Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A fast classification scheme and its application to face recognition

Xiao-hu MA[†], Yan-qi TAN, Gang-min ZHENG

(*School of Computer Science and Technology, Soochow University, Suzhou 215006, China*)

[†]E-mail: xhma@suda.edu.cn

Received Dec. 29, 2012; Revision accepted May 10, 2013; Crosschecked June 6, 2013

Abstract: To overcome the high computational complexity in real-time classifier design, we propose a fast classification scheme. A new measure called 'reconstruction proportion' is exploited to reflect the discriminant information. A novel space called the 'reconstruction space' is constructed according to the reconstruction proportions. A point in the reconstruction space denotes the case of a sample reconstructed using training samples. This is used to search for an optimal mapping from the conventional sample space to the reconstruction space. When the projection from the sample space to the reconstruction space is obtained, a new sample after mapping to the new discriminant space would be classified quickly according to the reconstruction proportions in the reconstruction space. This projection technique results in a diversion of time-consuming calculations from the classification stage to the training stage. Though training time is prolonged, it is advantageous in that classification problems such as identification can be solved in real time. Experimental results on the ORL, Yale, YaleB, and CMU PIE face databases showed that the proposed fast classification scheme greatly outperforms conventional classifiers in classification accuracy and efficiency.

Key words: Reconstruction proportion, Reconstruction space, Real-time classification, Face recognition
doi:10.1631/jzus.CIDE1309 **Document code:** A **CLC number:** TP39

1 Introduction

Classifier design is an essential component in pattern recognition. An effective classification such as the nearest neighbor (NN) classifier may be used in a variety of fields, such as cluster analysis (Zhang and Srihari, 2004), object recognition (Stefan *et al.*, 2009), pattern matching (Athitsos *et al.*, 2008), data mining (Zhang *et al.*, 2010), and machine learning (Chen, 2010). As a classical problem in a cross field, face recognition has been relying heavily on the performance of classifiers (Zhao *et al.*, 2003; Abate *et al.*, 2007; Jiang *et al.*, 2008; Zhang and Gao, 2009; Singh *et al.*, 2012). It is well observed that efficiency and accuracy are fundamental but contradicting issues in classification. Theoretically an optimal classifier with high precision is not necessarily the best practical choice if it is of high complexity. NN and *K*-nearest neighbors (KNN) classifiers are two most popular classifiers in pattern recognition (Psaltis *et al.*, 1994; Bax, 2000; Mullin and Sukthankar, 2000; Duda *et al.*, 2001). NN classifies the new sample based on the minimum distance to a single training sample, whereas KNN classifies the new sample based on the *K* minimum distances to the training samples. Though simple and efficient, these minimum distance based classifiers are of poor accuracy (Zhang *et al.*, 2006; Meo *et al.*, 2012), yielding poor generalization capability with a limited number of samples (Vincent and Bengio, 2002) and are less effective due to the 'curse of dimensionality' (Jain and Chandrasekaran, 1982). To improve the generalization capability, numerous techniques have been proposed, such as the KNN classifier based on adaptive nonparametric separability (Keller *et al.*, 1985), *K*-local hyperplane distance nearest neighbor (HKNN) (Vincent and Bengio, 2002), and nearest bounding hyperdisk (NHD) (Cevikalp *et al.*, 2008). However, they are similar to the classical KNN classifier which possesses only the classification phase but no training phase. The computational complexity of these classifiers can be prohibitive for classifications on large datasets.

Subspace classification methods classify a query sample into the class whose subspace is the closest. In a subspace based classifier, each class is considered as a variable smooth manifold. Compared with NN, KNN, and their improvements aforementioned, a subspace classifier has better performance but with more computational complexity. The nearest subspace (NS) classifier which is based on the minimum distance to the subspace is a representative subspace classification (Ho *et al.*, 2003). The nearest feature line (NFL) classifies samples based on the best affine representation in terms of a pair of training samples (Li and Lu, 1999). NS and NFL consider local discriminant information, but are of low robustness. Inspired from the recently developed compressive sensing (CS) theory (Donoho, 2006; Baraniuk, 2007; Candes and Wakin, 2008), Wright *et al.* (2009) proposed a sparse representation based classification (SRC) algorithm for robust face recognition. SRC represents a test sample in an overcomplete dictionary whose base elements are also the training samples. It is convincing that the theory of sparse representation and compressed sensing can solve the two fundamental problems in automatic face recognition, namely feature extraction and poor accuracy due to occlusion. SRC can enhance the robustness and improve the recognition performance; however, the very high performance complexity of the SRC algorithm significantly limits its use in practical applications.

It is obvious that subspace classifiers can be regarded as a method based on reconstruction, because samples are predicted based on the reconstruction error. They differ only in the choice of the reconstruction basis. For example, for NS and NFL, samples of each class are treated as a reconstruction dictionary, while for SRC all training samples are treated as a reconstruction dictionary. In fact, even NN may be generalized as a reconstruction based classifier. Each training sample is selected only as the basis for the calculation of the reconstruction error. Inspired by the above observation, a new measure called 'reconstruction proportion', reflecting the similarity of samples, is applied to improve these classifiers.

To deal with high dimensional pattern recognition such as face recognition, feature extraction or feature selection is a viable way. Nonlinear subspace methods, such as isometric mapping (ISOMAP) (Tenenbaum *et al.*, 2000), locally linear embedding (Roweis and Saul, 2000), and Laplacian eigenmaps (Belkin and Niyogi, 2003) are good accomplishments, where facial images are empirically viewed as a highly nonlinear manifold in the observation space. However, these methods suffer from the out-of-sample problem; that is, when a new sample is augmented, it needs to be re-calculated. Increased computational complexity limits their applications in real-time recognition. It seems that most of the complex classifiers such as SRC suffer from a similar complexity issue. To achieve effective and efficient face recognition, many linear subspace methods have been proposed, including subspace algorithms such as principal component analysis (PCA) (Turk and Pentland, 1991), linear discriminant analysis (LDA) (Belhumeur *et al.*, 1997), locality preserving projections (LPP) (He and Niyogi, 2003), neighborhood preserving embedding (NPE) (He *et al.*, 2005), maximum margin criterion (MMC) (Li *et al.*, 2006), and sparsity preserving projections (SPP) (Qiao *et al.*, 2010). The technique of mapping samples to a subspace by linear projection can significantly reduce the calculation in the recognition process. Motivated by this, we explore a linear projection from a sample class to a new manifold space, called the 'reconstruction space', to reduce the time complexity. The projection may be obtained by a training phase so that we need only to perform limited calculations in the reconstruction space mapped from a new sample, instead of comprehensive calculations in the sample space. As a result, the high performance complexity in the recognition phase is transferred to the preprocessing phase of the training stage.

2 Reconstruction proportion

As mentioned above, each class is considered as a variable smooth manifold space. In the local smooth manifold, a sample can be well reconstructed by samples from the same class. It is reasonable to assume that any point in the manifold is a placeholder of the class. Therefore, the distance from a query sample to the manifold can be defined as the classification metric (Liu *et al.*, 2011). To quantify the classification metric, we define the following measure of the distance from a sample to the class.

Definition 1 (Distance from sample to class, SCD) Consider a local smooth manifold $\boldsymbol{X}_i=[\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,n_i}]$ whose points belong to the same class *i*. The distance from a sample $\boldsymbol{y}$ to this class is defined as

$$\mathrm{SCD}(\boldsymbol{y}, \boldsymbol{X}_i)=\min\left\|\boldsymbol{y}-\psi_i\boldsymbol{X}_i\right\|_2, \tag{1}$$

where ψ_i is the reconstruction coefficient. The smaller the SCD, the higher the probability that sample $\boldsymbol{y}$ belongs to class *i*. To illustrate the relation, a simple example is shown in Fig. 1, where $\boldsymbol{x}_{i,1}$ and $\boldsymbol{x}_{i,2}$ are samples from the same class *i*, $\boldsymbol{y}$ is a new sample, parallelogram *oabc* is the local manifold spanned by vectors $\boldsymbol{x}_{i,1}$ and $\boldsymbol{x}_{i,2}$, and the norm of *bd* is the SCD with $\boldsymbol{y}$ to class *i*.

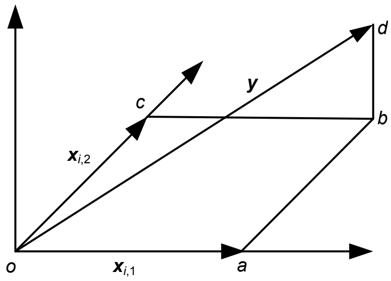


Fig. 1 Distance from sample to class (SCD)

Then we introduce the reconstruction proportion and illustrate it in the face recognition process.

In appearance-based techniques, a 2D face image of size *p*×*q* is represented by a point in an *M*-dimensional space (*M*=*pq*), which is called the 'sample space' or 'image space'. Given sufficient training samples, we assume that a new (test) sample $\boldsymbol{y}\in\mathbb{R}^M$ from the *i*th object class approximately lies in the linear span of the training sample of the same class associated with object *i*:

$$\boldsymbol{y}=a_{i,1}\boldsymbol{x}_{i,1}+a_{i,2}\boldsymbol{x}_{i,2}+\cdots+a_{i,n_i}\boldsymbol{x}_{i,n_i}, \tag{2}$$

where $a_{i,j}$ (*j*=1, 2, …, n_i) represents the reconstruction contribution of $\boldsymbol{y}$ with training samples $\boldsymbol{x}_{i,j}$. In fact, the contribution contains natural discrimination information. Based on this observation, we apply the contribution to evaluate the similarity of two samples.

Given *N* training samples, we define the reconstruction proportion as follows:

$$\begin{aligned}&\min_{\boldsymbol{s}_i}\left\|\boldsymbol{x}_i-\boldsymbol{A}\boldsymbol{s}_i\right\|_2\\&\text{s.t. } \mathbf{1}^{\mathrm{T}}\boldsymbol{s}_i=1,\end{aligned} \tag{3}$$

where $\boldsymbol{A}$ is the dictionary matrix, $\boldsymbol{s}_i=[a_{i,1}, a_{i,2}, \ldots, a_{i,N}]^{\mathrm{T}}\in\mathbb{R}^N$ is a point in the *N*-dimensional reconstruction space with $a_{i,j}$ being the reconstruction proportion of sample $\boldsymbol{x}_i$ with sample $\boldsymbol{x}_j$. $\mathbf{1}$ is a vector of all ones. It is straightforward to prove that $\boldsymbol{s}_i$ is invariant to rotations, rescaling, and translations due to the sum-to-one constraint (He *et al.*, 2005; Li *et al.*, 2006).

Obviously, the reconstruction proportion $a_{i,j}$ defined to reflect the similarity will change along with the changes in dictionary matrix $\boldsymbol{A}$ (the different construction methods of the dictionary matrix will be introduced in Section 3). A high reconstruction proportion value denotes a high probability of two samples belonging to the same class. Then we seek a reasonable linear projection $\boldsymbol{W}\in\mathbb{R}^{N\times N}$ from the initial sample space to the reconstruction space through the following minimization problem:

$$\min_{\boldsymbol{W}}\sum_{i=1}^{N}\left\|\boldsymbol{W}\boldsymbol{x}_i-\boldsymbol{s}_i\right\|_2. \tag{4}$$

It is assumed that a (test) sample can be represented as a linear combination of samples from the same object class in the sample space by Eq. (2); that is, samples from the same class lie on a local linear subspace (the so-called 'face subspace'). In fact, applying $\boldsymbol{W}$ to both sides of Eq. (2) yields

$$\begin{aligned}&\boldsymbol{W}\boldsymbol{y}=a_{i,1}\boldsymbol{W}\boldsymbol{x}_{i,1}+a_{i,2}\boldsymbol{W}\boldsymbol{x}_{i,2}+\cdots+a_{i,n_i}\boldsymbol{W}\boldsymbol{x}_{i,n_i}\\&\Rightarrow\tilde{\boldsymbol{y}}=a_{i,1}\tilde{\boldsymbol{s}}_{i,1}+a_{i,2}\tilde{\boldsymbol{s}}_{i,2}+\cdots+a_{i,n_i}\tilde{\boldsymbol{s}}_{i,n_i}.\end{aligned} \tag{5}$$

Consequently, this relationship will be preserved in the reconstruction space due to the linearity of projection $\boldsymbol{W}$. As in the sample space, the reconstruction proportions can comprise the regional clusters in the reconstruction space. We can classify the samples efficiently based on the distributions after mapping to the reconstruction space due to the coherency.

However, it is insufficient to express the sample space in terms of *N* points. Thus, mapping an *M*-dimensional point in the sample space to a new

N-dimensional point in the reconstruction space using dictionary matrix $\boldsymbol{A}\in\mathbb{R}^{M\times N}$ will result in much information loss if N is much smaller than M. Similar to sparse representation, we can make the dictionary consisting of points be over completed before mapping the points to the reconstruction space. An alternative method is to find the Q-dimensional ($Q\leq N$) feature subspace of the original sample space with feature extraction or dimension reduction algorithms such as PCA and LDA.

3 Improved classification scheme

3.1 Improved NN and KNN classifiers

The aim of a classifier in object recognition is to use labeled training samples from c distinct object classes to correctly determine the class to which a new test sample belongs. The NN classifier tries to find the minimum distance from the test sample to a single training sample:

$$j=\min_{i} d(\boldsymbol{y},\boldsymbol{x}_i),\quad i=1, 2, ..., N, \tag{6}$$

where $d(\boldsymbol{y}, \boldsymbol{x}_i)$ is the distance from $\boldsymbol{y}$ to $\boldsymbol{x}_i$. Then we can assign $\boldsymbol{y}$ to the object class to which $\boldsymbol{x}_j$ belongs, whereas the KNN classifier finds the K nearest neighbors and assigns $\boldsymbol{y}$ to the object class to which the majority of K samples found belong. Though NN and KNN classifiers are simple, the search for the most similar samples can be very time-consuming, especially with mass data. Inspired by the principle of NN and KNN classification, we project a training sample into the reconstruction space in which this training sample has the highest similarity to itself. Thus, the dictionary matrix in Eq. (3) is defined as $\boldsymbol{A}$=[$\boldsymbol{0}$, …, $\boldsymbol{0}$, $\boldsymbol{x}_i$, $\boldsymbol{0}$, …, $\boldsymbol{0}$]. Obviously, $\boldsymbol{s}_i$ is an N-dimensional unit vector with the ith element being 1 and all other elements 0. So, improved NN and KNN classifiers search for the projection $\boldsymbol{W}$ by solving the least squares problem defined in Eq. (4). According to the coherency, a test sample after mapping to the reconstruction space will be well reconstructed by its nearest neighbor. The distance from a query sample to the subspace is the distance from the origin to the point in the reconstruction space after mapping to each coordinate axis, because each training sample is considered to be independent of the NN and KNN schemes. That is, the ith reconstruction proportion value is the maximum if the test sample belongs to the same object class as $\boldsymbol{x}_i$. In Fig. 2, test samples are mapped to the reconstruction space. We select N=2 for illustration.

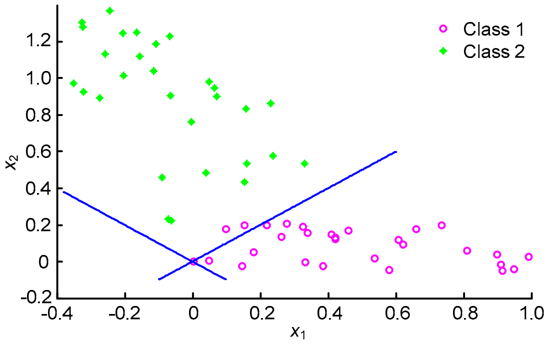


Fig. 2 Data visualization in a 2D reconstruction space

Based on the above discussion, we summarize the procedure of the proposed improved NN and KNN classifiers as Algorithm 1.

Algorithm 1 Improved NN classifier and KNN classifier (INN and IKNN)

Input: a matrix of training samples $\boldsymbol{X}$=[$\boldsymbol{x}_1$, $\boldsymbol{x}_2$, …, $\boldsymbol{x}_N$]$\in\mathbb{R}^{M\times N}$ for c classes, and a test sample $\boldsymbol{y}\in\mathbb{R}^{M}$.

1. If N<<M, map M-dimensional samples of $\boldsymbol{X}$ and $\boldsymbol{y}$ to Q-dimensional samples of $\tilde{\boldsymbol{X}}\in\mathbb{R}^{Q\times N}$ and $\tilde{\boldsymbol{y}}\in\mathbb{R}^{Q}$ ($Q\leq N$), respectively.

2. Let $\boldsymbol{E}$ be the N-dimensional identity matrix. Then solve the following optimization Frobenius norm problem:

$$\hat{\boldsymbol{W}}=\min_{\boldsymbol{W}}\left\|\boldsymbol{W}\tilde{\boldsymbol{X}}-\boldsymbol{E}\right\|_{\mathrm{F}}. \tag{7}$$

3. Compute the mapped data $\hat{\boldsymbol{y}}=\hat{\boldsymbol{W}}\tilde{\boldsymbol{y}}=[a_1,a_2,\ldots,a_N]$, and the improved NN classifier finds the maximum element of $\hat{\boldsymbol{y}}$:

$$j=\max_{i}|a_i|,\quad i=1, 2, ..., N, \tag{8}$$

while the improved KNN classifier finds the K maximum elements of $\hat{\boldsymbol{y}}$ (assume $|a_{j_1}|<|a_{j_2}|<\ldots<|a_{j_K}|$).

Output: For the NN classifier, the predicted class of $\boldsymbol{y}$ is set equal to the class to which sample $\boldsymbol{x}_j$ belongs, while for the KNN classifier, the predicted class of $\boldsymbol{y}$ is set equal to the most frequent true class among K training samples $\boldsymbol{x}_{j_1},\boldsymbol{x}_{j_2},\cdots,\boldsymbol{x}_{j_K}$.

3.2 Improved nearest subspace classifier

In contrast to the NN classifier, the NS classifier aims to find the minimum distance to the subspace spanning all training samples from each object class. Let $\boldsymbol{X}=[\boldsymbol{X}_1,\boldsymbol{X}_2,\ldots,\boldsymbol{X}_c]=[\boldsymbol{x}_{1,1},\ldots,\boldsymbol{x}_{1,n_c},\boldsymbol{x}_{2,1},\ldots,\boldsymbol{x}_{2,n_c},\ldots,\boldsymbol{x}_{c,1},\ldots,\boldsymbol{x}_{c,n_c}]$ be the matrix for the entire training set as the concatenation of the N training samples of all c object classes. NS classifies the new sample based on the best linear representation in terms of all the training samples in each class:

$$j=\min_i \left\|\boldsymbol{y}-\boldsymbol{X}_i\boldsymbol{v}_i\right\|. \tag{9}$$

We then assign $\boldsymbol{y}$ to the jth class. Though in the NS classifier there is no need to calculate the distance from the test sample to each training sample, searching for the linear combination can be very time-consuming. To avoid this problem, we explore to more efficiently obtain the coefficients of linear representation. For each training sample $\boldsymbol{x}_{i,j}$ from the ith class, we define dictionary matrix $\boldsymbol{A}$=[$\boldsymbol{0}$, …, $\boldsymbol{0}$, $\boldsymbol{x}_{i,1}$, …, $\boldsymbol{x}_{i,j-1}$, $\boldsymbol{0}$, $\boldsymbol{x}_{i,j+1}$, …, $\boldsymbol{x}_{i,n_i}$, $\boldsymbol{0}$, …, $\boldsymbol{0}$]. $\hat{\boldsymbol{s}}_i$ calculated using Eq. (3) is a vector whose only nonzero entries are the entries associated with the ith class. The improved NS classifier finds the projection $\boldsymbol{W}$ by solving the optimization problem defined in Eq. (4). Then, a test sample from the ith class after mapping to the reconstruction space can be well reconstructed by samples from the same class with a test sample. In other words, using only the reconstruction proportion associated with the ith class, the given test sample can be well approximated. Fig. 3 illustrates the improved NS classifier. We summarize the procedure of the proposed improved NS classifier as Algorithm 2.

Algorithm 2 Improved NS classifier (INS)

Input: a matrix of training samples $\boldsymbol{X}$=[$\boldsymbol{X}_1$, $\boldsymbol{X}_2$, …, $\boldsymbol{X}_c$]$\in\mathbb{R}^{M\times N}$ for c classes, and a test sample $\boldsymbol{y}\in\mathbb{R}^M$.

1. If N<<M, map M-dimensional samples of $\boldsymbol{X}$ and $\boldsymbol{y}$ to Q-dimensional samples of $\tilde{\boldsymbol{X}}$=[$\tilde{\boldsymbol{X}}_1,\tilde{\boldsymbol{X}}_2,\ldots,\tilde{\boldsymbol{X}}_c$]$\in\mathbb{R}^{Q\times N}$ and $\tilde{\boldsymbol{y}}\in\mathbb{R}^Q$ ($Q\leq N$), respectively.

2. For each training sample $\tilde{\boldsymbol{x}}_i^j$ in the feature subspace which denotes sample $\tilde{\boldsymbol{x}}_i$ from the jth class (i=1, 2, …, N), compute the reconstruction proportion:

$$\begin{aligned}\hat{\boldsymbol{s}}_i &= \min_{\boldsymbol{s}_i}\left\|\tilde{\boldsymbol{x}}_i^j-\boldsymbol{A}\boldsymbol{s}_i\right\|_2 \\ \text{s.t.}\quad & \boldsymbol{1}^{\mathrm{T}}\boldsymbol{s}_i=1,\end{aligned} \tag{10}$$

where $\boldsymbol{A}=[\boldsymbol{0},\ldots,\boldsymbol{0},\tilde{\boldsymbol{X}}'_j,\boldsymbol{0},\ldots,\boldsymbol{0}]$, and $\tilde{\boldsymbol{X}}'_j$ denotes training samples in which $\tilde{\boldsymbol{x}}_i^j$ is removed from $\tilde{\boldsymbol{X}}_j$.

3. Let $\boldsymbol{S}=[\hat{\boldsymbol{s}}_1,\hat{\boldsymbol{s}}_2,\ldots,\hat{\boldsymbol{s}}_N]$. Then solve the following optimization Frobenius norm problem:

$$\hat{\boldsymbol{W}}=\min_{\boldsymbol{W}}\left\|\boldsymbol{W}\tilde{\boldsymbol{X}}-\boldsymbol{S}\right\|_{\mathrm{F}}. \tag{11}$$

4. Compute the mapped data: $\boldsymbol{r}_{\mathrm{p}}=\hat{\boldsymbol{W}}\tilde{\boldsymbol{y}}=[a_1,a_2,\ldots,a_N]$. The improved NS classifier finds the minimum residuals:

$$l=\min_i\left\|\tilde{\boldsymbol{y}}-\tilde{\boldsymbol{X}}\boldsymbol{\delta}_i(\boldsymbol{r}_{\mathrm{p}})\right\|,\quad i=1,2,\ldots,c, \tag{12}$$

where $\boldsymbol{\delta}_i(\cdot)$ is the characteristic function used to select the coefficients associated with the ith class; that is, $\boldsymbol{\delta}_i(\boldsymbol{r}_{\mathrm{p}})$ is a new vector whose only nonzero elements are elements in $\boldsymbol{r}_{\mathrm{p}}$ that are associated with class i.

Output: the predicted class of $\boldsymbol{y}$ is set equal to l.

3.3 Improved sparse representation based classification

To obtain a more discriminative classifier, SRC provides new insight into two fundamental issues in classification, feature extraction and robustness to occlusion. Based on global sparse representation, SRC aims to find the sparse solution by solving the following stable l_1-minimization problem:

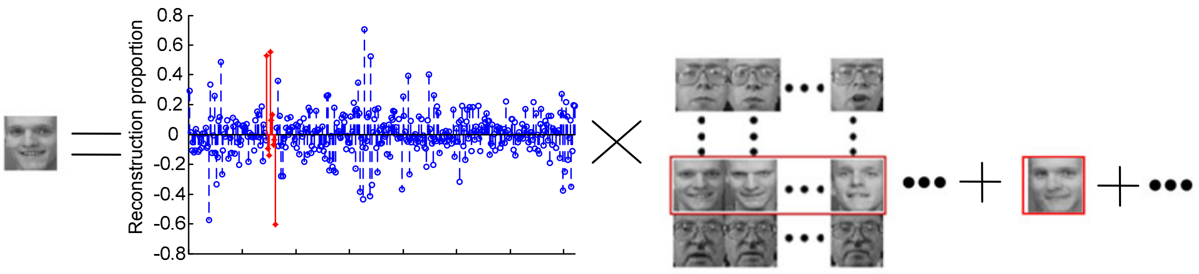


Fig. 3 Representation of a test image with reconstruction proportion by the improved nearest subspace (INS) classifier

$$\begin{aligned}&\hat{\boldsymbol{s}} = \min\|\boldsymbol{s}\|_1\\ &\text{s.t.}\ \ \|\boldsymbol{X}\boldsymbol{s} - \boldsymbol{y}\|_2 \le \varepsilon.\end{aligned} \tag{13}$$

SRC then classifies $\boldsymbol{y}$ based on these approximations by assigning it to the object class that minimizes the residual:

$$\min_i r_i(\boldsymbol{y}) = \left\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\delta}_i(\hat{\boldsymbol{s}})\right\|_2 . \tag{14}$$

Unfortunately, the computational complexity of the SRC algorithm is very high, which restricts its applications in practical recognition problems. To reduce the complexity of the robust algorithm, we try to avoid the sparse representation in the recognition stage. For each training sample $\boldsymbol{x}_i$, we define a dictionary matrix $\boldsymbol{A}=[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{0}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_n]$. $\hat{\boldsymbol{s}}_i$ calculated using Eq. (13) is a vector whose ith entry is zero. Improved SRC finds the projection $\boldsymbol{W}$ by solving the least squares problem defined in Eq. (4). Then, we can obtain the approximate reconstruction proportion of the test sample with training samples by mapping the test sample from the ith class to the reconstruction space. Using only the reconstruction proportion associated with the ith class, the given test sample can be accurately approximated. Fig. 4 illustrates the improved SRC.

We may summarize the procedure of the proposed improved SRC as Algorithm 3.

Algorithm 3 Improved SRC (ISRC)

Input: a matrix of training samples $\boldsymbol{X}=[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]\in\mathbb{R}^{M\times N}$ for c classes, and a test sample $\boldsymbol{y}\in\mathbb{R}^M$.

1. If $N<<M$, map M-dimensional samples of $\boldsymbol{X}$ and $\boldsymbol{y}$ to Q-dimensional samples of $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, \ldots, \tilde{\boldsymbol{x}}_N] \in \mathbb{R}^{Q\times N}$ and $\tilde{\boldsymbol{y}} \in \mathbb{R}^Q$ ($Q<<N$), respectively.

2. For each training sample $\tilde{\boldsymbol{x}}_i$ (i=1, 2, …, N), a sparse restriction for Eq. (3) is added to compute the reconstruction proportion:

$$\begin{aligned}&\hat{\boldsymbol{s}}_i = \min_{\boldsymbol{s}_i}\|\tilde{\boldsymbol{x}}_i - \boldsymbol{A}\boldsymbol{s}_i\|_2\\ &\text{s.t.}\ \ \|\boldsymbol{s}_i\|_0 \le \varsigma,\ \ \boldsymbol{1}^{\mathrm{T}}\boldsymbol{s}_i = 1,\end{aligned} \tag{15}$$

where $\boldsymbol{A} = [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_{i-1}, \boldsymbol{0}, \tilde{\boldsymbol{x}}_{i+1}, \ldots, \tilde{\boldsymbol{x}}_n]$, and ς is the small threshold value which restricts $\hat{\boldsymbol{s}}_i$ to be sparse enough.

3. Let $\boldsymbol{S}=[\hat{\boldsymbol{s}}_1, \hat{\boldsymbol{s}}_2, \ldots, \hat{\boldsymbol{s}}_N]$. Then solve the following optimization Frobenius norm problem:

$$\hat{\boldsymbol{W}} = \min_{\boldsymbol{W}}\|\boldsymbol{W}\tilde{\boldsymbol{X}} - \boldsymbol{S}\|_{\mathrm{F}} . \tag{16}$$

4. Compute the mapped data: $\boldsymbol{r}_{\mathrm{p}} = \hat{\boldsymbol{W}}\tilde{\boldsymbol{y}} = [a_1, a_2, \ldots, a_N]$. The improved NS classifier finds the minimum residuals:

$$l = \min_i \|\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{X}}\boldsymbol{\delta}_i(\boldsymbol{r}_{\mathrm{p}})\|, \quad i = 1, 2, \ldots, c. \tag{17}$$

Output: the predicted class of $\boldsymbol{y}$ is set equal to l.

To simplify the calculation, rewrite Eq. (15) as

$$\begin{aligned}&\hat{\boldsymbol{s}}_i = \min_{\boldsymbol{s}_i}\|\boldsymbol{s}_i\|_1\\ &\text{s.t.}\ \ \|\tilde{\boldsymbol{x}}_i - \boldsymbol{A}\boldsymbol{s}_i\|_2 \le \varepsilon,\ \ \boldsymbol{1}^{\mathrm{T}}\boldsymbol{s}_i = 1,\end{aligned} \tag{18}$$

where an optional error tolerance ε>0.

4 Experimental results

The experimental platform is configured with a general Intel Core 2 Duo CPU (E7500) 2.93 GHz, 2 GB RAM and MATLAB 7.11. To test the recognition accuracy of the proposed method, we perform comprehensive experiments on four famous facial databases:

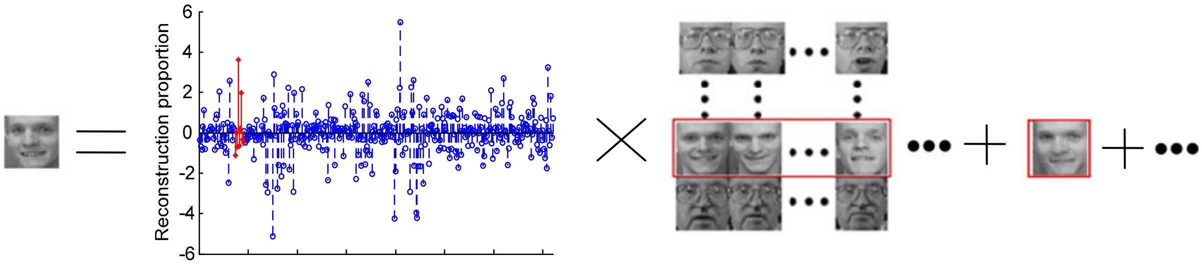


Fig. 4 Representation of a test image with reconstruction proportion by the improved sparse representation classifier

ORL: the ORL database consists of 400 frontal face images of 40 individuals. For some subjects, the images are taken at different times. The facial expressions (open or closed eyes, smiling or no smiling) and facial details (glasses or no glasses) also vary.

Yale: the Yale face database contains 165 images of 15 individuals (each person has 11 different images) with various facial expressions and lighting conditions.

YaleB: the Extended Yale B database consists of 2414 frontal-face images of 38 individuals. The face images were captured under various laboratory-controlled lighting conditions.

CMU PIE: the CMU PIE face database includes 68 subjects with 41 368 face images as a whole. The face images were captured by 13 synchronized cameras and 21 flashes, with varying pose illuminations and expressions.

To show the computational efficiency of the proposed methods, we use the cropped images with resolution 32×32, downloaded from http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html. Thus, each image can be represented by a 1024-dimensional vector, normalized to [0, 1]. Fig. 5 shows some face images from the four face databases. We perform each experiment 10 times and show an average of 10 experimental results for analysis.

Based on efficient linear projection to the reconstruction proportion, the complexity can be significantly reduced in the classification process. To evaluate the efficiency of classification, we select different numbers of samples as a training set and compare the performance between conventional classifiers and the proposed classifiers. Many classifiers, such as PCA, LPP, NPE, MMC, and SPP, may be adopted after feature extraction. In our approach we use the simple PCA algorithm as the preprocessor prior to applying the classification to ensure $Q<<N$. The feature subspace is spanned by the eigenvectors corresponding to Q eigenvalues where Q=150. The computational complexity of classifying a test sample with different numbers of training samples from the CMU PIE face database are shown in Fig. 6. For KNN and IKNN classifiers, we empirically set neighborhood size K equal to the training sample size per subject.

Fig. 6 gives the computational complexities of conventional and improved classifiers. It can be seen that the computational complexity of ISRC is much smaller than that of SRC. When there are 10000 training samples, the time consumption is about 78 s for SRC, but just about 0.3 s for ISRC. From the subgraph of Fig. 6, we can see that even the simple

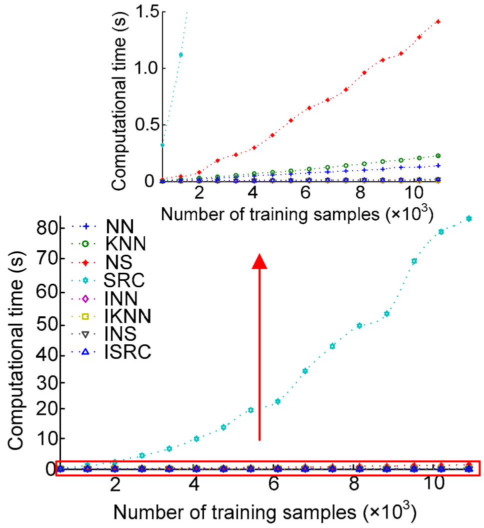


Fig. 6 The computational time of classifiers with different numbers of training samples

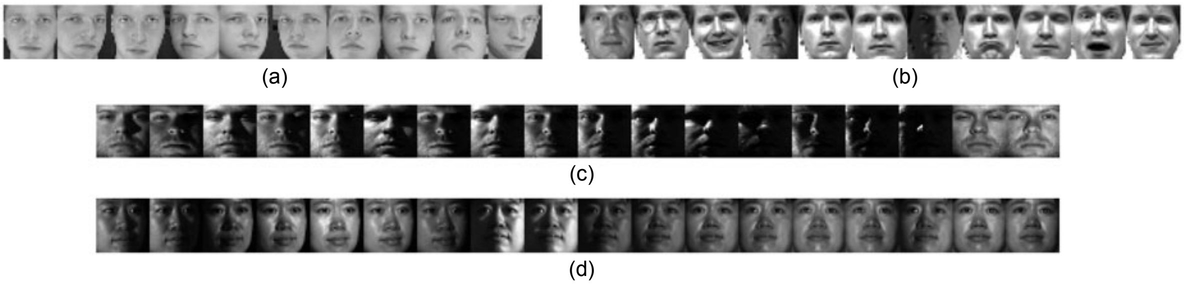


Fig. 5 Partial images of an individual from the ORL, Yale, YaleB, and CMU PIE face databases

(a) Ten face images of an individual from the ORL database; (b) Eleven face images of an individual from the Yale face database; (c) Partial images of an individual from the YaleB face database; (d) Partial images of an individual from the CMU PIE face database

NN and NS classifiers suffer from high complexity when the number of training samples gets large, but the proposed classifiers can significantly improve the efficiency as the number of training samples increases. Even when the number of training samples reaches 10 000, the time consumption of the proposed classifiers is almost 0 s. Comparing the complexity with the number of training samples for different algorithms, it is evident that the proposed improved NN, KNN, NS, and SRC classifiers significantly outperform traditional classifiers in complexity, particularly on large face databases, and maintain a slow computational complexity increase over the number of training samples.

The KNN classifier is always sensitive to the neighborhood size; that is, an inappropriate neighborhood size will lead to poor performance. Fig. 7 shows the recognition accuracy of KNN and IKNN classifiers with different neighborhood sizes on the YaleB database. The IKNN classifier greatly outperforms the KNN classifier in efficiency and is insensitive to the neighborhood size. The recognition accuracy of IKNN remains stable at about 95%, whereas the recognition accuracy of KNN sharply reduces from 65.72% with three neighbors to 47.20% with 32 neighbors.

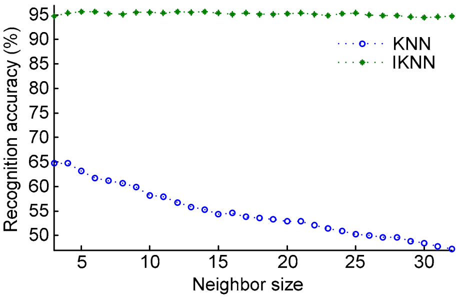


Fig. 7 The recognition accuracy of KNN and IKNN

To further verify the effectiveness and efficiency of the proposed classifiers, we evaluate the performance of the proposed algorithm and compare it to those of several conventional classifiers on ORL, Yale, YaleB, and CMU PIE face databases after applying different feature extraction algorithms such as PCA, LPP, NPE, MMC, and SPP. To better evaluate the performance, we also consider the supervised LPP and NPE (denoted by S-LPP and S-NPE, respectively) where a priori knowledge is used to obtain the nearest neighbors. We randomly select half of the image set per class for training (i.e., 5, 6, 32, and 85 images per subject for ORL, Yale, YaleB, and CMU PIE face databases, respectively) and the remaining images for test. For KNN and IKNN classifiers, we empirically set neighborhood size K=3. The best average recognition accuracies of the classifiers in conjunction with different conventional holistic face feature extractors are summarized in Tables 1–4, from which we have the following observations:

1. For PCA, LPP, etc., INN, IKNN, and INS classifiers outperform the conventional NN, KNN, and NS classifiers respectively on all the four face databases. For example, on the small Yale face database, PCA+INN, PCA+IKNN, and PCA+INS lead to 4.5%, 2.9%, and 13.7% of average improvement in comparison with PCA+NN, PCA+KNN, and PCA+NS, respectively; on the large YaleB face database, PCA+INN, PCA+IKNN, and PCA+INS lead to 27.0%, 28.6%, and 2.3% of average improvement in comparison with PCA+NN, PCA+KNN, and PCA+NS, respectively.

Table 1 Best average recognition accuracy on the ORL face database using different feature extraction algorithms

Method	Recognition accuracy (%)						
	PCA	LPP	NPE	S-LPP	S-NPE	MMC	SPP
NN	88.7±0.70	91.0±2.45	91.3±2.80	92.2±4.20	93.8±1.75	94.3±1.75	91.4±2.40
INN	91.5±2.45	91.3±2.80	91.5±2.45	93.6±2.10	94.5±3.50	94.3±1.75	91.6±1.55
KNN	85.7±0.70	89.1±3.40	89.2±3.35	92.5±3.50	93.8±1.75	91.5±0.50	89.3±3.25
IKNN	92.1±4.55	91.3±2.80	91.5±2.45	93.9±3.85	94.5±3.50	93.3±1.75	91.5±2.45
NS	92.8±2.80	92.6±2.10	93.0±2.45	94.7±3.15	95.0±2.45	93.5±3.50	92.3±1.75
INS	95.3±1.75	94.5±2.45	94.6±2.10	94.5±3.50	94.6±2.10	94.1±2.10	94.6±2.10
SRC	96.3±2.80	96.5±2.45	96.1±2.10	94.8±2.80	94.7±3.15	96.2±3.15	95.9±1.90
ISRC	96.0±3.50	96.5±2.45	96.5±2.45	94.4±3.85	95.0±2.45	96.5±2.45	96.5±2.45

NN: nearest neighbor; KNN: K-nearest neighbor; NS: nearest subspace; SRC: sparse representation based classification. INN, IKNN, INS, and ISRC are the improved NN, KNN, NS, and SRC, respectively. PCA: principal component analysis; LPP: locality preserving projections; NPE: neighborhood preserving embedding; S-LPP: supervised LPP; S-NPE: supervised NPE; MMC: maximum margin criterion, SPP: sparsity preserving projections

Table 2 Best average recognition accuracy on the Yale face database using different feature extraction algorithms

Method	Recognition accuracy (%)						
	PCA	LPP	NPE	S-LPP	S-NPE	MMC	SPP
NN	66.0±4.67	67.3±4.66	67.7±3.73	79.5±1.86	79.1±0.93	78.0±4.67	68.7±4.67
INN	70.5±6.53	70.1±7.46	71.1±8.40	79.5±1.86	79.2±1.87	81.3±9.33	71.5±7.47
KNN	67.1±8.40	67.3±4.66	68.3±5.60	78.4±3.73	79.1±0.93	77.9±1.87	68.3±5.60
IKNN	70.0±4.67	70.0±4.67	70.4±3.73	79.2±1.87	79.2±1.87	79.1±8.40	71.3±4.66
NS	70.7±5.34	71.5±7.47	71.9±6.54	81.5±2.80	82.4±3.73	69.1±3.74	79.5±7.47
INS	84.4±8.40	81.3±9.33	81.7±8.40	81.5±2.80	82.8±2.80	80.0±9.33	82.1±7.46
SRC	83.5±7.47	80.9±5.60	79.9±6.54	81.5±2.80	82.8±2.80	78.0±4.67	79.5±7.47
ISRC	85.1±3.74	82.0±4.67	82.4±3.73	81.5±2.80	82.8±2.80	79.1±8.40	82.8±2.80

NN: nearest neighbor; KNN: *K*-nearest neighbor; NS: nearest subspace; SRC: sparse representation based classification. INN, IKNN, INS, and ISRC are the improved NN, KNN, NS, and SRC, respectively. PCA: principal component analysis; LPP: locality preserving projections; NPE: neighborhood preserving embedding; S-LPP: supervised LPP; S-NPE: supervised NPE; MMC: maximum margin criterion, SPP: sparsity preserving projections

Table 3 Best average recognition accuracy on the YaleB face database using different feature extraction algorithms

Method	Recognition accuracy (%)						
	PCA	LPP	NPE	S-LPP	S-NPE	MMC	SPP
NN	66.8±0.87	67.1±2.10	67.4±1.43	88.4±0.12	90.2±0.57	92.4±0.69	67.4±1.45
INN	93.8±0.64	93.9±0.91	94.0±0.79	87.4±0.40	93.9±0.52	92.4±0.86	94.1±0.27
KNN	66.0±0.87	67.3±1.75	68.1±2.13	88.5±0.12	89.5±1.03	91.9±0.40	68.1±1.19
IKNN	94.6±0.5	94.6±0.14	94.7±0.09	88.5±0.17	93.8±0.57	92.6±0.29	94.5±0.30
NS	95.2±0.34	95.2±0.40	95.3±0.48	88.3±0.07	94.0±0.69	74.3±0.86	95.4±0.58
INS	97.5±0.17	97.6±0.40	97.7±0.36	91.9±0.20	94.8±0.52	93.4±0.93	97.7±0.30
SRC	98.0±0.30	98.2±0.25	98.1±0.30	86.3±0.86	87.3±0.86	86.3±0.86	97.9±1.11
ISRC	98.1±0.30	98.2±0.34	98.1±0.30	86.3±0.92	87.3±0.92	86.3±0.88	98.2±0.18

NN: nearest neighbor; KNN: *K*-nearest neighbor; NS: nearest subspace; SRC: sparse representation based classification. INN, IKNN, INS, and ISRC are the improved NN, KNN, NS, and SRC, respectively. PCA: principal component analysis; LPP: locality preserving projections; NPE: neighborhood preserving embedding; S-LPP: supervised LPP; S-NPE: supervised NPE; MMC: maximum margin criterion, SPP: sparsity preserving projections

Table 4 Best average recognition accuracy on the CMU PIE face database using different feature extraction algorithms

Method	Recognition accuracy (%)						
	PCA	LPP	NPE	S-LPP	S-NPE	MMC	SPP
NN	89.4±0.76	90.2±0.51	89.6±0.17	93.1±0.58	74.8±0.77	94.9±0.29	90.3±0.58
INN	94.0±0.18	93.3±0.13	94.1±0.11	94.2±0.99	94.1±0.20	97.1±0.16	94.3±0.11
KNN	85.1±0.84	89.7±0.25	89.6±0.52	93.2±0.78	69.3±0.44	94.8±0.26	89.8±0.17
IKNN	94.3±0.50	93.5±0.03	94.0±0.40	94.1±1.71	94.0±0.19	96.8±0.12	93.4±0.14
NS	97.2±0.29	94.7±0.12	96.3±0.05	94.9±0.66	97.3±0.17	70.4±2.29	94.8±0.11
INS	97.8±0.14	94.8±0.18	97.1±0.06	95.6±0.36	97.3±0.21	94.1±0.14	94.9±0.13
SRC	97.6±0.27	98.3±0.35	97.4±0.30	97.4±0.23	97.4±0.11	96.2±0.06	96.7±0.31
ISRC	98.1±0.12	98.3±0.44	97.5±0.40	97.5±0.23	97.5±0.14	97.7±0.08	97.2±0.38

NN: nearest neighbor; KNN: *K*-nearest neighbor; NS: nearest subspace; SRC: sparse representation based classification. INN, IKNN, INS, and ISRC are the improved NN, KNN, NS, and SRC, respectively. PCA: principal component analysis; LPP: locality preserving projections; NPE: neighborhood preserving embedding; S-LPP: supervised LPP; S-NPE: supervised NPE; MMC: maximum margin criterion, SPP: sparsity preserving projections

2. SRC almost always outperforms NN, KNN, and NS because of its properly harnessed sparsity. For PCA, LPP, etc., ISRC stays competitive and even slightly outperforms the highly effective SRC. For example, on the small Yale face database, PCA+ISRC achieves an average recognition accuracy of 85.1% compared to 83.5% of PCA+SRC; on the large YaleB face database, PCA+ISRC achieves an average recognition accuracy of 98.1% compared to 98.0% of PCA+SRC.

3. The supervised classifier usually possesses better accuracy in classification compared with unsupervised ones. In most cases, reconstruction based NS and SRC achieve higher recognition accuracy than distance based NN and KNN, and INS and ISRC achieve higher recognition accuracy than INN and IKNN. For example, on the Yale database, PCA+NS and PCA+SRC lead to 4.7% and 17.5% of average improvement in comparison with PCA+NN, respectively; PCA+INS and PCA+ISRC lead to 13.9% and 14.6% of average improvement in comparison with PCA+INN, respectively.

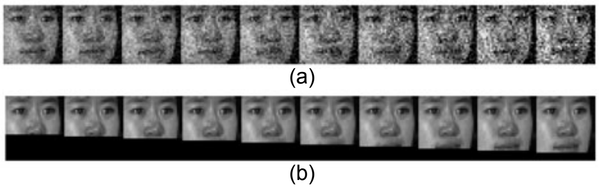
4. It is well known that different classifiers significantly affect the recognition performance of the feature extractor. For example, on the YaleB face database, PCA+NN achieves the best average recognition accuracy of 66.8% compared to 95.2% for PCA+NS. However, the feature extractors are insensitive to our proposed improved classifiers. For example, on the YaleB face database, PCA+INN, PCA+IKNN, PCA+INS, and PCA+ISRC achieve the best average recognition accuracies of 93.8%, 94.6%, 97.5%, and 98.1%, respectively.

Robustness to corruption and occlusion is a critical issue in face recognition (Wright *et al.*, 2009) and is the reason why SRC outperforms NN, KNN, and NS. To verify whether our proposed improved classifiers have the virtue of robustness to corruption and occlusion, we design the recognition experiment under random corruption and occlusion conditions. We process the test data set with white Gaussian noise and crops, respectively, and some facial images with corruption and occlusion are shown in Fig. 8.

Fig. 9 illustrates the recognition accuracy of the classifiers on the CMU PIE database with different intensities of white Gaussian noise and different occlusion sizes. Here, PCA is adopted as the feature extractor, and the signal-to-noise ratio (SNR) is used to denote the intensity of noise:

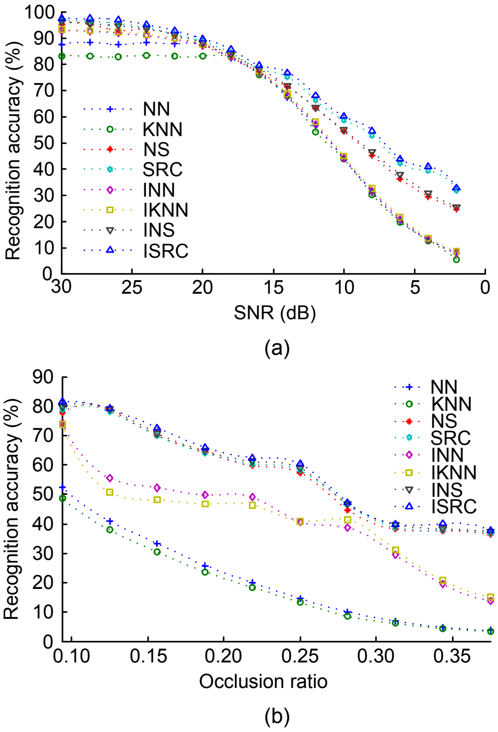
$$\mathrm{SNR} = 10\lg(S/N), \tag{19}$$

where S is the signal intensity and N is the noise intensity.


(a)

(b)

Fig. 8 Facial images with white Gaussian noise (a) or occlusion (b)



(a)

(b)

Fig. 9 Comparison of recognition accuracy with different noise intensities (a) or different occlusion sizes (b) on the CMU PIE face database

From Fig. 9 we have the following observations:

1. The robustness of the SRC algorithm has been proved experimentally on the face database with noise and occlusions in Wright *et al.* (2009). As shown in Fig. 9, SRC outperforms NN, KNN, and NS in recognition accuracy. For example, when SNR is set to 2 dB, NN, KNN, and NS achieve recognition accuracies of 6.12%, 5.36%, and 22.35%,

respectively, while SRC achieves a recognition accuracy of 28.9%; when the occlusion ratio is set to 0.375, NN, KNN, and NS achieve recognition accuracies of 3.82%, 3.51%, and 36.38%, respectively, while SRC achieves a recognition accuracy of 36.73%.

2. ISRC maintains the good robustness of SRC and outperforms INN, IKNN, and INS. For example, when SNR is set to 2 dB, INN, IKNN, and INS achieve recognition accuracies of 6.49%, 6.57%, and 23.35%, respectively, while ISRC achieves a recognition accuracy of 30.64%; when the occlusion ratio is set to 0.375, INN, IKNN, and INS achieve recognition accuracies of 13.85%, 15.13%, and 37.35%, respectively, while ISRC achieves a recognition accuracy of 37.79%.

3. NN and INN, KNN and IKNN, NS and INS, as well as SRC and ISRC, have the same attenuation characteristics respectively when the intensity of noise or the occlusion size is increased, which indicates that the improved classifiers properly inherit the traits of conventional classifiers.

5 Conclusions

In this paper we propose a novel fast classification scheme. A distance from sample to class (SCD) is defined as the classification metric. We exploit a new measure called the reconstruction proportion to reflect discriminant information and obtain a new space called the 'reconstruction space', by solving an optimization Frobenius norm problem. Samples are classified after being mapped to the reconstruction space. By this method, the high complexity calculations, such as calculation of distances and reconstruction of samples, are shifted from the recognition or classification phase to the training phase. The classification of a new sample can be efficiently performed as long as the projection from the sample space to the reconstruction space is obtained. We focus the proposed scheme on four popular classifiers, namely nearest neighbor (NN), *K*-nearest neighbor (KNN), nearest subspace (NS), and sparse representation based classifier (SRC), due to their representativeness. Improved nearest neighbor (INN), improved *K*-nearest neighbor (IKNN), improved nearest subspace (INS), and improved sparse representation based classifier (ISRC) can identify a facial image more accurately and efficiently when high complexity calculations are performed in the preprocessing phase instead of the recognition phase. Experimental results show that INN, IKNN, and INS outperform the conventional NN, KNN, and NS in recognition efficiency and classification accuracy. It is evident that ISRC is not only computationally efficient, but is also of high robustness. We will explore in our future work whether this framework can be extended to other non-Euclidean distance based classifiers.

References

Abate, A.F., Nappi, M., Riccio, D., Sabatino, G., 2007. 2D and 3D face recognition: a survey. *Pattern Recogn. Lett.*, **28**(14):1885-1906. [doi:10.1016/j.patrec.2006.12.018]

Athitsos, V., Alon, J., Sclaroff, S., Kollios, G., 2008. Boostmap: an embedding method for efficient nearest neighbor retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**(1):89-104. [doi:10.1109/TPAMI.2007.1140]

Baraniuk, R.G., 2007. Compressive sensing. *IEEE Signal Process. Mag.*, **24**(4):118-121. [doi:10.1109/MSP.2007.4286571]

Bax, E., 2000. Validation of nearest neighbor classifiers. *IEEE Trans. Inf. Theory*, **46**(7):2746-2752. [doi:10.1109/18.887892]

Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J., 1997. Eigenfaces vs. Fisher-faces: recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **19**(7):711-720. [doi:10.1109/34.598228]

Belkin, M., Niyogi, P., 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neur. Comput.*, **15**(6):1373-1396. [doi:10.1162/089976603321780317]

Candes, E.J., Wakin, M.B., 2008. An introduction to compressive sampling. *IEEE Signal Process. Mag.*, **25**(2):21-30. [doi:10.1109/MSP.2007.914731]

Cevikalp, H., Triggs, B., Polikar, R., 2008. Nearest Hyperdisk Methods for High-Dimensional Classification. Proc. 25th Int. Conf. on Machine Learning, p.120-127. [doi:10.1145/1390156.1390172]

Chen, C.J., 2010. Development of a Neural Classifier with Genetic Algorithm for Structural Vibration Suppression. Proc. Int. Conf. on Machine Learning and Cybernetics, p.2788-2795. [doi:10.1109/ICMLC.2010.5580789]

Donoho, D.L., 2006. Compressed sensing. *IEEE Trans. Inf. Theory*, **52**(4):1289-1306. [doi:10.1109/TIT.2006.871582]

Duda, R., Hart, P., Stork, D., 2001. Pattern Classification (2nd Ed.). John Wiley & Sons, New York.

He, X.F., Niyogi, P., 2003. Locality Preserving Projections. Advances in Neural Information Processing System 16.

He, X.F., Cai, D., Yan, S.C., Zhang, H.J., 2005. Neighborhood Preserving Embedding. IEEE Int. Conf. on Computer Vision, p.1208-1213. [doi:10.1109/ICCV.2005.167]

Ho, J., Yang, M., Lim, J., Lee, K., Kriegman, D., 2003. Clustering Appearances of Objects under Varying Illumination Conditions. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, p.11-18. [doi:10.1109/CVPR.2003.1211332]

Jain, A., Chandrasekaran, B., 1982. Dimensionality and Samples Size Considerations in Pattern Recognition Practice. Classification Pattern Recognition and Reduction of Dimensionality p.835-855. [doi:10.1016/S0169-7161(82)02042-2]

Jiang, X.D., Mandal, B., Kot, A., 2008. Eigenfeature regularization and extraction in face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**(3):383-394. [doi:10.1109/TPAMI.2007.70708]

Keller, J.M., Gray, M.R., Givens, J.A., 1985. A fuzzy *K*-nearest neighbor algorithm. *IEEE Trans. Syst. Man Cybern.*, **15**(4):580-585. [doi:10.1109/TSMC.1985.6313426]

Li, H.F., Jiang, T., Zhang, K.S., 2006. Efficient and robust feature extraction by maximum margin criterion. *IEEE Trans. Neur. Networks*, **17**(1):157-165. [doi:10.1109/TNN.2005.860852]

Li, S.Z., Lu, J., 1999. Face recognition using the nearest feature line method. *IEEE Trans. Neur. Networks*, **10**(2):439-443. [doi:10.1109/72.750575]

Liu, Y.G., Sam, S.Z., Li, C.G., You, Z.S., 2011. K-NS: a classifier by the distance to the nearest subspace. *IEEE Trans. Neur. Networks*, **22**(8):1256-1268. [doi:10.1109/TNN.2011.2153210]

Meo, R., Bachar, D., Ienco, D., 2012. LODE: a distance-based classifier built on ensembles of positive and negative observations. *Pattern Recogn.*, **45**(4):1409-1425. [doi:10.1016/j.patcog.2011.10.015]

Mullin, M., Sukthankar, R., 2000. Complete Cross-Validation for Nearest Neighbor Classifiers. Proc. Int. Conf. on Machine Learning, p.639-646.

Psaltis, D., Snapp, R., Venkatesh, S., 1994. On the finite sample performance of the nearest neighbor classifier. *IEEE Trans. Inf. Theory*, **40**(3):820-837. [doi:10.1109/18.335893]

Qiao, L.S., Chen, S.C., Tan, X.Y., 2010. Sparsity preserving projections with applications to face recognition. *Pattern Recogn.*, **43**(1):331-341. [doi:10.1016/j.patcog.2009.05.005]

Roweis, S.T., Saul, L.K., 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, **290**(5500):2323-2326. [doi:10.1126/science.290.5500.2323]

Singh, C., Walia, E., Mittal, N., 2012. Robust two-stage face recognition approach using global and local features. *Vis. Comput.*, **28**(11):1085-1098. [doi:10.1007/s00371-011-0659-7]

Stefan, A., Athitsos, V., Yuan, Q., Sclaroff, S., 2009. Reducing Jointboost-Based Multiclass Classification to Proximity Search. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, p.589-596. [doi:10.1109/CVPR.2009.5206687]

Tenenbaum, J.B., de Silva, V., Langford, J.C., 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, **29**(5500):2319-2323. [doi:10.1126/science.290.5500.2319]

Turk, M., Pentland, A., 1991. Eigenface for recognition. *J. Cogn. Neurosci.*, **3**(1):71-86. [doi:10.1162/jocn.1991.3.1.71]

Vincent, P., Bengio, Y., 2002. *K*-Local Hperplane and Convex Distance Nearest Neighbor Algorithms. Advances in Neural Information Processing Systems, p.985-992.

Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y., 2009. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **31**(2):210-227. [doi:10.1109/TPAMI.2008.79]

Zhang, B., Srihari, S.N., 2004. Fast *k*-nearest neighbor classification using cluster-based trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**(4):525-528. [doi:10.1109/TPAMI.2004.1265868]

Zhang, D.Q., Chen, S.C., Zhou, Z.H., 2006. Learning the kernel parameters in kernel minimum distance classifier. *Pattern Recogn.*, **39**(1):133-135. [doi:10.1016/j.patcog.2005.08.001]

Zhang, P., Zhu, X.Q., Tan, J.L., Guo, L., 2010. Classifier and Cluster Ensembles for Mining Concept Drifting Data Streams. Proc. IEEE Int. Conf. on Data Mining, p.1175-1180. [doi:10.1109/ICDM.2010.125]

Zhang, X.Z., Gao, Y.S., 2009. Face recognition across pose: a review. *Pattern Recogn.*, **42**(11):2876-2896. [doi:10.1016/j.patcog.2009.04.017]

Zhao, W., Chellappa, R., Rosenfeld, A., Phillips, P.J., 2003. Face recognition: a literature survey. *ACM Comput. Surv.*, **35**(4):399-458. [doi:10.1145/954339.954342]