian distribution under small training sample size situations may not be true, which often happens in hyperspectral imagery (HSI).

The nearest neighbor (NN) [2], [3] classifier, one of the simplest yet effective classification methods, has been widely used in HSI analysis. This nonparametric classifier does not need any prior knowledge about the density distribution of the data. The principle behind the NN classifier is to find a predefined number (e.g.,  $k$ ) of training samples closest in distance to the testing sample and assign the majority category label according to its  $k$  nearest training samples. The distance usually employs the standard Euclidean distance. Several extensions of this classifier have been studied. In [4], the  $k$ -NN rule was extended

to a local mean-based NN (LMNN) classifier. In [2], Euclidean metric in a low-dimensional space was modified to minimize the variance of given classes for the  $k$ -NN classifier. In [5], cosine-based nonparametric feature extraction was developed to include a weight function in the within- and between-class scatter matrices for the  $k$ -NN classifier. In [6], a variant of the  $k$ -NN classifier based on the maximal margin principle has been discussed. In [3], local manifold learning has been combined with  $k$ -NN to improve HSI classification.

Recently, sparse representation-based classification [7] has been proposed for robust face recognition. The basic idea is that a testing sample can be represented as a linear combination of all the training samples with a sparseness constraint, and the representation is recovered by an  $\ell_1$ -norm minimization, and the final label is assigned by checking against each class with minimum reconstruction error. Reference [8] argued that it is the “collaborative” nature of the approximation instead of “competitive” nature imposed by the sparseness constraint that actually improves classification accuracy. Collaborative representation (CR)-based classification has also been successfully applied in HSI analysis. In [9], a CR-based classifier, called nearest regularized subspace, was proposed for HSI classification. Note that the collaboration mentioned here means the atoms in a dictionary collaborate together to represent a single pixel; it is different from the “collaboration” reinforced in sparse unmixing in [10], where all the pixels collaborate together to choose the same set of atoms in the dictionary, if possible. In the latter, the data and the corresponding weights under consideration have to be matrices in the mathematical form, and the weight matrix is column-wise sparse; whereas in the former, they are vectors, and the weight vector is not sparse. The joint sparse models considering neighboring pixels for classification in [11] and [12] belong to the latter, where the term “joint” has the same meaning as “collaborative” as in “collaborative sparse unmixing” in [10].

In this letter, we propose two novel representation-based NN classifiers for HSI—CR-based NN (CRNN) and local within-class CR-based NN (LRNN). The proposed CRNN can be accomplished by two main steps: in the first step, a testing sample is represented as a linear combination of all the available training samples, and the weights for representation are estimated by an  $\ell_2$ -norm minimization-derived closed-form solution with Tikhonov regularization; in the second step, the label of the testing sample is determined by majority voting [13] of those with  $k$  largest representation weights. As an alternative, the proposed LRNN calculates the representation of the testing sample using the local class-specific training samples, which are obtained from the  $k$ -NN training samples

Manuscript received March 18, 2014; revised May 26, 2014 and July 15, 2014; accepted July 25, 2014. This work was supported by the National Natural Science Foundation of China under Grant NSFC-61302164.

W. Li, F. Zhang, and W. Hu are with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China (e-mail: liwei089@ieee.org).

Q. Du is with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762 USA (e-mail: du@ece.msstate.edu).

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/LGRS.2014.2343956

with Euclidean distance for each class; the testing pixel is assigned to the class producing the minimum representation residual. In both CRNN and LRNN, it is assumed that a larger weight means the corresponding training sample having more similar spectral characteristics to the testing sample, and the collaborative coefficients (in a weight vector) tend to have better discriminative ability compared with the Euclidean distance. The proposed algorithms are compared with several existing algorithms, such as the original  $k$ -NN and LMNN.

## II. RELATED WORK

Consider a data set with  $n$  training samples  $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$  in  $\mathbb{R}^d$  ( $d$ -dimensional feature space) and  $C$  class labels  $\omega_i \in \{1, 2, \dots, C\}$ . Let  $n_l$  be the number of available training samples for the  $l$ th class and  $\sum_{l=1}^C n_l = n$ .

### A. $k$ -NN

The NN classifier attempts to find the training sample nearest to the testing sample according to a given distance measure and assigns the former's class label to the latter. Commonly, Euclidean distance is used to measure the similarity between a training sample  $\mathbf{x}_i$  and a testing sample  $\mathbf{y}$ , i.e.,

$$d(\mathbf{x}_i, \mathbf{y}) = \|\mathbf{x}_i - \mathbf{y}\|_2^2. \quad (1)$$

The  $k$ -NN classifier [6] is a straightforward extension of the original NN classifier. Instead of using only one sample closest to the testing point  $\mathbf{y}$ , the  $k$ -NN classifier chooses the  $k$  nearest samples from training data  $\mathbf{X}$ , and majority voting [13] is employed to decide the class label of  $\mathbf{y}$ .

### B. LMNN

The LMNN [4] classifier is an extension of the original  $k$ -NN. The essence behind the method is that the local mean vector of the  $k$ -NNs in each class is used for classifying the testing sample. First, the  $k$ -nearest training samples per class from  $\mathbf{X}$  are selected via (1). Then, the local mean vector, represented as  $\tilde{\mathbf{y}}_l$ , is calculated using the  $k$ -NN training samples in the  $l$ th class, represented as  $\{\mathbf{x}_l^{(1)}, \mathbf{x}_l^{(2)}, \dots, \mathbf{x}_l^{(k)}\}$ , i.e.,

$$\tilde{\mathbf{y}}_l = \frac{1}{k} \sum_{j=1}^k \mathbf{x}_l^{(j)}. \quad (2)$$

Finally, after obtaining the local mean vector per class, the class label of  $\mathbf{y}$  is then determined according to the class that minimizes the residual. That is

$$\text{class}(\mathbf{y}) = \arg \min_{l=1, \dots, C} r_l(\mathbf{y}) \quad (3)$$

where  $r_l(\mathbf{y}) = \|\tilde{\mathbf{y}}_l - \mathbf{y}\|_2^2$  is the residual between the mean vector and the corresponding testing sample. Note that LMNN is equivalent to the 1-NN classifier when  $k = 1$ .

## III. PROPOSED CLASSIFIERS

### A. Proposed CRNN

CR [8] is based on the concept that the testing sample  $\mathbf{y}$  can be represented as a linear combination of all the training samples  $\mathbf{X}$ . Assume  $\boldsymbol{\alpha}$  is an  $n \times 1$  vector of weighting coefficients. The weight vector  $\boldsymbol{\alpha}$  for the linear combination is solved by an  $\ell_2$ -norm regularization, i.e.,

$$\boldsymbol{\alpha} = \arg \min_{\boldsymbol{\alpha}^*} \|\mathbf{y} - \mathbf{X}\boldsymbol{\alpha}^*\|_2^2 + \lambda \|\boldsymbol{\Gamma}_y \boldsymbol{\alpha}^*\|_2^2 \quad (4)$$

where  $\boldsymbol{\Gamma}_y$  is a biasing Tikhonov matrix, and  $\lambda$  is a global regularization parameter that balances the minimization between the residual part and the regularization term. Note that  $\boldsymbol{\alpha}^*$  is an estimate of  $\boldsymbol{\alpha}$  with size of  $n \times 1$ . Specifically, the regularization term is designed in the form of

$$\boldsymbol{\Gamma}_y = \begin{bmatrix} \|\mathbf{y} - \mathbf{x}^{(1)}\|_2 & & 0 \\ & \ddots & \\ 0 & & \|\mathbf{y} - \mathbf{x}^{(n)}\|_2 \end{bmatrix} \quad (5)$$

where  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$  are the columns of matrix  $\mathbf{X}$ . Then, the weight vector  $\boldsymbol{\alpha}$  can be determined in a closed-form solution as

$$\boldsymbol{\alpha} = (\mathbf{X}^T \mathbf{X} + \lambda^2 \boldsymbol{\Gamma}_y^T \boldsymbol{\Gamma}_y)^{-1} \mathbf{X}^T \mathbf{y}. \quad (6)$$

After obtaining  $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ , the  $k$  largest elements are found, from which the number of weights belonging to the  $l$ th class (denoted as  $N_l(\boldsymbol{\alpha})$ ). The final label of the testing sample is determined by majority voting as

$$\text{class}(\mathbf{y}) = \arg \max_{l=1, \dots, C} N_l(\boldsymbol{\alpha}). \quad (7)$$

As we can see, the weight vector  $\boldsymbol{\alpha}$  has a closed-form solution, resulting in low computational cost. Furthermore, the distance-weighted measurement adaptively adjusts the regularization when calculating a representation for each testing sample. If a weight in  $\boldsymbol{\alpha}$  is near 0, it indicates that the corresponding training sample is dissimilar to the testing sample; comparatively, the training sample who provides the largest weight is the most similar to the testing sample and can be viewed as the NN.

### B. Proposed LRNN

The proposed LRNN classifier can be considered as a further extension of LMNN. In LRNN, the local mean vector  $\tilde{\mathbf{y}}_l$  of (2) is replaced by a local within-class CR. First of all, the  $k$ -nearest training samples from class-specific  $\mathbf{X}_l$  (of size  $d \times n_l$ ) are selected via (1), represented as  $\mathbf{X}_l^{(k)} = \{\mathbf{x}_l^{(1)}, \mathbf{x}_l^{(2)}, \dots, \mathbf{x}_l^{(k)}\}$  (of size  $d \times k$ ). An approximation of the testing sample  $\mathbf{y}$  is represented via a linear combination of the chosen training samples per class,  $\mathbf{X}_l^{(k)}$ . That is, for each class, only from the training samples in class  $l$ , the new local "mean" vector  $\tilde{\mathbf{y}}_l$  is calculated as  $\tilde{\mathbf{y}}_l = \mathbf{X}_l^{(k)} \boldsymbol{\alpha}_l$ , and  $\boldsymbol{\alpha}_l$  is an  $k \times 1$  vector of weight

coefficients. The vector  $\alpha_l$  for the linear combination is solved by an  $\ell_2$ -norm regularization, i.e.,

$$\alpha_l = \arg \min_{\alpha_l^*} \left\| \mathbf{y} - \mathbf{X}_l^{(k)} \alpha_l^* \right\|_2^2 + \lambda \left\| \Gamma_{l,y} \alpha_l^* \right\|_2^2 \quad (8)$$

where  $\Gamma_{l,y}$  is a biasing Tikhonov matrix specific to class  $l$  and the current testing sample  $\mathbf{y}$ , and  $\lambda$  is a global regularization parameter that balances the minimization between the residual and the regularization term. Note that  $\alpha_l^*$  is an estimate of  $\alpha_l$  with size of  $k \times 1$ . Specifically, the regularization term is designed in the form

$$\Gamma_{l,y} = \begin{bmatrix} \left\| \mathbf{y} - \mathbf{x}_l^{(1)} \right\|_2 & & 0 \\ & \ddots & \\ 0 & & \left\| \mathbf{y} - \mathbf{x}_l^{(k)} \right\|_2 \end{bmatrix}. \quad (9)$$

Then, the weight vector  $\alpha_l$  can be recovered in a closed-form solution, i.e.,

$$\alpha_l = \left( \mathbf{X}_l^{(k)T} \mathbf{X}_l^{(k)} + \lambda^2 \Gamma_{l,y}^T \Gamma_{l,y} \right)^{-1} \mathbf{X}_l^{(k)T} \mathbf{y}. \quad (10)$$

Once we obtain the weight vector, the adaptive weighted representation of  $\mathbf{y}$  is  $\tilde{\mathbf{y}}_l = \mathbf{X}_l^{(k)} \alpha_l$ . The class label of  $\mathbf{y}$  is determined according to (3). Essentially, LMNN can be viewed as a special case of the proposed LRNN when all values in  $\alpha_l$  are the same with the sum equal to 1.

Note that although both the proposed CRNN and LRNN adopt CR, they are essentially different. On the one hand, the proposed CRNN belongs to distance-based classifiers, such as the  $k$ -NN classifier, which can be viewed as a simple implementation of locating the NNs in the feature space. The NNs are determined according to some similarity metric, such as the Euclidean distance in the  $k$ -NN classifier or the weight vector defined in CRNN. Comparatively, the proposed LRNN belongs to residual-based classifiers, as the LMNN, which assign the label of a testing pixel according to the minimum residual between the testing sample and its approximation. On the other hand, CR in CRNN needs all the training samples. However, in LRNN, a local within-class CR is employed; in other words, only some chosen training samples per class are used.

## IV. EXPERIMENTS AND ANALYSIS

### A. Experimental Data

The first experimental data set employed was collected by the Reflective Optics System Imaging Spectrometer sensor. The image, covering the city of Pavia, Italy, was collected under the HySens project managed by the German Aerospace Agency (DLR). The data have a spectral coverage from 0.43 to 0.86  $\mu\text{m}$  and a spatial resolution of 1.3 m. The scene used in the first experiment is the university area that has 103 spectral bands with a spatial coverage of  $610 \times 340$  pixels. There are nine classes (i.e., *Asphalt*, *Meadows*, *Gravel*, *Trees*, *Metal sheets*, *Bare soil*, *Bitumen*, *Bricks*, and *Shadow*), and 50 training samples and 650 testing samples per class were randomly selected from the ground truth map.

The second experimental hyperspectral data set employed was acquired using a HyMap sensor. The scenario is an area close to Purdue University cropped into a subimage with  $377 \times$

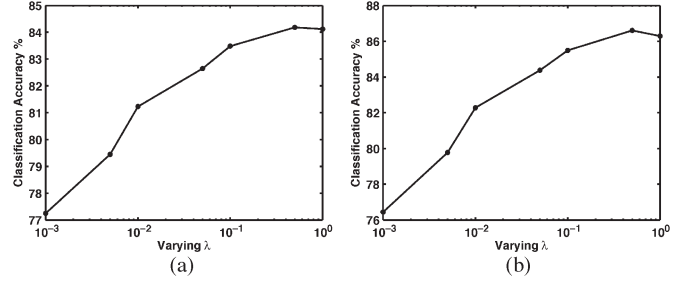


Fig. 1. Parameter tuning of  $\lambda$  for the proposed CRNN and LRNN using the University of Pavia data. (a) Proposed CRNN. (b) Proposed LRNN.

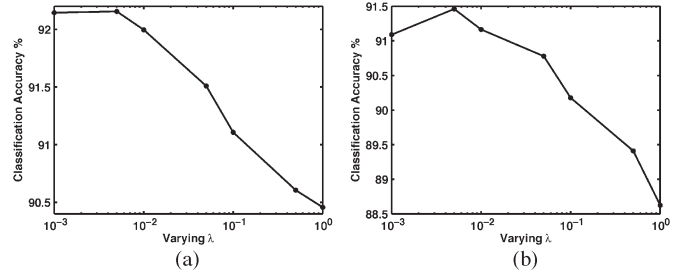


Fig. 2. Parameter tuning of  $\lambda$  for the proposed CRNN and LRNN using the HyMap Purdue data. (a) Proposed CRNN. (b) Proposed LRNN.

512 pixels and 126 bands spanning the wavelength interval 0.45–2.5  $\mu\text{m}$  with a spatial resolution of 3.5 m. In our experiment, there are six classes (i.e., *Road*, *Grass*, *Shadow*, *Soil*, *Tree*, and *Roof*), with a total of 60 training samples (10 per class) and 5635 testing samples (1287, 1114, 219, 379, 1351, and 1285 per class, respectively).

### B. Parameter Tuning of $\lambda$

We compare the classification performance of the proposed CRNN and LRNN with the traditional  $k$ -NN and one of its extension LMNN [4]. For CRNN and LRNN, the regularization parameter  $\lambda$  is nonnegligible. A preset range of  $\lambda$  is  $\{0, 1e-3, 5e-3, 1e-2, 5e-2, 1e-1, 5e-1, 1\}$ , and leave-one-out cross validation based on available labeled samples was conducted for parameter tuning. When  $\lambda = 0$ , the accuracy values of CRNN and LRNN are 40.62% and 66.89%, respectively, for the University of Pavia data; the accuracy values of CRNN and LRNN are 61.59% and 89.59%, respectively, for the HyMap Purdue data. Classification performance with other values of  $\lambda$  is illustrated in Figs. 1 and 2, where the optimal  $\lambda$  is 0.5 for both CRNN and LRNN using the University of Pavia data, and the optimal  $\lambda$  is  $5e-3$  for both CRNN and LRNN using the HyMap Purdue data.

### C. More Analysis on CR

Different from the traditional  $k$ -NN, the proposed CRNN chooses the NNs according to the weight coefficients of CR but not the simple Euclidean distance. Similarly, the difference between the proposed LRNN and the existing LMNN is that the local “mean” vector is calculated by an adaptive weight vector obtained from local within-class CR instead of equal weights. Here, we take CRNN, for example, to demonstrate the clear benefits of CR. Fig. 3 illustrates performance discrepancy when using the weight vector  $\alpha$  of the proposed CRNN and the



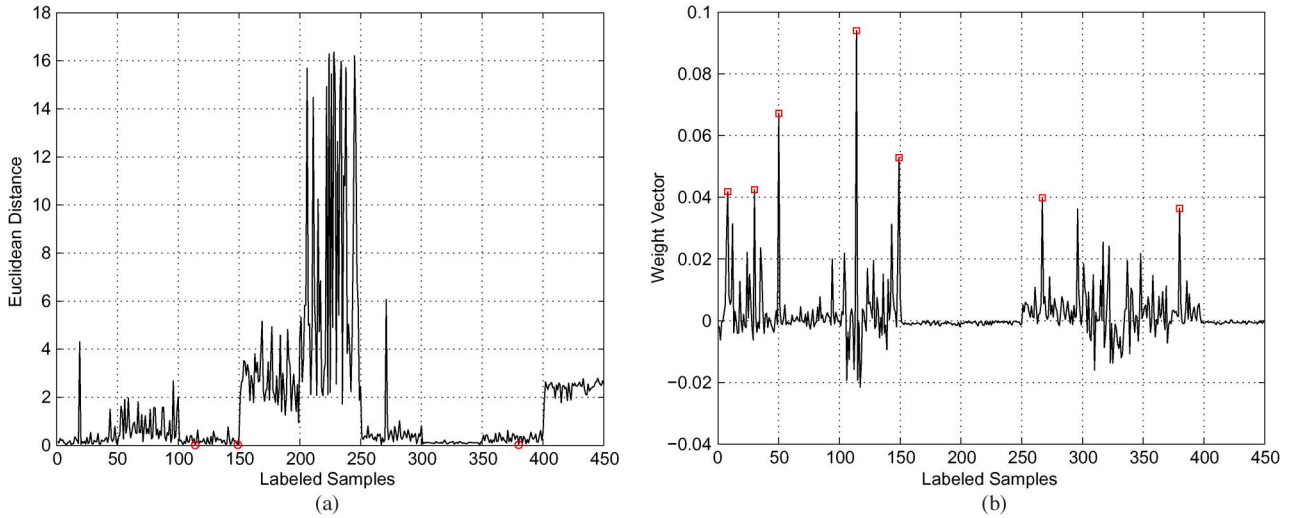


Fig. 3. Example of how the distance-weighted Tikhonov matrix  $\Gamma_y$  affects the weight vector  $\alpha$  for the proposed CRNN using the nine-class University of Pavia data with 50 training samples per class. The testing sample is selected from class 1. (a) Euclidean distance for  $k$ -NN. (b) Weight vector  $\alpha$  for CRNN.

TABLE I  
KL DIVERGENCE BETWEEN COLLABORATIVE COEFFICIENTS IN CRNN AND EUCLIDEAN DISTANCES IN  $k$ -NN WHEN SEPARATING PAIRWISE CLASSES IN THE UNIVERSITY OF PAVIA DATA

class	2	3	4	5	6	7	8	9
1	22.42	23.77	41.68	5.82	5.70	25.46	22.67	28.23
2	-	1.18	39.92	4.99	5.25	1.67	4.16	66.15
3	-	-	41.72	4.97	5.85	5.17	8.13	71.61
4	-	-	-	122.1	85.30	85.18	82.72	118.4
5	-	-	-	-	5.01	6.49	6.33	6.87
6	-	-	-	-	-	8.50	7.25	12.43
7	-	-	-	-	-	-	14.99	79.83
8	-	-	-	-	-	-	-	76.36

conventional distance measurement of  $k$ -NN for the University of Pavia data. From the definition of the weighted-distance Tikhonov matrix  $\Gamma_y$  [e.g., (9)], we notice that the Euclidean distances of the original  $k$ -NN classifier take exactly the same value as in the diagonal elements of matrix  $\Gamma_y$  of the proposed CRNN.

In Fig. 3(b), the locations exhibited the first seven largest weights (marked by red squares) are 114 (class 3), 50 (class 1), 149 (class 3), 30 (class 1), 8 (class 1), 267 (class 6), and 380 (class 8), and in Fig. 3(a), the locations exhibited the first three smallest Euclidean distances (marked by red circles) are 114 (class 3), 380 (class 8), and 149 (class 3). The optimal  $k$  is chosen to be 7 for the proposed CRNN and  $k = 3$  for  $k$ -NN according to our following experiments. The right class is 1, for which the proposed CRNN finds three weights in the first seven largest weights, whereas  $k$ -NN finds none. From these two figures, it is evident that the determination using collaborative weight vector  $\alpha$  to the right class is more robust than using the Euclidean distance.

Next, we investigate the benefits of the proposed CRNN. Kullback–Leibler (KL) divergence is employed to measure the dissimilarity between the obtained collaborative coefficients and the Euclidean distance. Table I shows the ratios of KL divergence for any two classes using CRNN and  $k$ -NN. All the values, which are larger than 1, confirm that collaborative coefficients tend to have better discriminative ability than the Euclidean distance.

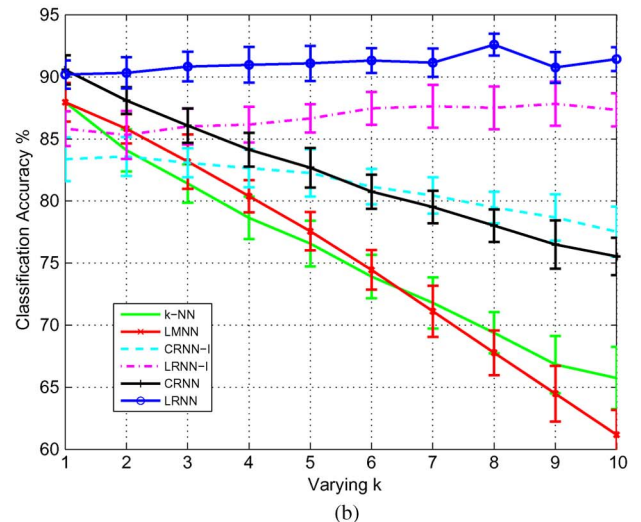
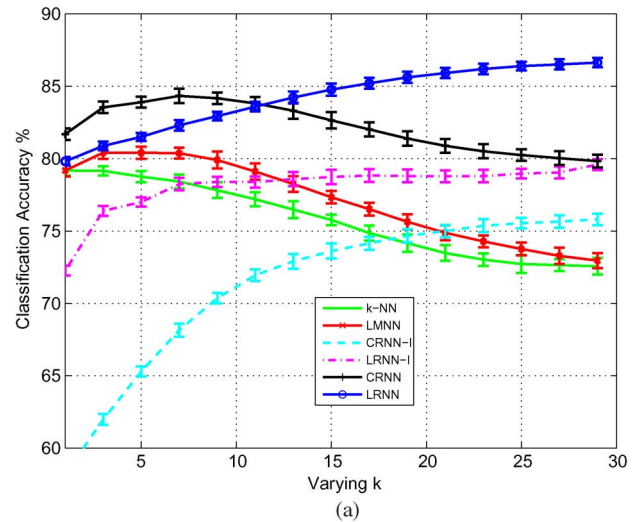


Fig. 4. Classification accuracy with standard deviation versus varying  $k$  for the proposed CRNN and LRNN using the two experimental data sets. (a) University of Pavia data. (b) HyMap Purdue data.

TABLE II  
CLASSIFICATION ACCURACY (%) PER CLASS AND OA WITH STANDARD DERIVATION OF FOUR CLASSIFIERS UNDER OPTIMAL PARAMETERS FOR THE UNIVERSITY OF PAVIA DATA

class	$k$ -NN	LMNN	CRNN	LRNN
1	67.23±2.23	69.38±1.68	71.54±1.53	<b>78.00±1.45</b>
2	66.62±2.51	69.08±1.35	<b>75.23±1.09</b>	71.85±1.50
3	68.00±3.18	69.85±2.05	77.54±1.63	<b>84.77±1.08</b>
4	91.23±2.02	93.54±0.89	91.54±0.80	<b>94.00±0.69</b>
5	99.23±0.12	99.23±0.20	<b>99.54±0.12</b>	99.38±0.08
6	65.23±2.80	70.62±1.28	81.23±1.98	<b>91.69±1.93</b>
7	87.85±1.02	86.46±0.81	<b>92.15±0.91</b>	91.85±0.56
8	67.85±2.25	<b>73.54±2.19</b>	70.15±1.27	72.15±1.45
9	99.85±0.07	99.85±0.04	<b>100±0</b>	98.77±0.02
OA	79.18±0.71	81.28±0.54	84.58±0.41	<b>86.63±0.37</b>

#### D. Classification Performance

To show the sensitivity of aforementioned methods to the number of NNs, we investigate the classification accuracy with standard deviation versus different values of  $k$ , as illustrated in Fig. 4. In order to reveal the efficiency of the designed Tikhonov matrix, we also test the performance of CRNN with identity matrix [CRNN-I, using identity matrix  $\mathbf{I}$  to replace the matrix  $\Gamma_y^T \Gamma_y$  in (6)] and LRNN with identity matrix [LRNN-I, using identity matrix  $\mathbf{I}$  to replace the matrix  $\Gamma_{l,y}^T \Gamma_{l,y}$  in (10)]. CRNN-I and LRNN-I are implemented with optimal  $\lambda$  after parameter tuning. To avoid any bias, we randomly choose samples, repeat the experiments 20 times, and report the average classification accuracy. The results provide evidence for the excellent performance of the proposed CRNN and LRNN with Tikhonov regularization.

For the University of Pavia data in Fig. 4(a), the optimal  $k$  is 7 for the proposed CRNN, 30 for the proposed LRNN, 3 for  $k$ -NN, and 5 for LMNN. Under the optimal  $k$ , the accuracy values are 84.58%, 86.63%, 79.18%, and 81.28%, respectively. It is apparent that the proposed LRNN achieves the best performance compared with the other methods. Furthermore, it is interesting to observe that the performance of other methods tends to deteriorate when  $k$  is very large (e.g., 20); however, the accuracy of the proposed LRNN remains stable. We notice that CRNN-I and LRNN-I both perform badly, sometimes even worse than the traditional  $k$ -NN and LMNN. For the HyMap Purdue data in Fig. 4(b), the optimal  $k$  is 1 for the proposed CRNN, 8 for the proposed LRNN, 1 for  $k$ -NN, and 1 for LMNN. Under the optimal  $k$ , the accuracy values are 90.57%, 92.59%, 87.96%, and 87.96%, respectively. The proposed CRNN and LRNN outperform the other two, and the difference is enlarged when  $k$  becomes greater. LMNN is reduced to  $k$ -NN when  $k$  is equal to 1, and it performs worse than  $k$ -NN when  $k$  is larger than 6, which is due to the fact that only ten training samples per class were employed.

Tables II and III further list the classification accuracy per class and overall accuracy (OA) with standard derivation of the proposed classifiers and the traditional two classifiers for the experimental data. From Tables II and III, it is obvious that most of the highest accuracy per class is persistently produced by the proposed CRNN and LRNN.

TABLE III  
CLASSIFICATION ACCURACY (%) PER CLASS AS WELL AS OA WITH STANDARD DERIVATION OF FOUR CLASSIFIERS UNDER OPTIMAL PARAMETERS FOR THE HYMAP PURDUE DATA

class	$k$ -NN	LMNN	CRNN	LRNN
1	73.74±3.75	73.74±3.86	79.50±3.71	<b>82.52±3.53</b>
2	96.32±3.63	96.32±1.85	<b>96.77±1.60</b>	95.60±1.85
3	97.72±0.62	<b>97.72±0.52</b>	96.80±0.69	95.42±0.58
4	96.57±1.14	96.57±1.31	97.10±2.44	<b>97.63±1.85</b>
5	95.04±2.32	95.04±3.12	99.41±1.58	<b>99.41±0.73</b>
6	83.04±4.46	83.04±4.37	86.32±4.40	<b>89.81±4.49</b>
OA	87.96±1.10	87.96±1.66	90.57±1.08	<b>92.59±1.04</b>

## V. CONCLUSION

In this letter, we have proposed two CR-based NN classifiers, e.g., CRNN and LRNN. They both exploit the discriminative nature of CR for classification. Compared with the traditional Euclidean distance, the weight coefficients of CR are more powerful in finding the true nearest training samples or local “mean” vector for each single testing sample. The experimental results demonstrated that the proposed CRNN and LRNN achieve higher classification accuracy than the traditional  $k$ -NN and LMNN.

## REFERENCES

- [1] W. Li, S. Prasad, J. E. Fowler, and L. M. Bruce, “Locality-preserving dimensionality reduction and classification for hyperspectral image analysis,” *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 4, pp. 1185–1198, Apr. 2012.
- [2] L. Samaniego, A. Bárdossy, and K. Schulz, “Supervised classification of recovery sensed image using a modified  $k$ -NN technique,” *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 7, pp. 2112–2125, Jul. 2008.
- [3] L. Ma, M. M. Crawford, and J. Tian, “Local manifold learning based  $k$ -nearest-neighbor for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.
- [4] Y. Mitani and Y. Hamamoto, “A local mean-based nonparametric classifier,” *Pattern Recognit. Lett.*, vol. 27, no. 10, pp. 1151–1159, Jul. 2006.
- [5] J. M. Yang, P. T. Yu, and B. C. Kuo, “A nonparametric feature extraction and its application to nearest neighbor classification for hyperspectral image data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 3, pp. 1279–1293, Mar. 2010.
- [6] E. Blanzieri and F. Melgani, “Nearest neighbor classification of remote sensing images with the maximal margin principle,” *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1804–1811, Jun. 2008.
- [7] 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.
- [8] L. Zhang, M. Yang, and X. Feng, “Sparse representation or collaborative representation: Which helps face recognition?” in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 471–478.
- [9] 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.
- [10] M. D. Iordache, J. M. Bioucas-Dias, and A. Plaza, “Collaborative sparse regression for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 341–354, Jan. 2014.
- [11] Y. Chen, N. M. Nasrabadi, and T. D. Tran, “Hyperspectral image classification using dictionary-based sparse representation,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.
- [12] J. Li, H. Zhang, Y. Huang, and L. Zhang, “Hyperspectral image classification by nonlocal joint collaborative representation with a locally adaptive dictionary,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3707–3719, Jun. 2014.
- [13] W. Li, S. Prasad, and J. E. Fowler, “Decision fusion in kernel-induced spaces for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3399–3411, Jun. 2014.