International Journal of Innovative
Computing, Information and Control
Volume **9**, Number **2**, February **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 543–554

AN IMPROVEMENT TO THE NEAREST NEIGHBOR CLASSIFIER AND FACE RECOGNITION EXPERIMENTS

Yong Xu[1], Qi Zhu[1], Yan Chen[1] and Jeng-Shyang Pan[2]

[1]Bio-Computing Research Center
[2]Innovative Information Industry Research Center
Harbin Institute of Technology Shenzhen Graduate School
HIT Campus of Shenzhen University Town, Xili, Shenzhen 518055, P. R. China
laterfall2@yahoo.com.cn; ksqiqi@sina.com; jadechenyan@gmail.com; jspan@cc.kuas.edu.tw

Received November 2011; revised March 2012

Abstract. *The conventional nearest neighbor classifier (NNC) directly exploits the distances between the test sample and training samples to perform classification. NNC independently evaluates the distance between the test sample and a training sample. In this paper, we propose to use the classification procedure of sparse representation to improve NNC. The proposed method has the following basic idea: the training samples are not uncorrelated and the "distance" between the test sample and a training sample should not be independently calculated and should take into account the relationship between different training samples. The proposed method first uses a linear combination of all the training samples to represent the test sample and then exploits modified "distance" to classify the test sample. The method obtains the coefficients of the linear combination by solving a linear system. The method then calculates the distance between the test sample and the result of multiplying each training sample by the corresponding coefficient and assumes that the test sample is from the same class as the training sample that has the minimum distance. The method elaborately modifies NNC and considers the relationship between different training samples, so it is able to produce a higher classification accuracy. A large number of face recognition experiments on three face image databases show that the maximum difference between the accuracies of the proposed method and NNC is greater than 10%.*

Keywords: Face recognition, Nearest neighbor classifier, Sparse representation, Classification

1. **Introduction.** The image recognition technique [1-3] can be used for a variety of applications such as objection recognition, personal identification and facial expression recognition [4-14]. For many years researchers in the field of image recognition have adopted the following procedures to perform recognition: image capture, feature selection or feature extraction and classification. Usually these procedures are consecutively implemented and each process is necessary. The nearest neighbor classifier (NNC) is an important classifier. NNC is also one of the oldest and simplest classifiers [15,17]. The nearest neighbors of the sample were used in a number of fields such as image retrieval, image coding, motion control and face recognition [19,22]. NNC first identifies the training sample that is the closest to the test sample and assumes that the test sample is from the same class as this training sample. Since "close" means "similarity", we can also say that NNC exploits the "similarity" of the test sample and each training sample to perform classification. To determine the nearest neighbors of the sample is the first step of NNC, so it is very crucial. In the past, various ideas and algorithms were proposed for determining the nearest neighbors. For example, D. Omercevic et al. proposed the idea of meaningful nearest neighbors [23]. H. Samet proposed the MaxNearestDist algorithm for

finding K nearest neighbors [24]. J. Toyama et al. proposed a probably correct approach for greatly reducing the searching time of the nearest neighbor search method [25]. The focus of this approach is to devise the correct set of k-nearest neighbors obtained in high probability. Y.-S. Chen et al. proposed a fast and versatile algorithm to rapidly perform nearest neighbor searches [26]. Besides the methods in these works, many other methods [26-30] have also been developed for computationally efficiently searching the nearest neighbors. We note that most of these methods focus on improving the computation efficiency of the nearest neighbor search.

We note that recently a distinctive image recognition method, the sparse representation (SR) method was proposed [31]. The applications of SR on image recognition such as face recognition have obtained a promising performance [32-34]. However, it seems that it is not very clear why SR can outperform most of previous face recognition methods and different researchers attribute the good performance of SR to different factors. In our opinion one of remarkable advantages of SR is that it uses a novel procedure to classify the test sample. Actually, this method first represents a test sample by using a linear combination of a subset of the training samples. Then it takes the weighted sum of the training sample as an approximation to the test sample and regards the coefficients of the linear combination as the weights. SR calculates the deviation of the test sample from the weighted sum of all the training samples from the same class and classifies the test sample into the class with the minimum deviation. As the weighted sum of a class is also the sum of the contribution in representing the test sample of this class, we refer to this classification procedure as representation-contribution-based classification procedure (RCBCP). We also say that SR consists of a representation procedure and a classification procedure.

The main rationale of RCBCP is that when determining the distances between the test sample and training samples, it takes into account the relationship of different training samples. If some training samples are collinear, RCBCP will use the weights to reflect the collinear nature and will classify the test sample into the class the weighted sum of which provides the best approximation to the test sample. However, the conventional NNC usually separately evaluates the distances between the test sample and training samples, ignoring the similarity and potential relationship between different training samples. The following example is very helpful to illustrate this difference between RCBCP and the conventional NNC: if two training samples have the same minimum Euclidean distances to the test sample, then NNC will be confused in classifying the test sample. However, under the same condition, RCBCP usually obtains two different "distances" and is still able to determine which training sample is closer to the test sample.

In this paper, motivated by RCBCP, we propose to exploit RCBCP to modify NNC. The basic idea is to use a dependent way to determine the "distances" between the test sample and training samples. We first use all of the training samples to represent the test sample, which leads to a linear system. We directly solve this system to obtain the least-squares solution and then exploit the solution and the classification procedure of NNC to classify the test sample. Differing from the conventional NNC, the proposed method calculates the distance between the test sample and the result of multiplying each training sample by the corresponding weight (i.e., a component of the solution vector) and assumes that the test sample is from the same class as the training sample with the minimum distance. The proposed method is very simple and computationally efficient. Our experiments show that the proposed method always achieves a lower rate of classification errors than NNC. This paper also shows that one modification of the proposed method is identical to NNC.

This paper not only proposes an improvement to NNC but also has the following contributions: it confirms that RCBCP is very useful for achieving a good face recognition

performance. Moreover, it also somewhat illustrates that RCBCP is one of the most important advantages of SR.

The rest of the paper is organized as follows. Section 2 describes our method. Section 3 shows the difference between NNC and the proposed method. Section 4 presents the experimental results. Section 5 offers our conclusion.

2. **Problem Statement and Preliminaries.** Let $A_1, \ldots, A_n$ denote all n training samples in the form of column vectors. We assume that in the original space test sample Y can be approximately represented by a linear combination of all of the training samples. That is,

$$Y \approx \sum\nolimits_{i=1}^{n} \beta_i A_i. \tag{1}$$

β_i is the coefficient of the linear combination. We can rewrite (1) as

$$Y = A\beta, \tag{2}$$

where $\beta = (\beta_1, \ldots, \beta_n)^T$, $A = (A_1, \ldots, A_n)$.

As we know, if A^TA is not singular, we can obtain the least squares solution of (2) using $\beta = (A^TA)^{-1}A^TY$. If A^TA is nearly singular, we can solve β by $\beta = (A^TA + \mu I)^{-1}A^TY$, where μ is a positive constant and I is the identity. After we obtain β, we calculate Y' using $Y' = A\beta$ and refer to it as the result of the linear combination of all of the training samples.

From (1), we know that every training sample makes its own contribution to representing the test sample. The contribution that the ith training sample makes is $\beta_i A_i$. Moreover, the ability, of representing the test sample, of the ith training sample A_i can be evaluated by the deviation between $\beta_i A_i$ and Y, i.e., $e_i = ||Y - \beta_i A_i||^2$. Deviation e_i can be also viewed as a measurement of the distance between the test sample and the ith training sample. We consider that the smaller e_i is, the greater ability of representing the test sample the ith training sample has. We identify the training sample that has the minimum deviation from the test sample and classify Y into the same class as this training sample.

3. **Analysis of Our Method.** In this section, we show the characteristics and rationale of our method.

3.1. **Difference between our method and NNC.** Superficially, our method performs somewhat similarly with NNC, because both of them first evaluate the "distances" between the test sample and each training sample and classify the test sample into the same class as the training sample that has the minimum "distance". However, our method is different from NNC as follows: it does not directly compute the distance between the test sample and each training sample but calculates the distance between the test sample and the result of multiplying each training sample by the corresponding coefficient. Since the sum of all the training samples weighted by the corresponding coefficients well approximates to the original test sample, the result of multiplying each training sample by the corresponding coefficient can be viewed as an optimal approximation, to the original test sample, generated from the training sample. Thus, the deviation between this approximation and the original test sample can be taken as the "distance" between the training sample and test sample. Intuitively, the smaller the "distance", the more "similar" to the test sample the training sample.

As the weighted sum (i.e., a linear combination) of all the training samples well represents the test sample, we say that all the training samples provide a good representation for the test sample in a competitive way. According to the classification procedure of our

method, the training sample that has the minimum deviation from the test sample wins in the competition. This has the following rationale: the training sample that has the minimum deviation is most similar to the test sample, because it can represent the test sample with the minimum error. Figure 1 shows the flowchart of our method.

It should be pointed out that our method exploits all of the training samples to represent the test sample. As a result, it is very different from SR and is not a sparse representation method at all. It is clear that our method only needs to solve one linear system and is computationally efficient.

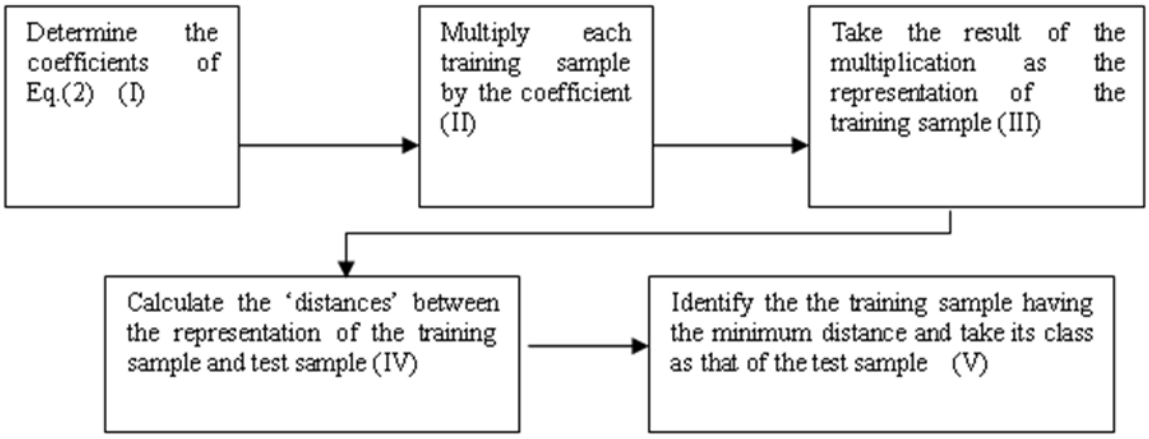

FIGURE 1. Flowchart of our method. Here distance is also the deviation of the test sample from the representation of the training sample.

3.2. **More exploration.** This subsection presents an alternative algorithm of the nearest neighbor classifier (AANNC), which is helpful for formally showing the difference between our method and the nearest neighbor classifier. AANNC first uses each training sample to express the test sample and then exploits the error of expression to classify the test sample. The formula to use the ith training sample A_i to express test sample Y is

$$Y = \gamma_i A_i + E_i, \quad i = 1, \ldots, n, \tag{3}$$

where γ_i is the coefficient and E_i denotes the error vector. Equation (3) shows that the test sample can be expressed as the sum of a training sample weighted by a coefficient and the error vector. We can convert (3) into

$$A_i^T Y = \gamma_i A_i^T A_i + A_i^T E_i, \quad i = 1, \ldots, n. \tag{4}$$

Further, we solve (4) using

$$\gamma_i = \frac{A_i^T Y}{A_i^T A_i}, \quad i = 1, \ldots, n. \tag{5}$$

If all the samples are unit vectors with length of 1, then we have

$$\gamma_i = A_i^T Y, \quad i = 1, \ldots, n. \tag{6}$$

AANNC then evaluates the ability of expressing the test sample of each training sample using the following distance

$$d_i = ||Y - \gamma_i A_i||^2, \quad i = 1, \ldots, n, \tag{7}$$

where γ_i is solved using (5). AANNC considers that the smaller distance d_i is, the better ability of expressing the test sample the ith training sample has. As a result, AANNC identifies the class-label of the training sample that has the minimum distance

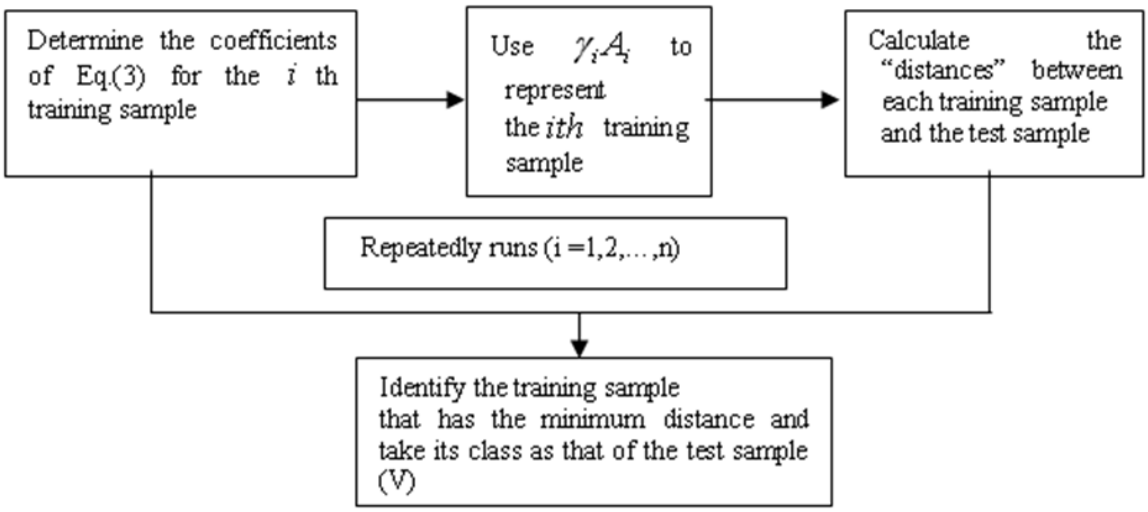

FIGURE 2. Flowchart of the modification of our method

$d = \min d_i$ and classifies the test sample into the same class. We use Figure 2 to show the flowchart of AANNC. This figure clearly shows that AANNC repeatedly solves Equation (3) and solving Equation (3) at a time produces only the coefficient for a training sample. However, our method shown in Section 2 obtains the coefficients for all of the training samples by solving Equation (2) at a time.

The following shows that AANNC is identical to NNC. As all of the samples are unit vectors, we can transform (7) into

$$d_i = 1 + \gamma_i^2 - 2\gamma_i A_i^T Y = 1 - \gamma_i^2 = 1 - (A_i^T Y)^2, \quad i = 1, \ldots, n. \tag{8}$$

If all the samples are unit vectors, the distance between each training sample and the test sample can be formulated as

$$dd_i = ||Y - A_i||^2 = 2 - 2A_i^T Y, \quad i = 1, \ldots, n. \tag{9}$$

Since NNC classifies the test sample based on the distance metric as shown in (9), it is sure that the classification based on (8) has the same classification decision as NNC. As a result, under the condition that all the samples are unit vectors, AANNC is identical to NNC.

4. **Experimental Results.** We conducted a number of experiments using the ORL [35], Yale [36] and AR [37] face databases. Later we will show the mean of the rates of the classification errors of our method, NNC, the center-based nearest neighbor classifier (CBNNC) proposed in [39] and the nearest neighbor line (NNL) classifier proposed in [40] on the three databases. The codes are available at http://www.yongxu.org/lunwen/.html. CBNNC and NNL were proposed respectively in 2007 and 2004 as two improvements to conventional NNC [39,40]. Previous literature shows that these two improvements can obtain a better performance than NNC in some cases [39,40]. The ORL database [35] includes 400 face images from 40 subjects. The images include variations in facial expression (smiling/not smiling, open/closed eyes) and facial detail. The subjects are in an upright, frontal position with tolerance for some tilting and rotation of up to 20°. Each of the face images contains 112×92 pixels. The Yale database contains face images with a variety of expressions such as normal, sad, happy, sleepy, surprised, and winking, all obtained under different lighting conditions. Some faces also wear glasses. From the very large scale AR face database, we used 3120 gray face images from 120 subjects, each providing 26 images. These images were taken in two sessions [39] and show faces

with different facial expressions, in varying lighting conditions and occluded in several ways. For the ORL and Yale databases, if s samples of all the n samples per class are used for training, there are $C_q^p = \frac{p(p-1)\cdots(p-q+1)}{q(q-1)\cdots 1}$ possible combinations. We use the same combinations to determine training samples and test samples for all the classes. As a result, there are C_n^s training sets and C_n^s testing sets. We dealt with the Yale database in the same way. Using this experiment scheme, we can make the obtained experimental result be representative. Table 1 shows from the ORL and Yale databases, how many training sample sets per class were used.

TABLE 1. Number of training sample sets. The number of the test sets is the same.

Number of training samples per class	1	2	3	4
ORL	10	45	120	210
Yale	11	55	165	330

We conducted experiments for all the training sets and testing sets of the ORL and Yale databases. As the AR face database contains too many samples, we took the first 2, 4, 6 and 8 training samples per class and the others as training samples and test samples, respectively. We then resized each face image of the AR database to a 40 by 50 image by using the down-sampling algorithm presented in [41]. The face images of the ORL database were also preprocessed in the same way. Before carrying out all the methods, we first converted each sample into the vector with the norm of 1. We then converted each image into a one-dimensional column vector before we implemented either of NNC and our method. We solve Equation (2) using $\beta = (A^T A + \mu I)^{-1} A^T Y$ with $\mu = 0.001$.

Figure 3 shows some face images of one subject from the AR face database. Figure 4 shows original test images of two subjects from the ORL database and the images corresponding to the result of the linear combination of all of the training samples for representing the test sample. As shown in Section 2, the result of the linear combination of all of the training samples, i.e., $Y' = A\beta$ is a column vector. In order to obtain the images shown in the second and fourth rows of Figure 4, we first converted Y' into a two-dimensional matrix with the same size as the original face image. Figure 5 shows a case where our method correctly classified a test sample, whereas NNC failed to do so. Figure 6 shows the distances between the test sample shown in Figure 5 and all of the training samples. Figure 7 shows the deviations between the test sample shown in Figure 5 and the result of multiplying each training sample by the corresponding coefficient presented in Section 2. From Figures 6 and 7, we see that though the first training sample is not the

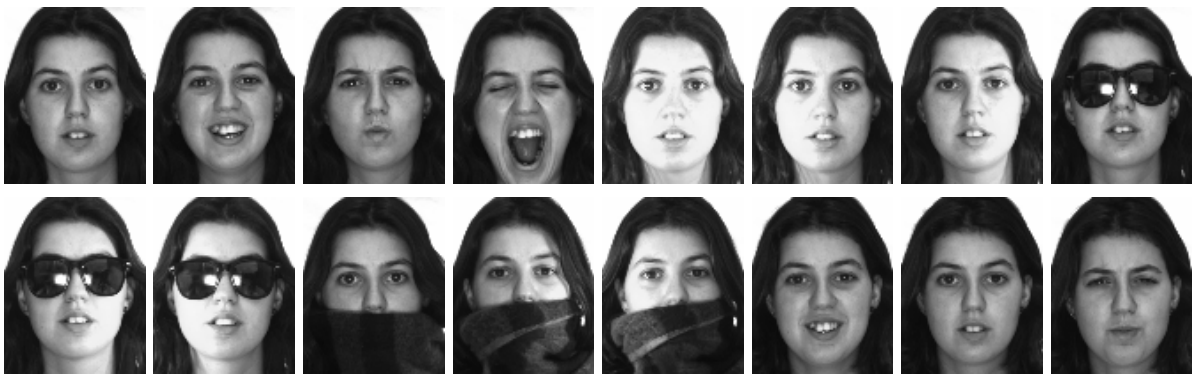

FIGURE 3. Some samples of one subject from the AR database

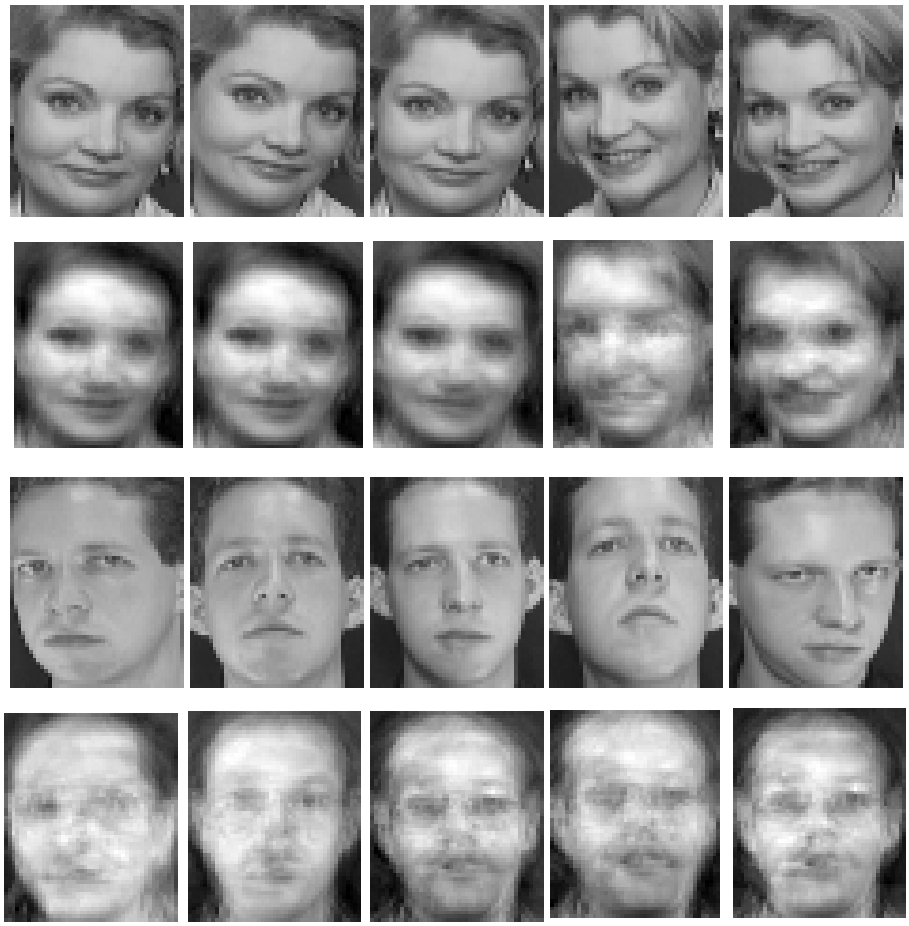

FIGURE 4. Original test images of two subjects from the ORL database and the images corresponding to the result of the linear combination of all of the training samples for representing the test sample. The first 5 images per subject were used as the training samples and the others were used as test samples. The first and third rows show the original test images and the second and fourth rows show the images corresponding to the result of the linear combination, respectively.

training sample that is the closest to the test sample, in our method it has the minimum deviation from the test sample. As a result, our method can correctly classify the test sample into the class that the first training sample belongs to, which is the genuine class of the test sample.

Tables 2-4 show the experimental results. From these tables, we see that our method almost always classifies more accurately than NNC, CBNNC proposed in [39] and NNL in [40] for all the databases. For example, for the AR database, when the first two images per class were used as training samples and the others were used as test sample, the ratios of the classification errors obtained using our method, NNC, CBNNC proposed in [39] and NNL proposed in [40] are 30.38%, 39.93%, 40.03% and 40.80% respectively. Moreover, we see that the maximum value of the difference between the rates of classification errors of NNC and our method is 11.14%. We also see that as NNL obtained a lower rate of classification errors than CBNNC, NNL seems to be a better improvement to conventional NNC in comparison with CBNNC.

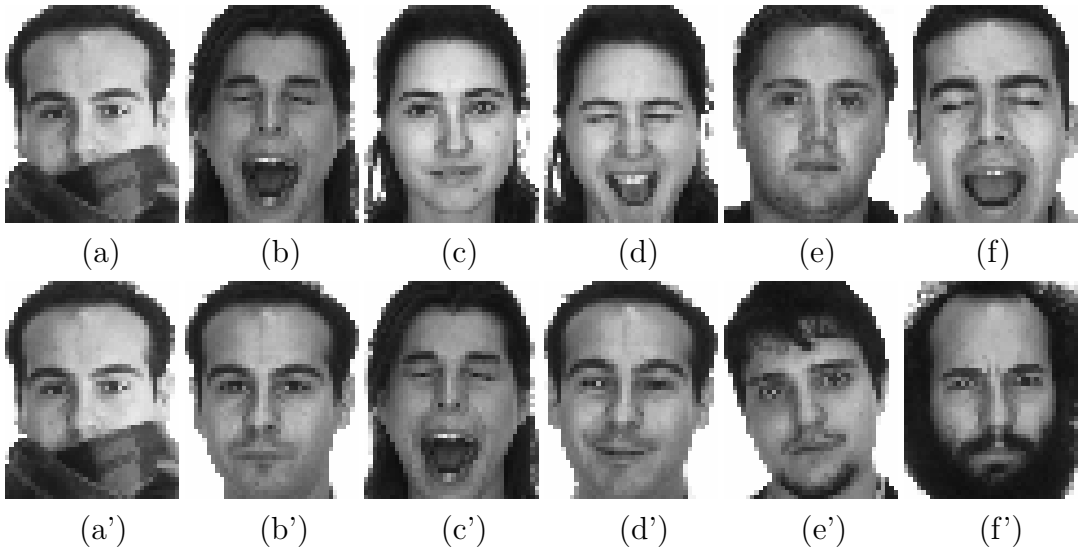

FIGURE 5. One original test image of one subject from the AR database and the first 5 nearest images obtained using NNC and our method, respectively. In the first row, while (a) denotes the test image, (b), (c), (d), (e) and (f) respectively stand for the first to fifth nearest images obtained using NNC. In the second row, while (a') denotes the test image (same as (a)), (b'), (c'), (d'), (e') and (f') respectively stand for the first to fifth nearest images obtained using our method. It is clear that our method correctly classified this test ample, whereas NNC did not. In this case, the first 4 face images per class were used as training samples and the others were used as testing samples.

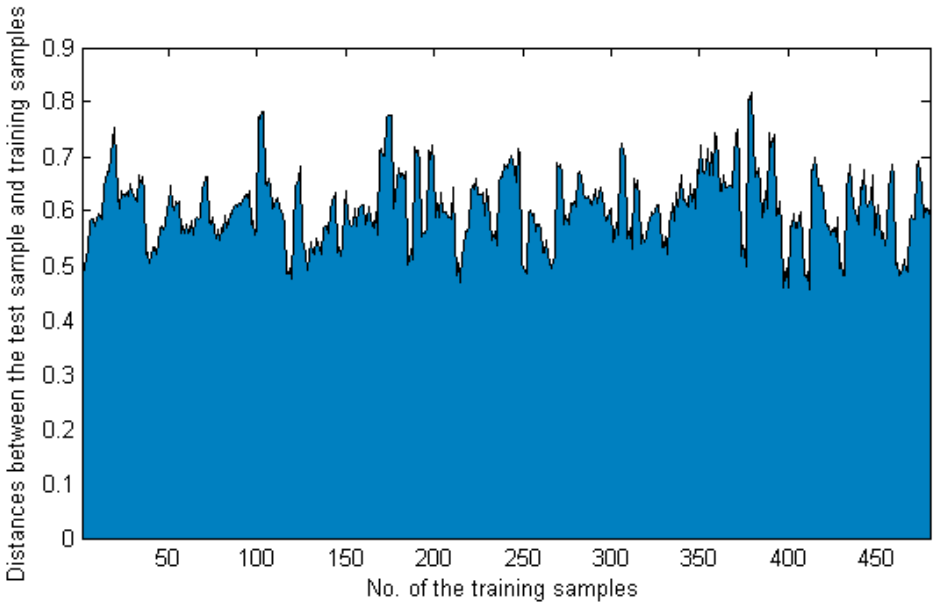

FIGURE 6. The distances between the test sample shown in Figure 5 and all of the training samples

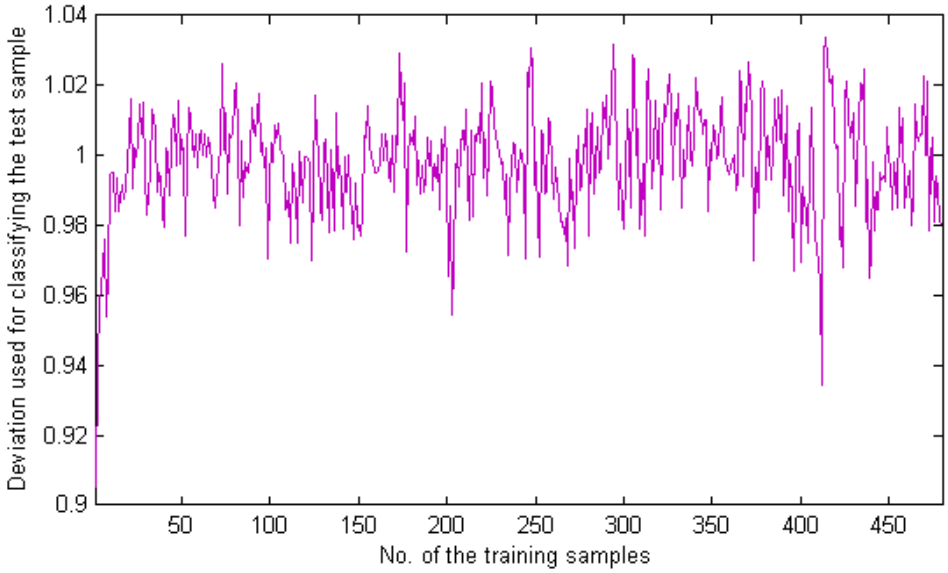

FIGURE 7. The deviations between the test sample shown in Figure 5 and the result of multiplying each training sample by the corresponding coefficient presented in Section 2

TABLE 2. Means of the rates of the classification errors (%) of our method and NNC on the Yale database

Number of training samples per class	1	2	3	4
Our method	15.52	5.67	4.17	3.69
NNC	18.97	9.17	5.89	4.71
CBNNC	18.85	9.44	6.23	5.11
NNL	¥	7.03	4.97	4.16
Rate difference between our method and NNC	3.45	3.5	1.72	1.02
Rate difference between our method and CBNNC	3.33	3.77	2.06	1.42
Rate difference between our method and NNL	¥	1.36	0.8	0.47

TABLE 3. Means of the rates (%) of the classification errors of our method and NNC on the ORL databases

Number of training samples per class	1	2	3	4
Our method	30.06	17.78	11.92	8.73
NNC	33.94	20.54	13.83	9.98
CBNNC	34.11	20.60	13.84	9.89
NNL	¥	19.40	12.03	8.03
Rate difference between our method and NNC	3.88	2.76	1.91	1.25
Rate difference between our method and CBNNC	4.05	2.82	1.92	1.16
Rate difference between our method and NNL	¥	1.62	0.11	−0.7

5. **Conclusions.** The proposed method elaborately modifies NNC and exploits the ability, of representing the test sample, of the training sample rather than only a simple distance to classify the test sample. This ability is related to the "similarity" between the test sample and each training sample. We say that the proposed method evaluates the "similarity" between the test sample and each training sample in a "competitive" way, whereas NNC directly calculates the "similarity" between the test sample and each

TABLE 4. Rates of the classification errors (%) of our method and NNC on the AR database. We took the first 2, 4, 6 and 8 training samples per class and the others as training samples and test samples, respectively.

Database	Training samples	Our method	NNC	CBNNC	NNL	Difference between our method and NNC	Difference between our method and CBNNC	Difference between our method and NNL
AR	2 per class	30.38	39.93	40.03	40.80	9.55	9.65	10.42
AR	4 per class	31.55	42.69	42.69	42.50	11.14	11.14	10.95
AR	6 per class	30.92	38.88	38.92	38.25	7.96	8.0	7.33
AR	8 per class	33.89	41.76	41.76	41.34	7.87	7.87	7.45

training sample. When computing the distance between the test sample and each training sample, the proposed method not only exploits these two samples but also takes into account the relationship between different training samples. As a result, the proposed method can identify the training sample that has the greatest contribution in representing the test sample. A large number of face recognition experiments show that our method always achieves a higher classification accuracy than NNC and the maximum difference between the accuracies of our method and NNC is greater than 10%.

Acknowledgment. This article is partly supported by Program for New Century Excellent Talents in University (Nos. NCET-08-0156 and NCET-08-0155), NSFC under Grant nos. 61071179, 61173086, 61020106004, 61001037 and 61173086 as well as the Fundamental Research Funds for the Central Universities (HIT.NSRIF. 2009130).

REFERENCES

[1] D. Zhang, X. Jing and J. Yang, Biometric image discrimination technologies, *Idea Group Publishing*, Hershey, USA, 2006.

[2] D. Zhang, F. Song, Y. Xu and Z. Liang, Advanced pattern recognition technologies with applications to biometrics, *IGI Global*, 2008.

[3] A. Khotanzad and Y. C. Hong, Invariant image recognition by Zernikemoments, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.12, no.5, pp.489-497, 1990.

[4] J. Yang and C. Liu, Color image discriminant models and algorithms for face recognition, *IEEE Transactions on Neural Networks*, vol.19, no.12, pp.2088-2098, 2008.

[5] Y. Xu, L. Yao and D. Zhang, Improving the interest operator for face recognition, *Expert System with Applications*, vol.36, no.6, pp.9719-9728, 2009.

[6] C. Liu and J. Yang, ICA color sSpace for pattern recognition, *IEEE Transactions on Neural Networks* vol.20, no.2, pp.248-257, 2009.

[7] F. Song, D. Zhang, D. Mei and Z. Guo, A multiple maximum scatter difference discriminant criterion for facial feature extraction, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol.37, no.6, pp.1599-1606, 2007.

[8] M. Wan, Z. Lai, J. Shao and Z. Jin, Two-dimensional local graph embedding discriminant analysis (2DLGEDA) with its application to face and palm biometrics, *Neurocomputing*, vol.73, no.1-3, pp.197-203, 2009.

[9] Q. Gao, L. Zhang, D. Zhang and H. Xu, Independent components extraction from image matrix, *Pattern Recognition Letters*, vol.31, no.3, pp.171-178, 2010.

[10] Y. Xu, D. Zhang and J. Y. Yang, A feature extraction method for use with bimodal biometrics, *Pattern Recognition*, vol.43, pp.1106-1115, 2010.
[11] J. Wang, W. Yang, Y. Lin and J. Yang, Two-directional maximum scatter difference discriminant analysis for face recognition, *Neurocomputing*, vol.72, no.1-3, pp.352-358, 2008.
[12] H. Sellahewa and S. A. Jassim, Image-quality-based adaptive face recognition, *IEEE Transactions on Instrumentation and Measurement*, vol.59, no.4, 2010.
[13] J. G. Wang and E. Sung, Facial feature extraction in an infrared image by proxy with a visible face image, *IEEE Transactions on Instrumentation and Measurement*, vol.56, no.5, 2007.
[14] X. Gao, J. Zhong, D. C. Tao and X.-L. Li, Local face sketch synthesis learning, *Neurocomputing*, vol.71, no.10-12, pp.1921-1930, 2008.
[15] T. Cover and P. Hart, Nearest neighbor pattern classification, *IEEE Transactions in Information Theory*, pp.21-27, 1967.
[16] K. Weinberger, J. Blitzer and L. Saul, Distance metric learning for large margin nearest neighbor classification, *Advances in Neural Information Processing Systems*, vol.18, pp.1473-1480, 2006.
[17] P. Y. Simard, Y. LeCun and J. Decker, Efficient pattern recognition using a new transformation distance, in *Advances in Neural Information Processing Systems 6*, S. Hanson, J. Cowan and L. Giles (eds.), Morgan Kaufman, San Mateo, CA, 1993.
[18] H. Li and T. Gao, Improving the B3LYP absorption energies by using the neural network ensemble and k-nearest neighbor approach, *ICIC Express Letters, Part B: Applications*, vol.2, no.5, pp.1075-1080, 2011.
[19] J. Zou, K. Yuan, S. Jiang and W. Chen, Brain CT image retrieval combined nonnegative tensor factorization with k-nearest neighbor, *Journal of Information and Computational Science*, vol.6, no.1, pp.273-281, 2009.
[20] J. Nagasue, Y. Konishi, N. Araki, T. Sato and H. Ishigaki, Slope-walking of a biped robot with k nearest neighbor method, *ICIC Express Letters*, vol.4, no.3(B), pp.893-898, 2010.
[21] Y. Xu and F. Song, Feature extraction based on a linear separability criterion, *International Journal of Innovative Computing, Information and Control*, vol.4, no.4, pp.857-865, 2008.
[22] Z.-M. Lu, S.-C. Chu and K.-C. Huang, Equal-average equal-variance equal-norm nearest neighbor codeword search algorithm based on ordered Hadamard transform, *International Journal of Innovative Computing, Information and Control*, vol.1, no.1, pp.35-41, 2005.
[23] D. Omercevic, O. Drbohlav and A. Leonardis, High-dimensional feature matching: Employing the concept of meaningful nearest neighbors, *IEEE the 11th International Conference on Computer Vision*, pp.1-8, 2007.
[24] H. Samet, K-nearest neighbor finding using MaxNearestDist, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.30, no.2, pp.243-252, 2008.
[25] J. Toyama, M. Kudo and H. Imai, Probably correct k-nearest neighbor search in high dimensions, *Pattern Recognition*, vol.43, pp.1361-1372, 2010.
[26] Y. S. Chen, Y. P. Hung, T. F. Yen and C. S. Fuh, Fast and versatile algorithm for nearest neighbor search based on a lower bound tree, *Pattern Recognition*, vol.40, pp.360-375, 2007.
[27] S. Berchtold, B. Ertl, D. A. Keim, H.-P. Kriegel and T. Seidl, Fast nearest neighbor search in high-dimensional spaces, *Proc. of the 14th IEEE Conference on Data Engineering*, pp.23-27, 1998.
[28] J. McNames, A fast nearest-neighbor algorithm based on a principal axis search tree, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.23, no.9, pp.964-976, 2001.
[29] S. A. Nene and S. K. Nayar, A simple algorithm for nearest neighbor search in high dimensions, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.19, pp.989-1003, 1997.
[30] A. Djouadi and E. Boutache, A fast algorithm for the nearest-neighbor classifier, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.19, no.3, pp.277-282, 1997.
[31] J. Wright, A. Y. Yang, A. Ganesh et al., Robust face recognition via sparse representation, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.31, no.2, pp.210-227, 2009.
[32] J. Wright, Y. Ma, J. Mairal et al., Sparse representation for computer vision and pattern recognition, *Proc. of IEEE*, pp.1-8, 2009.
[33] C. X. Ren and D. Q. Dai, Sparse representation by adding noisy duplicates for enhanced face recognition: An elastic net regularization approach, *Chinese Conference on Pattern Recognition*, Nanjing, China, 2009.
[34] Y. Xu and Q. Zhu, A simple and fast representation-based face recognition method, *Neural Computing and Applications*, 2012.
[35] *http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.*
[36] *http://cvc.yale.edu/projects/yalefaces/yalefaces.html.*

[37] *http://cobweb.ecn.purdue.edu/∼aleix/aleix_face_DB.html.*
[38] J. Yang, D. Zhang, A. F. Frangi and J.-Y. Yang, Two-dimensional PCA: A new approach to appearance-based face representation and recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, no.1, pp.131-137, 2004.
[39] Q-B Gao, Z. Z. Wang, Center-based nearest neighbor classifier, *Pattern Recognition*, vol.40, pp.346-349, 2007.
[40] W. Zheng, L. Zhao and C. Zou, Locally nearest neighbor classifiers for pattern classification, *Pattern Recognition*, vol.37, pp.1307-1309, 2004.
[41] Y. Xu and Z. Jin, Down-sampling face images and low-resolution face recognition, *The 3rd International Conference on Innovative Computing, Information and Control*, Dalian, China, pp.392, 2008.