

Research Article

A Novel Face Recognition Approach Based on Two-Step Test Sample Representation

Zun-xiong Liu, Zhi-qiang Huang, and Heng Zhang

School of Information Engineering, East China Jiaotong University, Nanchang 330013, China

Correspondence should be addressed to Zun-xiong Liu; liuzunx@tom.com

Received 1 November 2012; Revised 10 January 2013; Accepted 11 February 2013

Academic Editor: P. Liatsis

Copyright © 2013 Zun-xiong Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The two-step test sample representation method is proposed for face recognition. It first identifies k “representative” samples from each category training samples for the test sample then produces a weighted sum of all the “representative” samples that well approximates the test sample. This method assigns the test sample to the class whose training samples have the smallest deviation from the test sample. As the method proposed in this paper is able to reduce the side-effect of the other training samples that are very “far” from the test sample on the recognition decision of the test sample, the high recognition rates can be obtained.

1. Introduction

Face recognition has attracted much attention in the computer-vision-related fields which has a wide range of applications including identity authentication, access control, and surveillance [1–3]. The K nearest neighbor (KNN) is a simple and popular machine learning approach for classification and has been widely used for face recognition [4–6]. The basic idea of KNN is to use the K nearest neighbors of the test sample from the training set to classify the test sample. In particular, it assigns the test sample to the class which has the more nearest neighbors for the test sample than all the other classes. The principle of nearest neighbor classification was first proposed by Skellam [7], it takes the ratio of expected and observed mean value of the nearest neighbor distances to determine whether a data set is clustered or not.

The following works can be viewed as recent improvements of KNN. In 1999, Li and Lu proposed the nearest feature line (NFL) classifier [8]. Actually, NFL is a method that exploits virtue training samples to improve KNN. In other words, NFL treats the line between two sample points from the same class as the so-called feature line and views all points on feature lines as virtue training samples. Suppose that some certain class has n training samples, there are $n(n-1)/2$ feature lines with it. Chien and Wu [9] proposed the nearest feature plane (NFP) and the nearest feature space (NFS) classifiers. With n training samples from some certain

class, there are $n(n-1)(n-2)/6$ feature planes and one feature space, respectively. In 2004, Zheng et al. [10] proposed both the nearest neighbor line (NNL) and the nearest neighbor plane (NNP) classifiers. Moreover, Orozco-Alzate and Castellanos-Domínguez [11] proposed the k nearest feature line (KNFL) and the k nearest feature plane (KNFP) classifiers based on the k nearest feature lines and the k nearest feature planes, respectively. Du and Chen [12] proposed the rectified nearest feature line segment (RNFLS) classifier. Fu et al. [13] presented the k -nearest-neighbor simplex (KNNS) classifier where k is the number of vertexes in a simplex. Moreover, Majumdar [14] proposed the nearest subspace (NS) classifier.

There exists some differences among the nearest-neighbor-based classifiers. The first difference is that they have different-dimensional coefficient vectors. Specifically, the dimensionalities of the coefficient vectors in NFL, NFP, and NFS are 2, 3, and n , respectively. n stands for the number of training samples of the given class. The second difference is that the coefficients of some approaches satisfy the constraint that their sum are equal to one whereas the other methods do not. In particular, NFL is subject to this constraint, but NFP and NFS are not. The third difference is that all the coefficients should be nonnegative with RNFLS and KNNS, whereas NFL, NFP, and NFS do not have the requirement.

Recently, the sparse representation-based classification (SRC) [15] has been proposed and can be considered as an

improvement to KNN. SRC treats a weighted sum of all the training samples from a class as the center of this class and classifies the test sample into the class whose center is the nearest to the test sample. In SRC, the weights can be learned using some algorithms. Moreover, SRC outperforms most of the state-of-the-art facial classifiers.

The common characteristic of the classifiers presented earlier is that all of them use the deviation between the test sample and the linear combination of the training samples from some certain class to classify the test sample. They assign the test sample to the class that has the smallest deviation. Different from the other improvements of KNN, SRC exploits a sparse linear combination of all the training samples to represent the test sample. ‘‘Sparse’’ means that some coefficients of the linear combination are equal or close to zeros.

In this paper, we propose the two-step test sample representation method for face recognition. This method first identifies k ‘‘representative’’ samples from every category training samples, the representative samples are more similar to the test sample. The method also exploits a linear combination of all the ‘‘representative’’ samples from all the classes to represent and classify the test sample. Here, the proposed method is employed for face recognition, and the experiments are conducted with different face databases.

The remainder of this paper is organized as follows. Section 2 overviews previous improvements of KNN. Section 3 presents the proposed method. Section 4 shows the experimental results. Finally, Section 5 summarizes the paper.

2. Improvements of KNN

2.1. Nearest Feature Line. With the nearest feature line (NFL), any two feature points of the same class are generalized by the feature line (FL) passing through the two points. So more variation of available samples can be captured, and the classification is based on the nearest distance from the query point to each FL. Let $X^{(i)} = [X_1^{(i)}, \dots, X_{N_i}^{(i)}] \in R^{d \times N_i}$ denote the matrix composed of all training samples from the i th class, $i = 1, \dots, c$ (c is the number of class). $Y \in R^{d \times 1}$ is a given test sample. For each pair of training samples $X_p^{(i)}$ and $X_q^{(i)}$ from the i th class, there exist feature lines $X_p^{(i)} X_q^{(i)}$ defined as

$$X_p^{(i)} X_q^{(i)} = \{X \mid X = \alpha X_p^{(i)} + (1 - \alpha) X_q^{(i)}, \alpha \in R\}. \quad (1)$$

The NFL classifier defines the distance between Y and feature lines $X_p^{(i)} X_q^{(i)}$ as

$$d(Y, X_p^{(i)} X_q^{(i)}) = \min_{\alpha \in R} \|Y - \alpha X_p^{(i)} - (1 - \alpha) X_q^{(i)}\|_2. \quad (2)$$

The distance between Y and the i th class is calculated using

$$d_L^i(Y) = \min_{1 \leq p < q \leq N_i} d(Y, X_p^{(i)} X_q^{(i)}). \quad (3)$$

If $l = \arg \min_{1 \leq i \leq c} d_L^i(Y)$, then the NFL classifier assigns Y to the l th class. l is one element of the set of the class labels, that is, $\{1, \dots, c\}$.

It is clear that the NFL classifier indeed evaluates the deviation between a linear combination of two training samples from the same class and the test sample, and the smallest deviations generated from all the classes can be obtained. If the smallest deviation of a class is smaller than those of all other classes, then the NFL classifier assigns the test sample into this class.

2.2. Nearest Feature Space. The NFS classifier first defines the feature space spanned by all training samples from the i th class using

$$S_i = \left\{ X \mid X = \sum_{j=1}^{N_i} \alpha_j X_j^{(i)}, \alpha_j \in R, j = 1, \dots, N_i \right\}. \quad (4)$$

This classifier takes the minimum value of the distances between test sample Y and the elements of S_i as the distance between Y and the i th class. That is, the distance between Y and the i th class is calculated using

$$d_S^i(Y) = \min_{\alpha_j \in R, j=1, \dots, N_i} \left\| Y - \sum_{j=1}^{N_i} \alpha_j X_j^{(i)} \right\|_2. \quad (5)$$

If $l = \arg \min_{1 \leq i \leq c} d_S^i(Y)$, then the NFS classifier assigns Y to the l th class.

2.3. Sparse Representation-Based Classification. Let $X = [X^{(1)}, \dots, X^{(c)}] \in R^{d \times N}$ stands for the matrix consisting of all training samples, and let N be the number of training samples. Let Y be the test sample. SRC aims to use a sparse linear combination of all training samples to represent Y . In other words, if α denotes the coefficient vector of the sparse linear combination, SRC has the following objective function:

$$\min_{\alpha} \|\alpha\|_1 \quad \text{subject to } \|Y - X\alpha\|_2 \leq \varepsilon, \quad (6)$$

where ε is a user-defined small positive constant. Let $\hat{\alpha}$ be the solution of α . The distance between the test sample Y and the i th class is calculated using

$$d_i(Y) = \|Y - X\delta_i(\hat{\alpha})\|_2, \quad (7)$$

where $\delta_i(\hat{\alpha}) \in R^{N \times 1}$ is a vector whose nonzero entries are the entries in $\hat{\alpha}$ that are associated with the i th class, $i = 1, \dots, c$. If $l = \arg \min_{1 \leq i \leq c} d_i(Y)$, SRC assigns test sample Y to the l th class.

3. The Proposed Method

3.1. Description of the Proposed Method. The proposed method is composed of two steps. The first step determines k ‘‘representative’’ training samples from each category for the test sample. The c classes will have $k * c$ ‘‘representative’’ samples in total. The second step represents the test sample via a linear combination of the $k * c$ representative samples. Let P_1, \dots, P_{kc} be all the $k * c$ representative samples of the c classes, the test sample can be represented with the linear

combination of the $k * c$ representative samples. Finally, the second step calculates the deviation between the test sample and the linear combination of each class and assigns the test sample to the class that has the minimum deviation.

In the first step, the test sample is represented with a linear combination of all the training samples of the i th class ($i = 1, \dots, c$). The coefficients of the obtained linear combination will be used to compute the deviation between the test sample and the i th class ($i = 1, \dots, c$). It then computes the deviation between the test sample and the linear combination of arbitrary k ($k < n_i$) training samples of the i th class ($i = 1, \dots, c$), that is, the deviation between the test sample and the i th class.

N_i training samples from the i th class is formulated as a matrix $X^{(i)} = [X_1^{(i)}, \dots, X_{N_i}^{(i)}]$, the test sample Y can be approximated as

$$Y = X^{(i)} A^{(i)}. \quad (8)$$

$A^{(i)}$ is the combination coefficients of the i th class samples which represent the test sample, can be solved using $\hat{A}^{(i)} = (X^{(i)T} X^{(i)} + \lambda I)^{-1} X^{(i)T} Y$. λ and I are a small positive constant and the identity matrix, respectively. Let z_1, \dots, z_k denote arbitrarily k samples of $X_1^{(i)}, \dots, X_{N_i}^{(i)}$, the deviation between the test sample and z_1, \dots, z_k can be calculated using

$$d_i(z_1, \dots, z_k) = \left\| y - \sum_{g=1}^k z_g \hat{b}_g^{(i)} \right\|, \quad (9)$$

$\hat{b}_1^{(i)} \dots \hat{b}_k^{(i)}$ are k coefficients of $\{\hat{a}_1^{(i)} \dots \hat{a}_{N_i}^{(i)}\}$ related to z_1, \dots, z_k , respectively. $d_i(z_1, \dots, z_k)$ is the so-called deviation between the test sample and some i th class representation. If k training samples are selected from all N_i training samples of the i th class, there will be $(N_i(N_i-1) \dots (N_i-k+1))/k!$ possible cases. Also, there will be $(N_i(N_i-1) \dots (N_i-k+1))/k!$ different d_i . For a class, the k training samples that produce the minimum $d_i(z_1, \dots, z_k)$ are selected as the "representative" samples of the i th class for the test sample. In the experimental section, we also perform a classification based on the first step of the proposed method as follows: the minimum deviation of every class from the test sample can be computed, and the test sample is assigned to the class with which the minimum deviation is the smallest among all the classes.

After the $k * c$ representative samples are determined, the test sample is approximated with them as

$$Y = \sum_{j=1}^{k*c} \eta_j P_j, \quad (10)$$

where $P = [P_1, \dots, P_{k*c}]$ and $\eta = [\eta_1, \dots, \eta_{k*c}]^T$. The solution of (10) is obtained with

$$\hat{\eta} = [\hat{\eta}_1, \dots, \hat{\eta}_{k*c}]^T = (P^T P + \gamma I)^{-1} P^T Y, \quad (11)$$

I is the identity matrix, and γ stands for a small positive constant. The final deviation between the test sample and the i th class is calculated using

$$d_i = \|Y - G_i \hat{\eta}_i\|, \quad (12)$$



FIGURE 1: One test sample (the first image) and its 34 neighbors from the set of training samples.



FIGURE 2: The three training samples from the first three classes closest to the test sample obtained using the proposed method.



FIGURE 3: The three training samples from the first three classes closest to the test sample obtained using the first step of our method (the test sample misclassified only with them).

where $\hat{\eta}_i = [\hat{\eta}_{(i-1)k+1}, \dots, \hat{\eta}_{(i-1)k+k}]^T$, $G_i = [P_{(i-1)k+1}, \dots, P_{(i-1)k+k}]$ is composed of the k representative samples of the i th class. If $h = \arg \min_i d_i$, then the test sample is assigned to the h th class.

3.2. Insight into the Proposed Method

3.2.1. Why the Proposed Method Is Reasonable. This section will show why the proposed method is reasonable. The first step of the proposed method is indeed a step to identify the k "representative" samples that are the most similar to the test sample from each class. Then, only the training samples that are similar to the test sample are used to represent and classify the test sample. It is known that the face images of the same subject have much variation owing to the varying pose, facial expression, and illumination [16, 17], and the training samples dissimilar to the test sample might be or not be from the genuine subject of the test sample. As a result, it is not feasible to use the "dissimilar" training samples to distinguish which class the test sample is truly from. On the contrary, it is very helpful to use only the training samples that are similar to the test sample for classification.

The second step of the proposed method also enables all the classes to represent the test sample in a competitive way. Actually, if one class contributes much, that is, has a small deviation in representing the test sample, the other classes will contribute less. The experiments show that the genuine class of the test sample usually has the smallest deviation with the test sample; thus, the proposed method allows the test sample to be classified with a high accuracy.

The first step is somewhat able to remove the "outlier" of a class, that is, the training samples that are very close to other classes. Thus, it is useful for obtaining a high accuracy. Figures 1, 2, and 3 show a test sample from the AR face database and the classification result of our method on this test sample (the occluded image in the first column is the test sample). In these three figures, the first image always stands

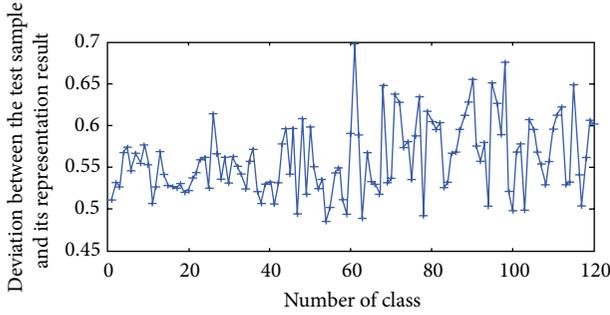


FIGURE 4: Deviation between the test sample and its representation result obtained using the first step of our method.

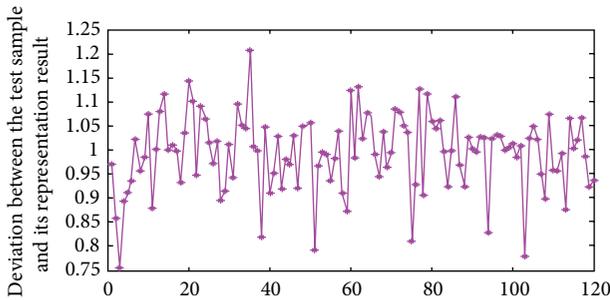


FIGURE 5: Deviation between the test sample and its representation result obtained using the second step of our method.

for the same test sample. In this example, our method takes the first six face images and the remaining face images of each subject as training and test samples, respectively, and k is set to 3. Figure 1 shows a sample and its 34 neighbors from the set of training samples (these neighbors were determined using the Euclidean distance between the training sample and test sample). Among these 34 neighbors only the last one is from the same subject as the test sample, and the second to sixth images are the first five neighbors of the test sample.

Figure 2 shows that the second step of our method correctly assigns the test sample to its genuine class, the second to fourth images are the three training samples of the first closest class to the test sample, the fifth to seventh images and eighth to tenth images are the three training samples of the second and third classes that are the closest to the test sample, respectively. On the other hand, Figure 3 shows that if the results from the first step of our method are directly employed to perform classification, the test sample is misclassified. The three training samples form three closest class to the test sample are also presented, the genuine class of the test sample is only the seventeenth class among all the classes that are “close” to the test sample. In other words, if the test sample is assigned to the class with the smallest $d_i(z_1, \dots, z_k)$ (defined in (9)), it will be misclassified.

Figures 4 and 5, respectively, show the difference between the same test sample as shown in Figures 1, 2, and 3 and its representation results obtained using the first and second steps of our method. In these two figures, the horizontal

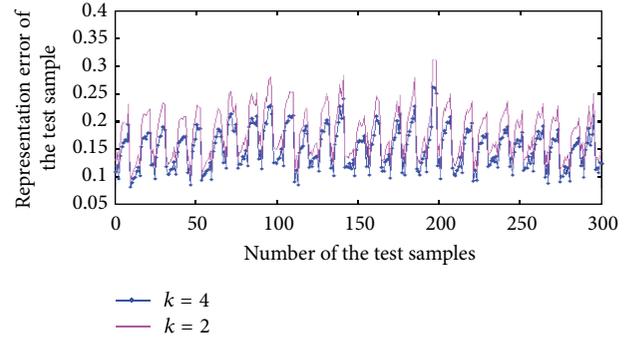


FIGURE 6: Representation errors of the first 300 test samples in the AR database. $k = 4$ means that all the available training samples were used for training. $k = 2$ means that the second step of our method uses 2 representative samples from each subject.

and vertical axes show the number of class (class label) and difference, respectively. The genuine class label of the test sample is 3. Figure 5 shows that in terms of the representation result obtained using the second step, the test sample has the smallest deviation from its representation result generated from the third class. As a result, the second step will correctly assign the test sample to the third class. However, Figure 4 shows that the test sample is assigned to the fifty-fourth class only with the deviations from the first step of our method, and the class that has the minimum deviation is not the genuine class (the genuine class is the third class).

Differing from conventional l_1 -norm-based sparse representation method, our method is an l_2 -norm-based. The l_2 -norm based representation method proposed by Xu et al. not only is computationally much more efficient than conventional sparse representation methods but also can lead to a higher face recognition accuracy [18].

3.2.2. Explore the Relationship between the Accuracy and the Representation Error. We will show whether the accuracy is directly related to the representation error or not. As shown earlier, if the representation result generated from a class has the minimum deviation from the test sample, the test sample is assigned to this class. Let $\hat{Y} = P\hat{\eta}$, $e = \|Y - \hat{Y}\|$, e is the so-called representation error. If the number of the available training samples is equal to or greater than the dimensionality of the sample vector, the representation error of the test sample can be zero. Under the other conditions, the more the available training samples the lower the representation error. In the real-world face recognition applications, the condition that the number of the available training samples is equal to or greater than the dimensionality of the sample vector is usually not satisfied. Nevertheless, the experiment shows that a lower representation error usually does not lead to a higher recognition accuracy. Moreover, our method usually has a high representation error, whereas it obtains a high recognition accuracy. Figure 6 shows the representation errors of the first 300 test samples in the case where the first four face images of each subject in the AR

TABLE 1: Experimental results of our method on the Lab2 database.

	Our method			
Number of representative samples	6	7	8	9
Number of erroneously classified test samples	216	194	195	190
	The first step of our method			
Number of representative samples	6	7	8	9
Number of erroneously classified test samples	419	409	402	362

TABLE 2: Experimental results of NFL, NFP, and NFS on the Lab2 database.

Method	NFL	NFS	NFP
Number of erroneously classified test samples	319	278	289



FIGURE 7: Some visible face images of one subject in the Lab2 database.

database are used as training samples and the remaining face images are used as test samples. It is clear that $k = 4$ obtains a lower representation error than $k = 2$; however, $k = 2$ obtains a higher classification accuracy. Thus, the representation error is not a good index of the accuracy. However, the accuracy is directly related to the deviation between the test sample and the representation result of the genuine class of the test sample. Moreover, it seems that the smaller the deviation, the better the classification performance.

4. Experimental Results

4.1. Experiment on the Lab2 Data. The first experiment is conducted on the Lab2 database [19]. This database contains 50 subjects and 1000 visible face images. Each subject is provided with twenty face images. These images were acquired under the following four different illumination conditions: (a) the environmental illumination (referred to as “normal illumination”) condition, (b) the environmental illumination puls the illumination of the left incandescent lamp (referred to as “left illumination”), (c) the environmental illumination puls the illumination of the right incandescent lamp (referred to as “right illumination”), and (d) the environment illumination puls the illumination of the left and the right incandescent lamps (referred to as “both illumination”). The face images also have a variation in facial expression and pose. Figure 7 shows several visible face images of one subject in the Lab2 database. We used the ten visible face images captured with “both illumination” and “left illumination” of every subject as training samples and took the remaining images as test samples. Table 1 shows the experimental results

of our method on the Lab2 database. Table 2 shows the experimental results of NFL, NFP, and NFS on the Lab2 database. It is shown that our method obtained much fewer classification errors than NFL, NFP, and NFS. Table 1 also illustrates that if we depend on only the first step of our method to perform classification, a higher error rate will be obtained.

4.2. Experiments on the AR Face Database. The second experiment is conducted on the AR database [19]. The face images of this database were obtained under the condition of varying pose, facial expression, or lighting. Occluded face images are also included in the AR face database. There are 120 subjects and 3120 gray face images captured in two sessions. We resized each image to a 40 by 50 image. A series of experiments are carried out, where the first certain number of images (set to 4, 6, 8, 12, and 14, resp.) for each subject constitute the training samples, all the other images make the test samples. The number of representative samples are selected in the view of Tables 3 and 4, and the experimental results are also shown there. The table shows the comparison results with another three KNN algorithms (NFL, NFS, and NFP) on the AR database. “ER” is the abbreviation of “error rate”.

From Tables 3, 4, and 5, it is known that our method leads to the lowest error rate than other methods. These tables also show that the first step of our method obtains a much higher error rate than the second step. Thus, the second step plays an important role in representing and classifying the test sample. This is mainly owing to that the second step evaluates the contribution of every class to representing the test sample, in a competitive way.

5. Conclusions

The two steps of the proposed method play different and positive roles in performing classification. The first step indeed identifies the training samples which are the most similar to the test sample. This is beneficial to reduce the side-effect on classification of the other dissimilar training samples. The second step allows the “representative” samples of every class to represent the test sample in a competitive way by using a linear combination of them. This is useful for the method to better evaluate the deviation between the test sample and each class than NFL, NFP, and NS. The proposed method is easy to implement and achieve better performance than the conventional nearest feature classifiers. But the selection of all possible subsets of training samples

TABLE 3: The experimental results on the AR database.

Number of training samples	4				6				8			
Number of representative samples	1	2	3	4	2	3	4	2	3	4	8	
ER of the first step of our method	42.69%	41.89%	41.93%	41.86%	37.12%	37.37%	37.00%	40.09%	40.19%	40.28%	40.37%	
ER of our method	32.39%	32.46%	32.63%	32.46%	28.54%	28.54%	28.33%	29.17%	28.47%	28.80%	30.05%	

TABLE 4: Experimental results of our method on the AR database (complementary of Table 3).

Number of training samples	12			14		
Number of representative samples	2	3	4	2	3	3
ER of the first step of our method	33.04%	32.20%	32.20%	16.32%		13.75%
ER of our method	25.12%	24.58%	25.18%	9.86%		9.24%

TABLE 5: ER of NFL, NFS, and NFP on the AR database.

Number of training samples	4	6	8	12	14
NFL	42.12%	37.29%	40.46%	32.92%	15.97%
NFS	41.86%	37.46%	40.46%	32.02%	12.99%
NFP	41.93%	41.86%	40.09%	37.98%	17.02%

with each class needs more computation that will bring some negative effects.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant no. 61065003) and Social Sciences Foundation of the State Education Ministry (Grant no. 10YJC630379).

References

- [1] Y. Yan, L. A. Osadciw, and P. Chen, "Confidence interval of feature number selection for face recognition," *Journal of Electronic Imaging*, vol. 17, no. 1, Article ID 011002, 2008.
- [2] W. Hizem, Y. Ni, and B. Dorizzi, "Near infrared sensing and associated landmark detection for face recognition," *Journal of Electronic Imaging*, vol. 17, no. 1, Article ID 011005, 2008.
- [3] Y. Xu, D. Zhang, and J. Y. Yang, "A feature extraction method for use with bimodal biometrics," *Pattern Recognition*, vol. 43, no. 3, pp. 1106–1115, 2010.
- [4] S. Arya, T. Malamatos, and D. M. Mount, "Space-time tradeoffs for approximate nearest neighbor searching," *Journal of the ACM*, vol. 57, no. 1, article 1, 2009.
- [5] O. Virtajoki and P. Fränti, "Fast pairwise nearest neighbor based algorithm for multilevel thresholding," *Journal of Electronic Imaging*, vol. 12, no. 4, pp. 648–659, 2003.
- [6] A. Andoni, P. Indyk, R. Krauthgamer, and H. L. Nguyen, "Approximate line nearest neighbor in high dimensions," in *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '09)*, pp. 293–301, 2009.
- [7] J. G. Skellam, "Studies in statistical ecology. I. Spatial pattern," *Biometrika*, vol. 39, pp. 346–362, 1952.
- [8] S. Z. Li and J. Lu, "Face recognition using the nearest feature line method," *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp. 439–443, 1999.
- [9] J. T. Chien and C. C. Wu, "Discriminant waveletfaces and nearest feature classifiers for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1644–1649, 2002.
- [10] W. Zheng, L. Zhao, and C. Zou, "Locally nearest neighbor classifiers for pattern classification," *Pattern Recognition*, vol. 37, no. 6, pp. 1307–1309, 2004.
- [11] M. Orozco-Alzate and C. G. Castellanos-Domínguez, "Comparison of the nearest feature classifiers for face recognition," *Machine Vision and Applications*, vol. 17, no. 5, pp. 279–285, 2006.
- [12] H. Du and Y. Chen, "Rectified nearest feature line segment for pattern classification," *Pattern Recognition*, vol. 40, no. 5, pp. 1486–1497, 2007.
- [13] Y. Fu, S. Yan, and T. S. Huang, "Classification and feature extraction by simplexization," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 91–100, 2008.
- [14] A. Majumdar, *Compressive classification for face recognition [M.S. thesis]*, Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada, 2009.
- [15] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.
- [16] Y. Xu, A. Zhong, J. Yang, and D. Zhang, "LPP solution schemes for use with face recognition," *Pattern Recognition*, vol. 43, no. 12, pp. 4165–4176, 2010.
- [17] Z. Xie and G. Liu, "Infrared face recognition based on intensity of local micropattern-weighted local binary pattern," *Optical Engineering*, vol. 50, no. 7, Article ID 077201, 2011.
- [18] Y. Xu, A. Zhong, J. Yang, and D. Zhang, "Bimodal biometrics based on a representation and recognition approach," *Optical Engineering*, vol. 50, no. 3, 2011.
- [19] <http://cobweb.ecn.purdue.edu/~aleix/aleix-face-DB.html>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

