Neural Comput & Applic (2013) 22:1543–1549
DOI 10.1007/s00521-012-0833-5

ORIGINAL ARTICLE

A simple and fast representation-based face recognition method

Yong Xu · Qi Zhu

Received: 29 March 2011 / Accepted: 5 January 2012 / Published online: 1 February 2012

Abstract In this paper, we propose a very simple and fast face recognition method and present its potential rationale. This method first selects only the nearest training sample, of the test sample, from every class and then expresses the test sample as a linear combination of all the selected training samples. Using the expression result, the proposed method can classify the testing sample with a high accuracy. The proposed method can classify more accurately than the nearest neighbor classification method (NNCM). The face recognition experiments show that the classification accuracy obtained using our method is usually 2–10% greater than that obtained using NNCM. Moreover, though the proposed method exploits only one training sample per class to perform classification, it might obtain a better performance than the nearest feature space method proposed in Chien and Wu (IEEE Trans Pattern Anal Machine Intell 24:1644–1649, 2002), which depends on all the training samples to classify the test sample. Our analysis shows that the proposed method achieves this by modifying the neighbor relationships between the test sample and training samples, determined by the Euclidean metric.

Keywords Pattern recognition · Face recognition · Computer vision · Biometrics

Y. Xu (✉) · Q. Zhu
Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China
e-mail: laterfall286@yahoo.com

Y. Xu · Q. Zhu
Key Laboratory of Network Oriented Intelligent Computation, Shenzhen, China

1 Introduction

It is remarkable that recently the researchers in the field of pattern recognition proposed a promising image recognition method, i.e. the so-called sparse representation method [2–4]. This method does not classify images in the conventional way. It first uses a linear combination of a subset of the training samples to express the test sample. Then, it bases on the expression result to classify the test sample. This method has obtained a very good performance and has been commented as a face recognition breakthrough [5]. However, the sparse representation method has a very high computational cost. This is mainly because it depends on an iterative algorithm to obtain its solution. The sparse representation method was also used for breast cancer biomarker identification and classification [6], signal processing [7] and image decomposition [8].

The main difference between the sparse representation method and previous face recognition methods is as follows: previous face recognition methods are usually composed of three stages: the feature extraction, classifier selection and classification. The feature extraction stage is usually implemented by a transform method such as the methods based on the independent component analysis methodology [9–11], on the principal component analysis methodology [12–17] and on the discriminant analysis methodology [18–25]. However, the sparse representation method adopts a noticeable novel way to address the face recognition problem. It does not contain the feature extraction and classifier selection stages. Instead, it first attempts to express the test sample as a sparse linear combination of the training samples. Hereafter, the term 'sparse linear combination of the training samples' means that if the test sample is expressed as a linear combination of all the training samples, the majority of the components

of the combination are zero. It seems that the sparse representation method attempts to seek the potential effects of different training samples on 'constructing' the test sample and intends to classify the test sample into the class whose training samples has the maximum effect. In other words, this method assumes that the test sample can be approximated by the sum of the effects of the training samples from different classes and the class that has the maximum effect is the most similar to the test sample.

In this paper, we propose a very simple and fast face recognition method. This method is partially similar to the sparse representation method, whereas it is computationally much more efficient. Moreover, our method has a distinctive characteristic that its solution can be solved with ease. The proposed method first selects only one neighbor sample, for the test sample, from each class and then expresses the test sample as a linear combination of all the selected neighbor samples. Finally, the method constructs a classification procedure on the basis of the expression result. The proposed method performs well in face recognition applications.

As the proposed method uses only L training samples to express and classify the test sample (L is the number of all the classes), it is also a sparse representation method and is able to inherit the advantages of this kind of method. The 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 L training samples to express and classify the test sample, we will obtain a higher classification error. The experimental result shows that the proposed method outperforms the nearest neighbor classification method (NNCM). NNCM is indeed a special form of the k nearest neighbor classifier [26]. NNCM first determines the training sample that is the nearest to the test sample and then classifies the test sample into the class of the training sample. It seems that the proposed method achieves this by modifying the neighbor relationships, between the test sample and training samples, determined by the Euclidean metric. Moreover, though our method depends on much fewer training samples to perform classification than the NFS method proposed in [1], it might obtain a better performance.

The rest of the paper is organized as follows: Sect. 2 describes our method. Section 3 provides our analysis of the proposed method. Section 4 presents our experiments and results. Section 5 offers our conclusion.

2 The proposed method

In this section, we formally describe our proposed method. This method consists of two processes. The first process selects the nearest training sample, of the test sample, from every class. Supposing there are L classes, we obtain L nearest training samples (NTS), for the test sample, each being from one class. The second process expresses the test sample as a linear combination of all the selected L NTSs and exploits the determined linear combination to classify the test sample. Hereafter, we assume that each training and testing samples are all column vectors.

The first process of our proposed method works as follows: let A_i^k ($i = 1, 2, \cdots n_k$, $k = 1,2, \cdots L$) denote the ith training samples of the k th class and n_k be the number of the training samples of the k th class. This process calculates the distance between test sample y and A_i^k using

$$d_i^k = ||A_i^k - y||^2. \tag{1}$$

If $j = \arg\min_i d_i^k$, then A_j^k is identified as the nearest training sample from the kth class. We denote A_j^k by NTS_k. Once the first process identifies all the NTS_k, $K = 1, 2, \cdots, L$, we define matrix $S = [\text{NTS}_1 \cdots \text{NTS}_L]$.

The second process of our proposed method works as follows: it assumes that test sample y can be approximately represented by a linear combination of all the NTS_k, $K = 1, 2, \cdots, L$. In other words, it assumes that the following equation is approximately satisfied:

$$y = \sum_{i=1}^{L} \beta_i \text{NTS}_i. \tag{2}$$

Eq. (2) can be rewritten into

$$y = S\beta, \tag{3}$$

where $\beta = (\beta_1 \cdots \beta_L)^T$. If $S^T S$ is not singular, we can obtain the least squares solution of (3) using $\beta = (S^T S)^{-1} S^T Y$. If $S^T S$ is nearly singular, we can solve β using $\beta = (S^T S + \mu I)^{-1} S^T Y$, where μ is a positive constant and I is the identity. We refer to this solution scheme as regularized solution scheme of our method. After the β is obtained, we use $\hat{y}$ to denote $S\beta$, i.e. $\hat{y} = S\beta$. We refer to $\hat{y}$ as the expression result of our method. In Sect. 4, we will convert the one-dimensional expression result into a two-dimensional image, which allow us to see how the expression result is close to the original test sample.

Eq. (2) shows that each NTS makes its own contribution to representing the test sample and the contribution that the ith NTS makes is β_i NTS_i. Moreover, the ability, of representing the test sample, of the ith NTS can be evaluated by the deviation between β_i NTS_i and Y. We define the deviation as $e_i = ||Y - \beta_i \text{NTS}_i||^2$. Our method regards that the smaller e_i is, the greater ability of representing the test sample the ith NTS has. The second process identifies the NTS that has the minimum deviation from the test sample and classifies the test sample into the class of the identified NTS. It should be pointed out that each class has only one

NTS and the ith NTS 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.

3 Analysis of our method

In this section, we analyze our method for exploring its characteristics. Our method differs from the sparse representation method in [2] as follows: our method has a very simple solution scheme, whereas the iterative solution scheme of the sparse representation method in [2] is computationally much inefficient. On the other hand, our method can be viewed as a special sparse representation method. Indeed, if the linear combination in our method is compulsorily rewritten as a linear combination of all the training samples, then the coefficients of the linear combination of all the training samples except for NTSs should be zero.

Although both our method and the method in [2] work as sparse representation methods, they achieve the sparseness in two different ways. Our method uses the first process to produce the sparseness and it is known how sparse the coefficients are (i.e. there are how many zero coefficients) and which coefficients are zero. However, the sparse representation method proposed in [2] achieves the sparseness by its iterative solution scheme and it is not clearly known which coefficients of the linear combination are equal or close to zero. We also refer to our method as "hard" sparse representation method. On the other hand, the method in [2] can be referred to as "soft" sparse representation method.

Our method has two advantages. The first is that it exploits only a small number of training samples to express and classify the test sample. As a result, our method can solve (3) computationally efficiently. When solving the linear system shown in (3), our method needs a time complexity of $O\ (L^2M + L^3 + LM)$, where M is the dimension of the sample vector. The NFS method in [1] needs a time complexity of $O(L(n^2M + n^3 + nM)) = O(nNM + Nn^2 + NM)$ to solve the L linear systems. n and N are, respectively, the numbers of training samples of each class and all the training samples. The second advantage of our method is that though it is simple and partially similar to NNCM, it can perform better than NNCM as shown in Sect. 4. NNCM can be described as a method that exploits the nearest neighbor from each class to classify the test sample. That is, we can present NNCM as follows: NNCM first selects the nearest neighbor from each class for the test sample. NNCM identifies the sample that is the closest to the test sample among the L selected nearest neighbors (NTSs) and assumes that the test sample is from the same class. L is the number of all the classes. It seems that the main reason why our method outperforms NNCM is to modify the neighbor relationships between the test sample and training samples, determined by the Euclidean metric. As shown in Sect. 2, in our method, the neighbor relationships, between the test sample and training samples are ultimately determined by the deviation between the test sample and the contribution to representing the test sample of the NTS. In other words, we can say that our method consists of the following two procedures: the first procedure uses the Euclidean metric to select neighbor samples from each class for the test sample. The second procedure using exploits the deviation defined in Sect. 2 to reorder the neighbor relationship between the test sample and the neighbor samples determined by the first procedure. That is, the second procedure uses the deviation between the test sample and the contribution to expressing the test sample of the NTS as the metric. Using this metric, the second procedure determines the 'final nearest neighbor' and classifies the test sample into the class that the 'final nearest neighbor' belongs to. We show the flowchart of our method using Fig. 1, which clearly shows that our method identifies the RTS having the minimum deviation and assumes that the test sample is from the class of the RTS identified. We can also say that the classification in our method is equivalent to the nearest neighbor classification based on the metric of 'deviation'.

When our method attempts to exploit L training samples to express the test sample, it indeed uses the most suitable and significant L samples. In other words, among all the training samples, the L nearest neighbors (NTSs) are the most L important training samples in terms of the ability of expressing the test sample. Actually, as shown in Sect. 4, if we use other L training samples to express and classify the test sample, the classification performance might be very poor.

4 Experimental results

We used the ORL [27], Yale [28] and AR [29] face databases to test our method. The face images in the ORL database 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. The AR face database is a large-scale database. We used 3,120 gray face images from 120 subjects from this database, each providing 26 images. These

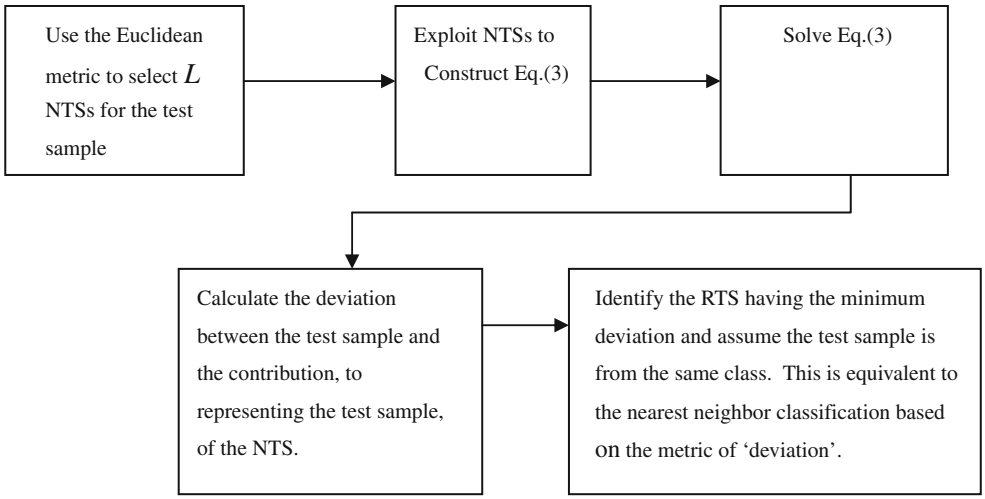


Fig. 1 Flowchart of our method. It is clear that the classifier in our method is equivalent to the nearest neighbor classifier based on the metric of 'deviation'

images were taken in two sessions [30] and show faces with different facial expressions, in varying lighting conditions and occluded in several ways. For the ORL and Yale databases, when s samples of all the n samples per class were used for training, we conducted experiments on all the C_n^s training sets and C_n^s testing sets. Before implementing all the methods, we converted each sample vector into a unit vector with the length of 1 in advance. We used the regularized solution scheme to obtain the solution of our method and set μ to 0.01. We also tested the NNCM and the NFS method proposed in [1]. Moreover, to show the reasonability of the way of selecting neighbors for the test sample, we also modified our method by selecting the first or last training sample from each class for the test sample and by using the selected first or last training sample to express and classify the test sample. We refer to the methods using the first and last training samples as the method using the first sample and the method using the last sample, respectively. We also modified our method by selecting the furthest neighbor, from each class, for the test sample and exploited them to express and classify the test

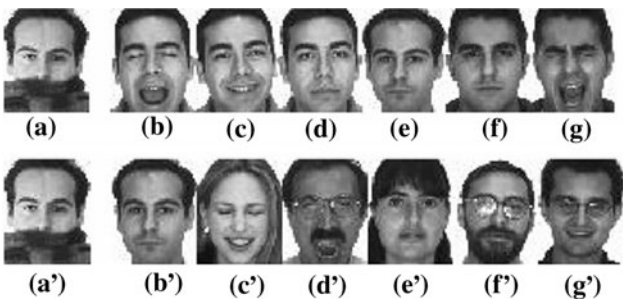
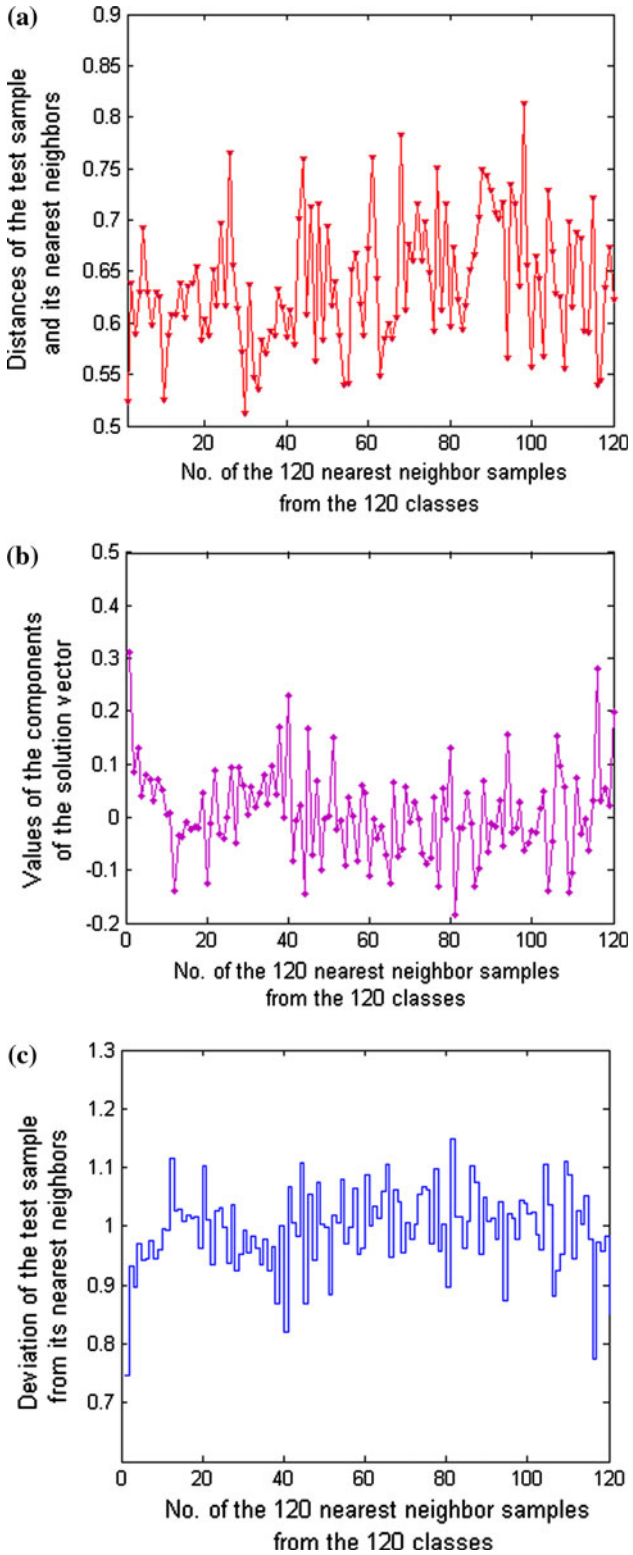


Fig. 2 The first six nearest neighbors, of one test sample of the first subject in the AR database, determined by NNCM and our method. Both (**a**) and (**a'**) denote the same test sample from the AR database. **b–g** denote the first six 'nearest neighbors' determined by NNCM. **b'–g'** denote the first six nearest neighbors determined by our method (these samples have the first six smallest deviations from the test sample). It is clear that our method can correctly classify the test sample, whereas NNCM will fail to do so. In this case, the first four images of each subject were used as training samples and the remaining samples were used as testing samples

Fig. 3 Some original test samples and the two-dimensional images corresponding to the expression result of our method. The *first and second rows* show some original test samples and the two-dimensional images corresponding to the expression result of our method, respectively. The *third and forth rows* show some other original test samples and the two-dimensional images corresponding to the expression result of our method, respectively. In this case, we also used the first four images of each subject as training samples and took the remaining samples as testing samples



◀ **Fig. 4** The distances (**a**) between the test sample shown in Fig. 2 and all the 120 nearest neighbor samples, the components (**b**) of the solution vector of our method and the deviation (**c**) of the test sample from its nearest neighbors. This figure shows that the proposed method can 'reorder' the 120 nearest neighbor training samples. **a** shows that the nearest neighbor from the 30th class is the closest to the test sample (as a result, NNCM will classify this test sample into the 30th class). **b** shows that the first component of the solution vector of our method has the maximum absolute value. **c** shows that the nearest neighbor from the first class has the minimum deviation from the test sample. As a result, our method will correctly classify the test sample. The deviation is defined in Sect. 2

sample. We refer to this method as the method using the furthest neighbor.

Figure 2 shows the first six nearest neighbors, of one test sample, determined by NNCM and our method on the AR database. This figure shows that the 'final nearest neighbor' determined by our method [shown in 2-(b')] is from the same subject as the test sample, but the nearest neighbor determined by NNCM [shown in 2-(b)] is not from the same subject as the test sample. As a result, our method can correctly classify the test sample, whereas NNCM will fail to do so. Figure 3 shows some original test samples and the two-dimensional images corresponding to the expression result of our method. It seems that when the face was not occluded, the two-dimensional image corresponding to the expression result was very similar with the original test sample. On the other hand, when the face was occluded, the two-dimensional image corresponding to the expression result was not very close to the original test sample. Figure 4 shows the distances between the test sample shown in Fig. 2 and all the 120 NTSs, the components of the solution vector of our method and the deviation of the test sample from the 120 NTSs. Figure 4a clearly (a) shows that the NTS from the 30th class is the closest to the test sample. Thus, NNCM will classify this test sample into the 30th class. However, Fig. 4c shows that the NTS from the first class has the minimum deviation from the test sample and our method will correctly classify the test sample. Figure 4 visually shows that our method can 'reorder' the neighbor relationships between the test sample and NTSs by using the deviation between the test samples and NTSs.

Tables 1, 2 and 3 show the experimental results. They show that our method always classifies more accurately than NNCM and the method using the furthest neighbor. For example, when the first four images per class from the AR database were used as training samples and the others were used as test samples, the ratios of the classification errors obtained using our method, NNCM, the NFS method proposed in [1] and the method using the furthest neighbor are 31.67, 42.69, 41.86 and 45.64%, respectively. We see that the difference between the rates of classification errors of NNCM and our method is 11.02%. In this case, the rate of classification errors of our method is also 10.19% lower than that of the NFS method. Moreover, the fact that our method always obtains a much lower error rate than the method using the furthest neighbor, the method using the first sample and the method using the last sample also

Table 1 Means of the rates of the classification errors of the nearest neighbor method and the proposed method on the AR database

Training samples	The proposed method (%)	NNCM (%)	The method using the furthest neighbor (%)	The method using the first training sample (%)	The method using the last training sample (%)	The NFS method in [1] (%)
2 per class	30.49	39.93	38.19	31.35	38.40	40.49
4 per class	31.67	42.69	45.64	33.67	44.05	41.86
6 per class	31.12	38.88	47.46	36.67	38.17	37.63

Table 2 Means of the rates of the classification errors of the nearest neighbor method and the proposed method on the Yale database

Number of training samples per class	1 (%)	2 (%)	3 (%)	4 (%)
The proposed method	14.73	6.38	4.59	3.80
NNCM	18.97	9.17	5.89	4.71
The method using the furthest neighbor	14.73	19.31	23.24	26.06
The method using the first training sample	–	14.80	13.76	12.73
The method using the last training sample	–	14.65	13.54	12.42
The NFS method in [1]	18.30	6.32	4.12	3.29
Number of training sets	11	55	165	330

Table 3 Means of the rates of the classification errors of the nearest neighbor method and the proposed method on the ORL database

Number of training samples per class	1 (%)	2 (%)	3 (%)	4 (%)
The proposed method	32.50	18.52	11.97	8.48
NNCM	33.94	20.54	13.83	9.98
The method using the furthest neighbor	32.50	39.43	42.40	44.20
The method using the first training sample	–	31.75	31.32	30.91
The method using the last training sample	–	33.25	33.57	33.86
The NFS method in [1]	33.94	18.88	11.54	7.42
Number of training sets	10	45	120	210

demonstrates that though these methods use the same number of training samples to express and classify the test sample, our method uses the most suitable training samples. Compared with the NFS, our method on the AR database classified much more accurately. For the ORL and Yale databases, the classification performance of our method is close to that of the NFS method.

5 Conclusion

NNCM can be described as a method that exploits the nearest neighbor from each class to classify the test sample as follows: if there are L classes, NNCM first selects L nearest neighbors, each being from one class, for the test sample. NNCM then identifies the sample that is the closest to the test sample among the L selected nearest neighbors and assumes that the test sample is from the same class. The method proposed in this paper also uses these nearest neighbors from each class to express and classify the test sample. It is remarkable that this method is able to achieve a much lower error rate than NNCM, and the maximum difference between the accuracies of the two methods might be greater than 10%. Our method achieves this by exploiting the ability of repressing the test sample of the training sample rather than only a simple distance to classify the test sample, which has been proven to be a good way to produce a high classification accuracy. The analysis also shows that for our method, the L nearest neighbors used are the most suitable training samples to express and classify the test sample. Moreover, though our method is 'sparser' than the NFS method proposed in [1], it might obtain a better performance. Besides the method design and experimental analysis, this paper also visually presents the rationale and characteristics of the proposed method.

Acknowledgments This article is partly supported by Key Laboratory of Network Oriented Intelligent Computation, Program for New Century Excellent Talents in University (Nos. NCET-08-0156 and NCET-08-0155), Natural Scientific Research Innovation

Foundation in Harbin Institute of Technology (HIT. NSRIF. 2009130), National Nature Science Committee of China (Nos. 61071179, 60803090, 60902099 and 61001037).

References

1. Chien JT, Wu CC (2002) Discriminant waveletfaces and nearest feature classifiers for face recognition. IEEE Trans Pattern Anal Machine Intell 24:1644–1649
2. Wright J, Yang AY, Ganesh A et al (2009) Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intell 31(2):210–227
3. Wright J, Ma Y, Mairal J, et al. (2009) Sparse. Representation for computer vision and pattern recognition. In: Proceedings of IEEE, pp 1–8
4. Xu Y, Zhang D, Yang J, Yang J-Y (2011) A two-phase test sample sparse representation method for use with face recognition. IEEE Trans Circuits Syst Video Technol. doi:10.1109/TCSVT.2011.2138790
5. Kroeker KL (2009) Face recognition breakthrough. Commun ACM 52(8):18–19
6. Shi Y, Dai D, Liu C, Yan H (2009) Sparse discriminant analysis for breast cancer biomarker identification and classification. Nat Sci 19(11):1635–1642
7. Candes E, Romberg J (2005) l_1-magic: recovery of sparse signals via convex programming http://www.acm.caltech.edu/l1magic/
8. Geiger D, Liu T, Donahue M (1999) Sparse representations for image decompositions. Int J Comput Vis 33(2):139–156
9. Hyvärinen A (1999) Survey on independent component analysis. Neural Computing Surveys 2:94–128
10. Liu C, Yang J (2009) ICA color space for pattern recognition. IEEE Trans on Neural Netw 20(2):248–257
11. Zhang L, Gao Q, Zhang D (2008) Directional independent component analysis with tensor representation. June, Anchorage, Alaska, U.S. 2008, CVPR, pp 1–7, 23–28
12. Moon H, Phillips PJ (2001) Computational and performance aspects of PCA-based face recognition algorithms. Perception 30:303–321
13. Yang J, Zhang D, Yang J-Y (2006) Locally principal component learning for face representation and recognition. Neurocomputing 69(13–15):1697–1701
14. Xu Y, Zhang D, Yang J-Y (2010) A feature extraction method for use with bimodal biometrics. Pattern Recogn 43:1106–1115
15. Yang J, Zhang D, Frangi AF, Yang J-Y (2004) Two-dimensional PCA: a new approach to appearance-based face representation and recognition. IEEE Trans Pattern Anal Mach Intell 26(1):131–137
16. Yang W, Sun C, Ricanek K (2011) Sequential row–column 2DPCA for face recognition, Neural Comput Applications. 24 June 2011, doi:10.1007/s00521-011-0676-5, pp 1–7
17. Xu Y, Zhang D (2011) Accelerating the kernel-method-based feature extraction procedure from the viewpoint of numerical approximation. Neural Comput Appl 20:1087–1096
18. Song F, Zhang D, Mei D, Guo Z (2007) A multiple maximum scatter difference discriminant criterion for facial feature extraction. IEEE Trans on Syst Man Cybern Part B 37(6):1599–1606
19. Etemad K, Chellappa R (1997) Discriminant analysis for recognition of human face images. J Opt Soc Am A 14(8):1724–1733
20. Xu Y, Yang J-Y, Lu J, Yu D-J (2004) An efficient renovation on kernel Fisher discriminant analysis and face recognition experiments. Pattern Recogn 37(10):2091–2094
21. Liu C, Wechsler H (2002) Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. IEEE Trans on Image Process 11(4):467–476
22. Loog M, Wu X-J, Lu J-P (2008) A note on an extreme case of the generalized optimal discriminant transformation. Neurocomputing 72(1–3):664–665
23. Wu F, Wang W, Yang Y, Zhuang Y, Nie F (2010) Classification by semi-supervised discriminative regularization. Neurocomputing 73(10–12):1641–1651
24. Yang J, Yang J-Y (2003) Why can LDA be performed in PCA transformed space? Pattern Recogn 36(2):563–566
25. Xu Y, Yang J-Y, Jin Z (2004) A novel method for Fisher discriminant analysis. Pattern Recogn 37:381–384
26. Nanni L, Lumini A (2009) Particle swarm optimization for ensembling generation for evidential k-nearest-neighbour classifier. Neural Comput Appl 18(2):105–108
27. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
28. http://cvc.yale.edu/projects/yalefaces/yalefaces.html
29. http://cobweb.ecn.purdue.edu/∼aleix/aleix_face_DB.html
30. Xu Y, Jin Z (2008) Down-sampling face images and low-resolution face recognition. The third international conference on innovative computing, information and control, 18–20 June, Dalian, China, pp 392–395