Neural Process Lett
DOI 10.1007/s11063-013-9288-7

Face Recognition and Micro-expression Recognition Based on Discriminant Tensor Subspace Analysis Plus Extreme Learning Machine

Su-Jing Wang · Hui-Ling Chen · Wen-Jing Yan · Yu-Hsin Chen · Xiaolan Fu

Abstract In this paper, a novel recognition algorithm based on discriminant tensor subspace analysis (DTSA) and extreme learning machine (ELM) is introduced. DTSA treats a gray facial image as a second order tensor and adopts two-sided transformations to reduce dimensionality. One of the many advantages of DTSA is its ability to preserve the spatial structure information of the images. In order to deal with micro-expression video clips, we extend DTSA to a high-order tensor. Discriminative features are generated using DTSA to further enhance the classification performance of ELM classifier. Another notable contribution of the proposed method includes significant improvements in face and micro-expression recognition accuracy. The experimental results on the ORL, Yale, YaleB facial databases and CASME micro-expression database show the effectiveness of the proposed method.

Keywords Face recognition · Micro-expression recognition · Locality preserving projection · Discriminant information · Tensor subspace · Extreme learning machine

1 Introduction

In the real world, a face image or a micro-expression [7] video clip exists in a high-dimensional space. In order to handle high-dimensional face image or micro-expression video clip adequately, their dimensionality needs to be reduced [26].

S.-J. Wang (✉)· W.-J. Yan · Y.-H. Chen · X. Fu
State Key Laboratory of Brain and Cognitive Science, Institute of Psychology,
Chinese Academy of Sciences, Beijing 100101, China
e-mail: wangsujing@psych.ac.cn

S.-J. Wang
College of Computer Science and Technology, Jilin University, Changchun 130012, China

H.-L. Chen
College of Physics and Electronic Information, Wenzhou University, Wenzhou 325035, Zhejiang , China

Published online: 27 February 2013

As one of the most widely used dimensionality reduction methods, principal component analysis (PCA) [33] seeks the optimal projection directions according to maximal variances. Linear discriminant analysis (LDA) [1] uses discriminant information to search for the directions which are most effective for discrimination by maximizing the ratio between the between-class and within-class scatters. Different to LDA, max–min distance analysis (MMDA) [2] duly considers the separation of all classes pairs. To deal with the general case of data distribution, Bian and Tao [2] also extended MMDA to kernel MMDA (KMMDA). To overcome the non-smooth max–min optimization problem with orthogonal constraints which is introduced by MMDA/KMMDA, they developed a sequential convex relaxation algorithm to solve it approximately. Zhang et al. [45] proposed a patch alignment framework, which consists of two stages: part optimization and whole alignment. The framework reveals that (1) algorithms are intrinsically different in the patch optimization stage and (2) all algorithms share an almost identical whole alignment stage. As an application of this framework, they developed a new dimensionality reduction algorithm, namely discriminative locality alignment (DLA), by imposing discriminative information into the part optimization stage. DLA can (1) attack the distribution nonlinearity of measurements; (2) preserve the discriminative ability; and (3) avoid the small-sample-size problem. Li and Tao [23] proposed the simple exponential family PCA (SePCA) to employ exponential family distributions to handle general types of observations. The method also automatically discovers the number of essential principal components by using Bayesian inference. Zhou et al. [46] used elastic net to find the optimal sparse solution of the dimensionality reduction algorithm which is based on manifold learning.

Nonnegative matrix factorization (NMF) is a powerful matrix decomposition technique that approximates a nonnegative matrix by the product of two low-rank nonnegative matrix factors. In order to overcome occlusions and noises, Guan et al. minimized the Manhattan distance between $\mathbf{X}$ and $\mathbf{X}^T\mathbf{H}$ in NMF [10] and used manifold learning and discriminant information to improve NMF [11]. They [12] also presented a non-negative patch alignment framework to unify popular NMF related dimension reduction algorithms. Guan et al. [13] used Nesterov's optimal gradient method to solve NMF. NMF was also applied to deal with streaming data [14].

On other hand, locality preserving projections (LPP) [16] aims to preserve the local structure of the original space in the projective subspace. Discriminant locality preserving projections (DLPP) [43] encodes discriminant information into LPP to further improve the discriminant performance of LPP for face recognition. Potential shortages of these methods are singularity of within-class scatter matrices, limited available projection directions, high computational cost and a loss of the underlying spatial structure information of the images.

To overcome the above problems, some researchers have attempted to treat the image as a matrix instead of a vector. Yang et al. [41] proposed a 2D-PCA algorithm to compute the image scatter matrix from the image matrix representations directly. Li and Yuan [24] presented a 2D-LDA to extend LDA using the idea of the image matrix representations. Chen et al. [4] developed a 2D-LPP which directly extracts the proper features from image matrix representations by preserving the local structure of samples. Xu et al. [38] used discriminant information to construct the adjacency graph based on 2D-LPP, and Yu further developed [42] 2D-DLPP, a variation of 2D-LPP which uses discriminant information. These two-dimensional methods not only reduce the complexities of time and space but also preserve spatial structure information of the 2D images.

However, one disadvantage of two-dimensional methods (compared to one-dimensional methods) is that more feature coefficients are needed to represent an image, due to the fact that two-dimensional methods only employ single-sided transformations. Recently, tensor

methods, which employ two-sided transformation for a gray image, attract more attention in the field of feature extraction and dimension reduction, since many objects can be represented by multidimensional arrays, *i.e.* tensors. The number of dimensions is called the *order* of the tensor and each dimension defines one of the so-called *modes*. For example, a gray image is a second-order tensor, then its rows are called mode-1 of the tensor and its columns are called mode-2 of the tensor.

For second-order tensor, He et al. [15] proposed an algorithm, tensor subspace analysis (TSA), which preserves the local structure of samples using two-sided transformations. Inheriting the merits from TSA and 2D-DLPP, discriminant tensor subspace analysis (DTSA) is proposed in our previous work [37]. Its advantages include:

1. The discriminant information can further improve recognition performance.
2. More spatial information of the images are preserved by presenting the image as a tensor and higher compression ratios are achieved with the use of two-sided transformations.
3. Local structure of samples distribution is preserved.

For Nth-order tensor, Liu et al. [25] extended PCA from vector to tensor. In order to encode the discriminant information into the tensor subspace, GTDA [39] and DATER [31] extended LDA and MSD [29] from vector to tensor, respectively. Tao et al. [32] also used Bayesian tensor to model 3-D face. For more knowledge about tensor, please refer to [30]. Wang et al. [36] treated an color facial images as a 3rd-order tensor and proposed tensor discriminant color space (TDCS) model. They [35] also used elastic net to propose sparse tensor discriminant color space.

On the other hand, the choice of classifier plays an important role in solving face recognition problems. In most face recognition research, nearest neighbor classifier (NNC) and support vector machine (SVM) are perhaps amongst the most frequently employed techniques. NNC is a simple yet powerful classifier, and it has been shown to be one of the most successful and robust classifiers on many data sets. Compared with NNC, SVM is a much more sophisticated classifier which is based on the statistical learning theory [34], which has found its application in a wide range of fields [3,5,22,28]. More recently, a new learning algorithm for a single hidden layer feed-forward neural networks (SLFNs) was proposed, called extreme learning machine (ELM) [20], which has shown to have extremely fast learning speed and obtain excellent generalization capability as well as keeping parameter tuning-free. Owing to its universal approximation capability, it has become more and more popular for solving a large number of benchmark problems and applications from regression and classification areas [19,21,27,44]. In order to construct an effective and efficient face recognition system, there is a pressing demand for more computationally efficient and high-accuracy classification techniques. In this paper we attempt to investigate the effectiveness of ELM approach in conducting face recognition and micro-expression recognition tasks. In order to deal with micro-expression video data , we extend DTSA to a high-order tensor.

The proposed recognition method is comprised of two stages. The first stage aims at constructing the discriminant features based on DTSA and high-order DTSA. Later on, switching from feature extraction to model construction, in the second stage, the obtained features are fed into the designed ELM classifier to conduct face recognition tasks. The effectiveness of the proposed method has been rigorously evaluated against the ORL and Yale face databases, which are commonly used among researchers who use pattern recognition methods for face recognition. We also conducted 3rd-order DTSA (DTSA3) and ELM on the Chinese Academy of Sciences Micro-Expression (CASME) database. The rest of this paper is organized as follows: in Sect. 2, we will introduce the discriminant tensor subspace analysis and extend it to a high-order tensor; in Sect. 3, we use ELM to classify the discriminant

features extracted by using DTSA; in Sect. 4, the experimental results are reported and analyzed; finally in Sect. 5, conclusions are drawn and several issues for future works are described.

2 Algorithm

2.1 DTSA

We have a set $\mathcal{X}$ consisting of N samples coming from C classes:

$$\mathcal{X} = \left\{ \mathbf{X}_1^1, \mathbf{X}_2^1, \ldots, \mathbf{X}_{N_1}^1, \mathbf{X}_1^2, \mathbf{X}_2^2, \ldots, \mathbf{X}_{N_2}^2, \ldots, \mathbf{X}_1^C, \mathbf{X}_2^C, \ldots, \mathbf{X}_{N_C}^C \right\} \tag{1}$$

where $\mathbf{X}_i^c \in \mathbb{R}^{I_1 \times I_2}$ means the ith sample in the cth class. N_c is the number of samples in the cth class, and $N_1 + N_2 + \ldots + N_C = N$ is satisfied. The task is to learn the two matrices $\mathbf{U}$ and $\mathbf{V}$ which project those N samples to

$$\mathbf{Y}_i^c = \mathbf{U}^T \mathbf{X}_i^c \mathbf{V}, \quad i = 1, 2, \ldots, N_c, \quad c = 1, 2, \ldots, C. \tag{2}$$

where $\mathbf{Y}_i^c \in \mathbb{R}^{L_1 \times L_2}$.

If the two samples $\mathbf{X}_i^c$ and $\mathbf{X}_j^c$ in the same class are close, then the corresponding projected points $\mathbf{Y}_i^c$ and $\mathbf{Y}_j^c$ are close as well. A reasonable criterion for the projection is to minimize the following objective function:

$$\min \sum_{c=1}^{C} \sum_{i,j=1}^{N_c} \|\mathbf{Y}_i^c - \mathbf{Y}_j^c\|_F^2 W_{ij}^c \tag{3}$$

where $\mathbf{W}^c$ is the *within-class similarity matrix* of cth class, each entry W_{ij}^c is the similarity between the samples $\mathbf{X}_i^c$ and $\mathbf{X}_j^c$, and it is defined as: $W_{ij}^c = \exp(-\|\mathbf{X}_i^c - \mathbf{X}_j^c\|_F^2 / t)$, where $\|\cdot\|$ is the Frobenius norm of matrix, *i.e.* $\|\mathbf{A}\|_F = \sqrt{\sum_i \sum_j A_{ij}^2}$.

Additionally, a reasonable criterion for the projection is to maximize the following objective function:

$$\max \sum_{i,j=1}^{C} \|\overline{\mathbf{Y}}_i - \overline{\mathbf{Y}}_j\|_F^2 B_{ij} \tag{4}$$

where $\mathbf{B}$ is the *between-class similarity matrix*, each entry B_{ij} is the similarity between the mean samples $\overline{\mathbf{X}}_i$ and $\overline{\mathbf{X}}_j$, and it is defined as: $B_{ij} = \exp(-\|\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j\|_F^2 / t)$, where $\overline{\mathbf{X}}_i = \frac{1}{N_i} \sum_{k=1}^{N_i} \mathbf{X}_k^i$. A reasonable criterion function is as follows:

$$\max_{\mathbf{U},\mathbf{V}} \frac{\sum_{i,j=1}^{C} \|\overline{\mathbf{Y}}_i - \overline{\mathbf{Y}}_j\|_F^2 B_{ij}}{\sum_{c=1}^{C} \sum_{i,j=1}^{N_c} \|\mathbf{Y}_i^c - \mathbf{Y}_j^c\|_F^2 W_{ij}^c} \tag{5}$$

Since $\|\mathbf{A}\|_F^2 = tr(\mathbf{A}^T\mathbf{A})$, the denominator of Eq. (5) can be simplified as:

$$
\begin{aligned}
&\frac{1}{2}\sum_{c=1}^{C}\sum_{i,j=1}^{N_c}\|\mathbf{Y}_i^c-\mathbf{Y}_j^c\|_F^2 W_{ij}^c \\
&=\frac{1}{2}\sum_{c=1}^{C}\sum_{i,j=1}^{N_c} tr\left[(\mathbf{Y}_i^c-\mathbf{Y}_j^c)^T(\mathbf{Y}_i^c-\mathbf{Y}_j^c)\right]W_{ij}^c \\
&=tr\left[\frac{1}{2}\sum_{c=1}^{C}\sum_{i,j=1}^{N_c}(\mathbf{U}^T\mathbf{X}_i^c\mathbf{V}-\mathbf{U}^T\mathbf{X}_j^c\mathbf{V})^T(\mathbf{U}^T\mathbf{X}_i^c\mathbf{V}-\mathbf{U}^T\mathbf{X}_j^c\mathbf{V})W_{ij}^c\right] \\
&=tr\left\{\mathbf{V}^T\left[\frac{1}{2}\sum_{c=1}^{C}\sum_{i,j=1}^{N_c}(\mathbf{U}^T\mathbf{X}_i^c-\mathbf{U}^T\mathbf{X}_j^c)^T(\mathbf{U}^T\mathbf{X}_i^c-\mathbf{U}^T\mathbf{X}_j^c)W_{ij}^c\right]\mathbf{V}\right\} \\
&=tr\left\{\mathbf{V}^T\left\{\sum_{c=1}^{C}\sum_{i,j=1}^{N_c}\left[(\mathbf{U}^T\mathbf{X}_i^c)^T\mathbf{U}^T\mathbf{X}_i^c-(\mathbf{U}^T\mathbf{X}_i^c)^T\mathbf{U}^T\mathbf{X}_j^c\right]W_{ij}^c\right\}\mathbf{V}\right\} \\
&=tr\left\{\mathbf{V}^T\left\{\sum_{c=1}^{C}\left[\sum_{i=1}^{N_c}(\mathbf{U}^T\mathbf{X}_i^c)^T\mathbf{U}^T\mathbf{X}_i^c\sum_{j=1}^{N_c}W_{ij}^c-\sum_{i,j=1}^{N_c}(\mathbf{U}^T\mathbf{X}_i^c)^T\mathbf{U}^T\mathbf{X}_j^cW_{ij}^c\right]\right\}\mathbf{V}\right\} \\
&=tr\left\{\mathbf{V}^T\left\{\sum_{c=1}^{C}\left[(\mathbf{P}_U^c)^T(\mathbf{D}^c\otimes\mathbf{I}_{L_1})\mathbf{P}_U^c-(\mathbf{P}_U^c)^T(\mathbf{W}^c\otimes\mathbf{I}_{L_1})\mathbf{P}_U^c\right]\right\}\mathbf{V}\right\} \\
&=tr\left\{\mathbf{V}^T\mathbf{P}_U^T\left[(\mathbf{D}-\mathbf{W})\otimes\mathbf{I}_{L_1}\right]\mathbf{P}_U\mathbf{V}\right\} \\
&=tr\left[\mathbf{V}^T\mathbf{P}_U^T\left(\mathbf{L}\otimes\mathbf{I}_{L_1}\right)\mathbf{P}_U\mathbf{V}\right] \\
&=tr\left(\mathbf{V}^T\mathbf{S}_L^U\mathbf{V}\right)
\end{aligned}
\tag{6}
$$

where,

$$
\mathbf{L}=\begin{bmatrix}\mathbf{L}^1 & & & \\ & \ddots & & \\ & & \mathbf{L}^c & \\ & & & \ddots \\ & & & & \mathbf{L}^C\end{bmatrix} \tag{7}
$$

$\mathbf{L}^c=\mathbf{D}^c-\mathbf{W}^c$ is a Laplacian matrix, where $\mathbf{D}^c$ is a corresponding diagonal matrix and its entry $D_{ii}^c=\sum_j W_{ij}^c$.

$$
\mathbf{P}_U^c=\begin{bmatrix}\mathbf{U}^T\mathbf{X}_1^c \\ \mathbf{U}^T\mathbf{X}_2^c \\ \vdots \\ \mathbf{U}^T\mathbf{X}_{N_c}^c\end{bmatrix} \tag{8}
$$

$$\mathbf{P}_U = \begin{bmatrix} \mathbf{P}_U^1 \\ \mathbf{P}_U^2 \\ \vdots \\ \mathbf{P}_U^{N_c} \end{bmatrix} \tag{9}$$

and $\mathbf{S}_L^U = \mathbf{P}_U^T(\mathbf{L} \otimes \mathbf{I}_{L_1})\mathbf{P}_U$.

Similarly, the numerator of Eq. (5) can be simplified as:

$$\begin{aligned}
&\frac{1}{2}\sum_{i,j=1}^{C} \|\overline{\mathbf{Y}}_i - \overline{\mathbf{Y}}_j\|_F^2 B_{ij} \\
&= \frac{1}{2}\sum_{i,j=1}^{C} tr\left[(\overline{\mathbf{Y}}_i - \overline{\mathbf{Y}}_j)^T(\overline{\mathbf{Y}}_i - \overline{\mathbf{Y}}_j)\right] B_{ij} \\
&= \frac{1}{2}\sum_{i,j=1}^{C} tr\left[(\mathbf{U}^T\overline{\mathbf{X}}_i\mathbf{V} - \mathbf{U}^T\overline{\mathbf{X}}_j\mathbf{V})^T(\mathbf{U}^T\overline{\mathbf{X}}_i\mathbf{V} - \mathbf{U}^T\overline{\mathbf{X}}_j\mathbf{V})\right] B_{ij} \\
&= \frac{1}{2}\sum_{i,j=1}^{C} tr\left[\mathbf{V}^T(\mathbf{U}^T\overline{\mathbf{X}}_i - \mathbf{U}^T\overline{\mathbf{X}}_j)^T(\mathbf{U}^T\overline{\mathbf{X}}_i - \mathbf{U}^T\overline{\mathbf{X}}_j)\mathbf{V}\right] B_{ij} \\
&= tr\left\{\sum_{i,j=1}^{C} \mathbf{V}^T\left[(\mathbf{U}^T\overline{\mathbf{X}}_i)^T\mathbf{U}^T\overline{\mathbf{X}}_i - (\mathbf{U}^T\overline{\mathbf{X}}_i)^T\mathbf{U}^T\overline{\mathbf{X}}_j\right]\mathbf{V}B_{ij}\right\} \\
&= tr\left\{\sum_{i=1}^{C}\mathbf{V}^T\left[(\mathbf{U}^T\overline{\mathbf{X}}_i)^T\mathbf{U}^T\overline{\mathbf{X}}_i\sum_{j=1}^{C}B_{ij}\right]\mathbf{V} - \sum_{i,j=1}^{C}\mathbf{V}^T(\mathbf{U}^T\overline{\mathbf{X}}_i)^T B_{ij}\mathbf{U}^T\overline{\mathbf{X}}_j\mathbf{V}\right\} \\
&= tr\left\{\mathbf{V}^T\mathbf{Q}_U^T[(\mathbf{E} - \mathbf{B}) \otimes \mathbf{I}_{L_1}]\mathbf{Q}_U\mathbf{V}\right\} \\
&= tr\left[\mathbf{V}^T\mathbf{Q}_U^T(\mathbf{H} \otimes \mathbf{I}_{L_1})\mathbf{Q}_U\mathbf{V}\right] \\
&= tr\left(\mathbf{V}^T\mathbf{S}_H^U\mathbf{V}\right)
\end{aligned} \tag{10}$$

where $\mathbf{H} = \mathbf{E} - \mathbf{B}$. $\mathbf{E}$ is a diagonal matrix, and its entries are column (or row, since $\mathbf{B}$ is symmetric) sum of $\mathbf{B}$, $E_{ii} = \sum_j B_{ij}$. Here, $\mathbf{H}$ is also a real symmetric matrix. For the mean value of each class $\overline{\mathbf{X}}_c$ ($c = 1, 2, \ldots, C$),

$$\mathbf{Q}_U = \begin{bmatrix} \mathbf{U}^T\overline{\mathbf{X}}_1 \\ \mathbf{U}^T\overline{\mathbf{X}}_2 \\ \vdots \\ \mathbf{U}^T\overline{\mathbf{X}}_{N_c} \end{bmatrix} \tag{11}$$

And $\mathbf{S}_H^U = \mathbf{Q}_U^T(\mathbf{H} \otimes \mathbf{I}_{L_1})\mathbf{Q}_U$. Therefore, for a given $\mathbf{U}$, the solution to Eq. (5) can be converted into the following optimal problem about a variable $\mathbf{V}$:

$$\max_{\mathbf{V}} \frac{tr(\mathbf{V}^T\mathbf{S}_H^U\mathbf{V})}{tr(\mathbf{V}^T\mathbf{S}_L^U\mathbf{V})} \tag{12}$$

It is easy to see that the optimal $\mathbf{V}$ should be the generalized eigenvalues problem:

$$\mathbf{S}_H^U \mathbf{v} = \lambda \mathbf{S}_L^U \mathbf{v} \tag{13}$$

the matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{L_2}]$ consists of the L_2 generalized eigenvectors corresponding to the largest L_2 generalized eigenvalues of the matrix pencil $(\mathbf{S}_H^U, \mathbf{S}_L^U)$.

For a given $\mathbf{V}$, similarly, the solution to Eq. (5) can be converted into the following generalized eigenvalues problem:

$$\mathbf{S}_H^V \mathbf{u} = \lambda \mathbf{S}_L^V \mathbf{u} \tag{14}$$

the matrix $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{L_1}]$ consists of the L_1 generalized eigenvectors corresponding to the largest L_1 generalized eigenvalues of the matrix pencil $(\mathbf{S}_H^V, \mathbf{S}_L^V)$. $\mathbf{S}_H^V = \mathbf{Q}_V^T(\mathbf{H} \otimes \mathbf{I}_{L_2})\mathbf{Q}_V, \mathbf{S}_L^V = \mathbf{P}_V^T(\mathbf{L} \otimes \mathbf{I}_{L_2})\mathbf{P}_V, \mathbf{P}_V^c = \left[\mathbf{X}_1^c\mathbf{V}, \mathbf{X}_2^c\mathbf{V}, \ldots, \mathbf{X}_{N_c}^c\mathbf{V}\right], \mathbf{P}_V = \left[\mathbf{P}_V^1, \mathbf{P}_V^2, \ldots, \mathbf{P}_V^{N_c}\right]$ and $\mathbf{Q}_V = \left[\overline{\mathbf{X}}_1\mathbf{V}, \overline{\mathbf{X}}_2\mathbf{V}, \ldots, \overline{\mathbf{X}}_{N_c}\mathbf{V}\right]$.

From the above analysis, we see that the optimizations of $\mathbf{U}$ and $\mathbf{V}$ depend on each other. So, an iterative procedure can be utilized to solve Eq. (5).

2.2 Extend DTSA to High-Order Tensor

DTSA only deals with 2-order tensor. In this section, we extended DTSA to high-order tensor in order to deal with high-order data such as micro-expression. The task of N-order DTSA thus becomes learning N projection matrices $\mathbf{U}_n$ of size $I_n \times L_n$ ($L_n < I_n$) which map those M points to a set of new points

$$\begin{aligned} \mathcal{Y}_i^c &= \mathcal{X}_i^c \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \ldots \times_N \mathbf{U}_N^T, \\ & i = 1, 2, \ldots, M_c, \quad c = 1, 2, \ldots, C. \end{aligned} \tag{15}$$

Eq. (15) can be n-mode unfolded as follows:

$$\mathbf{Y}_{i(n)}^c = \mathbf{U}_n^T \mathbf{X}_{i(n)}^c (\mathbf{U}_N \otimes \ldots \otimes \mathbf{U}_{n+1} \otimes \mathbf{U}_{n-1} \otimes \ldots \otimes \mathbf{U}_1) = \mathbf{U}_n^T \mathbf{X}_{i(n)}^c \tilde{\mathbf{U}}_n \tag{16}$$

where $\tilde{\mathbf{U}}_n = \mathbf{U}_N \otimes \ldots \otimes \mathbf{U}_{n+1} \otimes \mathbf{U}_{n-1} \otimes \ldots \otimes \mathbf{U}_1$.

Similarly, we can define as follows:

$$\mathbf{P}_n^c = \left[\mathbf{X}_{1(n)}^c \tilde{\mathbf{U}}_n, \mathbf{X}_{2(n)}^c \tilde{\mathbf{U}}_n, \ldots, \mathbf{X}_{M_c(n)}^c \tilde{\mathbf{U}}_n\right], \tag{17}$$

$$\mathbf{P}_n = \left[\mathbf{P}_n^1, \mathbf{P}_n^2, \ldots, \mathbf{P}_n^C\right] \tag{18}$$

and

$$\mathbf{Q}_n = \left[\overline{\mathbf{X}}_{(n)}^1 \tilde{\mathbf{U}}_n, \overline{\mathbf{X}}_{(n)}^2 \tilde{\mathbf{U}}_n, \ldots, \overline{\mathbf{X}}_{(n)}^C \tilde{\mathbf{U}}_n\right] \tag{19}$$

So, the denominator of Eq. (5) can be reduced to:

$$\begin{aligned} & \frac{1}{2} \sum_{c=1}^{C} \sum_{i,j=1}^{M_c} \|\mathcal{Y}_i^c - \mathcal{Y}_j^c\|_F^2 W_{ij}^c \\ &= tr\left\{\mathbf{U}_n^T \mathbf{P}_n \left[(\mathbf{D} - \mathbf{W}) \otimes \mathbf{I}\right] \mathbf{P}_n^T \mathbf{U}_n\right\} \\ &= tr\left[\mathbf{U}_n^T \mathbf{P}_n \left(\mathbf{L} \otimes \mathbf{I}\right) \mathbf{P}_n^T \mathbf{U}_n\right] \\ &= tr\left(\mathbf{U}_n^T \mathbf{S}_L^{(n)} \mathbf{U}_n\right) \end{aligned}$$

where, $\mathbf{S}_L^{(n)} = \mathbf{P}_{(n)}^T(\mathbf{L} \otimes \mathbf{I})\mathbf{P}_{(n)}$ and $\mathbf{I}$ is an identity matrix. Similarly, the numerator of Eq. (5) can be reduced to:

$$\begin{aligned}
&\frac{1}{2}\sum_{i,j=1}^{C} \|\overline{\mathcal{Y}}^i - \overline{\mathcal{Y}}^j\|_F^2 B_{ij} \\
&= tr\left\{\mathbf{U}_n^T\mathbf{Q}_n[(\mathbf{E}-\mathbf{B}) \otimes \mathbf{I}]\mathbf{Q}_n^T\mathbf{U}_n\right\} \\
&= tr\left[\mathbf{U}_n^T\mathbf{Q}_n(\mathbf{H} \otimes \mathbf{I})\mathbf{Q}_n^T\mathbf{U}_n\right] \\
&= tr\left(\mathbf{U}_n^T\mathbf{S}_H^{(n)}\mathbf{U}_n\right)
\end{aligned}$$

where, $\mathbf{S}_H^{(n)} = \mathbf{Q}_{(n)}^T(\mathbf{H} \otimes \mathbf{I})\mathbf{Q}_{(n)}$.

So, given all the other projection matrices $\mathbf{U}_1, \ldots, \mathbf{U}_{n-1}, \mathbf{U}_{n+1}, \ldots, \mathbf{U}_N$, the criterion (5) can be written as follow:

$$\max \frac{tr\left(\mathbf{U}_n^T\mathbf{S}_H^{(n)}\mathbf{U}_n\right)}{tr\left(\mathbf{U}_n^T\mathbf{S}_L^{(n)}\mathbf{U}_n\right)} \tag{20}$$

According to Rayleigh quotient, Eq. (20) is maximized if and only if the matrix $\mathbf{U}_n$ consists of the L_n generalized eigenvectors corresponding to the largest L_n generalized eigenvalues of the matrix pencil $(\mathbf{S}_H^{(n)}, \mathbf{S}_L^{(n)})$.

Since the $\mathbf{S}_H^{(n)}$ and $\mathbf{S}_L^{(n)}$ depends on $\mathbf{U}_1, \ldots, \mathbf{U}_{n-1}, \mathbf{U}_{n+1}, \ldots, \mathbf{U}_N$, we can see that the optimization of $\mathbf{U}_n$ depends on the projections in other modes. An iterative procedure can be constructed to maximize Eq. (20).

3 Classification Using ELM

ELM as a new learning algorithm for single layer feed forward neural networks (SLFNs) as shown in Fig. 1, was first introduced by Huang et al. [20,21]. ELM seeks to overcome the challenging issues faced with the traditional SLFNs learning algorithms such as slow learning speed, trivial parameter tuning and poor generalization capability. ELM has demonstrated great potential in handling classification and regression tasks with excellent generalization performance. The learning speed of ELM is much faster than conventional gradient based iterative learning algorithms of SLFNs like back propagation algorithm while obtaining better generalization performance. ELM has several significant features [21] which distinguish itself from the traditional learning algorithms of SLFNs:

1. ELM is easily and effectively used by avoiding tedious and time-consuming parameter tuning.
2. ELM has extremely fast learning speed.
3. ELM has much better generalization performance than the gradient based iterative learning algorithms in most cases.
4. ELM is much simpler and without being involved in local minima and over-fitting.
5. ELM can be used to train SLFNs with many non-differentiable activation functions.

Given a training set $X = \{(x_i, t_i)|x_i \in R^n, t_i \in R^m, i = 1, 2, \ldots, N\}$, where x_i is the $n \times 1$ input feature vector and t_i is a $m \times 1$ target vector. The standard SLFNs which has

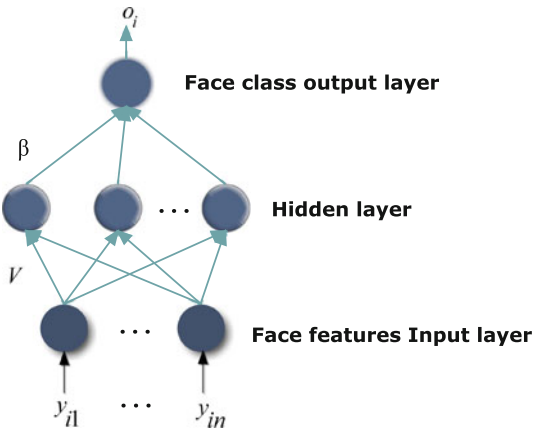


Fig. 1 The structure of ELM model

an activation function $g(x)$, and the number of hidden neurons $\tilde{N}$ can be mathematically modeled as follows:

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = o_j, \quad j = 1, 2, \ldots, N \tag{21}$$

where w_i is the weight vector between the ith neuron in the hidden layer and the input layer, b_i means the bias of the ith neuron in the hidden layer; β_i is the weight vector between the ith hidden neuron and the output layer; and o_j is the target vector of the jth input data. Here, $w_i \cdot x_j$ denotes the inner product of w_i and x_i.

If SLFNs can approximate these N samples with zero error, we will have $\sum_{j=1}^{N} ||o_j - t_j|| = 0$, i.e., there exist β_i, w_i, b_i such that $\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = t_j$, $j = 1, 2, \ldots, N$. The above equation can be reformulated compactly as:

$$H\beta = T \tag{22}$$

where

$$\mathrm{H}(w_{1,} \cdots, w_{\tilde{N}}, b_{1,} \cdots, b_{\tilde{N}}, x_{1,} \cdots, x_N) = \begin{pmatrix} g(w_1 \cdot x_1 + b_1) & \ldots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{pmatrix}_{N \times \tilde{N}} \tag{23}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \tag{24}$$

and

$$\mathrm{T} = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \tag{25}$$

As named by Huang et al. [18], H is called the hidden layer output matrix of the neural network, with the ith column of H being the ith hidden neuron output with respect to inputs

$x_1, x_2, \ldots, x_N$. Huang [17,19] has shown that the input weights and the hidden layer biases of SLFNs need not be adjusted at all and can be arbitrarily given. Under this assumption, the output weights can be analytically determined by finding the least square solution $\hat{\beta}$ of the linear system $\mathrm{H}\beta = \mathrm{T}$:

$$||\mathrm{H}(w_1, \cdots, w_{\tilde{N}}, b_1, \cdots, b_{\tilde{N}})\hat{\beta} - \mathrm{T}|| = \min_{\beta} ||\mathrm{H}(w_1, \cdots, w_{\tilde{N}}, b_1, \cdots, b_{\tilde{N}})\beta - \mathrm{T}|| \tag{26}$$

Eq. (26) can be easily accomplished using a linear method, such as the Moore-Penrose (MP) generalized inverse of H, as is shown in Eq. (27)

$$\mathrm{H}\beta = \mathrm{T} \Longrightarrow \hat{\beta} = \mathrm{H}^{\dagger}\mathrm{T} \tag{27}$$

where $\mathrm{H}^{\dagger}$ is the MP generalized inverse of the matrix H. The use of the MP generalized inverse method has led to the minimum norm least-squares (LS) solution, which is unique and has the smallest norm among all the LS solutions. As analyzed by Huang et al. [20], by using such MP inverse method, ELM tends to obtain a good generalization performance with a dramatically increased learning speed.

In summary, the learning steps of the ELM algorithm can be summarized by the following three steps:

Given a training set $\aleph = \{(x_i, t_i)|x_i \in R^n, t_i \in R^m, i = 1, 2, \ldots, N\}$, an activation function $g(x)$, and the number of hidden neurons $\tilde{N}$,

1. Randomly assign the input weights w_i and bias $b_i, i = 1, 2, \ldots, \tilde{N}$.
2. Calculate the hidden layer output matrix H.
3. Calculate the output weight $\beta = \mathrm{H}^{\dagger}\mathrm{T}, \mathrm{T} = [t_1, t_2, \ldots, t_n]^T$.

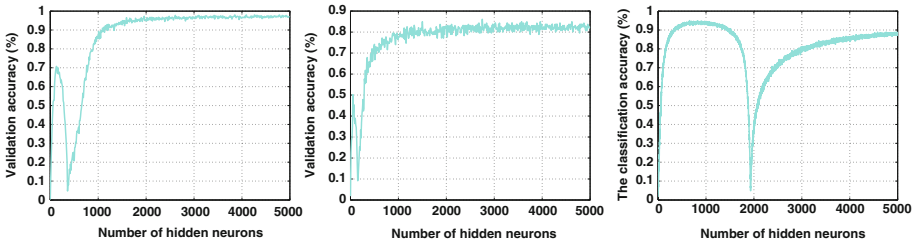
4 Experiments and Results

4.1 ELM Parameter Selection

One of the advantages of ELM over other methods is that the only parameter required to be determined is the number of hidden neurons. In this work, ELM[1] models are built via the stratified fivefold cross validation procedure on the two face databases through gradually increasing the number of hidden neurons from 1 to 5,000 in interval of 10. Namely the data were divided into ten subsets. Each time, one of the ten subsets is used as the test set and the other nine subsets are put together to form a training set. Then the average validation accuracy across all ten trials is computed. The advantage of this method is that all of the test sets are independent and the reliability of the results could be obtained.

The validation accuracy against the number of hidden neurons for ELM on the Yale database, ORL database and YaleB database are shown in Fig. 2a–c respectively. It is interesting to find that with the increasing of the number of hidden neurons, ELM first reaches local peak validation accuracy, and then increases gradually and finds the global optimal validation accuracy on the Yale database and ORL database. The highest validation accuracy has been achieved when the number of hidden neurons is equal to 2,751 and 4,091 for the Yale database and ORL database, respectively. The phenomena is opposite for the YaleB database, ELM finds the global optimal validation accuracy when the number of hidden neurons is equal to 661, after then the validation accuracy decreases gradually. Although after 1,900 hidden

[1] The matlab code can be downloaded from http://www3.ntu.edu.sg/home/egbhuang/.



(a) ORL **(b)** Yale **(c)** YaleB

Fig. 2 Validation accuracy vs. number of hidden neurons for ELM

Fig. 3 Sample images of one individual from the ORL database

Fig. 4 Sample images of one individual in the YALE database

neurons the validation accuracy increase gradually, ELM can not find the higher validation accuracy than the one at the 661 hidden neurons. Therefore, 2751, 4091 and 661 hidden neurons are chosen to create the training model for Yale database, ORL database and YaleB database in our implementations, respectively. The sigmoid activation function is used to compute the hidden layer output matrix.

4.2 Experiments on Face Recognition

Three well-known face database ORL[2], Yale[3] and the Extended Yale Face Database B [9] (denoted by YaleB hereafter) were used in our experiments.

The ORL database collects images from 40 individuals with 10 different images captured for each individual. For each individual, the images with different facial expressions and details are obtained at different times. Therefore, the face in the images may be rotated, scaled or tilted to a certain degree. Sample images of one individual from the ORL database are shown in Fig. 3.

There are a total of 165 gray-scale images for 15 individuals where each individual has 11 images in the Yale face database. The images demonstrate variations in lighting condition, facial expression (normal, happy, sad, sleepy, surprised, and wink). The sample images of one individual from the Yale database are shown in Fig. 4.

The YaleB database contains 21,888 images of 38 individuals under 9 poses and 64 illumination conditions. A subset containing 2,414 frontal pose images of 38 individuals

[2] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[3] http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

Fig. 5 Sample images of one individual from the YaleB database

under different illuminations per individual is extracted. Sample images of one individual from the YaleB database are shown in Fig. 5.

From each of the face database mentioned above, the image set is partitioned into the different gallery and probe sets. In this paper, the Gm/Pn indicates that m images per individual are randomly selected for training and the remaining n images are used for testing. For each partition, we use 50 random splits (20 random splits for YaleB) for cross-validation tests. All images are manually cropped and resized to 32×32 pixels. These cropped images and random splits can be downloaded from the Web.[4]

We investigate the performance of DTSA+ELM for face recognition. Two steps are adopted in face recognition: (1) dimensionality reduction; (2) classifier design. In this experiment, for dimensionality reduction, we apply the two algorithms TSA[5] and DTSA, which represent a 2D gray face image as a tensor. For classifier, we utilized the two classifiers nearest neighbor classifier and extreme learning machine. For TSA and DTSA, the heat kernel

[4] http://www.zjucadcg.cn/dengcai/Data/FaceData.html.

[5] The matlab code can be downloaded from http://www.zjucadcg.cn/dengcai/Data/data.html.

Table 1 Recognition accuracy (%) on ORL database (mean ± std)

Partitions	TSA + NNC	DTSA + NNC	TSA + ELM	DTSA + ELM
G2/P8	73.14 ± 3.06	83.76 ± 3.39	76.19 ± 2.77	86.85 ± 2.76
G3/P7	80.49 ± 2.59	91.61 ± 1.73	83.49 ± 2.13	93.89 ± 1.50
G4/P6	86.83 ± 1.84	94.55 ± 1.52	88.74 ± 1.59	96.76 ± 1.15
G5/P5	90.33 ± 1.80	96.68 ± 1.37	92.12 ± 1.66	98.35 ± 0.82
G6/P4	92.40 ± 1.75	97.46 ± 1.09	94.30 ± 1.78	99.19 ± 0.71
G7/P3	94.47 ± 2.10	98.03 ± 1.16	96.32 ± 1.66	99.47 ± 0.58
G8/P2	96.45 ± 1.70	98.60 ± 1.25	97.83 ± 1.53	99.78 ± 0.49

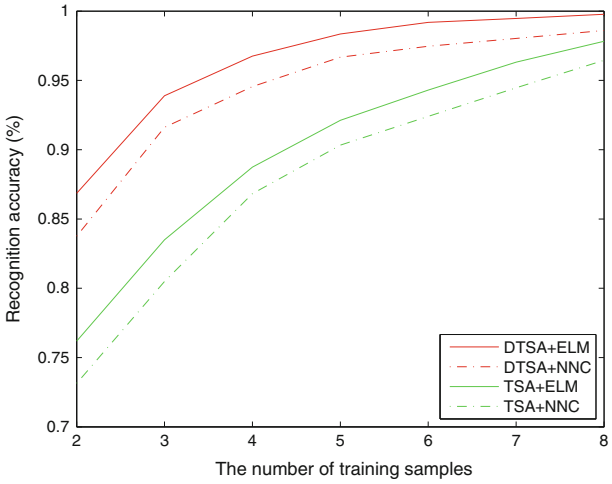


Fig. 6 Recognition accuracy (%) on ORL database (mean)

$\exp(-\|x - y\|^2/t)$ is used and t is set as 1,000. There are 10 iterations in both TSA and DTSA.

The first experiment is implemented on the ORL face database as described in Sect. 4.2. The maximum average recognition accuracy and the standard deviation across 50 runs of tests of each algorithm are shown in Table 1. The recognition accuracy curve versus the variation in size of training set is shown in Fig. 6. As we can see in Table 1, in every partition, the performance of DTSA + ELM are the highest, and the standard deviation of DTSA + ELM is the smallest. In situations where larger size of training set (*G6/P4*, *G7/P3* and *G8/P2*) are used, the recognition accuracy of DTSA + ELM may reach more than 99 %. As shown in Fig. 6, the two curves of DTSA are better than those of TSA. The results are constant with the theoretical analysis that the PCA subspace without discriminant information is not ideal for face recognition compared to LDA with discriminant information [43].

Compared to the ORL database, the Yale face database has different illuminations. The experimental setting is the same as that of the ORL database. The comparison results on the two databases are illustrated in Table 2. Figure 7 shows the recognition accuracy curves versus the variations of the size of training set. From Fig. 7, we can see that with smaller sizes of training set, the performance of DTSA is slightly worse than that of TSA. However,

Table 2 Recognition accuracy (%) on Yale database (mean)

Partitions	TSA + NNC	DTSA + NNC	TSA + ELM	DTSA + ELM
G2/P9	55.24 ± 4.03	52.00 ± 5.60	62.99 ± 3.95	58.07 ± 5.19
G3/P8	61.48 ± 4.12	63.47 ± 3.98	72.47 ± 3.46	71.58 ± 4.02
G4/P7	63.58 ± 3.49	68.69 ± 3.81	77.41 ± 3.28	78.90 ± 3.96
G5/P6	67.22 ± 3.42	71.09 ± 4.31	81.02 ± 2.23	83.36 ± 3.07
G6/P5	68.32 ± 3.63	74.21 ± 4.12	83.23 ± 2.72	85.68 ± 3.18
G7/P4	70.33 ± 3.79	76.47 ± 3.73	86.63 ± 2.58	88.53 ± 3.29
G8/P3	72.31 ± 4.76	80.00 ± 4.11	88.36 ± 3.22	91.56 ± 3.08

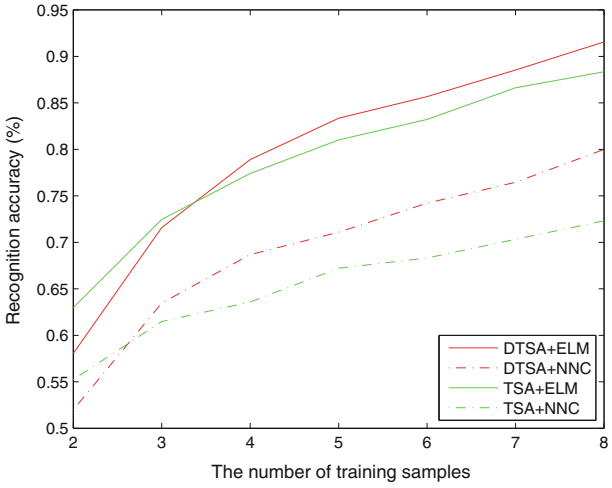


Fig. 7 Recognition accuracy (%) on Yale database (mean)

as the size of the training set increases, the performance of DTSA becomes better than those of TSA. This phenomenon occurs because as the size of the training set increases, there are more discriminant information in the training set. Therefore, the larger the size of the training set is, the better DTSA's performance is. As shown in Fig. 7, it can also be observed that as the size of the training set increases, the performance of ELM over NNC is becomes more and more obvious. The reason may lie in the fact that more discriminant information are needed for ELM in constructing an effective prediction model.

Furthermore, we implemented DTSA + ELM and DTSA + NNC on the more complex facial database Yale B. The results are illustrated in Table 3. Figure 8 shows that the performance of DTSA + ELM is superior to the ones of DTSA + NNC on this complex database. Moreover, we can see that the standard deviation of DTSA + ELM is smaller than that of DTSA + NNC from Table 3 (except for *G5/P55*).

4.3 Experiments on Micro-expression Recognition

The CASME database [40] includes 195 spontaneous facial micro-expressions recorded by two 60 fps cameras. These samples were selected from more than 1,500 facial expressions.

Table 3 Recognition accuracy (%) on YaleB database (mean ± std)

Partitions	DTSA + NNC	DTSA + ELM
G5/P55	72.06 ± 1.54	83.38 ± 1.58
G10/P50	82.97 ± 1.37	92.00 ± 1.16
G20/P40	90.22 ± 0.80	96.60 ± 0.52
G30/P30	93.19 ± 0.53	98.26 ± 0.33
G40/P20	94.97 ± 0.71	98.93 ± 0.35
G50/P10	95.92 ± 0.79	99.21 ± 0.34

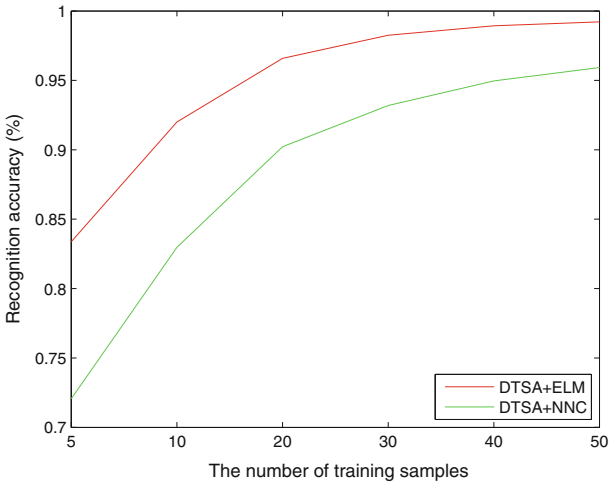


Fig. 8 Recognition accuracy (%) on YaleB database (mean ± std)

The selected micro-expressions either had a total duration less than 500 ms or an onset duration (time from onset frame to apex frame[6]) less than 250 ms. These samples are coded with the onset, apex and offset frames, furthermore tagged with action units (AUs) [8] and emotions (besides the six basic emotions according to Ekman [6], we also added other classes such as: attention, repression and tense, for the undefined AU combinations) (Fig. 9).

22 subjects (8 females, 14 males) participated in the study and had a mean age of 22.75 years (standard deviation: 2.01). The procedure of eliciting spontaneous micro-expressions is guided by psychologists. 17 video episodes were downloaded from the Internet, which were evaluated as highly positive or negative in valence. Since micro-expressions are presented when individuals try to conceal their emotions, we attempted to enhance their motivation of concealing emotions. The participants were firstly instructed that the purpose of the experiment was to test their ability to control emotions, which was highly related to their social success. The participants were also told that their payment was directly related to their performance. If they revealed any facial expressions during the experiment, 5 Chinese

[6] The onset is the first frame which changes from the baseline (usually neutral facial expressions). The apex is the one that reaches highest intensity of the facial expression. The offset is the last frame of the expression (before turning back to a neutral facial expression). Sometimes the facial expressions faded very slowly, and the changes between frames were very difficult to detect by eyes. For such offset frames, the coders only coded the last obvious frame as the offset frame while ignore the nearly imperceptible changing frames.

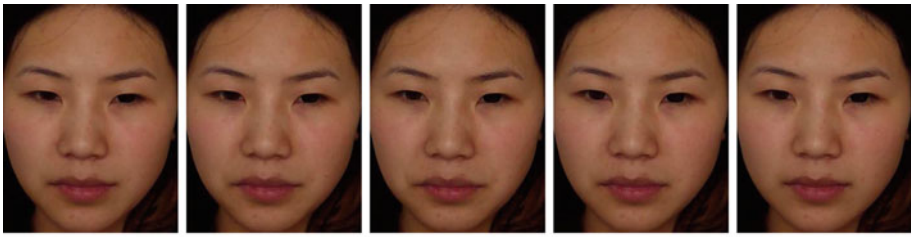


Fig. 9 An example of part of the frame sequence. The action unit is 15, which stands for lip corner depressor. **a** Onset frame; **b** a frame between (**a**) and (**c**); **c** apex frame; **d** a frame between (**c**) and (**e**); **e** offset frame

Table 4 Experimental results on micro-expression database with various optimal dimensionality

		*G*3	*G*6	*G*9	*G*12	*G*15
$20 \times 20 \times 20$	DTSA3+NNC	28.82	32.95	35.09	38.43	40.18
	DTSA3+ELM	30.17	34.69	38.60	40.76	43.33
$30 \times 30 \times 30$	DTSA3+NNC	30.35	34.34	36.71	41.11	41.07
	DTSA3+ELM	31.28	37.02	41.10	44.09	46.55
$40 \times 40 \times 40$	DTSA3+NNC	30.90	35.58	38.60	42.47	42.44
	DTSA3+ELM	30.21	37.48	40.92	43.64	46.90
$50 \times 50 \times 50$	DTSA3+NNC	30.21	35.50	39.96	42.42	42.44
	DTSA3+ELM	30.14	36.20	39.91	43.94	45.60
$60 \times 60 \times 60$	DTSA3+NNC	29.83	33.33	36.71	39.29	39.46
	DTSA3+ELM	29.79	33.80	38.55	42.32	43.93

Yuan (RMB) was deducted from their total payment each time as a punishment. In addition, they were not allowed to turn their eyes or head away from the screen while watching the video episodes.

From the CASME database, we selected 5 types of micro-expressions *attention*, *disgust*, *repression*, *surprise*, and *tense*. The micro-expression video set is partitioned into the different gallery and probe sets. *Gm* indicates that *m* samples per micro-expression are randomly selected for training and the remaining samples are used for testing. For each partition, we use 20 random splits for cross-validation tests. The CASME database provides color video clips of micro-expressions, in this paper we converted these color video clips into grey video clips. All samples are manually cropped and resized to $64 \times 64 \times 64$ pixels.

Micro-expression grey video clips can be treated as 3rd-order tensor. In this section, we investigate the performance of DTSA3+ELM for micro-expression recognition. The optimal dimensionality are $20 \times 20 \times 20, 30 \times 30 \times 30, 40 \times 40 \times 40, 50 \times 50 \times 50$ and $60 \times 60 \times 60$. Table 4 shows mean performances on these optimal dimensionality. The solid lines denote the performance of DTSA3+ELM, and the dotted lines denote the performance of DTSA3+NNC in Fig. 10. From the figure, we can see that the performance of DTSA3+ELM is superior to those of DTSA3+NNC on the CASME database.

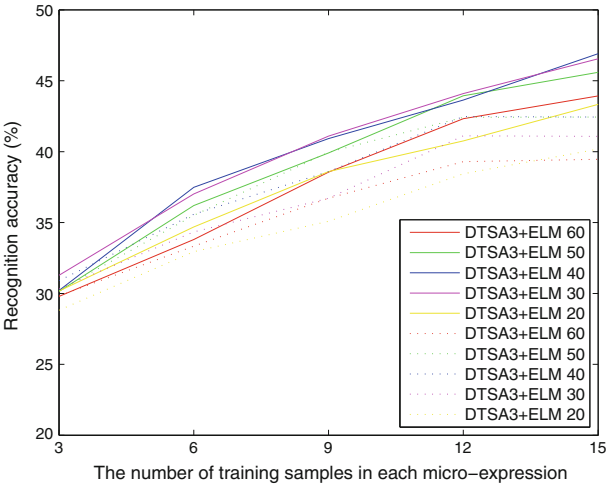


Fig. 10 Recognition accuracy (%) on CASME database (mean)

5 Conclusion

In this paper, we proposed an efficient recognition technique based on discriminant tensor subspace reduction dimensionality and extreme learning machine classifier. The 2D face images are first dimensionally reduced using DTSA to generate discriminant features, then the reduced features are fed into the ELM classifier to analytically learn an optimal model for recognition. In order to deal with micro-expression video clips, we extend DTSA to a high-order tensor. Experimental results on the ORL, Yale, YaleB face database and CASME micro-expression database show the efficiency of the proposed method. In addition, we also find that as the size of the training set increases, more discriminant information are obtained from the training set. Therefore, by increasing size of the training set, more benefits can be gained by both DTSA dimensionality reduction and ELM classification, especially on a complicated face database with various light variations.

Acknowledgments This work was supported in part by grants from 973 Program (2011CB302201), the National Natural Science Foundation of China (61075042, 61175023) and China Postdoctoral Science Foundation funded project (2012M520428).

References

1. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Machine Intell 19(7):711–720
2. Bian W, Tao D (2011) Max–min distance analysis by using sequential sdp relaxation for dimension reduction. IEEE Trans Pattern Anal Machine Intell 33(5):1037–1050
3. Chen H, Yang B, Liu J, Liu D (2011) A support vector machine classifier with rough set based feature selection for breast cancer diagnosis. Expert Syst Appl 38:9014–9022
4. Chen SB, Zhao HF, Kong M, Luo B (2007) 2D-LPP: a two-dimensional extension of locality preserving projections. Neurocomputing 70(4–6):912–921
5. Cortes C, Vapnik V (1995) Support-vector networks. Machine Learn 20(3):273–297
6. Ekman P (1992) An argument for basic emotions. Cogn Emotion 6(3–4):169–200
7. Ekman P, Friesen W (1969) Nonverbal leakage and clues to deception. Technical report, DTIC document

8. Ekman P, Friesen W, Hager J (2002) FACS: investigators guide. In: A Human Face. Research Nexus eBook, Salt Lake City
9. Georghiades A, Belhumeur P, Kriegman D (2001) From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans Pattern Anal Machine Intell 23(6):643–660
10. Guan N, Tao D, Luo Z, Shawe-Taylor J (2012) Mahnmf: Manhattan non-negative matrix factorization. arXiv, preprint arXiv:1207.3438. Accessed 12 Sept 2012
11. Guan N, Tao D, Luo Z, Yuan B (2011) Manifold regularized discriminative nonnegative matrix factorization with fast gradient descent. IEEE Trans Image Process 20(7):2030–2048
12. Guan N, Tao D, Luo Z, Yuan B (2011) Non-negative patch alignment framework. IEEE Trans Neural Netw 22(8):1218–1230
13. Guan N, Tao D, Luo Z, Yuan B (2012) Nenmf: an optimal gradient method for nonnegative matrix factorization. IEEE Trans Signal Process 60(6):2882–2898
14. Guan N, Tao D, Luo Z, Yuan B (2012) Online nonnegative matrix factorization with robust stochastic approximation. IEEE Trans Neural Netw Learn Syst 23(7):1087–1099
15. He X, Cai D, Niyogi P (2005) Tensor subspace analysis. In: Advances in neural information processing systems, vol 18. MIT Press, Cambridge
16. He XF, Niyogi P (2004) Locality preserving projections. In: Advances in neural information processing systems, vol 16. The MIT Press, Cambridge, pp 153–160
17. Huang G (2003) Learning capability and storage capacity of two-hidden-layer feedforward networks. IEEE Trans Neural Netw 14(2):274–281
18. Huang G, Babri H (1998) Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. IEEE Trans Neural Netw 9(1):224–229
19. Huang G, Chen L, Siew C (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17(4):879–892
20. Huang G, Zhu Q, Siew C (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: 2004 IEEE international joint conference on neural networks, vol 2. IEEE, New York, pp. 985–990
21. Huang G, Zhu Q, Siew C (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501
22. Joachims T (1998) Text categorization with support vector machines: Learning with many relevant features. Machine Learning. In: ECML-98, Springer Verlag, Heidelberg, pp 137–142
23. Li J, Tao D (2010) Simple exponential family PCA. J Mach Learn Res 9:453–460, Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, AISTATS 2010
24. Li M, Yuan BZ (2005) 2D-LDA: a statistical linear discriminant analysis for image matrix. Pattern Recogn Lett 26(5):527–532
25. Lu H, Plataniotis KN, Venetsanopoulos AN (2008) MPCA: multilinear principal component analysis of tensor objects. IEEE Trans Neural Netw 19(1):18–39
26. Van der Maaten L, Postma E, Van Den Herik H (2009) Dimensionality reduction: a comparative review. Tilburg University, Technical report
27. Nizar A, Dong Z, Wang Y (2008) Power utility nontechnical loss analysis with extreme learning machine method. IEEE Trans Power Syst 23(3):946–955
28. Osuna E, Freund R, Girosi F (1997) Training support vector machines: an application to face detection. In: 1997 IEEE computer society conference on computer vision and, pattern recognition, pp 130–136
29. Song F, Zhang D, Chen Q, Wang J (2007) Face recognition based on a novel linear discriminant criterion. Pattern Anal Appl 10(3):165–174
30. Tao D, Li X, Wu X, Hu W, Maybank S (2007) Supervised tensor learning. Knowl Inf Syst 13(1):1–42
31. Tao D, Li X, Wu, X Maybank S (2007) General tensor discriminant analysis and gabor features for gait recognition. IEEE Trans Pattern Anal Machine Intell 29:1700–1715
32. Tao D, Song M, Li X, Shen J, Sun J, Wu X, Faloutsos C, Maybank S (2008) Bayesian tensor approach for 3-d face modeling. IEEE Trans Circuits Syst Video Technol 18(10):1397–1410
33. Turk M, Pentland A (1991) Eigenfaces for recognition. J Cogn Neurosci 3(1):71–86
34. Vapnik V (2000) The nature of statistical learning theory. Springer Verlag, Berlin
35. Wang SJ, Yang J, Sun MF, Peng XJ, Sun MM, Zhou CG (2012) Sparse tensor discriminant color space for face verification. IEEE Trans Neural Netw Learn Syst 23(6):876–888
36. Wang SJ, Yang J, Zhang N, Zhou CG (2011) Tensor discriminant color space for face recognition. IEEE Trans Image Process 20(9):2490–2501
37. Wang SJ, Zhou CG, Zhang N, Peng XJ, Chen YH, Liu X (2011) Face recognition using second-order discriminant tensor subspace analysis. Neurocomputing 74(12–13):2142–2156
38. Xu Y, Feng G, Zhao YN (2009) One improvement to two-dimensional locality preserving projection method for use with face recognition. Neurocomputing 73(1–3):245–249

39. Yan S, Xu D, Yang Q, Zhang L, Tang X, Zhang HJ (2005) Discriminant analysis with tensor representation. Proc IEEE Comput Soc Conf Comput Vision Pattern Recogn 1:526–532
40. Yan WJ, Wu Q, Liu YJ, Wang SJ, FU X (2013) CASME Database: a dataset of spontaneous micro-expressions collected from neutralized faces. In: 10th IEEE conference on automatic face and gesture recognition, Shanghai
41. Yang J, Zhang D, Frangi AF, Yang JY (2004) Two-dimensional PCA: a new approach to appearance-based face representation and recognition. IEEE Trans Pattern Anal Machine Intell 26(1):131–137
42. Yu W (2009) Two-dimensional discriminant locality preserving projections for face recognition. Pattern Recogn Lett 30(15):1378–1383
43. Yu WW, Teng XL, Liu CQ (2006) Face recognition using discriminant locality preserving projections. Image Vis Comput 24(3):239–248
44. Zhang R, Huang G, Sundararajan N, Saratchandran P (2007) Multicategory classification using an extreme learning machine for microarray gene expression cancer diagnosis. IEEE/ACM Trans Comput Biol Bioinformatics 4(3):485–495
45. Zhang T, Tao D, Li X, Yang J (2008) Patch alignment for dimensionality reduction. IEEE Trans Knowl Data Eng 21:1299–1313
46. Zhou T, Tao D, Wu X (2010) Manifold elastic net: a unified framework for sparse dimension reduction. In: Data mining and knowledge discovery, Springer, Heidelberg, pp 1–32