Journal of Information Hiding and Multimedia Signal Processing ©2016 ISSN 2073-4212
Ubiquitous International Volume **7**, Number **6**, November **2016**

Support Tensor Machine Image Classification Algorithm Based on Tensor Principal Component Analysis

Qing-Zhu Wang, Wan-Jun Kang and Yan-Jiao Wang

School of Information Engineering
Northeast Dianli University
169 Changchun Road, Jilin, China
150681573@qq.com, 464760918@qq.com, 563274435@qq.com

Received April, 2016; revised July, 2016

Abstract. *To allow rapidly and accurately support tensor machine (STM) image classification, we developed an algorithm based on tensor principal component analysis (TPCA). This STM image classification algorithm based on TPCA was specifically developed for gait recognition. In our system, the STM processes the gait image as a data cube and then identifies the information classes in tensor space. TPCA is used for preprocessing to reduce the tensorial data redundancy and to maintain the tensorial structure information in high-order subspace. The algorithm steps are as follows: Firstly, third-order tensor image features are constructed based on the structural information of the data. Second, TPCA is used for feature extraction. Finally, STM classifier is used to directly classify the image. The experiments show that the proposed algorithm can not only improve the accuracy of image classification but also reduce the required computational time of the STM.*

Keywords: TPCA, STM, Image classification.

1. **Introduction.** In the study of computer vision, data mining, and pattern recognition, data objects are usually represented as tensors. To deal with tensor data, traditional classification algorithms usually convert them into vectors [1, 2]. However, vector conversion may lead to two problems. First, this process destroys the inherent higher order structure and correlation of the original data, leading to the loss of information. Second, it generates high-dimensional vectors and makes the subsequent learning process prone to over-fitting. Additionally, the dimensionality contributes to small sample size problems [3]. Maintenance of tensor data can eliminate the problems result from converting tensors into vectors. Thus, many researchers have proposed classification algorithms based on the tensor pattern and extending a support vector machine to support the tensor machine. In the field of machine learning, determining an effective model and attempting to design a quick and accurate tensor classification algorithm is a topic currently of high interest [4]. Principal component analysis (PCA) is a well-known unsupervised linear technique for dimensionality reduction. In [5], PCA was used to remove the data redundancy and SVM was used for data classification. The results show that the proposed approach is effective, and has a high probability of correct classification. Inspired by [5], we extended the study of TPCA and STM.

Typically, researchers construct a number of support vector machines to manage tensor data by projecting the tensor into multiple modal space. For example, a unified framework for supervised tensor learning was proposed by Tao [6]. Based on the STL framework, the least square tensor classifier was proposed by Cai [7] and Wang [8] for the two order tensor and Tao generalized least squares support vector machines to general tensor models. Because the optimization problem given by the STL framework is non-convex, the alternating projection is used to solve the model, which requires calculation, time, and memory space, and may result in a local optimal solution. The support tensor machine only requires solving a convex optimization problem to determine the solution of the model, which greatly reduces the computational time. Because tensor objects are usually high-dimensional and contain large amounts of redundancy, in the STM model, the HOSVD of the tensor is used to replace the original data. This better reflects the potential and close pattern representation of the tensor data, but does require storage of all the tensor data in the memory space, thus greatly reducing the required memory overhead.

Inspired by [5], we extend the PCA + SVM to tensor patterns and propose a novel tensor-based framework (TPCA + STM) for image classification in this paper. Specifically, analogous to PCA + SVM, the images are stacked as a feature tensor, and TPCA is used to reduce the information redundancy in the tensor. The reduced features are then inputting to an STM for classification. Compared with the previous traditional classification algorithms which will result in the following two problems: (1) breaking the natural structure and correlation in the original data; (2) leading to the curse of dimensionality and small sample size problem. The proposed method (TPCA + STM) that represents data with tensor can remain the natural structure and correlation of the data, and avoid information missing, over-fitting, curse of dimensionality and small sample size. And we integrate the tensor decomposition into the model to assist its inner product computation. The tensor decomposition, which is a low-rank approximation of the tensor, can better embody the structural information and intrinsic correlation of the data, obtain more compact and meaningful representations of the tensor objects, especially in the case of higher-order tensor, thus it can improve the effectiveness of inner product computation and save storage space and computational time.

In order to quickly and accurately classify image, TPCA is first used to conduct feature extraction and then the tensor is treated as the input of STM. Thus, we propose and experimentally test a support tensor machine image classification algorithm based on TPCA.

The remainder of this paper is organized as follows. A brief review of multilinear algebra is given in the Section II. The proposed STM image classification algorithm based on TPCA is described in Section III. Experimental results and comparison of TPCA, TPCA + SVM, PCA + SVM and TPCA + STM for image sequence are provided in Section IV.

2. **Tensor and Multilinear Algebra.**

The definitions and theorems used in the algorithm are as follows:

Definition 2.1 (Tensor)**.** *Tensor can be viewed as a multidimensional array,* $X \in R^{I_1 \times I_2 \times \cdots \times I_N}$ *represent a order tensor,* $x_{i_1, i_2, \cdots, x_n}$ *represent an element of* X*,* $1 \leqslant i_n \leqslant I_n$*,* $1 \leqslant n \leqslant N$*.*

Definition 2.2 (Tensor matrix expansion)**.** *Tensor matrix expansion is a process of rearrangement of the elements in a tensor to obtain a matrix and the N mode expansion matrix of* $X \in R^{I_1 \times I_2 \times \cdots \times I_N}$ *is represented as* $X_{(n)} \in R^{I_n \times (I_1 \cdots I_{n-1} I_n \cdots I_N)}$*. Fig.1 illustrates the 1-mode (column mode) unfolding of a third-order tensor* [9, 10].

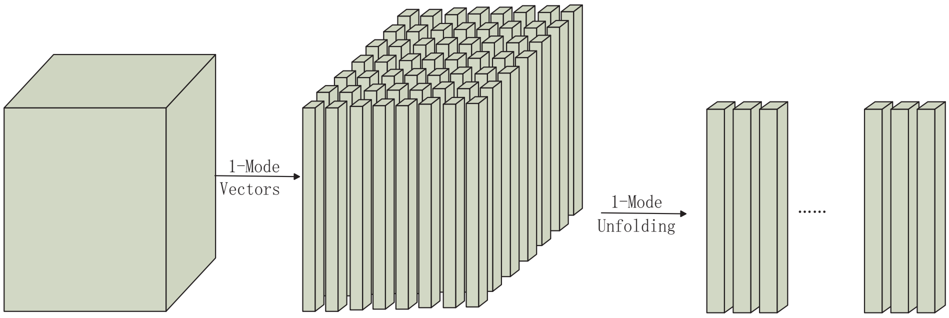

FIGURE 1. Visual illustration of the 1-mode unfolding of a third-order tensor

Definition 2.3 (Tensor multiplication). *An Nth-order tensor is denoted as $X \in R^{I_1 \times I_2 \times \cdots \times I_N}$. It is addressed by N indices i_n, $i = 1, 2, \cdots, N$, and each i_n addresses the n-mode of X. The n-mode product of a tensor X by a matrix $U \in R^{J_n \times I_n}$, denoted by $X \times_n U$, is a tensor with entries $(X \times_n U)_{i_1 i_2 \cdots j_n \cdots i_N} = \sum_{i_n} a_{i_1 i_2 \cdots j_n \cdots i_N} \cdot u_{j_n i_n}$. Fig. 2 provides a visual illustration of the multilinear projection* [9]*. In Fig. 2, a third-order tensor $A \in R^{10 \times 8 \times 6}$ is projected in the 1-mode vector space by a projection matrix $B^{(1)^{\mathrm{T}}} \in R^{5 \times 10}$, resulting in the projected tensor $A \times_1 B^{(1)^{\mathrm{T}}} \in R^{5 \times 8 \times 6}$. In the 1-mode projection, each 1-mode vector of A of length 10 is projected by $B^{(1)^{\mathrm{T}}}$ to obtain a vector of length 5 as the differently shaded vectors in Fig. 2.*

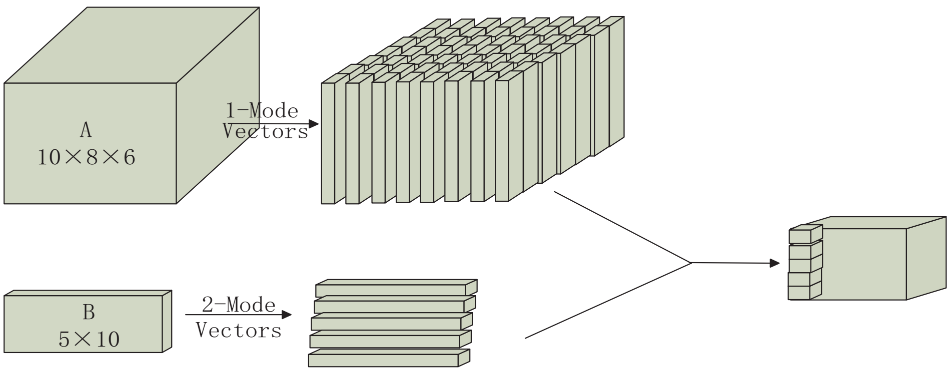

FIGURE 2. Visual illustration of multilinear projection

Theorem 2.1 (HOSVD). *Each tensor $X \in R^{I_1 \times I_2 \times \cdots \times I_N}$ can be uniquely decomposed into $X = T \times_1 U_1 \times_2 \times U_2 \times \cdots \times_N U_N$, and meets the following conditions:*

(1) $U_{(n)}$ is the orthogonal matrix of $I_n \times I_n$.

(2) The row of $T_{(n)}$ is orthogonal.

(3) For any n, $\|T_{i_n=1}\| \geqslant \|T_{i_n=2}\| \geqslant \cdots \geqslant \|T_{i_n=N}\| \geqslant 0$.

Tensor T is also called the core tensor, whose HOSVD [11, 12] *as follows:*

(1) For any n, $U_{(n)}$ is a left matrix which is n mode expansion matrix $X_{(n)}$ of X by SVD decomposition.

(2) Compute: $T = X \times_1 U_1^T \times_2 U_2^T \times \cdots \times_N U_N^T$.

3. STM Image Classification Algorithm based on TPCA.

3.1. **TPCA.** TPCA based on higher order singular value decomposition (HOSVD) is introduced in this paper.

The basic idea of TPCA [13, 14] is to extract the most significant signal components for all tensor modes while retaining the data variation of the original data set. TPCA formulation involves the acquisition of projection matrices and the low-dimensional tensor.

Let $\{\alpha_m, m = 1, 2, \cdots, M\}$ be a set of M tensor samples in $R^{I_1} \otimes R^{I_2} \otimes \cdots \otimes R^{I_N}$. The total scatter of these tensors is defined as $S_{Y_\Omega}^{(n)} = \sum_{m=1}^{M} (\alpha_{m(n)} - \bar{\alpha}_{(n)})^{\mathrm{T}}$, where $\overline{\alpha}$ is the mean tensor calculated as $\bar{\alpha} = \frac{1}{M} \sum_{m=1}^{M} \alpha_m$. The n-mode total scatter matrix of these samples is then defined as $S_{Y_\Omega}^{(n)} = \sum_{m=1}^{M} (\alpha_{m(n)} - \bar{\alpha}_{(n)})^{\mathrm{T}}$, where $\alpha_{m(n)}$ is the n-mode unfolded matrix of α_m.

The goal of TPCA is to define a multidimensional linear projection matrix $\{\tilde{U}(n) \in R^{I_n \times P_n}, n = 1, 2, \cdots, N\}$, which can project the data in the original tensor space $R^{I_1} \otimes R^{I_2} \otimes \cdots \otimes R^{I_N}$ into tensor subspace $R^{P_1} \otimes R^{P_2} \otimes \cdots \otimes R^{P_N} (P < I_n, n = 1, 2, \cdots, N)$. Because of $P < I_n$, the feature extraction of data is realized in the projection process and the key information of the original data is retained in the target tensor object.

$$\begin{aligned}
&\beta_m = \alpha_m \times_1 \tilde{U}^{(1)\mathrm{T}} \times_2 \tilde{U}^{(2)\mathrm{T}} \cdots \times_N \tilde{U}^{(N)\mathrm{T}} \; (m = 1, 2, \cdots, M) \\
&\beta_m \in R^{P_1} \otimes R^{P_2} \otimes \cdots \otimes R^{P_N}, m = 1, 2, \cdots, M
\end{aligned} \tag{1}$$

The optimal projection matrix should satisfy the following conditions:

$$\{\tilde{U}^{(n)} \in R^{I_n \times P_n}, P_n \prec I_n, n = 1, 2, \cdots, N\} = \arg \max_{\tilde{U}^{(1)}, \tilde{U}^{(2)}, \cdots, \tilde{U}^{(N)}} \Psi_\beta \tag{2}$$

$$\Psi_\beta = \sum_{m=1}^{M} \|\beta_m - \bar{\beta}\|_F^2$$

The optimum solution of (2) is $\{\tilde{U}^{(n)} \in R^{I_n \times p_n}, n = 1, 2, \ldots, N\}$, matrix $\tilde{U}^{(n)}$ contains maximum P_N feature vector corresponding to the eigenvalue of $\varphi^{(n)} = \sum_{m=1}^{M} (\alpha_{m(n)} - \bar{\alpha}_{(n)})$ $\tilde{U}_\varphi^{(n)\mathrm{T}} (\alpha_{m(n)} - \bar{\alpha}_{(n)})^{\mathrm{T}}$, where $\tilde{U}^{(n)}$ is defined as $\tilde{U}_\varphi^{(n)} = (\tilde{U}^{(n+1)} \otimes \tilde{U}^{(n+2)} \otimes \cdots \otimes \tilde{U}^{(N)} \otimes \tilde{U}^{(1)} \otimes \tilde{U}^{(2)} \otimes \cdots \otimes \tilde{U}^{(n-1)} \otimes \tilde{U}^{(n)}$.

3.2. **STM.** Traditional machine learning methods can only directly deal with the input vector mode. There are two ways to process the input samples of the image pattern. First, the matrix structure must be collapsed to make vector inputs to be used in the algorithm. One common way is connecting each row (or column) of a matrix to reformulate a vector. Second, the structure and parameters of the classifier can be improved so that the input samples of the image pattern can be processed directly.

It should be noted that the conventional vector-based analysis approach need to unfold the cubic features into vectors for feature description and classification owing to traditional classifiers can only process vector inputs. However, the proposed STM is able to directly model the gait image sequence tensor, which is naturally aligned as a cube for image interpretation. Compared with the traditional SVM, the characteristic presents a main methodological advantage of STM.

STM is a generalization of support vector machine, which uses the tensor as a running sample instead of vector. Taking the order tensor as an example, this paper briefly introduces the STM. Support tensor machine [15, 16] was designed in this paper based on some methods of two-dimensional classifier, which improves the construction of the classifier and also can classify directly tensor data. It is known a running sample has

N M-th order tensors $X_i \in R^{L_1\times L_2\times\cdots\times L_M}$ and its corresponding class is $y_i \in \{+1,-1\}$, $i = 1,2,\cdots,N$, so the optimal classification flat is

$$y(X) = X\prod_{k=1}^{M}\times_k\omega_k + b \tag{3}$$

It is easy to see from Eq. (3) that projection vector $\omega_k \in R^{nk}(k = 1,2,\cdots,N)$ and offset $b \in R$ are the unknown parameters in the classification flat. We can solve the optimal problem using Eq. (4):

$$\begin{aligned} &\min_{\omega_k,|_{k=1}^{M}b,\xi} \quad \frac{1}{2}\prod_{k=1}^{N}\|\bigotimes_{k=1}^{M}\omega_k\|^2 + c\sum_{i=1}^{N}\xi_i \\ &s\cdot t\cdot \quad y_i[X_i\prod_{k=1}^{M}\times_k\omega_k + b] \leqslant 1-\xi_i \\ &\xi_i \geqslant 0 \end{aligned} \tag{4}$$

As seen from Eq. (4), we solve the model as follows:

(1) Construct the Lagrangian function, where α_i and k_i are the Lagrange multiplier

$$\begin{aligned} L(\omega_k|_{k=1}^{M},b,\zeta,\alpha,\kappa) &= \frac{1}{2}\|\bigotimes_{k=1}^{M}\|^2 + c\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\kappa_i\xi_i - \sum_{i=1}^{N}\alpha_i(y_i[X_i\prod_{k=1}^{M}\times_k\omega_k + b] - 1 + \xi_i) \\ &= \frac{1}{2}\prod_{k=1}^{M}\omega_k^{\mathrm{T}}\omega_k + c\sum_{i=1}^{N}\xi_i - b\alpha^{\mathrm{T}}y + \sum_{i=1}^{N}\alpha_i - \alpha^{\mathrm{T}}\xi - \kappa^{\mathrm{T}}\xi - \sum_{i=1}^{N}\alpha_i y_i(X_i\prod_{k=1}^{M}\times_k\omega_k) \end{aligned} \tag{5}$$

(2) L in Eq. (5) respect to ω, b and ξ, taking the derivative of L with respect to ω, b and ξ, we have:

$$\begin{aligned} &\frac{\partial L}{\partial\omega} = 0 \Rightarrow \omega_j = \frac{1}{\prod_{k=1}^{k\neq j}\omega_k^{\mathrm{T}}\omega_k}\cdot\sum_{i=1}^{N}\alpha_i y_i(X_i\prod_{k=1}^{M}\times_j\omega_j) \\ &\frac{\partial L}{\partial b} = 0 \Rightarrow \alpha^{\mathrm{T}}y = 0 \\ &\frac{\partial L}{\partial\xi} = 0 \Rightarrow c-\alpha-\kappa = 0 \end{aligned} \tag{6}$$

(3) Its dual form is:

$$\max_{\alpha,\kappa}\min L(\omega_k|_{k=1,}^{M}b,\xi,\varepsilon,\kappa) \tag{7}$$

Eq. (7) is a linear programming problem. We solve the optimization problem in an alternative way.

3.3. **Newly proposed TPCA + STM.** Real tensor data, such as medical images, remote sensing images, video images and so on, have a lot of redundant information, which seriously affects the ability of the recognition of learning machine. The dimension of tensor data is higher, and the storage data requires a lot of memory space and disk space, therefore it takes a long time to train. In order to solve the above problems, support tensor machine image classification algorithm based on TPCA is proposed in this paper.

The advantage of this algorithm is that it can not only improves the recognition ability of the tensor machine, but also speeds up the learning speed of the support tensor machine. The processing procedure of the algorithm is as follows:

Step1: TPCA is used for the running set and test set to conduct feature extraction, and then obtain the tensor after feature extraction.

Step2: The tensor after feature extraction is used as input for model (4), using the Lagrange multiplier algorithm to solve the model.

Step3: The trained classifier is used to classify the test set.

4. **Experiments.** A total of six tensor databases are used in the experiments, where two of them (Yale, ORL) are second-order Face Recognition Database. And others (USF Gait17-$32\times32\times10$, USF Gait17-$128\times88\times20$, CASIA Gait-$190\times120\times20$, CASIA Gait-$240\times352\times20$) are third-order Gait Recognition Databases. To better understand the tensor structures of experimental data, we illustrate with one example for some databases that are shown in Fig.3 and Fig.4.

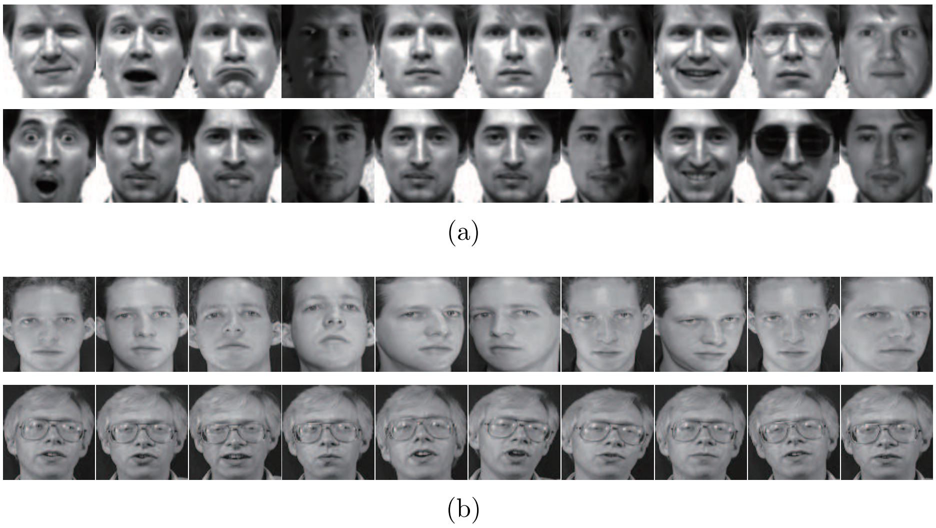

FIGURE 3. Second-order face recognition databases. (a) Yale samples. (b) ORL samples.

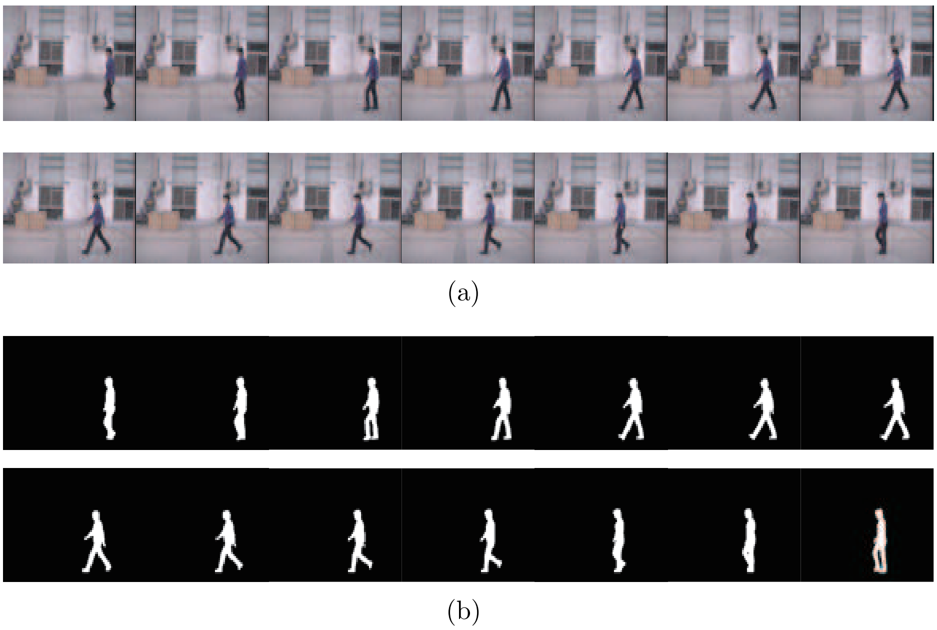

FIGURE 4. Gait silhouette sequence for third-order gait recognition database. (a) A frame sample of a video; (b) Image sequence of a frame sample.

To verify the effectiveness of the proposed tensor-based classification framework, the proposed TPCA + STM is compared with TPCA + SVM, TPCA and PCA + SVM. The difference of the four algorithms are as follows: TPCA is used to reduce the tensorial data redundancy and to maintain the tensorial structure information in high-order subspace, and then the reduced features are converted into vectors as the input of KNN for classification. TPCA + SVM is used to reduce the information redundancy in the tensor, then the tensor data is converted into vector model as the input of the SVM classifier. The TPCA + STM algorithm, which is a tensor-based dimensionality reduction method (TPCA), is used in the preprocessing of the STM classification. Specifically, analogous to TPCA + STM, PCA + SVM is used to reduce the information redundancy in the vector, then the reduced features are input to an SVM for classification.

The four algorithms, TPCA + STM, TPCA + SVM, TPCA and PCA + SVM, were compared for recognition rate and running time in different databases. The experimental results are shown in Table 1.

Table 1. Comparison of experimental results of the four algorithms

Database	Algorithms	Accuracy	Running Time
Yale	PCA + SVM	85.21%	1.383s
	TPCA	86.33%	0.732s
	TPCA + SVM	86.48%	0.642s
	TPCA + STM	87.33%	0.544s
ORL	PCA + SVM	96.50%	34.299s
	TPCA	97.75%	17.997s
	TPCA + SVM	97.84%	16.251s
	TPCA + STM	98.50%	13.208s
USFGait17-32×32×10	PCA + SVM	75.83%	20.869s
	TPCA	78.79%	12.962s
	TPCA + SVM	76.39%	12.371s
	TPCA + STM	79.60%	7.693s
USFGait17-128×88×20	PCA + SVM	75.69%	83.394s
	TPCA	79.83%	16.537s
	TPCA + SVM	77.53%	15.337s
	TPCA + STM	82.60%	8.214s
CASIA Gait-190×120×20	PCA + SVM	87.37%	218.471s
	TPCA	91.86%	19.333s
	TPCA + SVM	91.35%	18.363s
	TPCA + STM	91.67%	14.621s
CASIA Gait-240×352×20	PCA + SVM	87.37%	314.375s
	TPCA	92.27%	25.363s
	TPCA + SVM	91.89%	24.795s
	TPCA + STM	92.43%	18.836s

From table 1, we have the following observations:

1) In terms of recognition rate, the recognition rate of PCA + SVM is the smallest on all databases, and the recognition rate of TPCA + STM is the highest on the most databases except CASIA Gait-$190 \times 120 \times 20$. The recognition rates of all algorithms are increasing with the increasing of the input data on the third-order databases. PCA + SVM is compared with TPCA + STM, the recognition rate of the latter is higher than the former, where TPCA + STM has a better effect in the process of feature extraction

directly of tensor data. TPCA + SVM is compared with TPCA + STM, the recognition rate of the latter is higher than the former, where TPCA + STM has a better effect in the process of classification of tensor data.

2) In terms of running time, the running time of the four algorithms, PCA + SVM, TPCA, TPCA + SVM and TPCA + STM, is reduced in turn on the all databases. The running time of TPCA + STM is the shortest among the four algorithms when the amount of data is very large.

Table 2. Comparison of the memory of four algorithms

Input $d_1 \times d_2 \times d_3$	PCA+SVM	TPCA	TPCA+SVM	TPCA+STM
Each vector Occupied memory	$1 \times d_1^2 d_2^2 d_3^2$	$1 \times d_1 d_2 d_3$	$1 \times d_1 d_2 d_3$	$1 \times 1 \times d_1$ $1 \times d_2 \times 1$ $d_3 \times 1 \times 1$
Total memory	$d_1^2 d_2^2 d_3^2$	$d_1 d_2 d_3\ (d_1 d_2 d_3 - 1)/2$	$d_1 d_2 d_3$	$d_1 + d_2 + d_3$

The memory usage of the main operations link for the four methods were compared for the same input samples, as shown in Table 2. TPCA + STM used the smallest amount of memory, so when the amount of data is very large, the speed of TPCA + STM is fast and the running time of TPCA + STM is short.

We conclude that TPCA + STM offers improved performance for large-scale matrix data, especially if the amount of data is very large. The recognition rate of TPCA + STM is high and the running time of TPCA + STM is short, allowing rapidly and accurately image classification.

5. **Conclusion.** We have presented a new tensor-based framework for the classification of image sequence. This efficient STM extends the traditional vector-based feature representation and classification strategy to a tensor-based version. STM can be used for matrix data classification. We have used one left and two right projection vectors to formulate objective function and construct constraints. Because there are three weight vectors, STM can be classified directly for a three order tensor. In particular, tensor-based processing is naturally appropriate for image sequence, which has an intrinsic tensor data structure. Furthermore, our work reveals that information redundancy exists in the sparse and high-dimensional feature space for the data. As a result, redundancy reduction becomes a crucial issue, particularly when the tensor data representation is considered. Accordingly, we introduced an TPCA, which is a tensor-based dimensionality reduction algorithm, and then constructed a novel TPCA + STM classifier. TPCA + STM takes advantage of tensor feature extraction and at the same time solves the problem of information redundancy. The experimental results show that the proposed algorithm allows rapidly and accurately image classification with reduced computation time.

Acknowledgement. This work was supported by National Natural Science Foundation of China (61301257).

REFERENCES

[1] X. He, D. Cai, and P. Niyogi, Tensor subspace analysis, in Advances in Neural Information Processing Systems 18 (NIPS). *Cambridge, MA: MIT Press*, 2005.

[2] W. Jiang, B. R. Yang. Subspace Learning on Tensor Representation, Computer Application and System Modeling *(ICCASM), 2010 International Conference*, vol.14, pp.261-265, 2010.

[3] C. Hou, F. Nie, D. Yi, Y. Wu, Efficient image classification via multiple rank regression, *IEEE Transactions on Image Processing*, vol.22, no 1, pp.340-352, 2013.

[4] Y. Bar-Shalom, R. Osborne, P. Willett, F. Daum. Robust PCA micro-doppler classification using SVM on embedded systems, *IEEE Transactions on Aerospace and Electronic Systems*, vol.50, no 3, pp. 2304-2310, 2014.

[5] Z. Wang, S. Chen, D. Gao, A novel multi-view learning developed from single-view patterns, *Pattern Recognition*, vol.44, no 10, pp. 2395-2413, 2011.

[6] D. Tao, X. Li, W. Hu, et al. Supervised tensor learning. *In: Proceedings of the IEEE International Conference on Data Mining, Houston, Texas, Us*a, pp.450-457, 2005.

[7] D. Cai, X. He, J. Han, Learning with tensor representation, *Technical Report*, pp.2006-2716, 2006.

[8] Z. Wang, S. Chen. New least squares support vector machines based on matrix patterns, *Neural Process Letters*, vol.26, no.1, pp.41-56, 2007.

[9] Lu H., Plataniotis K. N., Venetsanopoulos A. N. MPCA: Multilinear Principal Component Analysis of Tensor Object[J], *IEEE Trans. on Pattern Neural Networks*, vol.19, no 1, pp. 18-39, 2008.

[10] C. Chen, Z. Shi-qing, C. Yue-fen, Face recognition based on MPCA. Industrial Mechatronics and Automation (ICIMA), 2*010 2nd International Conference on*, 2010

[11] Q. Li, X. Shi, D. Schonfeld. Robust HOSVD-Based Higher-Order Data Indexing and Retrieval, IEEE Signal Processing Letters, vol.20, pp.984-987, 2013.

[12] Z. Hao, L. He, B. Chen, X. Yang. A Linear Support Higher-Order Tensor Machine for Classification, *IEEE Trans. on Image Processing*, vol.22, pp.2911-2920, 2013.

[13] C. M. Chen, S. Q. Zhang, Y. F. Chen. Face recognition based on MPCA, Industrial Mechatronics and Automation (ICIMA), 2*010 2nd International Conference on*, pp.322-325, 2010.

[14] P. P. Paul, M. M. Monwar, M. L. Gavrilova. Face Detection Using Skin Color Recursive Clustering and Recognition Using Multilinear PCA, Cyberworlds (CW), *(2011) International Conference on*, pp.100-105, 2011.

[15] M. Lv, X. B. Zhao, L. J. Song, H. F. Shi, L. Jing. Least squares support tensor machine, Operations and its Applications in Engineering, Technology and Management, *2013 (ISORA 2013), 11th International Symposium on*, pp.1-6, 2013.

[16] X. Guo, X. Huang, L. Zhang, L. Zhang, A. Plaza, J. A. Benediktsson. Support Tensor Machines for Classification of Hyperspectral Remote Sensing Imagery, *IEEE Trans. on Geoscience and Remote Sensing*, pp.1-17, 2016.