Neurocomputing 214 (2016) 218–232

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Sparse tensor canonical correlation analysis for micro-expression recognition

Su-Jing Wang [a,*], Wen-Jing Yan [b], Tingkai Sun [c], Guoying Zhao [d], Xiaolan Fu [e]

[a] Key Laboratory of Behavior Sciences, Institute of Psychology, Chinese Academy of Sciences, Beijing 100101, China
[b] College of Teacher Education, Wenzhou University, Wenzhou 325035, China
[c] Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
[d] Center for Machine Vision Research, Department of Computer Science and Engineering, University of Oulu, P.O. Box 4500, FI-90014, Finland
[e] State Key Laboratory of Brain and Cognitive Science, Institute of Psychology, Chinese Academy of Sciences, Beijing 100101, China

ARTICLE INFO

Article history:
Received 1 March 2016
Received in revised form
14 May 2016
Accepted 25 May 2016
Available online 8 June 2016

Keywords:
Micro-expression recognition
Correlation analysis
Sparse representation
Tensor subspace

ABSTRACT

A micro-expression is considered a fast facial movement that indicates genuine emotions and thus provides a cue for deception detection. Due to its promising applications in various fields, psychologists and computer scientists, particularly those focus on computer vision and pattern recognition, have shown interest and conducted research on this topic. However, micro-expression recognition accuracy is still low. To improve the accuracy of such recognition, in this study, micro-expression data and their corresponding Local Binary Pattern (LBP) (Ojala et al., 2002) [1] code data are fused by correlation analysis. Here, we propose Sparse Tensor Canonical Correlation Analysis (STCCA) for micro-expression characteristics. A sparse solution is obtained by the regularized low rank matrix approximation. Experiments are conducted on two micro-expression databases, CASME and CASME 2, and the results show that STCCA performs better than the Three-dimensional Canonical Correlation Analysis (3D-CCA) without sparse resolution. The experimental results also show that STCCA performs better than three-order Discriminant Tensor Subspace Analysis (DTSA3) with discriminant information, smaller projected dimensions and a larger training set sample size. The experiments also showed that Multi-linear Principal Component Analysis (MPCA) is not suitable for micro-expression recognition because the eigenvectors corresponding to smaller eigenvectors are discarded, and those eigenvectors include brief and subtle motion information.

1. Introduction

A micro-expression is a brief facial expression that is typically less than 0.5 s in duration [2,3]. It can reveal an emotion that a person tries to conceal, especially in high-stakes situations [4,5]. Compared with ordinary facial expressions, micro-expressions have two significant characteristics: short duration and low intensity. The reputation of micro-expressions is derived from their potential practical applications as a cue of genuine emotions or lie detection in many fields, such as clinical diagnosis [6], national security [7], and interrogations [8]. Compared with polygraphs, lie detection based on micro-expressions is unobtrusive, and thus, the individuals being observed are less likely to develop countermeasures.

Micro-expressions were first discovered decades ago by Haggard and Isaacs. At the time, they were termed micro-momentary expressions and regarded as repressed (unconscious) emotions [9,10]. In 1969, Ekman and his colleague [11] analyzed video interviews of a depressed patient who attempted to commit suicide and found such a facial display, which was termed a micro-expression. From then on, several studies have focused on micro-expressions (mainly by Ekman's group) but few results have been published. According to Ekman [5], micro-expressions might be the most promising cue for deception detection.

However, human beings have difficulty detecting and recognizing micro-expressions. This difficulty may be due to their short duration, low intensity and fragmental action units [4,2]. Although there is a debate on the duration, the generally accepted upper limit duration is 0.5 s [2,3], which is considerably faster than ordinary facial expressions. Micro-expressions are typically extremely subtle because they are displayed with repression [2]. Additionally, micro-expressions typically only present part of the action units of full-stretched facial expressions; only the upper or lower face may show action units [11]. To improve human

* Corresponding author.
E-mail address: wangsujing@psych.ac.cn (S.-J. Wang).

http://dx.doi.org/10.1016/j.neucom.2016.05.083
0925-2312/

performance on recognizing micro-expressions, Ekman [12] developed the Micro-Expression Training Tool (METT), which trains people to better recognize micro-expressions in seven categories.[1] Computer scientists try to use computers to automatically recognize micro-expressions to better apply micro-expressions as a cue to reveal a person's emotions in practice.

Existing research on micro-expression recognition is rare. Polikovsky et al. [13] used a 3D-gradient descriptor for micro-expression recognition. Wang et al. [14] treated a micro-expression gray-scale video clip as a 3rd-order tensor and used Discriminant Tensor Subspace Analysis (DTSA) and an Extreme Learning Machine (ELM) to recognize micro-expressions. Pfister et al. [15] utilized a temporal interpolation model (TIM) [16] based on a Laplacian matrix to normalize the frame numbers of micro-expression video clips. In addition, the LBP-TOP [17] was used to extract the motion and appearance features of micro-expressions. The features are given to multiple kernel learning to recognize micro-expression. Huang et al. [18] proposed a Spatiotemporal Local Binary Pattern with Integral Projection (STLBP-IP), in which they used integral projection for extracting face shape information and subsequently employed 1-D and 2-D local binary pattern to face shape, for micro-expression recognition. They [19] also proposed Spatiotemporal Local Quantized Pattern (STCLQP), which exploits magnitude and orientation as complementary of sign information, for improving the performance of micro-expression recognition.

Gray video clips of micro-expressions are viewed as 3rd-order tensors, which are considered extensions of vectors and matrices. Tensors have been widely used [20–23]. The elements of a tensor are addressed by a number of indices [24], where the number of indices used in the description defines the order of the tensor object and each index defines one *mode* [25]. For micro-expression gray video clips, mode-1 and mode-2 represent the facial spatial information and mode-3 represents the temporal information. The characteristics of micro-expressions, such as short duration and subtlety, require micro-expressions to be captured by high-speed, high-resolution cameras. Therefore the captured micro-expression data are often in a high dimensional tensor space, and suffer from the so-called curse of dimensionality.[2] However, micro-expression data are typically highly constrained and belong to a subspace, a manifold of intrinsically low dimensions. Before analyzing high-dimensional data, their dimensionality must be reduced. Dimensionality reduction is the transformation of high-dimensional data into a lower dimensional data space. The tensor subspace dimensionality reduction has many advantages compared with the traditional vector subspace dimensionality reductions, including (1) keeping the spatial structure information of data, (2) avoiding the well-known small sample size (SSS) problem, and (3) lowering the computational complexity. Multi-linear Principal Component Analysis (MPCA) [26], a tensor version of PCA, applies PCA transformation to each mode (or dimensionality) of tensors. Similarly, Discriminant Analysis with Tensor Representation (DATER) [27], General Tensor Discriminant Analysis (GTDA) [28], Tensor Subspace Analysis (TSA) [29], and Discriminant Tensor Subspace Analysis (DTSA) [30] apply LDA, Maximum Scatter Difference (MSD) [31], LPP, and DLPP, respectively, to transform each mode of tensors. These methods use a certain vector subspace transformation method to transform every mode of tensors.

Pfister et al. [15] successfully used LBP-TOP [17] to extract the motion and appearance features of micro-expressions. In this paper, we seek a low dimensional subspace, in which the correlation between micro-expression data and their corresponding LBP code data is maximal. Canonical correlation analysis (CCA) [32] is an efficient method for seeking such a subspace. Moreover, some variations of the CCA method, such as kernel CCA (KCCA) [33] and Locality Preserving CCA (LPCCA) [34] were also developed in recent years. One approach analyzes the relationship between two tensor data sets rather than vector data sets. Tensor variations of the CCA have also been introduced. The two-dimensional CCA (2D-CCA) [35] is the first work along this line to analyze relations between two sets of image data without reshaping into vectors. It was further extended to local 2D-CCA [36] and 3D CCA (3D-CCA) [37]. However, CCA is generally sensitive to noise. In order to address the problem, Hardoon and Shawe-Taylor [38] used LASSO to propose Sparse CCA (SCCA). Yan et al. [39] used the low rank matrix approximation to propose Sparse 2D-CCA.

As noted above, micro-expressions exhibit two main characteristics: short duration and low intensity. Thus, micro-expression data are sparse in both the spatial and temporal domains. Accordingly, we will use sparse tensor analysis to reduce the dimensions of micro-expression data. Wang et al. [40] proposed Sparse Tensor PCA (STPCA) to recognize a face with occlusions. They [41] also proposed Sparse Tensor Discriminant Color Space (STDCS), in which the color of a face is recognized more robustly in noisy data sets.

Motivated by Sparse 2D-CCA, we proposed Sparse Tensor CCA (STCCA) in this paper. STCCA obtains a sparse solution via regularized low rank matrix approximation. STCCA is exploited to seek a subspace such that the correlation between micro-expression data and their corresponding LBP code data is maximal in the subspace. The micro-expression information in video clip is tiny. Some of these information are enhance in the corresponding LBP codes. So, STCCA can enhance the recognition accuracy of micro-expression.

The remainder of this paper is organized as follows. In Section 2, we provide the related definitions to tensors. In Section 3, we briefly review CCA and STCCA. In Section 4, the experiments are conducted with two micro-expression databases (CASME and CASME 2), and the results show that STCCA outperforms 3D-CCA and DTSA3. Finally in Section 5, conclusions are drawn and several issues for future works are described.

2. Tensor fundamentals

A tensor is a multidimensional array. It is the higher-order generalization of a scalar(zero-order tensor), vector(1st-order tensor), and matrix (2nd-order tensor). In this paper, lowercase italic letters (*a, b,* …) denote scalars, bold lowercase letters (**a**, **b**, …) denote vectors, bold uppercase letters (**A**, **B**, …) denote matrices, and calligraphic uppercase letters ($\mathcal{A}$, $\mathcal{B}$, …) denote tensors. The formal definition is given below [42,43]:

Definition 1. The *order* of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is N. An element of $\mathcal{A}$ is denoted by $\mathcal{A}_{i_1 i_2 \ldots i_N}$ or $a_{i_1 i_2 \ldots i_N}$, where $1 \le i_n \le I_n$, $n = 1, 2, \ldots, N$.

Definition 2. The *n-mode vectors* of $\mathcal{A}$ are the I_n-dimensional vectors obtained from $\mathcal{A}$ by fixing every index but index i_n.

Definition 3. The *n-mode unfolding matrix* of $\mathcal{A}$, denoted by $(\mathcal{A})_{(n)} \in \mathbb{R}^{I_n \times (I_1 \cdots I_{n-1} I_{n+1} \cdots \times I_N)}$, contains the element $a_{i_1 \ldots i_N}$ at the i_nth row and jth column, where

$$j = 1 + \sum_{k=1, k \neq n}^{N} (i_k - 1) J_k, \quad \text{with}$$

$$J_k = \begin{cases} 1, & \text{if } k = 1 \quad \text{or if } k = 2 \quad \text{and} \quad n = 1; \\ \prod_{m=1, m \neq n}^{k-1} I_m, & \text{otherwise.} \end{cases} \tag{1}$$

[1] Contempt was added in addition to the basic six emotions.

[2] The curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces (often with hundreds or thousands of dimensions) that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience.

We can generalize the product of two matrices to the product of a tensor and a matrix.

Definition 4. The *n-mode product* of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$, denoted by $\mathcal{A} \times_n \mathbf{U}$, is an $(I_1 \times I_2 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N)$-tensor, the entries of which are given by

$$(\mathcal{A} \times_n \mathbf{U})_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} \stackrel{\text{def}}{=} \sum_{i_n} a_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} u_{j_n i_n}. \tag{2}$$

Definition 5. The *scalar product* of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, denoted by $\langle \mathcal{A}, \mathcal{B} \rangle$, is defined in a straightforward manner as $\langle \mathcal{A}, \mathcal{B} \rangle \stackrel{\text{def}}{=} \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} a_{i_1 i_2 \ldots i_N} b_{i_1 i_2 \ldots i_N}$. The *Frobenius norm* of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is then defined as $\| \mathcal{A} \|_F \stackrel{\text{def}}{=} \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$

From the definition of the n-mode unfolding matrix, we have

$$\| \mathcal{A} \|_F = \| (\mathbf{A})_{(n)} \|_F \tag{3}$$

By using tensor decomposition, any tensor $\mathcal{A}$ can be expressed as the product

$$\mathcal{A} = C \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_N \mathbf{U}_N \tag{4}$$

where $\mathbf{U}_n$, $n = 1, 2, \ldots, N$, is an orthonormal matrix and contains the ordered principal components for the nth mode. C is called the *core tensor*. Unfolding the above equation, we have

$$\mathbf{A}_{(n)} = \mathbf{U}_n \mathbf{C}_{(n)} (\mathbf{U}_N \otimes \cdots \otimes \mathbf{U}_{n+1} \otimes \mathbf{U}_{n-1} \otimes \cdots \otimes \mathbf{U}_1)^T \tag{5}$$

where the operator $\otimes$is the Kronecker product of the matrices.

3. Sparse tensor canonical correlation analysis

3.1. Canonical correlation analysis

Consider two sets of vectors $\{\mathbf{x}_t \in \mathbb{R}^{D_x}, t = 1, \ldots, M\}$ and $\{\mathbf{y}_t \in \mathbb{R}^{D_y}, t = 1, \ldots, M\}$. Their mean vectors are denoted by $\bar{\mathbf{x}} = \frac{1}{M} \sum_{t=1}^{M} \mathbf{x}_t$ and $\bar{\mathbf{y}} = \frac{1}{M} \sum_{t=1}^{M} \mathbf{y}_t$, such that the centered vectors are represented by $\tilde{\mathbf{x}}_t = \mathbf{x}_t - \bar{\mathbf{x}}$ and $\tilde{\mathbf{y}}_t = \mathbf{y}_t - \bar{\mathbf{y}}$, respectively.

CCA seeks a pair of linear transforms $\mathbf{w}_x \in \mathbb{R}^{D_x}$ and $\mathbf{w}_y \in \mathbb{R}^{D_y}$, such that correlations between the transformed $\mathbf{w}_x^T \mathbf{x}$ and $\mathbf{w}_y^T \mathbf{y}$ are maximized. In other words, their Pearson's coefficients are maximized:

$$\frac{cov(\mathbf{w}_x^T \mathbf{x}, \mathbf{w}_y^T \mathbf{y})}{\sqrt{var(\mathbf{w}_x^T \mathbf{x}) var(\mathbf{w}_y^T \mathbf{y})}} = \frac{\mathbf{w}_x^T \mathbf{C}_{xy} \mathbf{w}_y}{\sqrt{(\mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x)(\mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y)}} \tag{6}$$

where $cov(,)$ and $var(,)$ are the mean covariance and variance, respectively. $\mathbf{C}_{xy} = \frac{1}{M} \sum_{t=1}^{M} \tilde{\mathbf{x}}_t \tilde{\mathbf{y}}_t^T$ represents the statistical expectation. Similarly, $\mathbf{C}_{xx} = \frac{1}{M} \sum_{t=1}^{M} \tilde{\mathbf{x}}_t \tilde{\mathbf{x}}_t^T$ and $\mathbf{C}_{yy} = \frac{1}{M} \sum_{t=1}^{M} \tilde{\mathbf{y}}_t \tilde{\mathbf{y}}_t^T$. Then, CCA criterion can be written as follows:

$$\underset{\mathbf{w}_x, \mathbf{w}_y}{\operatorname{argmax}} \mathbf{w}_x^T \mathbf{C}_{xy} \mathbf{w}_y \tag{7}$$

subject to $\mathbf{w}_x^T \mathbf{C}_{xx} \mathbf{w}_x = 1$ and $\mathbf{w}_y^T \mathbf{C}_{yy} \mathbf{w}_y = 1$. Its solution can be obtained by the following generalized eigenvalue problem:

$$\begin{bmatrix} 0 & \mathbf{C}_{xy} \\ \mathbf{C}_{yx} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_x \\ \mathbf{w}_y \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{C}_{xx} & 0 \\ 0 & \mathbf{C}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{w}_x \\ \mathbf{w}_y \end{bmatrix}, \tag{8}$$

where $\mathbf{C}_{yx} = \mathbf{C}_{xy}^T$.

3.2. Sparse tensor canonical correlation analysis

In this section, we will extend CCA from vector space to tensor space and make the transformed matrices sparse. We consider two sets of Nth-order tensors $\{\mathcal{X}_t \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}, t = 1, \ldots, M\}$ and $\{\mathcal{Y}_t \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}, t = 1, \ldots, M\}$.

Similarly, their mean tensors are defined by

$$\bar{\mathcal{X}} = \frac{1}{M} \sum_{t=1}^{M} \mathcal{X}_t, \quad \bar{\mathcal{Y}} = \frac{1}{M} \sum_{t=1}^{M} \mathcal{Y}_t, \tag{9}$$

and the centered tensors are defined by

$$\tilde{\mathcal{X}}_t = \mathcal{X}_t - \bar{\mathcal{X}}, \quad \tilde{\mathcal{Y}}_t = \mathcal{Y}_t - \bar{\mathcal{Y}}. \tag{10}$$

The STCCA seeks two transform sets $\{\mathbf{u}^{(1)} \in \mathbb{R}^{I_1}, \mathbf{u}^{(2)} \in \mathbb{R}^{I_2}, \ldots, \mathbf{u}^{(N)} \in \mathbb{R}^{I_N}\}$ and $\{\mathbf{v}^{(1)} \in \mathbb{R}^{J_1}, \mathbf{v}^{(2)} \in \mathbb{R}^{J_2}, \ldots, \mathbf{v}^{(N)} \in \mathbb{R}^{J_N}\}$ such that correlations between $x = \mathcal{X}_t \times_1 \mathbf{u}^{(1)} \times_2 \mathbf{u}^{(2)} \cdots \times_N \mathbf{u}^{(N)}$ and $y = \mathcal{Y}_t \times_1 \mathbf{v}^{(1)} \times_2 \mathbf{v}^{(2)} \cdots \times_N \mathbf{v}^{(N)}$ are maximized and $\mathbf{u}^{(n)}$ and $\mathbf{v}^{(n)}$ are sparse. Here, sparsity indicates that $\mathbf{u}^{(n)}$ and $\mathbf{v}^{(n)}$ have only a small number of nonzero elements or numerous zero elements. Therefore, the criterion function of STCCA is defined as:

$$\begin{aligned} &\arg\max_{\mathbf{u}_n, \mathbf{v}_n} cov(x, y) \\ &\text{s. t.} \quad var(x) = 1, var(y) = 1, \\ &Card(\mathbf{u}^{(n)}), Card(\mathbf{v}^{(n)}) < K_n \end{aligned} \tag{11}$$

where $Card(\cdot)$ denotes the number of nonzero elements of $\mathbf{u}^{(n)}$ and $\mathbf{v}^{(n)}$. Given two transform sets with $N-1$ vectors $\{\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(n-1)}, \mathbf{u}^{(n+1)}, \ldots, \mathbf{u}^{(N)}\}$ and $\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n-1)}, \mathbf{v}^{(n+1)}, \ldots, \mathbf{v}^{(N)}\}$, we denote

$$\mathbf{x}^{(n)} = \mathcal{X} \times_1 \mathbf{u}^{(1)} \cdots \times_{n-1} \mathbf{u}^{(n-1)} \times_{n+1} \mathbf{u}^{(n+1)} \cdots \times_N \mathbf{u}^{(N)} \tag{12}$$

and

$$\mathbf{y}^{(n)} = \mathcal{Y} \times_1 \mathbf{v}^{(1)} \cdots \times_{n-1} \mathbf{v}^{(n-1)} \times_{n+1} \mathbf{v}^{(n+1)} \cdots \times_N \mathbf{v}^{(N)} \tag{13}$$

Then, we define

$$\mathbf{C}_{xy}^{(n)} = \frac{1}{M} \sum_{t=1}^{M} \tilde{\mathbf{x}}_t^{(n)} \tilde{\mathbf{y}}_t^{(n)T}, \tag{14}$$

$$\mathbf{C}_{xx}^{(n)} = \frac{1}{M} \sum_{t=1}^{M} \tilde{\mathbf{x}}_t^{(n)} \tilde{\mathbf{x}}_t^{(n)T}, \tag{15}$$

and

$$\mathbf{C}_{yy}^{(n)} = \frac{1}{M} \sum_{t=1}^{M} \tilde{\mathbf{y}}_t^{(n)} \tilde{\mathbf{y}}_t^{(n)T}, \tag{16}$$

where, $\tilde{\mathbf{x}}_t^{(n)}$ means the result of Eq. (12) for the tth sample $\mathcal{X}_t$ Thus, we have

$$cov^{(n)}(x, y) = \mathbf{u}^{(n)T} \mathbf{C}_{xy}^{(n)} \mathbf{v}^{(n)} \tag{17}$$

Based on the above notations, Eq. (11) can be rewritten as

$$\begin{aligned} &\arg\max_{\mathbf{u}_n, \mathbf{v}_n} \mathbf{u}^{(n)T} \mathbf{C}_{xy}^{(n)} \mathbf{v}^{(n)} \\ &\text{s.t.} \quad \mathbf{u}^{(n)T} \mathbf{C}_{xx}^{(n)} \mathbf{u}^{(n)} = 1, \mathbf{v}^{(n)T} \mathbf{C}_{yy}^{(n)} \mathbf{v}^{(n)} = 1, \\ &Card(\mathbf{u}^{(n)}), Card(\mathbf{v}^{(n)}) < K_n \end{aligned} \tag{18}$$

Without the condition of $Card(\mathbf{u}^{(n)}), Card(\mathbf{v}^{(n)}) < K_n$, the Lagrangian of Eq. (18) is

$$L = \mathbf{u}^{(n)T} \mathbf{C}_{xy}^{(n)} \mathbf{v}^{(n)} - \lambda_1 (\mathbf{u}^{(n)T} \mathbf{C}_{xx}^{(n)} \mathbf{u}^{(n)} - 1) - \lambda_2 (\mathbf{v}^{(n)T} \mathbf{C}_{yy}^{(n)} \mathbf{v}^{(n)} - 1) \tag{19}$$

Taking the partial derivative of L with respect to $\mathbf{u}^{(n)}$ and $\mathbf{v}^{(n)}$ and setting them equal to zeros, we have

$$\mathbf{C}_{xy}^{(n)}\mathbf{v}^{(n)} = \lambda_1\mathbf{C}_{xx}^{(n)}\mathbf{u}^{(n)} \tag{20}$$

$$\mathbf{C}_{yx}^{(n)}\mathbf{u}^{(n)} = \lambda_2\mathbf{C}_{yy}^{(n)}\mathbf{v}^{(n)} \tag{21}$$

where $\mathbf{C}_{yx}^{(n)} = \mathbf{C}_{xy}^{(n)T}$. Replacing $\mathbf{v}^{(n)}$ in Eq. (21) with Eq. (20), we have

$$\mathbf{C}_{xy}^{(n)}\mathbf{C}_{yy}^{(n)-1}\mathbf{C}_{yx}^{(n)}\mathbf{u}^{(n)} = \lambda_1\lambda_2\mathbf{C}_{xx}^{(n)}\mathbf{u}^{(n)} \tag{22}$$

Additionally replacing $\mathbf{u}^{(n)}$ in Eq. (20) with Eq. (21), we obtain

$$\mathbf{C}_{yx}^{(n)}\mathbf{C}_{xx}^{(n)-1}\mathbf{C}_{xy}^{(n)}\mathbf{v}^{(n)} = \lambda_1\lambda_2\mathbf{C}_{yy}^{(n)}\mathbf{v}^{(n)} \tag{23}$$

Thus, $\mathbf{u}^{(n)}$ and $\mathbf{v}^{(n)}$ can be obtained by solving two generalized eigenvalues problems.

However, $\mathbf{u}^{(n)}$ and $\mathbf{v}^{(n)}$ are not sparse. In order to make $\mathbf{u}^{(n)}$ and $\mathbf{v}^{(n)}$ sparse, we use firstly the regularized SVD on $\mathbf{C}^{(n)}$. For convenience, we denote

$$\mathbf{C}^{(n)} = \mathbf{C}_{xx}^{(n)-\frac{1}{2}}\mathbf{C}_{xy}^{(n)}\mathbf{C}_{yy}^{(n)-\frac{1}{2}}. \tag{24}$$

Using SVD on $\mathbf{C}^{(n)}$, we have

$$\mathbf{C}^{(n)} = \mathbf{U}^{(n)}\mathbf{D}^{(n)}\mathbf{V}^{(n)T} = \sum_{i=1}^{r} d_i\mathbf{u}_i^{(n)}\mathbf{v}_i^{(n)T} \tag{25}$$

where, $d_1^2 \geq d_2^2 \geq \cdots \geq d_r^2 > 0$ are the positive eigenvalues of $\mathbf{C}^{(n)}\mathbf{C}^{(n)T}$. $\mathbf{u}_i^{(n)}$ and $\mathbf{v}_i^{(n)}$ are the eigenvectors of $\mathbf{C}^{(n)}\mathbf{C}^{(n)T}$ and $\mathbf{C}^{(n)T}\mathbf{C}^{(n)}$, respectively.

To make $\mathbf{u}_i^{(n)}$ and $\mathbf{v}_i^{(n)}$ sparse, we define the closest rank-l matrix approximation of $\mathbf{C}^{(n)}$ as follows:

$$\mathbf{C}^{(n)\{l\}} = \sum_{i=1}^{l} d_i\mathbf{u}_i^{(n)}\mathbf{v}_i^{(n)T} \tag{26}$$

where $l \leq r$ and the term *closest* means that the squared Frobenius norm between $\mathbf{C}^{(n)}$ and $\mathbf{C}^{(n)\{l\}}$ is minimal, where the Frobenius norm is defined as

$$\| \mathbf{C}^{(n)} - \mathbf{C}^{(n)\{l\}} \|_F^2 = \mathrm{tr}\{(\mathbf{C}^{(n)} - \mathbf{C}^{(n)\{l\}})(\mathbf{C}^{(n)} - \mathbf{C}^{(n)\{l\}})^T\} \tag{27}$$

Then, the rank-1 matrix approximation of $\mathbf{C}^{(n)}$ can be formulated as solving the following optimization form:

$$\min_{\tilde{\mathbf{u}}^{(n)},\tilde{\mathbf{v}}^{(n)}} \| \mathbf{C}^{(n)} - \tilde{\mathbf{u}}^{(n)}\tilde{\mathbf{v}}^{(n)T} \|_F^2 \tag{28}$$

Then, the low rank approximation property of SVD implies that the solution is

$$\tilde{\mathbf{u}}^{(n)} = \mathbf{u}_1^{(n)}, \quad \tilde{\mathbf{v}}^{(n)} = d_1\mathbf{v}_1^{(n)}. \tag{29}$$

The subsequent pairs $(\mathbf{u}_i^{(n)}, d_i\mathbf{v}_i^{(n)})$, $i > 1$, provide the best rank one approximations of the corresponding residual matrices. For example, $d_2\mathbf{u}_2^{(n)}\mathbf{v}_2^{(n)T}$ is the best rank one approximation of $\mathbf{C}^{(n)} - d_1\mathbf{u}_1^{(n)}\mathbf{v}_1^{(n)T}$.

To make $\tilde{\mathbf{u}}^{(n)}$ and $\tilde{\mathbf{v}}^{(n)}$ sparse, ℓ_1-norm penalties are imposing on Eq. (28),

$$\min_{\tilde{\mathbf{u}}^{(n)},\tilde{\mathbf{v}}^{(n)}} \| \mathbf{C}^{(n)} - \tilde{\mathbf{u}}^{(n)}\tilde{\mathbf{v}}^{(n)T} \|_F^2 + \lambda_u \| \tilde{\mathbf{u}}^{(n)} \|_1 + \lambda_v \| \tilde{\mathbf{v}}^{(n)} \|_1 \tag{30}$$

where $\lambda_u > 0$ and $\lambda_v > 0$ are tradeoff parameters.

Given a fixed $\tilde{\mathbf{u}}^{(n)}$ and supposing $\| \tilde{\mathbf{u}}^{(n)} \| = 1$, Eq. (30) can be written as

$$\begin{aligned}
&\min_{\tilde{\mathbf{v}}^{(n)}} \sum_{i,j=1}^{I_n} \left(C_{ij}^{(n)} - \tilde{u}_i^{(n)}\tilde{v}_j^{(n)} \right)^2 + \lambda_u \| \tilde{\mathbf{u}}^{(n)} \|_1 + \lambda_v \sum_{j=1}^{I_n} |\tilde{v}_j^{(n)}| \\
&= \min_{\tilde{\mathbf{v}}^{(n)}} \sum_{j=1}^{I_n} \left(\sum_{i=1}^{I_n} C_{ij}^{(n)2} - 2\sum_{i=1}^{I_n} C_{ij}^{(n)}\tilde{u}_i^{(n)}\tilde{v}_j^{(n)} + \sum_{i=1}^{I_n} \tilde{u}_i^{(n)2}\tilde{v}_j^{(n)2} \right) + \lambda_v \sum_{j=1}^{I_n} |\tilde{v}_j^{(n)}| \\
&\quad + \lambda_u \| \tilde{\mathbf{v}}^{(n)} \|_1 = \min_{\tilde{\mathbf{v}}^{(n)}} \sum_{j=1}^{I_n} \left(\sum_{i=1}^{I_n} C_{ij}^{(n)2} - 2(\mathbf{C}^{(n)T}\tilde{\mathbf{u}}_j^{(n)})\tilde{v}_j^{(n)} + \tilde{v}_j^{(n)2} + \lambda_v|\tilde{v}_j^{(n)}| \right) \\
&\quad + \lambda_u \| \tilde{\mathbf{u}}^{(n)} \|_1.
\end{aligned} \tag{31}$$

Thus, we only need to minimize $\sum_{j=1}^{I_n}(\tilde{v}_j^{(n)2} - 2(\mathbf{C}^{(n)T}\tilde{\mathbf{u}}_j^{(n)})\tilde{v}_j^{(n)} + \lambda_v|\tilde{v}_j^{(n)}|)$. The equation can be solved by Lemma 1.

Lemma 1. *Let* $\hat{\beta}_j$ *be the minimizer of* $\beta_j^2 - 2\alpha_j\beta_j + \lambda|\beta_j|$, $(\lambda > 0, j = 1, 2, \ldots, n)$. *Then,* $\hat{\beta}_j = \mathrm{sign}(\alpha_j)\max\{0, |\alpha_j| - \frac{\lambda}{2}\}$.

According to the lemma, the solution of Eq. (31) can be written as follows

$$\tilde{\mathbf{v}}^{(n)} = \mathrm{sign}(\mathbf{C}^{(n)T}\tilde{\mathbf{u}}^{(n)})\max\left\{ \mathbf{0}_{I_n}, |\mathbf{C}^{(n)T}\tilde{\mathbf{u}}^{(n)}| - \frac{\lambda_v}{2}\mathbf{1}_{I_n} \right\} \tag{32}$$

where $\mathbf{0}_{I_n}$ and $\mathbf{1}_{I_n}$ are the $I_n \times 1$ vectors with all entries equal to 0 and 1, respectively. Similarly, given a fixed $\tilde{\mathbf{v}}^{(n)}$ and supposing $\| \tilde{\mathbf{v}}^{(n)} \| = 1$, we have

$$\tilde{\mathbf{u}}^{(n)} = \mathrm{sign}(\mathbf{C}^{(n)}\tilde{\mathbf{v}}^{(n)})\max\left\{ \mathbf{0}_{I_n}, |\mathbf{C}^{(n)}\tilde{\mathbf{v}}^{(n)}| - \frac{\lambda_u}{2}\mathbf{1}_n \right\}. \tag{33}$$

Thus, the optimal $\tilde{\mathbf{u}}^{(n)}$ and $\tilde{\mathbf{v}}^{(n)}$ can be obtained by iteratively computing Eqs. (33) and (32).

Thus, we obtain

$$\mathbf{U}^{(n)} = \left\{ \mathbf{u}_1^{(n)}, \mathbf{u}_2^{(n)}, \ldots, \mathbf{u}_{L_n}^{(n)} \right\} \tag{34}$$

and

$$\mathbf{V}^{(n)} = \left\{ \mathbf{v}_1^{(n)}, \mathbf{v}_2^{(n)}, \ldots, \mathbf{v}_{L_n}^{(n)} \right\}. \tag{35}$$

The sparse transforms $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ can now be obtained by the above two equations. Because $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ depend on $\{\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(n-1)}, \mathbf{U}^{(n+1)}, \ldots, \mathbf{U}^{(N)}\}$ and $\{\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(n-1)}, \mathbf{V}^{(n+1)}, \ldots, \mathbf{V}^{(N)}\}$, respectively, the optimization of $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ depends on the projections in other modes. An iterative procedure can be constructed to maximize Eq. (11). We use $\{\mathbf{u}_{l_1}^{(1)}, \mathbf{u}_{l_2}^{(2)}, \ldots, \mathbf{u}_{l_N}^{(N)}\}$ and $\{\mathbf{v}_{l_1}^{(1)}, \mathbf{v}_{l_2}^{(2)}, \ldots, \mathbf{v}_{l_N}^{(N)}\}$ to project $\mathcal{X}$ and $\mathcal{Y}$ into the $(l_1, l_2, \ldots, l_N)$ feature dimension, which is determined by the l_nth column vector in the nth mode projected matrices $(n = 1, 2, \ldots, n)$:

$$x^{(l_1,l_2,\ldots,l_N)} = \mathcal{X} \times_1 \mathbf{u}_{l_1}^{(1)} \times_2 \mathbf{u}_{l_2}^{(2)} \times_3 \cdots \times_N \mathbf{u}_{l_N}^{(N)} \tag{36}$$

and

$$y^{(l_1,l_2,\ldots,l_N)} = \mathcal{Y} \times_1 \mathbf{v}_{l_1}^{(1)} \times_2 \mathbf{v}_{l_2}^{(2)} \times_3 \cdots \times_N \mathbf{v}_{l_N}^{(N)}. \tag{37}$$

The Pearson coefficient $\rho^{(l_1,l_2,\ldots,l_N)}$ in the $(l_1, l_2, \ldots, l_N)$ feature dimension is calculated by

$$\rho^{(l_1,l_2,\ldots,l_N)} = \frac{cov(x^{(l_1,l_2,\ldots,l_N)}, y^{(l_1,l_2,\ldots,l_N)})}{\sqrt{var(x^{(l_1,l_2,\ldots,l_N)})var(y^{(l_1,l_2,\ldots,l_N)})}}. \tag{38}$$

The aim of STCCA is to seek the sparse projected matrices to maximize these Pearson coefficients. The sum of these Pearson coefficients is denoted by

$$J = \sum_{l_1=1}^{L_1}\sum_{l_2=1}^{L_2}\cdots\sum_{l_N=1}^{L_N} \rho^{(l_1,l_2,\ldots,l_N)}. \tag{39}$$

We use the difference of J between the iteration and the previous iteration as the convergence criterion. The pseudocode of the

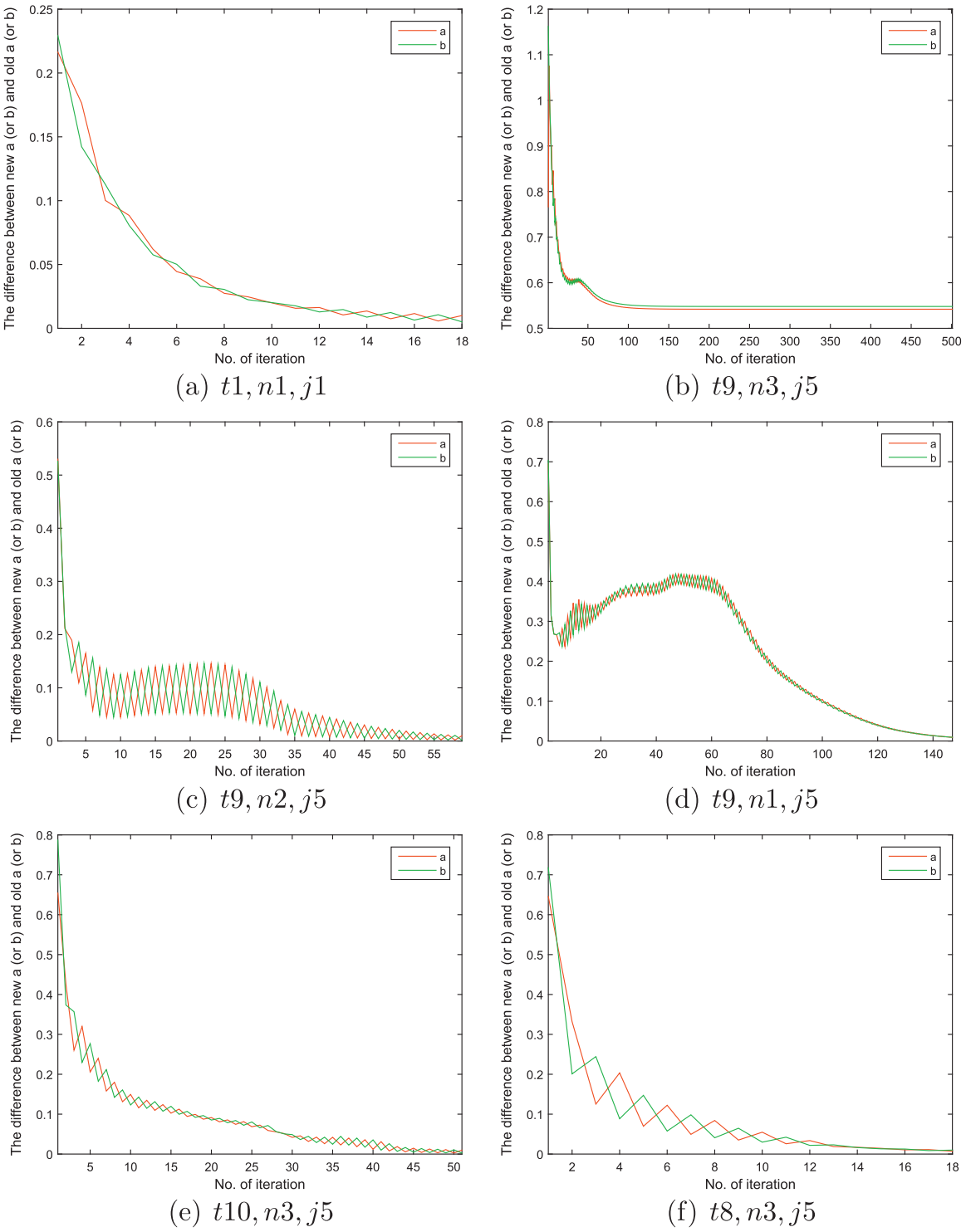


Fig. 1. Illustration of the convergence of the second iterative loop. t indicates the times of the first iterative loop. n and j are two variable at line 7 and line 12 in Algorithm 1, respectively.

proposed method is summarized in Algorithm 1.

Algorithm 1. STCCA.

1: **INPUT**: two sets of M tensor samples $\{\mathcal{X}_t \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}\}$ and $\{\mathcal{Y}_t \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_N}\}$, the number of reduced dimensions L_n, $n = 1, 2, \ldots, N$ and sparse tuning parameters λ_u, λ_v.

2: **OUTPUT**: two sets of sparse transformation matrices $\{\mathbf{U}_n \in \mathbb{R}^{I_n \times L_n}\}$ and $\{\mathbf{V}_n \in \mathbb{R}^{J_n \times L_n}\}$, where $n = 1, 2, \ldots, N$.

3: **Algorithm**:

4: Initialize $\{\mathbf{U}_n\}$ and $\{\mathbf{V}_n\}$ with identity matrices;

5: Calculate the centered tensors $\tilde{\mathcal{X}}_t$ and $\tilde{\mathcal{Y}}_t$ by Eqs. (9) and (10);

6: **repeat**

for $n=1$ to N **do**
  Calculate $\mathcal{X}^{(n)}$ and $\mathcal{Y}^{(n)}$ by Eqs. (12) and (13);
  Calculate $\mathbf{C}_{xy}^{(n)}$, $\mathbf{C}_{xx}^{(n)}$, and $\mathbf{C}_{yy}^{(n)}$ by Eqs. (14), (15), and (16), respectively;
  Calculate $\mathbf{C}^{(n)}$ by Eq. (24);
  Do SVD on $\mathbf{C}^{(n)} = \mathbf{UDV}^T = \sum_{i=1}^{L_n} d_i \mathbf{u}_i \mathbf{v}_i^T$;
  for $j=1$ to L_n **do**
    repeat

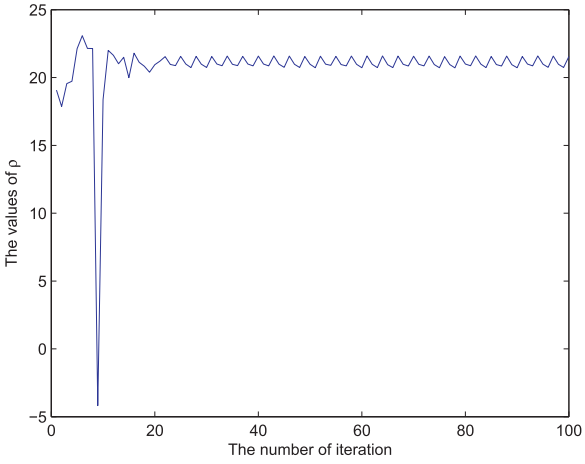


Fig. 2. Illustration of the convergence of the first iterative loop.

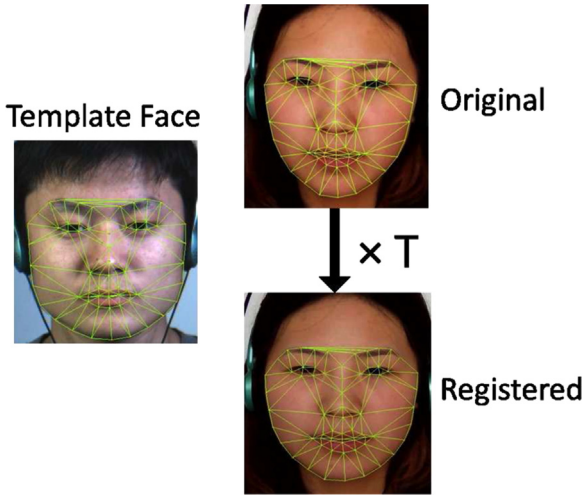


Fig. 3. The process of micro-expression registration.

Table 1
the numbers of samples in CASME database.

Micro-expressions	*Ha*	*Di*	*Su*	*Te*	*Re*
Number of samples	10	44	20	69	38

    $\tilde{\mathbf{u}}_{old} = \mathbf{u}_j$ and $\tilde{\mathbf{v}}_{old} = d_j \mathbf{v}_j$;
    $\tilde{\mathbf{v}}_{new} = \text{sign}(\mathbf{C}^{(n)T}\tilde{\mathbf{u}}_{old})\max\{\mathbf{0}_{J_n}, |\mathbf{C}^{(n)T}\tilde{\mathbf{u}}_{old}| - \frac{\lambda_v}{2}\mathbf{1}_{J_n}\}$;
    Normalize $\tilde{\mathbf{v}}_{new} = \tilde{\mathbf{v}}_{new}/\| \tilde{\mathbf{v}}_{new} \|$;
    $\tilde{\mathbf{u}}_{new} = \text{sign}(\mathbf{C}^{(n)}\tilde{\mathbf{v}}_{old})\max\{\mathbf{0}_{I_n}, |\mathbf{C}^{(n)}\tilde{\mathbf{v}}_{old}| - \frac{\lambda_u}{2}\mathbf{1}_{I_n}\}$;
    Normalize $\tilde{\mathbf{u}}_{new} = \tilde{\mathbf{u}}_{new}/\| \tilde{\mathbf{u}}_{new} \|$;
  until $\| \tilde{\mathbf{u}}_{new} - \tilde{\mathbf{u}}_{old} \| < \epsilon_u$ and $\| \tilde{\mathbf{v}}_{new} - \tilde{\mathbf{v}}_{old} \| < \epsilon_v$
  $\mathbf{u}_j^{(n)} = \tilde{\mathbf{u}}_{new}/\| \tilde{\mathbf{u}}_{new} \|$; $\mathbf{v}_j^{(n)} = \tilde{\mathbf{v}}_{new}/\| \tilde{\mathbf{v}}_{new} \|$
  update $\mathbf{C}^{(n)} = \mathbf{C}^{(n)} - \mathbf{u}_j^{(n)T}\mathbf{C}^{(n)}\mathbf{v}_j^{(n)}\mathbf{u}_j^{(n)}\mathbf{v}_j^{(n)T}$;
  end for
  $\mathbf{U}^{(n)} = [\mathbf{u}_1^{(n)}, \mathbf{u}_2^{(n)}, \ldots, \mathbf{u}_{L_n}^{(n)}]$;
  $\mathbf{V}^{(n)} = [\mathbf{v}_1^{(n)}, \mathbf{v}_2^{(n)}, \ldots, \mathbf{v}_{L_n}^{(n)}]$;
end for
Calculate J by Eq. (39)
until $\| J - J_{pre} \| < \epsilon$

3.3. Feature extraction and classification

For each sample $\mathcal{X}_t$ and its LBP code $\mathcal{Y}_t$, we project them into the feature space by

$$\mathcal{X}_t^{feature} = \mathcal{X}_t \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \cdots \times_N \mathbf{U}^{(N)} \tag{40}$$

and

$$\mathcal{Y}_t^{feature} = \mathcal{Y}_t \times_1 \mathbf{V}^{(1)} \times_2 \mathbf{V}^{(2)} \times_3 \cdots \times_N \mathbf{V}^{(N)} \tag{41}$$

Then, $\mathcal{X}_t^{feature}$ and $\mathcal{Y}_t^{feature}$ are vectorized and concatenated to form the final feature vector. Here, $\mathbf{f}_i^c$ is denoted as the feature vector of ith sample in cth class.

For a test sample $\mathcal{X}_{test}$ and its LBP code $\mathcal{X}_{test}$, we obtain its feature vector $\mathbf{f}_{test}$ by using the above method. We choose the nearest-neighbor classifier which uses the Euclidean distance to measure the similarity and identify the target label of each test sample:

$$d(\mathbf{f}_{test}, \mathbf{f}_i^c) = \| \mathbf{f}_{test} - \mathbf{f}_i^c \|_F \tag{42}$$

The $\mathcal{X}_{test}$ is assigned to cth class according to its closest class center $\mathbf{f}_i^c$.

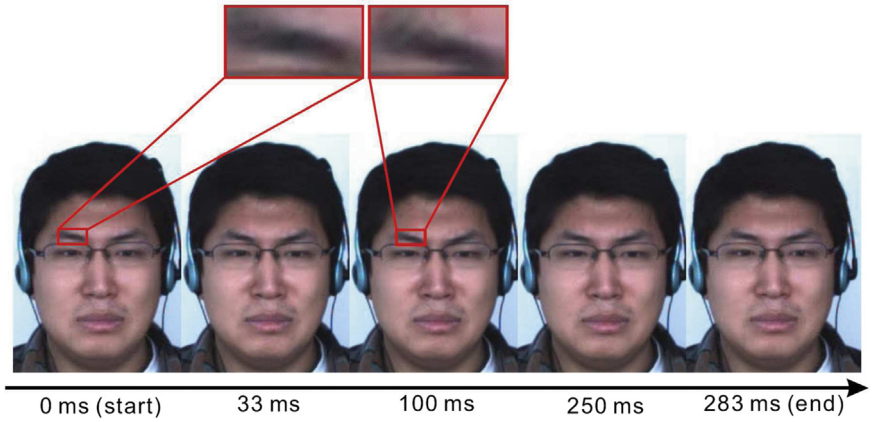


Fig. 4. A demonstration of the frame sequence in a micro-expression in CASME. The apex frame presents at approximately 100 ms.

Table 2
Micro-expression mean recognition accuracies (%) of STCCA, 3D-CCA, MPCA, and DTSA3 in CASME. (mean ± std. Bold numbers denote the highest recognition accuracies and `teletype numbers` denote the lowest recognition accuracies.)

q	Methods	Classifiers	5 × 5 × 5	10 × 10 × 10	20 × 20 × 20	30 × 30 × 30	40 × 40 × 40	50 × 50 × 50	60 × 60 × 60	Baseline
$q = 3$	STCCA	NNC	**33.31 ± 6.08**	**34.25 ± 8.06**	**35.00 ± 6.72**	**34.13 ± 7.50**	33.98 ± 7.51	34.07 ± 7.62	34.31 ± 7.70	32.68 ± 7.76
		SVM	21.14 ± 0.27	21.08 ± 0.00	21.08 ± 0.00	21.36 ± 0.53	24.49 ± 3.07	27.98 ± 5.28	29.85 ± 6.32	
	3D-CCA	NNC	21.54 ± 3.22	29.40 ± 7.16	24.43 ± 9.19	22.80 ± 11.33	22.08 ± 10.68	24.46 ± 9.97	27.83 ± 10.00	
		SVM	21.84 ± 5.55	29.61 ± 6.76	23.89 ± 10.27	22.95 ± 6.60	24.43 ± 7.85	28.10 ± 8.84	29.10 ± 8.35	
	MPCA	NNC	19.85 ± 3.39	19.25 ± 3.65	16.20 ± 12.16	16.90 ± 11.85	15.18 ± 11.90	14.19 ± 11.70	22.08 ± 10.40	
		SVM	20.42 ± 7.13	20.00 ± 7.92	18.34 ± 8.28	17.98 ± 8.24	17.47 ± 8.74	17.11 ± 8.94	16.14 ± 7.94	
	DTSA3	NNC	25.54 ± 3.13	26.57 ± 4.28	28.86 ± 6.39	32.47 ± 6.78	**36.17 ± 7.93**	**37.59 ± 7.23**	**37.53 ± 8.04**	
		SVM	24.46 ± 4.06	24.64 ± 4.43	26.84 ± 6.87	29.67 ± 7.14	31.23 ± 8.64	32.77 ± 8.68	33.28 ± 8.82	
$q = 4$	STCCA	NNC	**35.62 ± 5.80**	**36.12 ± 7.06**	**35.71 ± 7.28**	**35.81 ± 7.12**	35.90 ± 7.38	35.84 ± 7.34	35.84 ± 7.36	35.37 ± 7.40
		SVM	21.12 ± 0.00	21.12 ± 0.00	21.12 ± 0.00	21.77 ± 0.82	26.02 ± 2.40	30.84 ± 3.79	31.86 ± 5.15	
	3D-CCA	NNC	25.37 ± 4.85	31.96 ± 6.05	26.96 ± 10.00	26.06 ± 12.20	26.34 ± 11.22	29.94 ± 9.95	31.83 ± 7.16	
		SVM	22.92 ± 6.13	29.47 ± 8.46	27.39 ± 7.15	27.20 ± 5.17	29.75 ± 5.94	29.94 ± 6.51	28.57 ± 6.77	
	MPCA	NNC	20.75 ± 3.16	20.81 ± 2.98	15.25 ± 12.14	15.96 ± 13.12	15.90 ± 12.63	17.83 ± 14.10	23.88 ± 9.36	
		SVM	21.37 ± 6.98	23.35 ± 5.96	23.85 ± 8.86	21.74 ± 9.99	21.77 ± 10.44	23.35 ± 8.78	20.68 ± 9.38	
	DTSA3	NNC	25.43 ± 3.22	27.30 ± 3.61	30.34 ± 4.98	34.78 ± 4.01	**38.66 ± 4.58**	**40.40 ± 5.18**	**40.47 ± 5.51**	
		SVM	23.60 ± 5.37	24.47 ± 6.12	25.56 ± 7.54	28.14 ± 7.08	30.71 ± 6.51	33.23 ± 7.19	33.29 ± 5.97	
q=5	STCCA	NNC	**36.06 ± 5.56**	**37.85 ± 5.40**	**37.15 ± 5.69**	**36.28 ± 5.78**	36.35 ± 5.52	36.31 ± 5.62	36.47 ± 5.69	35.51 ± 5.49
		SVM	21.19 ± 0.14	21.15 ± 0.00	21.15 ± 0.00	21.67 ± 0.53	26.35 ± 1.81	30.16 ± 3.27	32.44 ± 4.46	
	3D-CCA	NNC	27.28 ± 4.49	32.63 ± 4.64	26.70 ± 11.23	25.67 ± 10.90	31.47 ± 9.55	36.60 ± 7.19	37.21 ± 5.54	
		SVM	25.77 ± 6.33	30.67 ± 6.23	24.17 ± 8.21	24.29 ± 7.60	27.02 ± 7.40	31.83 ± 6.48	32.56 ± 6.33	
	MPCA	NNC	19.97 ± 2.98	19.78 ± 4.44	17.15 ± 15.60	15.90 ± 16.17	14.13 ± 13.70	19.10 ± 14.80	23.62 ± 12.68	
		SVM	22.69 ± 5.08	22.53 ± 6.05	22.37 ± 8.14	21.89 ± 10.17	22.12 ± 10.06	23.46 ± 7.68	16.73 ± 10.37	
	DTSA3	NNC	26.76 ± 4.07	27.47 ± 3.99	30.00 ± 4.10	33.91 ± 4.26	**37.18 ± 5.51**	**39.39 ± 6.01**	**39.17 ± 6.37**	
		SVM	24.94 ± 6.66	22.79 ± 5.59	25.54 ± 6.55	28.94 ± 7.15	30.96 ± 8.57	32.82 ± 8.72	33.56 ± 8.91	
q=6	STCCA	NNC	**38.15 ± 5.54**	**37.68 ± 5.72**	**38.25 ± 5.09**	**36.49 ± 5.58**	36.52 ± 5.55	37.02 ± 5.44	36.92 ± 5.84	35.40 ± 5.58
		SVM	21.19 ± 0.00	21.13 ± 1.99	21.19 ± 0.00	22.12 ± 1.15	25.96 ± 2.76	30.46 ± 5.25	32.68 ± 6.49	
	3D-CCA	NNC	28.28 ± 4.57	35.50 ± 5.92	23.11 ± 12.89	22.85 ± 11.62	28.25 ± 11.61	33.74 ± 7.65	35.13 ± 6.31	
		SVM	25.17 ± 5.97	34.93 ± 6.12	27.15 ± 6.82	28.08 ± 6.23	30.70 ± 7.33	34.83 ± 7.65	35.76 ± 6.33	
	MPCA	NNC	20.43 ± 3.43	20.33 ± 3.96	9.44 ± 9.79	**10.03 ± 12.52**	**12.91 ± 14.53**	19.14 ± 17.30	23.54 ± 13.72	
		SVM	22.38 ± 5.22	25.33 ± 5.60	27.81 ± 6.73	26.52 ± 8.98	24.14 ± 7.80	24.07 ± 6.36	23.77 ± 7.19	
	DTSA3	NNC	26.42 ± 4.59	28.38 ± 3.47	31.75 ± 5.65	34.04 ± 5.09	**37.65 ± 6.07**	**39.97 ± 5.15**	**40.53 ± 5.99**	
		SVM	25.86 ± 7.23	24.01 ± 7.57	23.48 ± 6.72	25.73 ± 6.54	29.67 ± 8.30	31.56 ± 7.75	33.01 ± 8.14	
q=7	STCCA	NNC	**40.07 ± 5.06**	**41.20 ± 5.14**	**40.65 ± 5.61**	**39.59 ± 5.43**	39.76 ± 5.48	39.52 ± 5.64	39.76 ± 5.51	38.77 ± 6.05
		SVM	21.68 ± 1.84	20.79 ± 1.99	21.23 ± 0.00	22.12 ± 0.89	27.16 ± 2.87	32.50 ± 3.51	35.96 ± 5.10	
	3D-CCA	NNC	32.74 ± 4.80	38.39 ± 5.51	21.95 ± 11.66	21.92 ± 10.78	29.38 ± 10.44	37.43 ± 4.84	39.14 ± 3.63	
		SVM	27.57 ± 6.32	34.21 ± 5.40	25.65 ± 8.31	25.45 ± 9.04	31.64 ± 9.43	37.50 ± 6.46	38.39 ± 6.20	
	MPCA	NNC	**20.79 ± 3.38**	21.54 ± 4.18	11.95 ± 13.04	8.97 ± 11.51	10.03 ± 12.45	14.42 ± 14.12	23.29 ± 9.38	
		SVM	25.10 ± 5.64	25.00 ± 5.46	23.97 ± 5.93	25.62 ± 7.22	24.18 ± 8.10	26.10 ± 6.50	18.84 ± 11.00	
	DTSA3	NNC	28.46 ± 3.86	30.21 ± 4.91	34.14 ± 4.96	37.05 ± 4.47	**41.30 ± 5.02**	**42.95 ± 4.76**	**42.57 ± 4.92**	
		SVM	27.95 ± 7.38	25.79 ± 7.28	21.88 ± 7.23	25.34 ± 9.37	26.58 ± 11.37	31.64 ± 10.28	31.27 ± 9.53	

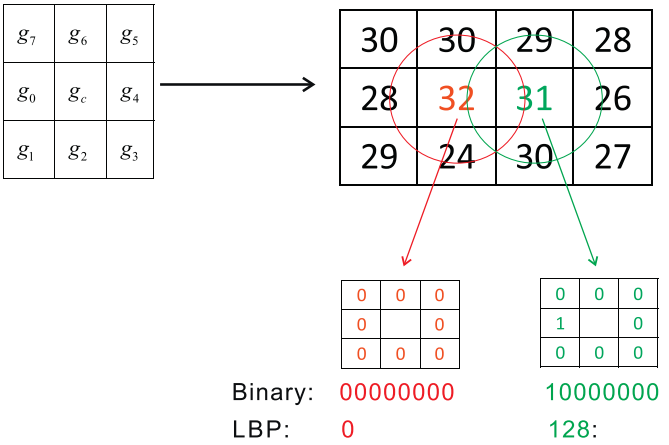


Fig. 5. A demonstration of small difference in gray value and large difference in LBP code. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

3.4. Discussion on convergence

There are two iterative loops in Algorithm 1. One is from line 6 to line 27, and another is from line 13 to line 19. We first discuss the convergence of the second iterative loop. We used all of the data in Section 4.2 as the training data to investigate the convergence behavior. The convergence threshold ϵ_u and ϵ_v are set as 0.01. The sparse tuning parameters λ_u and λ_v are set as 0.05. The number of reduced dimensions L_1, L_2 and L_3 are set as 5.

Fig. 1 illustrates the convergence of the second iterative loop. The loop is not converging in the $(t9, n3, j5)$ case. However, this does not influence the convergence of the corresponding next loop (the $(t10, n3, j5)$ case). Among the six cases illustrated in Fig. 1, except for the $(t9, n3, j5)$ case, the second iterative loop always converges. The loop does not converge in several cases. However, this does not influence the final result. In practice, we set a maximal iterative time to address the non-convergence case. The maximal iterative time is set at 500.

The first iterative loop is a general framework for tensor subspace analysis. In reality, the convergence of many tensor subspace analysis algorithms cannot be proven; therefore, the classification results based on these algorithms are shown to be stable after rounds of iterations as illustrated in these previous papers (e.g., DATER [27], 2D LDA [44]) [45]. Here, we demonstrate the convergence behavior in Fig. 2. After more than 30 iterations, the value of ρ shows periodic oscillation around the a larger value.

Table 3
The results of t-test between STCCA and DTSA3 in CASME.

(p,t)	$5\times5\times5$	$10\times10\times10$	$20\times20\times20$	$30\times30\times30$	$40\times40\times40$	$50\times50\times50$	$60\times60\times60$
$q=3$	**(0.000, 6.73)**	**(0.000, 5.11)**	**(0.000, 4.71)**	(0.195, 1.34)	(0.084, −1.82)	(0.007, −3.02)	(0.008, −2.98)
$q=4$	**(0.000, 8.85)**	**(0.000, 6.15)**	**(0.000, 4.79)**	(0.447, 0.78)	(0.037, −2.25)	**(0.001, −3.94)**	**(0.001, −4.02)**
$q=5$	**(0.000, 8.43)**	**(0.000, 8.04)**	**(0.000, 8.52)**	(0.019, 2.56)	(0.467, −0.74)	(0.014, −2.72)	(0.042, −2.18)
$q=6$	**(0.000, 8.63)**	**(0.000, 6.21)**	**(0.000, 4.77)**	(0.053, 2.07)	(0.359, −0.94)	(0.016, −2.64)	**(0.002, −3.60)**
$q=7$	**(0.000, 9.70)**	**(0.000, 9.33)**	**(0.000, 6.39)**	**(0.004, 3.26)**	(0.044, −2.16)	**(0.001, −3.75)**	**(0.001, −3.95)**

4. Experiments

4.1. Data preprocessing

To address the large variations in the spatial appearance of faces, all faces are normalized to a template face M, which is a frontal face image with a neutral expression. First, the template face is marked with 68 landmarks (ψ_M) using the Active Shape Model (ASM) [46]. Second, the first frame of a sample micro-expression clip was marked with 68 landmarks ψ_{f1}, and the 2D geometric transformation of the template face is estimated as $\psi_M = T\psi_{f1}$, where T is the transformation matrix. Third, the remaining frames are registered to the template face by applying the transformation T. Because there is only slight head movement in the video clip, the transformation T can be used in all of the frames in the same video clip. The sizes of each frame of samples are normalized to 163×134 pixels. Fig. 3 illustrates the registered process.

4.2. CASME

The Chinese Academy of Sciences Micro-Expression (CASME) database [47,48] includes 195 spontaneous facial micro-expressions recorded by two 60 fps cameras. These samples were selected from more than 1500 facial expressions. The selected micro-expressions either had a total duration less than 500 ms or an onset duration (time from onset frame to apex frame[3]) less than 250 ms. These samples are coded with the onset, apex and offset frames and are tagged with the action units (AUs) [49]. In this database, micro-expressions are classified into 7 categories (happiness, surprise, disgust, fear, sadness, repression and tense). An example is shown in Fig. 4.

We selected only 5 classes of the micro-expressions (happiness (*Ha*), surprise (*Su*), disgust (*Di*), repression (*Re*) and tense (*Te*)), amounting to 181 samples for the following experiments, because the number of samples of fear and sadness was insufficient. Table 1 lists the sample numbers of each selected class.

In these samples, the frame number of the shortest and longest samples are 10 and 68, respectively. The frame numbers of all samples are normalized to 70 using linear interpolation. The size of each frame is normalized to 163×134 pixels. Thus, each sample was normalized to a third-order tensor with a size of $163\times134\times70$. To utilize the 3D-CCA and STCCA methods, LBP-TOP is used on the micro-expression samples to obtain corresponding LBP video clips. The radii of axes X and Y were set as 1, and the radius of axis T was set as 2. The numbers of neighboring points in the XY, XT and YT planes were all set as 4. The basic LBP was used in the LBP coding. Thus, each LBP video clip may be treated as a third-order tensor with a size of $161\times132\times66$. We perform STCCA and 3D-CCA on the micro-expression video clips and their corresponding LBP video clips, and project them to $5\times5\times5$, $10\times10\times10$, $20\times20\times20$, $30\times30\times30$, $40\times40\times40$, $50\times50\times50$, and $60\times60\times60$ dimensions, respectively. We used two classifiers: Nearest Neighbor Classifier (NNC) and Support Vector Machine (SVM). The RBF kernel is adopted for the SVM.[4] MPCA and DTSA3 are also used to reduce the dimensions of the micro-expression video clips in this experiment. We randomly split the micro-expression samples so that q (q = 3, 4, 5, 6, 7) samples for each class are used as the training set, and the remaining samples are used as the testing set. This process is repeated 20 times. The mean recognition accuracies and standard deviations are listed in Table 2. Table 2 also lists the baseline, which is calculated by concatenated the samples and their LBP codes without dimensionality reduction. The accuracies of the baselines are almost lower 5% then the highest accuracies of the dimensionality reduction methods. Furthermore, the baselines need huge memory to calculate. For CASME 2, calculating the baseline leads to out of memory.

DTSA3 showed the best performances with larger projected dimensions. DTSA3 is a discriminant reduced dimension method and the discriminant information can further improve the recognition accuracy. However, STCCA obtains higher recognition accuracies than DTSA3 when the projected dimensions are smaller. STCCA outperforms DTSA3 for larger training sets. MPCA typically exhibit the worst performances. Thus, MPCA is not suitable for micro-expression recognition. This poor performance occurs because the eigenvectors corresponding to smaller eigenvectors are discarded, and these eigenvectors include the brief and subtle motion information. So, MPCA is not conducted on the following experiments. When solving STCCA, the eigenvectors corresponding to smaller eigenvectors are also discarded. Why does STCCA suit to recognizing micro-expression. The reason is there is a great difference in LBP codes between two small difference gray values. In Fig. 5, the difference between red point and green point is only 1. The corresponding difference in LBP code is 128. So, the eigenvectors corresponding to smaller eigenvectors in STCCA maybe not include the brief and subtle motion information of micro-expression.

STCCA and DTSA3 almost are the two best performances on each case. Here, we use t-test to perform statistical significance test between STCCA and DTSA3 on different q and reductional dimension cases. Table 3 lists (p, t) of t-test. When $p < 0.005$, (p, t) is Bold in the table. When the reductional dimensions are smaller, the performances of STCCA are significance better than those of DTSA3. Although the performance of DTSA3 is better than those of STCCA on the larger projected dimensions, these better performances are not always significance.

Fig. 6 shows the bar graphs of these mean recognition accuracies. For STCCA, 3DCCA and DTSA3, the performances of NNC are better than that of SVM. Thus, in the following experiments, we only use NNC as a classifier. With the increasing projected dimensions, the performance of DTSA3 is dramatically better in the NNC cases, perhaps because the eigenvectors of DTSA3 contain more discriminant information. The performance of STCCA is not particularly sensitive to the projected dimensions, which is why STCCA can achieve a good performance with relatively small

[3] The onset is the first frame, which changes from the baseline (typical neutral facial expressions). The apex is the frame that reaches the highest intensity of the facial expression. The offset is the last frame of the expression (before turning back into a neutral facial expression). In some cases, the facial expressions faded slowly, and the changes between frames were difficult to detect by the naked eye. For such offset frames, the coders only coded the last obvious frame as the offset frame while ignoring the nearly imperceptible changing frame(s).

[4] We used LIBSVM. The parameter is set as '$-t$ 2'

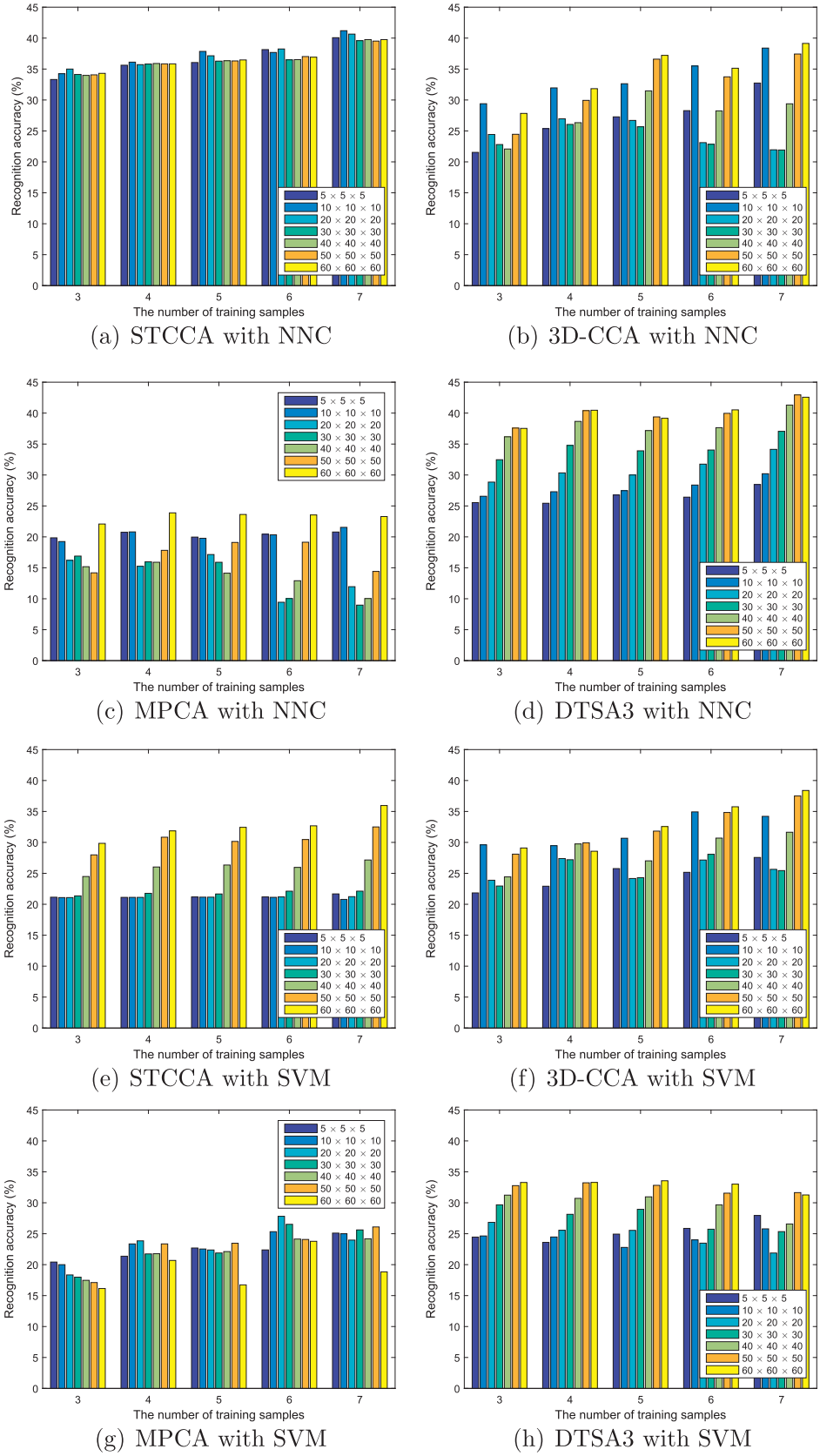


Fig. 6. Bar graphs of micro-expression mean recognition accuracies in CASME.

projected dimensions.

We also investigate the computational efficiencies of STCCA, 3D-CCA, MPCA and DTSA3. Fig. 7 illustrates the amount time required to project to dimensions of $5 \times 5 \times 5$. DTSA3 and MPCA are drastically more time-consuming than 3D-CCA and STCCA. 3D-CCA is more efficient than STCCA because STCCA has a sparsity

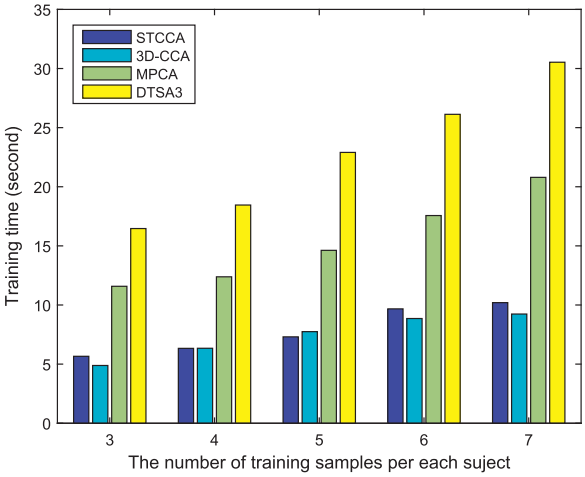


Fig. 7. The training times for training models on CASME.

procedure which can be time-consuming.

Now, some researchers [50,51] used optical flow to extract features of micro-expressions. Optical flow infers the motion of objects by detecting the changing intensity of pixels between two image frames over time. In a video clip, a pixel at location (x, y, t) with intensity $I(x, y, t)$ will have moved by Δx, Δy and Δt between the two frames. According to the brightness constancy constraint, we have

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{43}$$

Assuming that the movement is small, the image constraint at $I(x, y, t)$ can be developed with a Taylor series to obtain:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + \tau \tag{44}$$

where τ is a higher-order infinitesimal. From these equations it follows that:

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0 \tag{45}$$

and

$$\frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t}\frac{\Delta t}{\Delta t} = 0 \tag{46}$$

which results in

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0 \tag{47}$$

where V_x and V_y are the x and y components, respectively, of the velocity or optical flow of $I(x, y, t)$. Thus, between two frames with distance Δt, the optical flow value of a pixel at time t is expressed as a two-dimensional vector:

$$[V_x^t, V_y^t]^T \tag{48}$$

Many methods can be used to compute the optical flow field [52]. In our implementation, we use the method presented in [53].

Given a micro-expression video clip with $163 \times 134 \times 70$ size, we calculated its optical flow and got two 3rd-order tensors with $163 \times 134 \times 69$ size $\mathcal{U}$ and $\mathcal{V}$. STCCA is conducted on $\mathcal{U}$ and $\mathcal{V}$. Table 4 lists the results of STCCA between the optical flow features $\mathcal{U}$ and $\mathcal{V}$. The performances of STCCA between raw video clip and corresponding its LBP code are better than those between the optical flow features $\mathcal{U}$ and $\mathcal{V}$.

For the problem that the classes are not balanced in CASME, we use the F_1 score to address it. The F_1 score is defined as follows:

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{49}$$

Suppose there are C class, cp_c is the number of correct positive results of cth class, ap_c is the number of all positive results of cth class and rp_c is the number of positive results that should have been returned of c class (See Fig. 8). *Precision* and *Recall* are defined as follows:

$$Precision = \frac{1}{C}\sum_{c=1}^{C}\frac{cp_c}{ap_c} \tag{50}$$

and

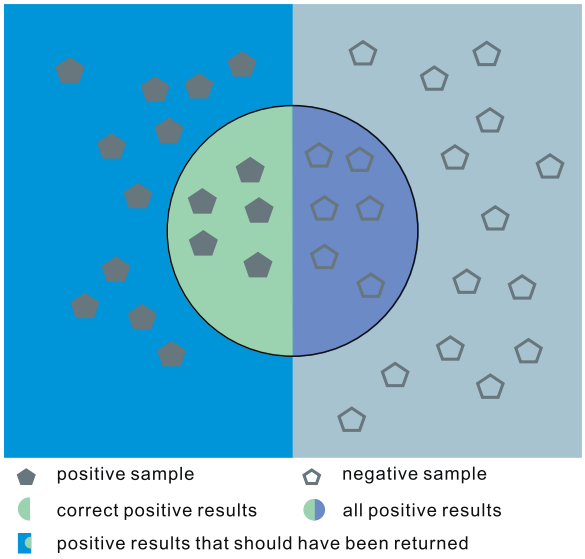


Fig. 8. A demonstration about the F_1 score.

Table 4
Micro-expression mean recognition accuracies (%) of STCCA in CASME. (mean ± std. RL means the result of STCCA between raw video clip and corresponding its LBP code. OF means the result of STCCA between the optical flow features $\mathcal{U}$ and $\mathcal{V}$.)

q	Codes	5 × 5 × 5	10 × 10 × 10	20 × 20 × 20	30 × 30 × 30	40 × 40 × 40	50 × 50 × 50	60 × 60 × 60
$q=3$	RL	**33.31 ± 6.08**	**34.25 ± 8.06**	**35.00 ± 6.72**	**34.13 ± 7.50**	**33.98 ± 7.51**	**34.07 ± 7.62**	**34.31 ± 7.70**
	OF	23.52 ± 4.37	23.67 ± 7.14	24.76 ± 9.78	25.45 ± 9.66	26.20 ± 9.58	26.78 ± 9.74	27.83 ± 9.25
$q=4$	RL	**35.62 ± 5.80**	**36.12 ± 7.06**	**35.71 ± 7.28**	**35.81 ± 7.12**	**35.90 ± 7.38**	**35.84 ± 7.34**	**35.84 ± 7.36**
	OF	23.98 ± 6.58	24.78 ± 8.44	24.41 ± 9.52	23.82 ± 9.46	24.75 ± 9.98	24.53 ± 9.82	24.47 ± 9.78
$q = 5$	RL	**36.06 ± 5.56**	**37.85 ± 5.40**	**37.15 ± 5.69**	**36.28 ± 5.78**	**36.35 ± 5.52**	**36.31 ± 5.62**	**36.47 ± 5.69**
	OF	22.21 ± 5.16	22.79 ± 8.16	25.13 ± 8.27	25.64 ± 8.99	25.45 ± 9.80	25.74 ± 10.04	26.92 ± 10.60
$q=6$	RL	**38.15 ± 5.54**	**37.68 ± 5.72**	**38.25 ± 5.09**	**36.49 ± 5.58**	**36.52 ± 5.55**	**37.02 ± 5.44**	**36.92 ± 5.84**
	OF	24.70 ± 6.20	25.07 ± 8.36	27.72 ± 9.26	27.72 ± 9.52	28.48 ± 10.47	27.65 ± 10.53	28.05 ± 11.36
$q=7$	RL	**40.07 ± 5.06**	**41.20 ± 5.14**	**40.65 ± 5.61**	**39.59 ± 5.43**	**39.76 ± 5.48**	**39.52 ± 5.64**	**39.76 ± 5.51**
	OF	23.08 ± 6.52	24.69 ± 8.87	25.17 ± 9.74	25.24 ± 11.12	24.45 ± 11.54	24.66 ± 12.10	24.83 ± 12.25

Table 5
The F_1 scores of STCCA, 3D-CCA, MPCA, and DTSA3 in CASME. (Bold numbers denote the highest scorers.)

q	Methods	5 × 5 × 5	10 × 10 × 10	20 × 20 × 20	30 × 30 × 30	40 × 40 × 40	50 × 50 × 50	60 × 60 × 60
$q=3$	STCCA	**0.1217**	**0.1255**	**0.1282**	**0.1251**	0.1245	0.1250	0.1258
	3D-CCA	0.0804	0.1064	0.0797	0.0582	0.0636	0.0849	0.1039
	MPCA	0.0730	0.0702	0.0386	0.0367	0.0320	0.0286	0.0497
	DTSA3	0.0925	0.0970	0.1033	0.1184	**0.1318**	**0.1372**	**0.1382**
$q=4$	STCCA	**0.1320**	**0.1324**	**0.1316**	**0.1318**	0.1323	0.1322	0.1321
	3D-CCA	0.0933	0.1141	0.0797	0.0659	0.0806	0.1012	0.1144
	MPCA	0.0775	0.0781	0.0370	0.0325	0.0306	0.0314	0.0428
	DTSA3	0.0940	0.0999	0.1120	0.1283	**0.1431**	**0.1493**	**0.1503**
$q=5$	STCCA	**0.1358**	**0.1407**	**0.1403**	**0.1376**	**0.1376**	0.1378	0.1384
	3D-CCA	0.1036	0.1222	0.0820	0.0610	0.1023	0.1300	0.1351
	MPCA	0.0753	0.0753	0.0409	0.0365	0.0331	0.0368	0.0494
	DTSA3	0.0995	0.1036	0.1134	0.1263	**0.1376**	**0.1442**	**0.1443**
$q=6$	STCCA	**0.1428**	**0.1419**	**0.1434**	**0.1351**	0.1358	0.1377	0.1374
	3D-CCA	0.1035	0.1300	0.0683	0.0696	0.0930	0.1261	0.1322
	MPCA	0.0753	0.0749	0.0217	0.0226	0.0233	0.0391	0.0477
	DTSA3	0.0991	0.1053	0.1196	0.1266	**0.1391**	**0.1478**	**0.1505**
$q=7$	STCCA	**0.1492**	**0.1491**	**0.1519**	**0.1475**	0.1488	0.1474	0.1484
	3D-CCA	0.1184	0.1411	0.0560	0.0531	0.0963	0.1366	0.1448
	MPCA	0.0790	0.0798	0.0320	0.0208	0.0224	0.0318	0.0478
	DTSA3	0.1046	0.1132	0.1265	0.1383	**0.1536**	**0.1580**	**0.1589**

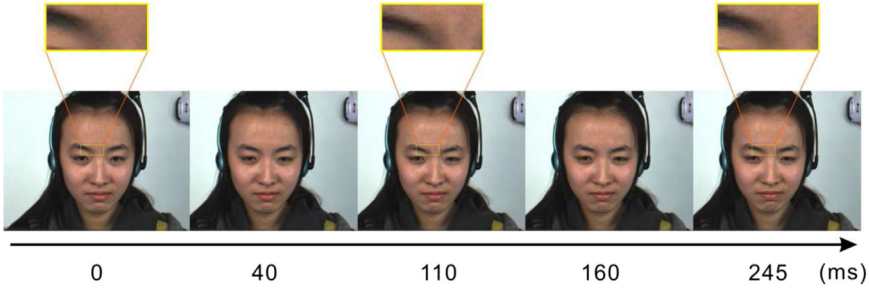


Fig. 9. A demonstration of the frame sequence in a micro-expression in CASME 2. The three rectangles above the images show the right inner brow (AU 4) in *zoom in* mode. The movement is more obvious in video play than in picture sequence.

Table 6
The numbers of samples in CASME 2 database.

Micro-expressions	*Ha*	*Di*	*Su*	*Te*	*Re*
Number of samples	32	60	25	102	27

$$Recall = \frac{1}{C}\sum_{c=1}^{C}\frac{cp_c}{rp_c} \tag{51}$$

The F_1 score reaches its best value at 1 and worst at 0. Table 5 lists the F_1 scores of STCCA, 3D-CCA, MPCA, and DTSA3 in CASME.

4.3. CASME2

The CASME2 database [54] includes 246 spontaneous facial micro-expressions recorded by two 200 fps cameras. These samples were selected from more than 2500 facial expressions. Compared with CASME, this spontaneous micro-expression database is improved in terms of an increased sample size, fixed illumination, and higher resolution (both temporal and spatial). This database selected micro-expressions that had either a total duration of less than 500 ms or an onset duration (time from onset frame to apex frame) of less than 250 ms. These samples are coded with the onset and offset frames, and tagged with the action units (AUs) and emotions. An example is shown in Fig. 9.

Table 7
Micro-expression mean recognition accuracies (%) of STCCA, 3D-CCA, and DTSA3 in CASME2. (mean ± std. Bold numbers denote the highest recognition accuracies and `teletype numbers` denote the lowest recognition accuracies.)

q	Methods	5 × 5 × 5	10 × 10 × 10	20 × 20 × 20	30 × 30 × 30	40 × 40 × 40	50 × 50 × 50	60 × 60 × 60
$q = 5$	STCCA	**29.16 ± 4.79**	**29.64 ± 4.60**	**30.70 ± 4.38**	**30.97 ± 4.68**	**31.02 ± 4.85**	30.48 ± 4.98	30.68 ± 4.70
	3D-CCA	22.74 ± 4.05	26.81 ± 5.05	27.47 ± 14.87	25.59 ± 14.95	26.67 ± 14.28	26.67 ± 14.96	27.58 ± 14.64
	DTSA3	25.70 ± 4.66	26.31 ± 4.23	27.15 ± 4.76	30.45 ± 4.86	30.72 ± 4.73	**31.56 ± 4.88**	**31.49 ± 4.93**
$q=10$	STCCA	**31.73 ± 3.78**	**34.62 ± 3.59**	**34.87 ± 4.14**	**35.36 ± 3.98**	**35.05 ± 4.18**	**35.13 ± 4.26**	**35.00 ± 3.98**
	3D-CCA	28.49 ± 5.27	33.98 ± 4.03	23.67 ± 14.95	24.18 ± 18.39	24.92 ± 17.44	23.47 ± 17.91	21.91 ± 17.73
	DTSA3	27.17 ± 3.04	29.54 ± 3.15	32.60 ± 2.99	32.93 ± 4.01	32.68 ± 4.52	33.44 ± 3.83	34.11 ± 3.97
$q=15$	STCCA	**35.82 ± 4.36**	**36.99 ± 3.53**	**36.26 ± 4.11**	**36.49 ± 4.50**	**36.75 ± 4.61**	**36.96 ± 4.38**	**36.70 ± 4.56**
	3D-CCA	32.28 ± 3.42	34.15 ± 4.22	31.46 ± 4.23	19.24 ± 13.99	18.77 ± 17.73	18.30 ± 17.71	17.95 ± 17.89
	DTSA3	28.60 ± 2.38	31.32 ± 3.95	33.16 ± 4.48	33.77 ± 3.73	35.06 ± 3.79	35.50 ± 3.85	35.56 ± 3.77
$q=20$	STCCA	35.10 ± 3.46	**37.60 ± 4.01**	**38.32 ± 4.56**	**37.77 ± 4.71**	**37.64 ± 4.39**	**38.39 ± 4.98**	**37.81 ± 4.34**
	3D-CCA	**35.75 ± 4.74**	36.51 ± 4.23	35.07 ± 4.26	29.66 ± 9.46	19.97 ± 13.35	14.01 ± 14.52	17.19 ± 15.41
	DTSA3	29.01 ± 2.60	32.71 ± 2.91	35.07 ± 3.40	35.24 ± 3.74	35.00 ± 3.46	36.23 ± 3.89	36.82 ± 4.04

Table 8
The results of t-test between STCCA and DTSA3 in CASME2.

(p,t)	$5\times5\times5$	$10\times10\times10$	$20\times20\times20$	$30\times30\times30$	$40\times40\times40$	$50\times50\times50$	$60\times60\times60$
$q=5$	(0.006, 3.11)	(0.017, 2.63)	(0.009, 2.90)	(0.693, 0.40)	(0.815, 0.24)	(0.359, −0.94)	(0.443, −0.78)
$q=10$	**(0.000, 4.52)**	**(0.000, 4.85)**	(0.032, 2.32)	(0.012, 2.78)	**(0.001, 3.81)**	(0.029, 2.35)	(0.220, 1.27)
$q=15$	**(0.000, 6.03)**	**(0.000, 4.83)**	(0.005, 3.14)	(0.006, 3.13)	(0.063, 1.97)	(0.148, 1.51)	(0.157, 1.47)
$q=20$	**(0.000, 6.96)**	**(0.000, 5.37)**	**(0.005, 3.20)**	(0.038, 2.23)	(0.010, 2.88)	(0.065, 1.96)	(0.259, 1.16)

Table 9
The results of t-test between STCCA and 3D-CCA in CASME2.

(p,t)	$5\times5\times5$	$10\times10\times10$	$20\times20\times20$	$30\times30\times30$	$40\times40\times40$	$50\times50\times50$	$60\times60\times60$
$q=5$	**(0.000, 6.39)**	(0.007, 3.05)	(0.354, 0.95)	(0.146, 1.52)	(0.239, 1.22)	(0.313, 1.04)	(0.405, 0.85)
$q=10$	(0.014, 2.72)	(0.530, 0.64)	**(0.002, 3.53)**	(0.008, 2.95)	(0.011, 2.84)	(0.005, 3.15)	**(0.002, 3.53)**
$q=15$	**(0.001, 3.78)**	(0.021, 2.52)	**(0.000, 4.21)**	**(0.000, 4.71)**	**(0.001, 4.10)**	**(0.000, 4.32)**	**(0.000, 4.24)**
$q=20$	(0.577, −0.57)	(0.238, 1.22)	**(0.004, 3.29)**	**(0.002, 3.55)**	**(0.000, 5.20)**	**(0.000, 6.75)**	**(0.000, 5.55)**

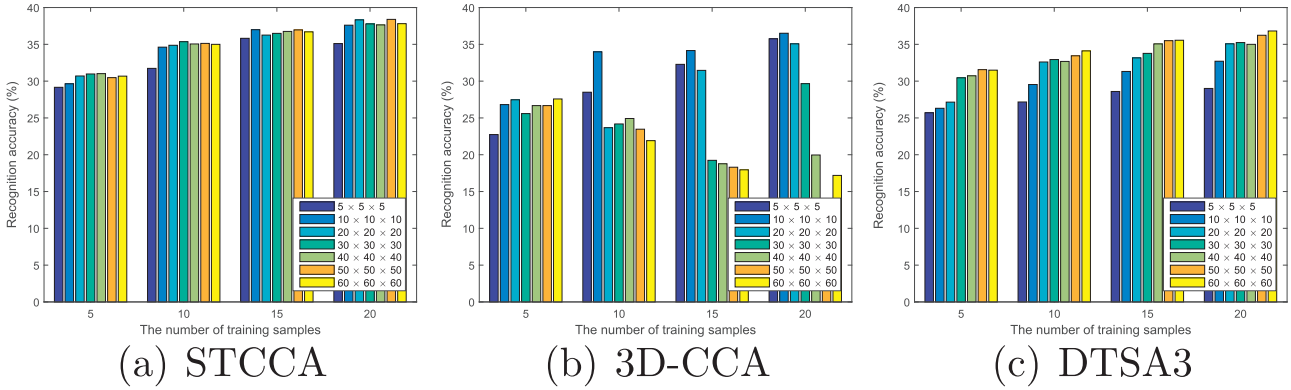


Fig. 10. The bar graphs of micro-expression mean recognition accuracies in CASME 2.

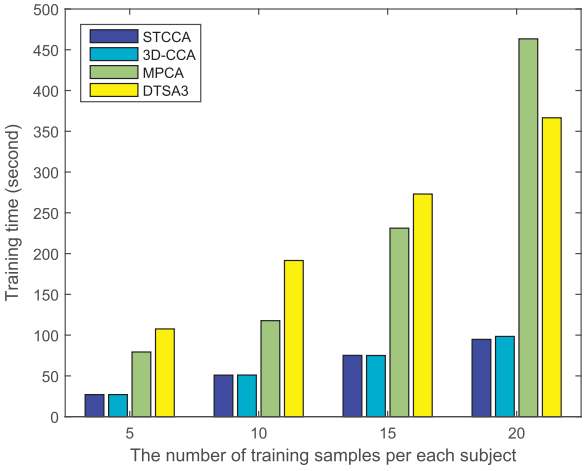


Fig. 11. The training times for training models in CASME 2.

In this database, micro-expressions are classified into 5 categories (happiness, surprise, disgust, repression and tense[5]). Table 6 lists the number of samples in each class. Among these samples, the frame numbers of the shortest and longest samples are 24 and 146, respectively. The frame numbers of all samples are normalized to 150 using linear interpolation. The sizes of each frame are normalized to 163×134 pixels. Thus, each sample was normalized to a third-order tensor with a size of $163\times134\times150$. The experimental setting is the same as that of the CASME database. We randomly split the micro-expression samples so that q ($q=5, 10, 15, 20$) samples for each class are used as the training set, and the remaining samples are used as the testing set. This process is repeated 20 times. The mean recognition accuracies and standard deviations are listed in Table 7.

3D-CCA only obtains the best performance in $5\times5\times5$ and $q=20$ case, and DTSA3 obtains the best performances in cases of larger projected dimensions and smaller sample size. In the majority of cases, STCCA performs the best. The CASME 2 database is larger than the CASME database. Thus, STCCA performs better than DTSA3 with the discriminant information when the training samples size is larger.

Table 8 lists the results of the t-test between STCCA and DTSA3. With lower reductional dimension and larger training sample size, the performances of STCCA are significance better than those of DTSA3. Table 9 lists the results of the t-test between STCCA and 3D-CCA. With larger reductional dimension and larger training sample size, the performances of STCCA are significance better than those of 3D-CCA.

The bar graphs of these mean recognition accuracies are illustrated in Fig. 10. Similarly, we also investigate their computational efficiencies for CASME 2. Fig. 11 illustrates the amount of time required project to dimensions of $5\times5\times5$. Compared to Fig. 7, the time consumption dramatically increases with the increasing number of training samples.

To further increase the size of the training set, we combine CASME and CASME 2. Each sample was normalized to a third-order tensor with a size of $163\times134\times150$. We randomly split the micro-expression samples so that q ($q=15, 20, 25, 30, 35$) samples for each class are used as the training set and the remaining are used as the testing set. This process is repeated 20 times. The mean

[5] The term "others" in CASME 2 is equivalent to "tense" in CASME, thus we use the same label as in CASME.

Table 10
Micro-expression mean recognition accuracies (%) of STCCA, 3D-CCA, and DTSA3 in CASME and CASME2. (mean ± std. Bold numbers denote the highest recognition accuracies and `teletype numbers` denote the lowest recognition accuracies.)

q	Methods	5 × 5 × 5	10 × 10 × 10	20 × 20 × 20	30 × 30 × 30	40 × 40 × 40	50 × 50 × 50	60 × 60 × 60
$q=15$	STCCA	31.78 ± 3.76	**35.48 ± 3.63**	**37.14 ± 3.98**	**37.68 ± 3.94**	**36.78 ± 3.81**	**36.69 ± 4.33**	**36.49 ± 4.14**
	3D-CCA	**31.90 ± 4.39**	34.74 ± 3.58	33.68 ± 2.87	29.36 ± 4.92	27.67 ± 4.35	28.76 ± 4.69	30.14 ± 3.95
	DTSA3	26.05 ± 2.22	29.05 ± 2.44	32.66 ± 2.92	33.68 ± 3.09	34.90 ± 3.64	36.14 ± 3.50	36.29 ± 3.41
$q = 20$	STCCA	**36.61 ± 2.88**	**38.32 ± 2.38**	**39.91 ± 2.54**	**39.97 ± 2.47**	**40.09 ± 2.61**	**39.76 ± 2.13**	38.12 ± 2.07
	3D-CCA	36.09 ± 3.85	37.58 ± 4.44	37.29 ± 4.17	35.02 ± 3.31	33.17 ± 3.12	33.90 ± 2.86	35.15 ± 2.53
	DTSA3	27.29 ± 2.96	30.67 ± 3.05	35.12 ± 2.14	35.73 ± 3.26	37.55 ± 3.19	38.29 ± 3.10	**39.20 ± 3.26**
$q=25$	STCCA	36.54 ± 2.24	**38.81 ± 2.61**	**39.47 ± 3.51**	**40.25 ± 3.39**	**40.56 ± 3.13**	**39.85 ± 3.27**	**39.24 ± 3.09**
	3D-CCA	**40.51 ± 4.26**	37.63 ± 3.29	39.01 ± 3.84	38.59 ± 3.28	36.95 ± 4.18	35.83 ± 3.88	36.04 ± 3.30
	DTSA3	27.88 ± 2.38	31.44 ± 2.54	35.23 ± 2.84	36.21 ± 3.16	36.71 ± 3.17	37.30 ± 2.37	38.20 ± 2.63
$q=30$	STCCA	37.64 ± 3.60	**40.22 ± 2.83**	**41.46 ± 2.49**	**41.79 ± 2.86**	**42.33 ± 2.79**	**42.04 ± 2.95**	**41.12 ± 2.90**
	3D-CCA	**41.77 ± 3.87**	38.75 ± 3.24	40.18 ± 3.05	40.92 ± 2.69	40.02 ± 2.88	38.52 ± 2.71	38.30 ± 2.60
	DTSA3	27.26 ± 3.05	32.11 ± 2.91	36.30 ± 2.93	37.09 ± 2.54	38.19 ± 3.45	38.36 ± 3.38	39.31 ± 3.17
$q=35$	STCCA	36.27 ± 7.42	**41.61 ± 2.44**	**42.36 ± 2.52**	**43.06 ± 2.30**	**43.47 ± 2.76**	**42.94 ± 2.76**	**42.12 ± 3.17**
	3D-CCA	**40.79 ± 3.51**	39.58 ± 3.17	42.16 ± 2.76	42.62 ± 2.71	41.61 ± 2.20	40.87 ± 2.75	39.46 ± 2.63
	DTSA3	28.77 ± 2.51	33.87 ± 3.07	38.39 ± 2.32	38.83 ± 2.40	40.06 ± 2.51	40.32 ± 1.51	41.61 ± 2.37

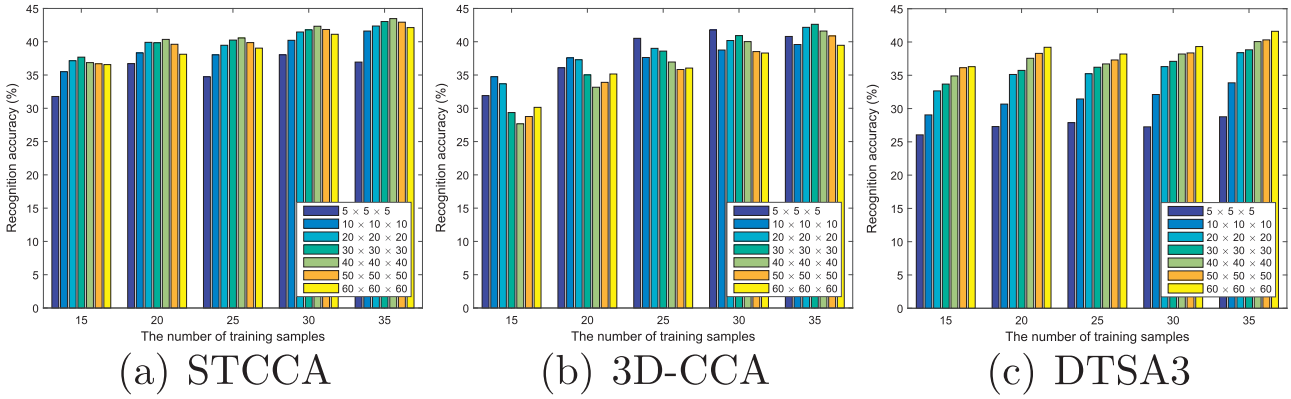


Fig. 12. Bar graphs of micro-expression mean recognition accuracies in CASME and CASME 2.

Table 11
The results of *t*-test between STCCA and DTSA3 in CASME and CASME2.

(*p*,*t*)	5 × 5 × 5	10 × 10 × 10	20 × 20 × 20	30 × 30 × 30	40 × 40 × 40	50 × 50 × 50	60 × 60 × 60
$q=15$	**(0.000, 6.30)**	**(0.000, 6.63)**	**(0.000, 5.13)**	**(0.000, 4.68)**	(0.018, 2.60)	(0.391, 0.88)	(0.710, 0.38)
$q=20$	**(0.000, 8.53)**	**(0.000, 8.20)**	**(0.000, 6.34)**	**(0.000, 5.25)**	(0.005, 3.14)	(0.077, 1.87)	(0.209, −1.30)
$q=25$	**(0.000, 11.61)**	**(0.000, 14.00)**	**(0.000, 6.20)**	**(0.000, 5.46)**	**(0.000, 6.39)**	**(0.000, 4.66)**	(0.073, 1.90)
$q=30$	**(0.000, 9.95)**	**(0.000, 11.31)**	**(0.000, 6.11)**	**(0.000, 6.30)**	**(0.000, 4.84)**	**(0.000, 4.67)**	(0.045, 2.14)
$q=35$	**(0.001, 4.17)**	**(0.000, 9.41)**	**(0.000, 4.92)**	**(0.000, 5.54)**	**(0.000, 4.68)**	**(0.000, 4.83)**	(0.497, 0.69)

Table 12
The results of *t*-test between STCCA and 3D-CCA in CASME and CASME2.

(*p*,*t*)	5 × 5 × 5	10 × 10 × 10	20 × 20 × 20	30 × 30 × 30	40 × 40 × 40	50 × 50 × 50	60 × 60 × 60
$q=15$	(0.903, −0.12)	(0.402, 0.86)	**(0.001, 4.03)**	**(0.000, 7.20)**	**(0.000, 7.86)**	**(0.000, 6.06)**	**(0.000, 8.14)**
$q=20$	(0.652, 0.46)	(0.383, 0.89)	(0.006, 3.06)	**(0.000, 5.82)**	**(0.000, 7.78)**	**(0.000, 7.94)**	**(0.000, 6.21)**
$q=25$	**(0.002, −3.70)**	(0.154, 1.49)	(0.341, 0.98)	**(0.000, 4.40)**	**(0.000, 4.90)**	**(0.000, 4.89)**	**(0.000, 4.98)**
$q=30$	**(0.001, −3.87)**	(0.071, 1.91)	(0.053, 2.06)	(0.316, 1.03)	**(0.002, 3.51)**	**(0.000, 6.98)**	**(0.000, 6.51)**
$q=35$	(0.017, −2.61)	**(0.004, 3.24)**	(0.718, 0.37)	(0.529, 0.64)	(0.006, 3.11)	(0.006, 3.06)	**(0.001, 3.95)**

recognition accuracies standard deviations are listed in Table 10, and their bar graphs are illustrated in Fig. 12. 3D-CCA exhibit the best performances in the $5 \times 5 \times 5$ cases, whereas STCCA performed the best in the other cases. STCCA performs worse than 3D-CCA particularly in cases of small projected dimensions, because when the projected dimensions are rather small, the sparsity cause many of the coefficients of the projected dimensions to be zeros, and important information may be lost in the this process. Tables 11 and 12 list the corresponding results of *t*-test. The similar conclusion can be drawn.

5. Conclusions

We have presented STCCA, a novel method of dimensionality reduction, to recognize micro-expressions. STCCA seeks a subspace in which the correlation between micro-expression data and their corresponding LBP code data is maximal. The solution of STCCA is sparse. The experiments on two micro-expression databases revealed that STCCA outperformed the existing methods. Furthermore, MPCA is not suitable for micro-expression recognition because the eigenvectors corresponding to smaller eigenvectors are discarded, and

those eigenvectors include brief and subtle motion information.

We will introduce discriminant information to STCCA to further improve recognition accuracy.

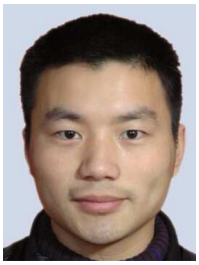
Acknowledgments

This paper is supported in part by grants from the National Natural Science Foundation of China (61379095, 61375009, 61472138, 31500875), the Beijing Natural Science Foundation (4152055), and the Academy of Finland, and Infotech Oulu.

References

[1] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[2] W.-J. Yan, Q. Wu, J. Liang, Y.-H. Chen, X. Fu, How fast are the leaked facial expressions: the duration of micro-expressions, J. Nonverbal Behav. (2013) 1–14.
[3] D. Matsumoto, H. Hwang, Evidence for training the ability to read microexpressions of emotion, Motiv. Emot. 35 (2) (2011) 181–191.
[4] P. Ekman, W. Friesen, Nonverbal Leakage and Clues to Deception, Technical Report, DTIC Document, 1969.
[5] P. Ekman, Lie catching and microexpressions, Philos. Decept. (2009) 118–133.
[6] M.G. Frank, M. Herbasz, A.K.K. Sinuk, C. Nolan, I see how you feel: training laypeople and professionals to recognize fleeting emotions, in: The Annual Meeting of the International Communication Association, New York, 2009.
[7] M. OSullivan, M. Frank, C. Hurley, J. Tiwana, Police lie detection accuracy: the effect of lie scenario, Law Hum. Behav. 33 (6) (2009) 530–538.
[8] M. Frank, C. Maccario, V. Govindaraju, Behavior and Security, Greenwood Publishing Group, Santa Barbara, California, 2009, pp. 86–106.
[9] E.A. Haggard, K.S. Isaacs, Micromomentary facial expressions as indicators of ego mechanisms in psychotherapy, in: Methods of Research in Psychotherapy, Appleton-Century-Crofts, New York, 1966, pp. 154–165.
[10] P. Ekman, Darwin, deception, and facial expression, Ann. N. Y. Acad. Sci. 1000 (1) (2006) 205–221.
[11] S. Porter, L. Ten Brinke, Reading between the lies identifying concealed and falsified emotions in universal facial expressions, Psychol. Sci. 19 (5) (2008) 508–514.
[12] P. Ekman, Microexpression training tool (METT), University of California, San Francisco, 2002.
[13] S. Polikovsky, Y. Kameda, Y. Ohta, Facial micro-expressions recognition using high speed camera and 3D-gradient descriptor, in: 3rd International Conference on Crime Detection and Prevention, IET, 2009, pp. 1–6.
[14] S.-J. Wang, H.-L. Chen, W.-J. Yan, Y.-H. Chen, X. Fu, Face recognition and micro-expression based on discriminant tensor subspace analysis plus extreme learning machine, Neural Process. Lett. (2013), http://dx.doi.org/10.1007/s11063-013-9288-7.
[15] T. Pfister, X. Li, G. Zhao, M. Pietikainen, Recognising spontaneous facial micro-expressions, in: 12th IEEE International Conference on Computer Vision, IEEE, 2011, pp. 1449–1456.
[16] Z. Zhou, G. Zhao, M. Pietikainen, Towards a practical lipreading system, in: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 137–144.
[17] G. Zhao, M. Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions, IEEE Trans. Pattern Anal. Mach. Intell. 29 (6) (2007) 915–928.
[18] X. Huang, S.J. Wang, G. Zhao, M. Piteikainen, Facial micro-expression recognition using spatiotemporal local binary pattern with integral projection, in: 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), 2015, pp. 1–9, http://dx.doi.org/10.1109/ICCVW.2015.10.
[19] X. Huang, G. Zhao, X. Hong, W. Zheng, M. Pietikinen, Spontaneous facial micro-expression analysis using spatiotemporal completed local quantized patterns, Neurocomputing 175 (Part A) (2016) 564–578, http://dx.doi.org/10.1016/j.neucom.2015.10.096, URL ⟨.
[20] Y. Mu, W. Ding, D. Tao, T.F. Stepinski, Biologically inspired model for crater detection, in: The 2011 International Joint Conference on Neural Networks (IJCNN), IEEE, 2011, pp. 2487–2494.
[21] Y. Mu, W. Ding, M. Morabito, D. Tao, Empirical discriminative tensor analysis for crime forecasting, in: Knowledge Science, Engineering and Management, Springer, 2011, pp. 293–304.
[22] Y. Mu, D. Tao, Biologically inspired feature manifold for gait recognition, Neurocomputing 73 (4) (2010) 895–902.
[23] Y. Mu, D. Tao, X. Li, F. Murtagh, Biologically inspired tensor features, Cognit. Comput. 1 (4) (2009) 327–341.
[24] L. De Lathauwer, B. De Moor, J. Vandewalle, On the best rank-1 and rank-(r1, r2,...,r-n) approximation of higher-order tensors, Siam J. Matrix Anal. Appl. 21 (4) (2000) 1324–1342.
[25] H. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition, IEEE Trans. Neural Netw. 20 (1) (2009) 103–123.
[26] H.P. Lu, N.P. Konstantinos, A.N. Venetsanopoulos, MPCA: multilinear principal component analysis of tensor objects, IEEE Trans. Neural Netw. 19 (1) (2008) 18–39.
[27] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, H. Zhang, Discriminant analysis with tensor representation, in: CVPR, 2005.
[28] D. Tao, X. Li, X. Wu, S. Maybank, General tensor discriminant analysis and gabor features for gait recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1700–1715.
[29] X. He, D. Cai, P. Niyogi, Tensor subspace analysis, in: Advances in Neural Information Processing Systems 18 (NIPS), MIT Press, 2005.
[30] S.-J. Wang, C.-G. Zhou, N. Zhang, X.-J. Peng, Y.-H. Chen, X. Liu, Face recognition using second-order discriminant tensor subspace analysis, Neurocomputing 74 (12–13) (2011) 2142–2156.
[31] F. Song, D. Zhang, Q. Chen, J. Wang, Face recognition based on a novel linear discriminant criterion, Pattern Anal. Appl. 10 (3) (2007) 165–174.
[32] H. Hotelling, Relations between two sets of variates, Biometrika 28 (3/4) (1936) 321–377.
[33] W. Zheng, X. Zhou, C. Zou, L. Zhao, Facial expression recognition using kernel canonical correlation analysis (KCCA), IEEE Trans. Neural Netw. 17 (1) (2006) 233–238.
[34] T. Sun, S. Chen, Locality preserving cca with applications to data visualization and pose estimation, Image Vis. Comput. 25 (5) (2007) 531–543.
[35] S.H. Lee, S. Choi, Two-dimensional canonical correlation analysis, IEEE Signal Process. Lett. 14 (10) (2007) 735–738.
[36] H. Wang, Local two-dimensional canonical correlation analysis, IEEE Signal Process. Lett. 17 (11) (2010) 921–924.
[37] L. Gang, Z. Yong, L. Yan-Lei, D. Jing, Three dimensional canonical correlation analysis and its application to facial expression recognition, in: Intelligent Computing and Information Science, Springer, 2011, pp. 56–61.
[38] D.R. Hardoon, J. Shawe-Taylor, Sparse canonical correlation analysis, Mach. Learn. 83 (3) (2011) 331–353.
[39] J. Yan, W. Zheng, X. Zhou, Z. Zhao, Sparse 2-d canonical correlation analysis via low rank matrix approximation for feature extraction, IEEE Signal Process. Lett. 19 (1) (2012) 51–54.
[40] S.-J. Wang, M.-F. Sun, Y.-H. Chen, E.-P. Pang, C.-G. Zhou, STPCA: sparse tensor principal component analysis for feature extraction, in: The 21st International Conference on Pattern Recognition, Tsukuba, Japan, 2012, pp. 2278–2281.
[41] S.-J. Wang, J. Yang, M.-F. Sun, X.-J. Peng, M.-M. Sun, C.-G. Zhou, Sparse tensor discriminant color space for face verification, IEEE Trans. Neural Netw. Learn. Syst. 23 (6) (2012) 876–888.
[42] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, Siam Rev. 51 (3) (2009) 455–500.
[43] E. Acar, D.M. Dunlavy, T.G. Kolda, M. Mørup, Scalable tensor factorizations with missing data, in: SDM, SIAM, 2010, pp. 701–712.
[44] J. Ye, R. Janardan, Q. Li, Two-dimensional linear discriminant analysis, in: Neural Information Processing Systems, 2004, pp. 1569–1576.
[45] S.-J. Wang, C.-G. Zhou, X. Fu, Fusion tensor subspace transformation framework, PLoS One 8 (7) (2013) e66647.
[46] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, Active shape models-their training and application, Comput. Vis. Image Underst. 61 (1) (1995) 38–59.
[47] W.-J. Yan, Q. Wu, Y.-J. Liu, S.-J. Wang, X. FU, CASME Database: a dataset of spontaneous micro-expressions collected from neutralized faces, in: 10th IEEE Conference on Automatic Face and Gesture Recognition, 2013.
[48] W.-J. Yan, S.-J. Wang, Y.-J. Liu, Q. Wu, X. Fu, For micro-expression recognition: database and suggestions, Neurocomputing 136 (2014) 82–87.
[49] P. Ekman, W. Friesen, J. Hager, Facs investigators guide, A human face (2002).
[50] Y.-J. Liu, J.-K. Zhang, W.-J. Yan, S.-J. Wang, G. Zhao, X. Fu, A main directional mean optical flow feature for spontaneous micro-expression recognition.
[51] S.-T. Liong, J. See, R. C.-W. Phan, A.C. Le Ngo, Y.-H. Oh, K. Wong, Subtle expression recognition using optical strain weighted features, in: Computer Vision—ACCV 2014 Workshops, Springer, 2014, pp. 644–657.
[52] D. Sun, S. Roth, M.J. Black, A quantitative analysis of current practices in optical flow estimation and the principles behind them, Int. J. Comput. Vis. 106 (2) (2014) 115–137.
[53] T. Senst, V. Eiselein, T. Sikora, Robust local optical flow for feature tracking, IEEE Trans. Circuits Syst. Video Technol. 22 (9) (2012) 1377–1387.
[54] W.-J. Yan, X. Li, S.-J. Wang, G. Zhao, Y.-J. Liu, Y.-H. Chen, X. Fu, CASME II: an improved spontaneous micro-expression database and the baseline evaluation, PLoS One 9 (1) (2014) e86041.

Su-Jing Wang received the Master's degree from the Software College of Jilin University, Changchun, China, in 2007. He received the Ph.D. degree from the College of Computer Science and Technology of Jilin University in 2012. He was a Postdoctoral Researcher in Institute of Psychology, Chinese Academy of Sciences from 2012 to 2015. He is now an Assistant Researcher in Institute of Psychology, Chinese Academy of Sciences. He has published more than 40 scientific papers. He is One of Ten Selectees of the Doctoral Consortium at International Joint Conference on Biometrics 2011. He was called as Chinese Hawkin by the Xinhua News Agency. His current research interests include pattern recognition, computer vision and machine learning. He serves as an Associate Editor of Neurocomputing (Elsevier). For more information, visit http://sujingwang.name.



Wen-Jing Yan received his Ph.D. degree from Institute of Psychology, Chinese Academy of Sciences, Beijing, China, in 2014. He is now an Assistant Professor in Department of Psychology in Wenzhou University, China. His interests include facial expression and deception. His research interests include interdisciplinary research on facial expression and affective computing.

Tingkai Sun received his B.Sc. degree from Nanjing University, Nanjing, China, in 1996. In 1999 he completed M.Sc. degree in Nanjing University, and in July 2007 he received his Ph.D. degree in Computer Applications Technology from Nanjing University of Aeronautics & Astronautics. Now as an Associate Professor he works at School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include pattern recognition and machine learning.

Guoying Zhao received the Ph.D. degree in Computer Science from the Chinese Academy of Sciences, Beijing, China, in 2005. She is currently an Associate Professor with the Center for Machine Vision Research, University of Oulu, Finland, where she has been a Researcher since 2005. In 2011, she was selected to the highly competitive Academy Research Fellow position. She has authored or co-authored more than 120 papers in journals and conferences, and has served as a Reviewer for many journals and conferences. She has lectured tutorials at ICPR 2006, ICCV 2009, and SCIA 2013, and authored/edited three books and four special issues in journals. Dr. Zhao was a Co-Chair of five International Workshops at ECCV, ICCV, CVPR and ACCV, and two special sessions at FG13 and FG15. She is a editorial board member for Image and Vision Computing Journal, International Journal of Applied Pattern Recognition and ISRN Machine Vision. She is a IEEE Senior Member. Her current research interests include image and video descriptors, gait analysis, dynamic-texture recognition, facial-expression recognition, human motion analysis, and person identification.

Xiaolan Fu received her Ph.D. degree in 1990 from Institute of Psychology, Chinese Academy of Sciences. Currently, she is a Senior Researcher at Cognitive Psychology. Her research interests include visual and computational cognition: (1) attention and perception, (2) learning and memory, and (3) affective computing. At present, she is the Director of Institute of Psychology, Chinese Academy of Sciences and Vice Director, State Key Laboratory of Brain and Cognitive Science.