Journal of Computers Vol. 30 No. 4, 2019, pp. 45-56
doi:10.3966/199115992019083004004

A New Extended Classifier to the NNC for Face Recognition

Jingyuan Zeng[1,2], Jianghong Guo[2], Xiansheng Wang[3], Tiansheng Hong[1*]

[1] School of engineering, South China Agriculture University, Guangzhou, China
zjy@jyu.edu.cn, tshong@scau.edu.cn

[2] School of computer, Jiaying University, Meizhou, China
g_jh@jyu.edu.cn

[3] School of Software Engineering, Shenzhen Institute of Information Technology, Shenzhen, China
wxs111111@163.com

Received 26 September 2017; Revised 22 March 2018; Accepted 31 March 2018

Abstract. In this paper, we propose a novel classifier which is based on subspaces of each class of training samples. This method has the following basic idea: the training samples among different classes are uncorrelated, but the distance between the test sample and the training samples in one class should be taking into account all training samples in this class. Compared with other methods, the contribution of this paper is that we use the skinny QR-decomposition for L small least square problems. From our analysis, we can find that our method is equivalent to the nearest neighbor classifier (NNC) when the training sample in any class is one. A large number of face recognition experiments on two face image databases show that our method can work efficiently and effectively.

Keywords: face recognition, least square problem, nearest neighbor classifier, skinny QR-decomposition

1 Introduction

For many years, face recognition has attracted much attention, and it includes two stages: feature selective (feature extraction) and classification [1-13]. A number of feature extraction methods have been widely used, and typically methods include principal component analysis (PCA) [10, 14-15], linear discriminant analysis (LDA) [16-18], mini mum squared error (MSE) method [12, 19], kernel LDA (KLDA) [20-22] and kernel (KPCA) [23-26]. The purpose of the transform methods is to transform samples into a new low-dimensional subspace in which some good properties can be hold. For example, after projecting the training samples into the low-dimensional subspace, PCA can make these samples in the new subspace have the most variance. LDA can get the maximum ration of the between-class distance to the within-class distance in the new subspace. The MSE method wants to get a space that can well transform the training samples into its class label. KPCA and KLDA are the extensions of PCA and LDA, thus, they have goals similar to those of PCA and LDA. The difference is that KPCA and KLDA perform nonlinear transformations, whereas PCA and LDA perform linear transformations. Compared with these traditional transformation methods, nonnegative matrix factorization is a very different way to represent the sample. It considers that the testing sample should be represented by nonnegative numbers and factorizes the training samples matrix into two matrices to represent the test sample [27]. All of these feature extraction methods show good performance in both classification and image retrieval.

From above, we know that transformation methods exploit only the training samples to implement their training stage. And they work as follows: they first use the training samples to get a low-dimensional subspace and then project the training samples and the test samples into the low-dimensional subspace. So, the low-dimensional subspaces are "optimal" for the training samples; however, they are

* Corresponding Author

not “optimal” for the test samples to be classified.

Since feature extraction and classification are consecutively implemented and each process is essential, classification is also very important for face recognition. The nearest neighbor classifier (NNC) is an important classifier, and it is one of the simplest and the most direct classifier [28-29]. NNC identifies the test sample is from the same class which contains the training sample that is the closest to the test sample. We will give a simple description of it in section 2.

In this paper, we propose a novel method for classification. Our method is different form nearest neighbor classifier, but it is equal to the nearest neighbor classifier in a special case. Our method is also a representation method, the first stage of this method is feature extraction, and the second stage of this method is partially similar to the representation method, whereas it is based on solving some least square problems. Another advantage is that our method has a distinctive characteristic that its solution can be solved with ease. The proposed method first gets a low-dimension orthogonal subspace from the training samples; and then it projects all the training samples and the test samples on the orthogonal subspace; and then the test sample can be expressed as a linear combination of all the training samples in one class. Finally, the method constructs a classification procedure on the basis of the expression result. And from the experiments in section 4, the proposed method performs well in face recognition applications.

Since the proposed method uses only n_i training samples to express and classify the test sample (n_i is the number of the training samples of the ith class), it is also a special representation method. We analysis also demonstrates that the training samples selected and used by our method are the most suitable ones that can produce the minimum classification error. In other words, if we use other n_i training samples to express and classify the test sample, we will obtain a higher classification error.

The rest of this paper is organized as follows. Section 2 describes two old methods. Section 3 presents our method, including the stages and the analysis of our method; besides these, we also give an extended algorithm, which has the similar experimental results. Section 4 shows the experimental results. Section 5 offers our conclusion.

2 The Old Methods

In this section, we introduce two old classifications. One is called nearest neighbor classifier (NNC); the other is an extension of nearest neighbor classifier, class-nearest neighbor classifier (CNNC) [30]. We assume that the training set $A=[A_1;A_2;...;A_L]$, and $A_i=[a_{1i};a_{2i};\ldots;a_{ni}]$ denotes the ith class of the training samples ($i=1,\ldots, L$, Lis the number of the class; n_i is the number of the training samples of the ith class).

2.1 Nearest Neighbor Classifier

Nearest neighbor classifier is the most straightforward classifier in the machine learning techniques, and the test samples are classified based on the class of their nearest neighbors. Thus, for a test sample a, the NNC computes the distances between the test sample and all the training samples and assigns the test sample to the class containing the nearest neighbor.

Then, we give a brief description of nearest neighbor classifier. We can define a function to measure the distance between the test sample a and the ith class:

$$e_i(a)=\min\left\|a-a_i^j\right\|, j=1,2,...,n_i \qquad \textbf{(1)}$$

Here, $\|\bullet\|$ is a norm, in this paper, we use Euclidian distance to find the nearest neighbor; if $e_j(a)=\min e_i(a)$; $i=1,2,...,L$, then $a\in A_j$ add $j=\arg\min e_i(a)$.

From above, we can see that the NNC is intuitive and simple.

2.2 Class-nearest Neighbor Classifier

In [30], authors proposed a new classifier which is based on the average squared Euclidean distance between the test sample and elements of a given class, measured in the approximate subspace (CNNC). For a test sample a, the distance between a and training class A_i can be defined:

$$e_i(a) = \frac{1}{n_i}\sum_{j=1}^{n_i}\left\|a - a_i^j\right\|, i = 1, 2, ..., L \tag{2}$$

Where n_i is the number of training samples in class i. Thus, the test sample a will then be classified the class j with the smallest $e_i(a)$. When all the n_i=1, class-nearest neighbor classifier is equivalent to nearest neighbor classifier; and this method can be identified as an extended method of nearest neighbor classifier. Since class-nearest neighbor classifier can exploit the information in a class, it can get better results than nearest neighbor classifier; in particular, when n_i is large.

3 The Proposed Method

Since CNNC has the advantage on methods before, we need to find a fast method to solute the formulate (2), so in this section we give a detailed description of our proposed method.

Assume that a given training dataset is composed of *nm*-dimensional column vectors a_i, where i=1, 2…, n. For simplicity, we can use *a* $m \times n$ matrix A to represent this dataset:

$$A = [a_1, a_2, \ldots, a_n].$$

Stage 1: Apply a feature extraction algorithm on the training set (PCA or LDA), and then get the orthogonal subspace U_r.

Stage 2: Reset the training samples, $\tilde{A} = U_r^T A \in R^{r \times n}$; for a given test sample a, we also project it on the orthogonal subspace U_r, thus, $\tilde{a} = U_r^T a \in R^r$.

Stage 3: Let $\tilde{A}_i = \left[\tilde{a}_i^1, \tilde{a}_i^2, \cdots, \tilde{a}_i^3,\right]$ (i = 1…, L, L is the number of the class) denotes the ith class of the training samples and n_i is the number of the training samples of the ith class. In this stage, we want to solve this equation:

$$\tilde{a}_i = x_i^1\tilde{a}_i^1 + x_i^2\tilde{a}_i^2 + \cdots + x_i^3\tilde{a}_i^3 \tag{3}$$

the compact matrix form of (3) can be written as

$$\tilde{a} = \tilde{A}_i x^i$$

where $x_i = (x_i^1, x_i^2, \cdots, x_i^{n_i})^T$.

Apply skinny QR-decomposition [31] to the matrix $\tilde{A}_i$ as $\tilde{A}_i = Q_i R_i$,

where $Q_i \in R^{r \times n_i}$, $R_i \in R^{n_i \times n_i}$.

Let $Q_i = (Q_i^1 Q_i^2)$ be a column partition of Q_i, where $Q_i^1 \in R^{r \times n_i^r}, Q_i^2 \in R^{r \times (n_i - n_i^r)}$, ($n_i^r$ is the rank of matrix $\tilde{A}_i$), then

$$\tilde{A}_i = (Q_i^1, Q_i^2)\begin{pmatrix} R_i^1 \\ 0 \end{pmatrix} = Q_i^1 R_i^1$$

Where $R_i = \begin{pmatrix} R_i^1 \\ 0 \end{pmatrix}$ is a row partition of R_i, and $R_i^1 \in R^{n_i^r \times n_i}$. So, we can get the solution:

$$x_i = (R_i^r)^{-1}(Q_i^1)^T \tilde{a}$$

Stage 4: This stage calculates the deviation between the test sample $\tilde{a}$ and the training samples of the ith class:

$$e_i = \left\|\tilde{a} - \tilde{A}_i x_i\right\|^2$$

Our method regards that the smaller e_i is, the greater ability of representing the test sample of the ith class has; thus, we can point out that each class has only one deviation and the ith deviation corresponds to the ith class. If e_t=min e_i, then the test sample is classified into the tth class. We refer to the nearest neighbor from the tth class as the nearest neighbor determined by our method.

From above, we can find that our propose method can exploit the subspace of each class; thus, it is an improved method of class-nearest neighbor classifier, and we call it CSNNC for simplicity.

3.1 Analysis of Our Method

In this subsection, we analyze our method for exploring its characteristics. Our method differs from other classifiers as follows: it does not directly compute the distance between the test sample and each training sample after feature extraction.

Our method can be viewed as a special representation; indeed, it exploits the subspace of every class and the test sample is represented by the linear combination of the training samples of every class. If the training samples in any class is one, our method is equivalent to the nearest neighbor classifier. Our method is also different with the sparse representation and the collaborative representation, which test sample is represented by the subspace of the whole training samples [32].

Our method has two advantages. The first is that it combines the feature extraction with classification, it is said that the classification exploits the low-dimensional subspace from training samples to express and classify the test sample. When solving the linear system in equation (3), instead of computing the singular value decomposition of A_i, we apply skinny QR-decomposition on A_i, which is much faster and more efficient, and we can solve the singularity of $\tilde{A}_i$ easily. So, from our analysis, for a test sample a, we only need $O(\sum_{i=1}^{L}(rn_i^2 + n_i^3))$ at stage 3 (L is the number of classes, r is the dimension of low-dimensional subspace and n_i is the number of training samples of ith class, both of r and n_i are small integers), which is very effective. The second advantage is that though it is very simple and partially similar to nearest neighbor classifier (when the training samples of any class is one), thus, it also can be viewed as an extended method of nearest neighbor classifier.

3.2 Extension of Our Method

This subsection presents an extended algorithm of our algorithm (EC-SNNC). Now, we can give a description of this algorithm.

At the stage 3 of our propose method, we can get x_i; so, we can denote $\tilde{a}_i = \tilde{A}_i x_i$. The formula to use $\tilde{a}_i$ to express the test sample $\tilde{a}$ is:

$$\tilde{a} = \sigma_i \tilde{a}_i + b_i \tag{4}$$

where σ_i is the coefficient and b_i is the error vector. So, the test sample $\tilde{a}$ can be presented by the reconstructed vector $\tilde{a}_i$ weighted by a coefficient and a error vector. Now we can convert (4) into

$$\tilde{a}_i^T \tilde{a} = \sigma_i \tilde{a}_i^T \tilde{a}_i + \tilde{a}_i^T b_i, i = 1,...,L \tag{5}$$

Since b_i is an error vector, we solve (5) using:

$$\sigma_i = \frac{\tilde{a}_i^T \tilde{a}}{\tilde{a}_i^T \tilde{a}_i}, i = 1,...,L \tag{6}$$

We can assume that all the samples are unit vectors (2-norm is one), then we have

$$\sigma_i = \tilde{a}_i^T \tilde{a}, i = 1,...,L \tag{7}$$

ECSNNC can express the distance of test sample and the reconstructed vector by

$$e_i = \|\tilde{a} - \sigma_i \tilde{a}_i\|^2, i = 1,...,L \tag{8}$$

Where σ_i is solved by (6). The smaller distance e_i is, the better ability of expressing the test sample the ith class of training samples has. Thus, we can classifies the test sample into the jth class, if $e_j = \min e_i$.

Assume that all of the samples are unit vectors, we can transform (8) into

$$\begin{aligned} e_i &= 1 - 2\sigma_i \tilde{a}_i^T \tilde{a} + \sigma_i^2 \\ &= 1 - \sigma_i^2 = 1 - \tilde{a}_i^T \tilde{a} \\ i &= 1,\ldots,L \end{aligned} \tag{9}$$

If only one training sample in ith class, the distance between ith training sample and the test sample can be formulated as

$$\begin{aligned} e_i &= \left\| \tilde{a} - \tilde{a}_i \right\|^2 = 2 - 2\tilde{a}_i^T \tilde{a}, \\ i &= 1,\ldots,L \end{aligned} \tag{10}$$

Since NNC classifies the test sample based on the distance metric as shown in (10), it is sure that the classification based on (9) has the same classification decision as NNC when only one training sample in ith class. So, under the condition that all the classes only have one training sample, our method is identical to NNC.

4 Numerical Experiments

In this section, we evaluate the effectiveness of our propose method. All computations are done using Matlab 2014b on an Intel CPU@3 GHz, 8 GB memory computer. And we mainly test our method on PCA [10] and DLDA [33]. Moreover, the experiments on four classification methods, i.e. NNC, CNNC, CSNNC and ECSNNC were also performed.

4.1 Experimental Results on the ORL Database

From the ORL face database [34], we used 400 face images from 40 persons (10 images per person). These images were manually cropped and resized to 112×92 pixels, with 256 grey levels per pixel. After the image cropping, most of the complex background has been excluded. Before all the methods were implemented, all the training samples and test samples were normalized as vectors with length 1.

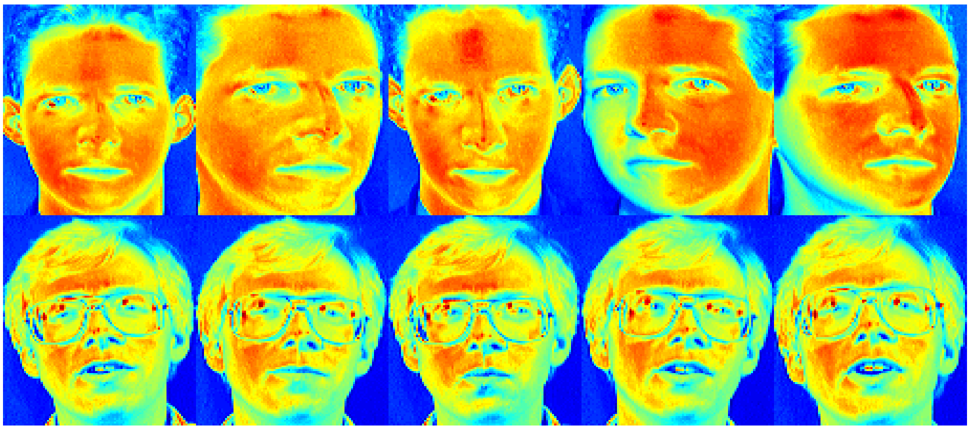


Fig. 1. Some face images of two subjects in the ORL database

Table 1 and Table 2 show the experimental results. It tells us that CSNNC and ECSNNC can obtain improved results in classification accuracy. Compared with the NNC, the other three methods, CNNC, CSNNC and ECSNNC can get much higher recognition rate. And in most cases, CSNNC and ECSNNC can get higher recognition rates than CNNC.

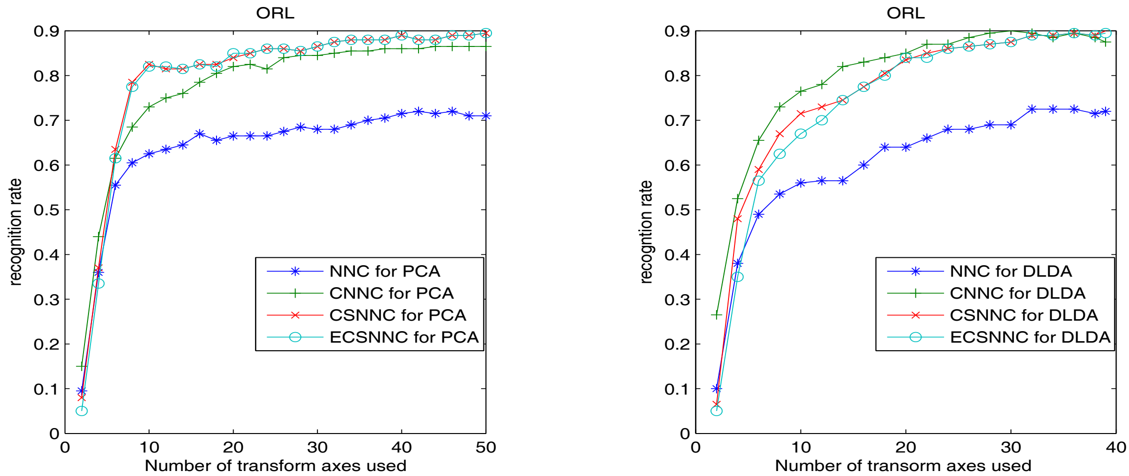
Table 1. Recognition rates (percent) of different methods on ORL database (The number of the transform axes used on PCA is equal to the total number of the subjects.)

Number of training samples per class	2	3	4	5
NNC for PCA	44.58	56.87	60.00	71.50
CNNC for PCA	76.67	76.04	78.13	86.00
CSNNC for PCA	75.83	82.92	80.94	89.00
ECSNNC for PCA	75.83	82.92	80.94	89.00

Table 2. Recognition rates (percent) of different methods on ORL database (The number of the transform axes used on DLDA is equal to the total number of the subjects minus one.)

Number of training samples per class	2	3	4	5
NNC for DLDA	42.50	55.67	70.00	72.00
CNNC for DLDA	80.94	82.29	90.00	87.50
CSNNC for DLDA	79.37	83.33	90.83	90.00
ECSNNC for DLDA	79.37	83.33	90.83	90.00

Fig. 2 shows the recognition rates of different classifiers. In our experiments, the training set consists of the first five images of per subject, and the test set consists of the remaining 5 images of per subject. From the left figure of Fig. 2, we can find that CSNNC and ECSNNC can get best results, and CNNC also has good recognition rates, while the NNC is the worst. From the right figure of Fig. 2, we can see that CNNC can get highest recognition rate when the dimensions are small, but CSNNC and ECSNNC can get similar results when dimensions are large, and the NNC always gets the worst results.



PCA DLDA

Fig. 2. Comparisons of recognition rate with different dimension OR

Fig. 3 shows Original test images of four subjects from the ORL database and the images correspondingtotheresultsofthelinearcombinationsofthefirstfourclassesoftraining samples for representing the test samples. The first column shows the original test images from the first four classes (we choose 1 image per subject), from the second column to the fifth column show the images corresponding to the result of the linear combination, respectively.

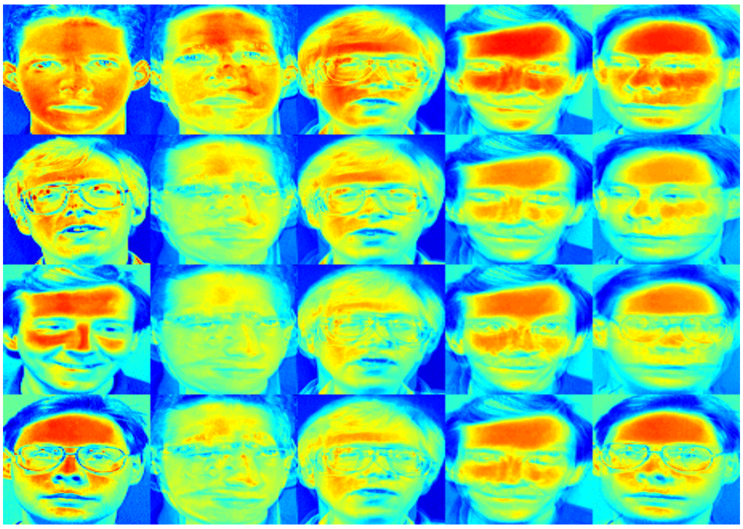
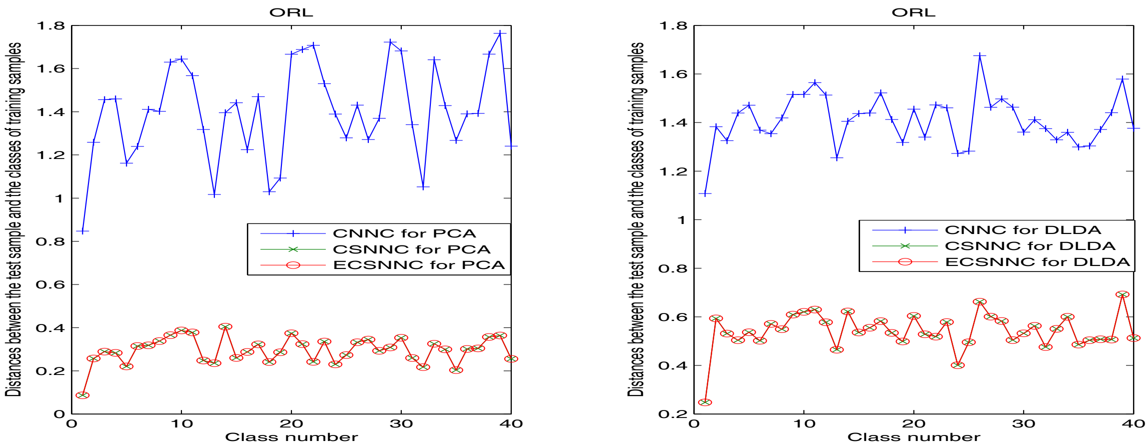


Fig. 3. Original test images and linear combinations images

The first row of Fig. 3, we choose the first image of ORL as an original image, and from the second image to the fifth image are reconstructed images by our method without feature extraction. In this example, we use the last five images of different classes as training set; thus, the second image of the first row is the linear combination by the last five images of the first subject and the third image of the first row is the linear combination by the second subject, etc. The images from the second row to the forth row are similar to the first row. From Fig. 3, we can find that the best reconstructed images are always be combined by the linear combination of their original class.

In Fig. 4, we show the distances between the test image of ORL and the linear combinations of all of the classes of training samples. In this example, the test sample is the first image of ORL, the training samples are constituted by the last five images of each subject, and the numbers of transform axes are the same as Table. 1 and Table. 2. From Fig. 4, we can find that CSNNC and ECSNNC can get the same distances, while CNNC gets larger distances.



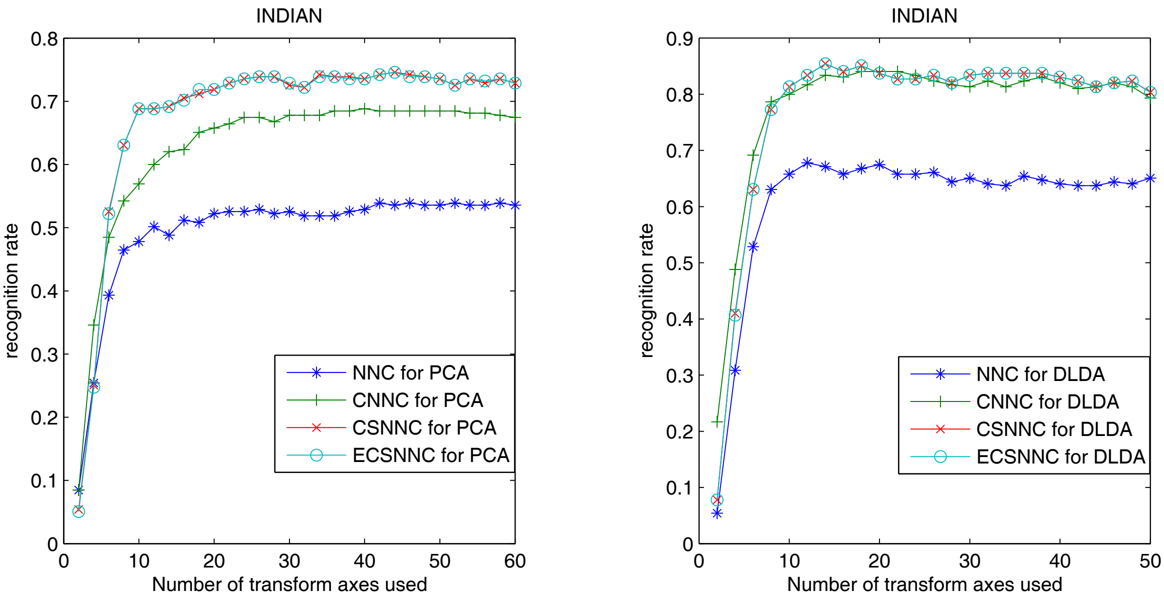
PCA DLDA

Fig. 4. The distances between the test image of ORL and the linear combinations of all of the different classes of training samples

4.2 Experimental Results on the INDIAN Database

The Indian face database [35] contains 22 females and 37 male subjects. For each of these 59 subjects, the database contains 11 face images (but we only select the first 10 of these images). Similar to ORL face database, these images were also manually cropped and resized to 92×112 pixels, with 256 grey levels per pixel. And most of the complex background of these images has been excluded; all the training samples and test samples were normalized as vectors with length 1 before all the methods were implemented.

Fig. 5. Some face images of two subjects in the INDIAN database



PCA DLDA

Fig. 6. Comparisons of recognition rate with different dimension (INDIAN)

From Table. 3, for PCA, compared with NNC and CNNC, CSNNC and ECSNNC always get the better results. For DLDA, from Table. 4, we can find that when the number of training samples per class is 2 or 3, CNNC can get the highest recognition rates; and when the number of training samples per class is 4 or 5, CSNNC and ECSNNC have higher recognition rate.

Table 3. Recognition rates (percent) of different methods on INDIAN database (The number of the transform axes used on PCA is equal to the total number of the subjects.)

Number of training samples per class	2	3	4	5
NNC for PCA	30.51	43.36	49.89	53.90
CNNC for PCA	63.35	64.41	68.64	67.80
CSNNC for PCA	63.56	69.07	72.56	73.90
ECSNNC for PCA	63.56	69.07	72.56	73.90

Table 4. Recognition rates (percent) of different methods on INDIAN database (The number of the transform axes used on DLDA is equal to the total number of the subjects minus one.)

Number of training samples per class	2	3	4	5
NNC for DLDA	32.50	46.61	57.10	61.36
CNNC for DLDA	64.19	71.47	75.64	77.97
CSNNC for DLDA	62.50	69.07	76.27	79.32
ECSNNC for DLDA	62.50	69.07	76.07	79.32

Fig. 7 shows original test images of our subjects from the INDIAN database and the images corresponding to the results of the linear combination of the first four classes of training samples for representing the test samples. The first column shows the original test images from different classes (we choose 1 image per subject), from the second column to the fifth column show the images corresponding to the result of the linear combination, respectively.

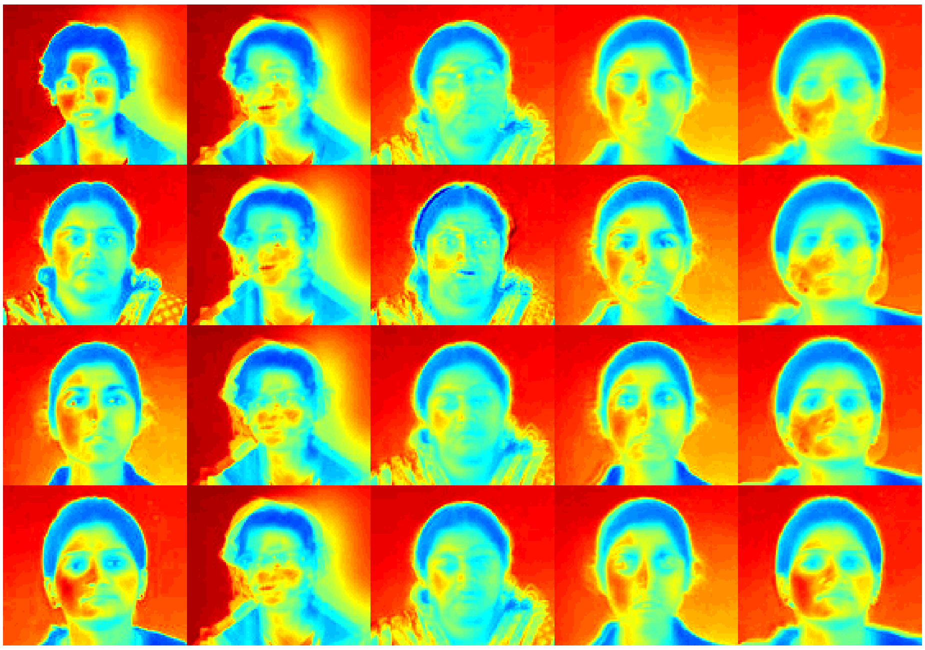
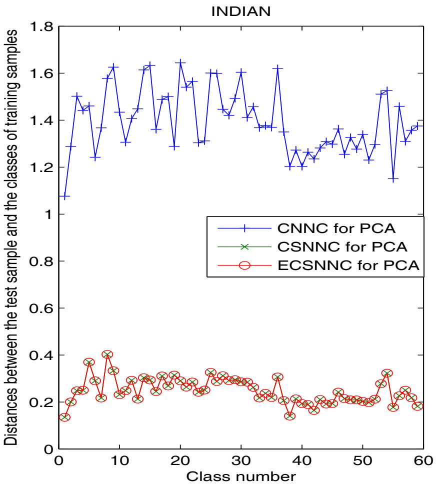


Fig. 7. Original test images and the linear combination of the samples

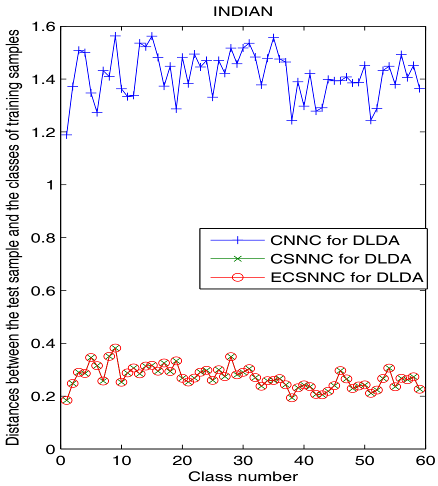
In this example, we take the first five images of each class as the training set, and the test samples are constituted by the left images. For PCA, we can find that CSNNC and ECSNNC get the best results, and NNC gets the worst results. Different with Fig. 2, for DLDA, CSNNC and ECSNNC can also get the better results than CNNC.

Original images are reconstructed by our method without feature extraction. This example is similar to Fig. 3, and the figure shows that the best reconstructed images are always be combined by the linear combination of their original class.

In Fig. 8, similar to Fig. 4, we also take the first image of INDIAN as the test sample and choosing the last 5 images of all of the classes as the training samples. And from Fig. 8, CSNNC and ECSNNC have the same distances, and CNNC has larger distances. For three methods, this figure tells us that the test sample is classified to the first class.



PCA



DLDA

Fig. 8. The distances between the test image of INDIAN and the linear combinations of all of the different classes of training samples

5 Conclusions

In this paper, we propose a novel classification method, which we call it CSNNC. This method modifies NNC and exploits the abilities, of representing the test sample by the linear combination of each class of the training samples. This ability is related to the "similarity" between the test sample and each class of the training samples. Our proposed method evaluates the "similarity" between the test sample and the training samples of each class, whereas NNC directly compute the "similarity" between the test sample and each training sample. On the other hand, compared with CNNC, our method is not only a class-nearest neighbor classifier, but our method is also a special representation method, which can exploit the subspace information between the training set and test sample. Thus, CSNNC uses both the training samples and test sample to obtain an "optimal" representation of the test sample that is very beneficial to classification; and it is the reason why CSNNC and ECSNNC can get higher recognition rate than CNNC. Another advantage of CSNNC is that we apply skinny QR-decomposition for least square problems, and this method can work very efficient and solve the singularity. The experimental results show that CSNNC obtain a great improvement in recognition rate.

Acknowledgements

This work is supported by: (1) China Agriculture Research System (CARS-27); (2) Quality Project of JiayingUniversity, 2015 (3-6-25); (3) Special Fund for Agro-scientific Research in the Public Interest of China (201203016); (4) Yang Fan Innovative & Entrepreneurial Research Team Project (201312G06). The financial contributions are gratefully acknowledged.

References

[1] A. Djouadi, E. Bouktache, A Fast algorithm for the nearest-neighbor classifier, IEEE Transactions on Pattern Analysis and Machine Intelligence 19(3)(1997) 277-282.

[2] C.-C. Chiang, Z.-W. Chen, Recognizing partially-occluded faces by recovering normalized facial appearance, International Journal of Innovative Computing, Information and Control 7(11)(2011) 6219-6234.

[3] X.-S. Wang, F. Glineur, L.-Z. Lu, P. V. Dooren, Extended Lanczos bidiagonalization algorithm for low rank approximation and its applications, Journal of Computational and Applied Mathematics, (301)(2016) 213-339.

[4] X.-S. Wang, Z. Chen, Z.-M. Teng, Determination of candidate subjects for better recognition of faces, Journal of Modern Optics (10)(2016)1-7.

[5] X.-S. Wang, Y. Chen, Differentiated representation and applications to face recognition, Optic, (140)(2017) 216-222.

[6] H. Zhang, A.C. Berg, M.M.J. Malik, SVM-KNN: Discriminative nearest neighbor classification for visual category recognition, Computer Vision and Pattern Recognition 2(2006) 2126-2136.

[7] J. Yang, C. Liu, Color image discriminant models and algorithms for face recognition, IEEE Transactions on Neural Networks 19(12)(2008) 2088-2098.

[8] J. Yang, D. Zhang, A.F. Frangi, J.-Y. Yang, Two-dimensional PCA: a new approach to appearance-based face representation and recognition, IEEE Transaction on Pattern Analysis and Machine Intelligence 26(1)(2004) 131-137.

[9] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd ed., Academic Press, 1990.

[10] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3(1991) 71-86.

[11] O. Kramer, Dimensionality Reduction with Unsupervised Nearest Neighbors, Springer, 2013.

[12] S.A. Nene, S.K. Nayar, A simple algorithm for nearest neighbor search in high dimensions, IEEE Transactions on Pattern Analysis & Machine Intelligence 19(9)(1997) 980-1003.

[13] S. Berchtold, B. Ertl, D.A. Keim, H.P. Kriegel, T. Seidl, Fast nearest neighbor search in high-dimensional spaces, in: Proc. of the 14th IEEE Conference on Data Engineering, 1998.

[14] S. Shan, B. Cao, Y. Su, L. Qing, Unified Principal Component Analysis with Generalized Covariance Matrix for Face Recognition, CVPR, 2008.

[15] Y. Xu, Q. Zhu, J. Wang, Breast cancer diagnosis based on a kernel orthogonal transform, Neural Computing and Applications 8(2012) 1865-1870.

[16] Y. Xu, D. Zhang, J.-Y. Yang, A feature extraction method for use with bimodal biometrics, Pattern Recognition 43(2010) 1106-1115.

[17] M. Kan, S. Shan, Y. Su, X. Chen, W. Gao, Adaptive discriminant analysis for face recognition from single sample per person, in: Proc. IEEE International Conference on Automatic Face and Gesture Recognition, 2011.

[18] S. Yan, D. Xu, Q. Yang, L. Zhang, Multilinear discriminant analysis for face recognition, IEEE Trans Image Process 16(1)(2007) 212-C220.

[19] Z. Fan, Y. Xu, D. Zhang, Local linear discriminant analysis framework using sample neighbors, IEEE Transactions on Neural Networks 22(2011) 1119-1132.

[20] Y. Xu, A new kernel mse algorithm for constructing efficient classification procedure, International Journal of Innovative Computing, Information and Control 5(2009) 2439-2447.

[21] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Computation 12(10)(2000) 2385-2404.

[22] S.A. Billings, K.-L. Lee, Nonlinear fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm, Neural Networks 15(2)(2002) 263-270.

[23] Y. Xu, D. Zhang, Z. Jin, M. Li, J.-Y. Yang, A fast kernel-based nonlinear discriminant analysis for multi-class problems, Pattern Recognition 39(2006) 1026-1033.

[24] K.I. Kim, K. Jung, H.J. Kim, Face recognition using kernel principal component analysis, IEEE Signal Processing Letters 9(2)(2002) 40-42.

[25] J. Li, X. Li, D. Tao, KPCA for semantic object extraction in images, Pattern Recognition 41(10)(2008) 3244-3250.

[26] Y. Xu, D. Zhang, F. Song, J.-Y. Yang, Z. Jing, M. Li, A method for speeding up feature extraction based on KPCA Neuro computing 70(2007) 1056-1061.

[27] N. Guan, D. Tao, Z. Luo, Non-negative patch alignment frame work, IEEE Trans Neural Networks 22(8)(2011) 1218-1230.

[28] P.Y. Simard, Y.-L. Cun, J. Decker, Efficient pattern recognition using a new transformation distance, in: Proc. Advances in Neural Information Processing Systems 6, 1993.

[29] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions in Information Theory, 13(1)(1967) 21-27.

[30] F. Glineur, L. Lu, P.V. Dooren, X.-S. Wang, Extended Lanczos bidiagonalization for dimension reduction in information retrieval, in: Proc. 2012 8th International Conference on Natural Computation, 2012.

[31] G.H. Golub, C.F.V. Loan, Matrix Computations, 4th ed., The Johns Hopkins University Press, 2013.

[32] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: Which helps face recognition?, in: Proc. the 2011 International Conference on Computer Vision, 2011.

[33] AT&T, face-database. <http://www.cl.cam.ac.uk/research/dtg/attarchive/face-database.html>, 2017 (accessed 17.02.20).

[34] Database, I. F., <http://vis-www.cs.umass.edu/vid-it/IndianFaceDatabase>, 2017 (accessed 17.02.20).