Support Tucker Machines

Irene Kotsia and Ioannis Patras
School of Electronic Engineering and Computer Science
Queen Mary University of London, Mile end Campus, UK, E14NS
{irene.kotsia, i.patras} @eecs.qmul.ac.uk

Abstract

In this paper we address the two-class classification problem within the tensor-based framework, by formulating the Support Tucker Machines (STuMs). More precisely, in the proposed STuMs the weights parameters are regarded to be a tensor, calculated according to the Tucker tensor decomposition as the multiplication of a core tensor with a set of matrices, one along each mode. We further extend the proposed STuMs to the $\mathbf{\Sigma}/\mathbf{\Sigma}_w$ STuMs, in order to fully exploit the information offered by the total or the within-class covariance matrix and whiten the data, thus providing invariance to affine transformations in the feature space. We formulate the two above mentioned problems in such a way that they can be solved in an iterative manner, where at each iteration the parameters corresponding to the projections along a single tensor mode are estimated by solving a typical Support Vector Machine-type problem. The superiority of the proposed methods in terms of classification accuracy is illustrated on the problems of gait and action recognition.

1. Introduction

Images, videos and color videos are all multidimensional objects that can be regarded as 2nd, 3rd and 4th order tensors (a 3rd order tensor can be seen in Fig. 1, where the 3rd mode corresponds to the time dimension). Within the last decade, the advantages of tensorial frameworks have attracted significant interest from the research community. Several fundamental algorithms have been extended to deal with tensors, such as Multilinear Principal Component Analysis (MPCA) [9], Multilinear Discriminant Analysis (MDA) [18], Support Tensor Machines (STMs) [16], Non-negative Tensor Factorization (NTF) [19] and Canonical Analysis Correlation of tensors (CAC) [7].

The introduction of tensors in the above mentioned algorithms resulted in considerable performance improvements, due to the fact that tensors retain information regarding the high dimensional space the data lie in an efficient way. This is not the case with vector-based methods, that function by stacking the rows or columns of the original tensorial input in an arbitrary way, thus creating vectors of huge dimensionality. This fact, in combination with the small number of available samples per class can lead to small size problems, something that tensors can handle efficiently [16].

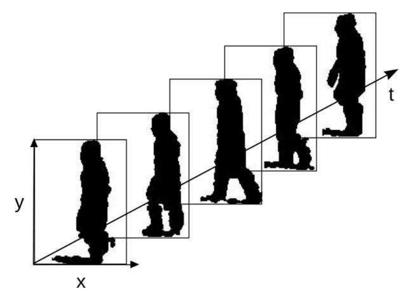

Figure 1. An example of a gait sequence as a 3rd order tensor.

In [16], the SVMs learning framework was extended to handle tensorial input. More precisely, a two class STM formulation was proposed, where the weights parameters were defined as rank one tensors, one for every mode of the input tensor. This is in contrast to the approach followed in this work where the weights parameters are defined as a tensor, that can be written as a multiplication of a core tensor with a set of matrices, one along each mode.

Vector-based classifiers that utilize a scatter matrix (either the total or the within-class) to achieve data whitening and better class separability, have been widely studied in the literature [2]. Lately, this concept has been used within the SVMs framework. More specifically, the minimization of the within-class variance has been reformulated taking under consideration that the Fisher's discriminant optimization problem for two classes is a constraint least-squares optimization problem [10]. In [15] ellipsoids were considered instead of hyperspheres in order to bound the available data, thus providing a better linear decision boundary for the classification. The SVMs methodology within the supervised tensor learning framework, was extended to handle more than one projection directions in [11, 17]. The low-rank SVMs formulation proposed in [17] minimized the rank of the projection matrix instead of the classical maximum-margin criterion. The bilinear SVMs that relaxed the orthogonality constraints on the columns of the weight matrix were proposed in [11].

In this paper we exploit the advantages of tensor-based frameworks for the classification problem. To this extent we formulate the Support Tucker Machines (STuMs), in which the weights parameters form a tensor, obtained using the Tucker tensor decomposition. This is in contrast to previous works in Support Tensor Classification that consider only rank-one tensor formulations, dealing with multiple weight vectors, one for each mode of the input tensor.

The advantages of the proposed scheme are twofold. First, the use of direct tensor representations for the parameter weights, in contrast to previous STMs that used one vector per mode, is intuitively closer to the idea of properly processing tensorial input data as the data topology is more efficiently retained. The second theoretical benefit lies in the use of the Tucker decomposition, a general decomposition that decomposes a tensor into a core tensor multiplied by a matrix along each mode (regarded as a principal component). The choice of a smaller dimension core tensor leads to dimensionality reduction taylored to the classification problem. If the factor matrices are properly chosen, the most significant "principal components" will be retained.

We proceed with proposing one extension, namely the $\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_w$ STuMs, in which the separating tensorplane takes into consideration the spread of the training data along the different tensor modes. The corresponding optimization problems are solved in an iterative manner, where at each iteration the parameters corresponding to the projections along a single tensor mode are estimated by solving a typical SVM-type optimization problem. We show the superiority of the proposed classifiers in terms of recognition accuracy on the problems of gait and action recognition.

The rest of the paper is organized as follows. In Section 2, we briefly present some useful notations that will be used throughout the paper. In Section 3, we introduce the novel Support Tucker Machines that are able to handle tensorial representations of the data. In detail, we present the two-class STuMs in Section 3.1 and introduce the $\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_w$ STuMs in Section 3.2. All of the algorithms presented in Section 3 assume that the weights are in a tensorial format, obtained using the Tucker tensor decomposition. The power of the proposed classifiers is demonstrated on the gait and actions recognition problems in Section 4. Finally, conclusions are drawn in Section 5.

2. Useful Notations in Multilinear Algebra

An n-th order tensor is a collection of measurements indexed by n indices, each one corresponding to a mode. Vectors and matrices are first and second order tensors, respectively [8]. In this paper, we will use lower case letters (e.g. x), boldface lowercase letters (e.g. $\mathbf{x}$) and boldface capital letters (e.g. $\mathbf{X}$) to denote scalars, vectors and matrices, respectively. Tensors of order 3 or higher will be denoted by boldface Euler script calligraphic letters (e.g. $\mathcal{X}$).

The i-th element of a vector $\mathbf{x} \in \mathbb{R}_+^I$ is denoted by x_i, $i = 1, 2, \ldots, I$. In a similar way, the elements of an n-th order tensor $\mathcal{X}$ will be denoted by $x_{i_1 i_2 \ldots i_n}$, $i_\ell = 1, 2, \ldots, I_\ell$, $\ell = 1, 2, \ldots, n$. To indicate the objects resulting by fixing one of the indices to a specific value, we introduce the generic subscript : and therefore denote by $\mathbf{x}_{i:}$ the i-th row of a matrix $\mathbf{X}$. Unless otherwise stated, the j-th column of a matrix $\mathbf{X}$ will be denoted compactly by $\mathbf{x}_j = \mathbf{x}_{:j}$.

The matricization (also unfolding/flattening) of a tensor is the reordering of its elements into a matrix. The n-mode matricization of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$, denoted by $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times (\prod_{k \neq n} I_k)}$, arranges the n-mode fibers to become the columns of the final matrix. Each tensor element $(i_1, i_2, \ldots, i_M)$ maps to the matrix element (i_n, j):

$$j = 1 + \sum_{k=1, k\neq n}^{M} (i_k - 1) J_k, \text{with } J_k = \prod_{l=1, l\neq n}^{k-1} I_l. \quad (1)$$

Let $\mathbf{a} \in \mathbb{R}_+^I$ and $\mathbf{b} \in \mathbb{R}_+^J$ be two non-negative real valued vectors. Their outer product yields a matrix $\mathbf{C} \in \mathbb{R}_+^{I \times J}$:

$$\mathbf{C} = \mathbf{a} \otimes \mathbf{b} \quad \text{with elements } c_{ij} = a_i b_j. \quad (2)$$

Similarly, the outer product of M vectors $\mathbf{a}_\ell \in \mathbb{R}_+^{I_\ell}$, $\ell = 1, 2, \ldots, M$, $\mathbf{a}_1 \otimes \mathbf{a}_2 \otimes \cdots \otimes \mathbf{a}_M = \bigotimes_{\ell=1}^{M} \mathbf{a}_\ell$ yields a tensor $\mathcal{A} \in \mathbb{R}_+^{I_1 \times I_2 \times \cdots \times I_M}$.

An important operation between a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ and a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_\ell}$ is the ℓ-mode product denoted by $\mathcal{X} \times_\ell \mathbf{U}$ which yields a tensor $\mathcal{Y}$ of size $I_1 \times \cdots \times I_{\ell-1} \times J \times I_{\ell+1} \times \cdots I_M$ having as elements [8]:

$$y_{i_1, \ldots, i_{\ell-1} j i_{\ell+1} i_M} = \sum_{i_\ell = 1}^{J} x_{i_1 i_2 \ldots i_M} u_{j i_\ell}, \quad j = 1, 2, \ldots, J \quad (3)$$

with $i_l = I_1, I_2, \ldots, I_l$ and $l = 1, 2, \ldots, M$. The product $\mathcal{X} \times_1 \mathbf{U}_1 \times_2 \cdots \times_M \mathbf{U}_M$ will be denoted in compact notation by $\mathcal{X} \times_{k=1}^{M} \mathbf{U}_k$. Let us also introduce the compact notation for the product of a tensor $\mathcal{X}$ with a matrix $\mathbf{U}$ in all modes besides the ℓ-mode as:

$$\mathcal{X} \overline{\times}_\ell \mathbf{U}_r \triangleq \mathcal{X} \times_1 \mathbf{U}_1 \cdots \times_{\ell+1} \mathbf{U}_{\ell+1} \times_{\ell-1} \mathbf{U}_{\ell-1} \cdots \times_M \mathbf{U}_M. \quad (4)$$

Let us define the inner product of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ as:

$$\langle \mathcal{X}, \mathcal{Y} \rangle \triangleq \sum_{i_1}^{I_1} \cdots \sum_{i_M = 1}^{I_M} x_{i_1 \ldots i_M} y_{i_1 \ldots i_M} \quad (5)$$

and the Frobenius norm of a tensor as $||\mathcal{X}, \mathcal{X}|| = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$.

In all of the classifiers that will be presented in this paper, the weights are regarded to form a tensor $\mathcal{W}$, obtained using the Tucker tensor decomposition. The weights tensor $\mathcal{W}$ is regarded to be the generalization of the weights vector $\mathbf{w}$ defined by the SVMs formulation. According to the Tucker tensor decomposition, an M order tensor $\mathcal{W} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_M}$, can be written as:

$$\begin{aligned} \mathcal{W} &= \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_n \mathbf{A}^{(M)} \\ &= [[\mathcal{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(M)}]] \end{aligned} \tag{6}$$

where $\mathcal{G}$ is a core tensor and $\mathbf{A}^{(1)} \ldots \mathbf{A}^{(M)}$ are a set of matrices that are multiplied to the core tensor $\mathcal{G}$ along each mode. The above decomposition is written elementwise as

$$w_{i_1,i_2,\ldots i_M} = \sum_{r_1,r_2,\ldots,r_M}^{R_1,R_2,\ldots,R_M} g_{r_1 r_2 \ldots r_M} \alpha^{(1)}_{i_1 r_1} \alpha^{(2)}_{i_2 r_2} \ldots \alpha^{(M)}_{i_M r_M}$$
$$\text{for} \quad i_n = 1, \ldots, I_n, n = 1, \ldots, M. \tag{7}$$

The Kronecker product of N matrices is defined as

$$\mathbf{A}_{\otimes} = \mathbf{A}^N \otimes \cdots \otimes \mathbf{A}^{(1)}. \tag{8}$$

Similarly, the Kronecker product of $N-1$ matrices (skipping the the n-th matrix $\mathbf{A}^{(n)}$) is given by

$$\mathbf{A}^{(n)}_{\otimes} = \mathbf{A}^N \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)}. \tag{9}$$

Therefore, the matricized version of (6) is

$$\begin{aligned} \mathbf{W}_{(j)} &= \mathbf{A}^{(j)} \mathbf{G}_{(j)} (\mathbf{A}^{(M)} \otimes \cdots \otimes \mathbf{A}^{(j+1)} \\ &\quad \otimes \mathbf{A}^{(j-1)} \otimes \cdots \otimes \mathbf{A}^{(1)})^T \\ &= \mathbf{A}^{(j)} \mathbf{G}_{(j)} \mathbf{A}^{(j)}_{\otimes} \end{aligned} \tag{10}$$

and the vectorized version of (10) is

$$\text{vec}(\mathbf{W}_{(1)}) = \mathbf{A}_{\otimes} \text{vec}(\mathbf{G}_{(1)}) \tag{11}$$

where vec$(\mathbf{B})$ denotes the vectorized form of the matrix $\mathbf{B}$.

The optimization problem to solve is the following

$$\min_{\mathcal{G}, \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(M)}} = \left|\left| \mathcal{W} - [[\mathcal{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, ..., \mathbf{A}^{(M)}]] \right|\right| \tag{12}$$

subject to $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_M}$, $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ and columnwise orthogonal for $n = 1, \ldots, M$.

Let us define the covariance matrices of the input tensor $\mathcal{X}$. The total scatter matrix of the j-th mode is given by:

$$\mathbf{\Sigma}_{(j)} = \frac{1}{M} \sum_{i=1}^{M} \left(\mathbf{X}_{(j)i:} - \overline{\mathbf{X}_{(j)}}\right)^T \left(\mathbf{X}_{(j)i:} - \overline{\mathbf{X}_{(j)}}\right) \tag{13}$$

where $\mathbf{X}_{(j)}$ is the matricization of $\mathcal{X}$ for the j-th mode and $\mathbf{X}_{(j)i:}$ denotes its i-th row. $\overline{\mathbf{X}_{(j)}}$ corresponds to the mean tensor. The mean of tensor samples belonging to the class $\mathcal{Q}_1$ is defined as

$$\overline{\mathbf{X}^{\mathcal{Q}_1}_{(j)}} = \frac{1}{M_{\mathcal{Q}_1}} \sum_{\mathbf{X}_{(j)i:} \in \mathcal{Q}_1} \mathbf{X}_{(j)i:}. \tag{14}$$

The mean $\overline{\mathbf{X}^{\mathcal{Q}_2}_{(j)}}$ of the tensor samples belonging to the class $\mathcal{Q}_2$ is similarly defined. Similarly, the within-class scatter matrix $\mathbf{\Sigma}_{w(j)}$ along the j-th mode is calculated as:

$$\begin{aligned} \mathbf{\Sigma}_{w,(j)} &= \sum_{\mathbf{X}_{(j)i:} \in \mathcal{Q}_1} \left(\mathbf{X}_{(j)i:} - \overline{\mathbf{X}^{\mathcal{Q}_1}_{(j)}}\right)^T \left(\mathbf{X}_{(j)i:} - \overline{\mathbf{X}^{\mathcal{Q}_1}_{(j)}}\right) \\ &\quad + \sum_{\mathbf{X}_{(j)i:} \in \mathcal{Q}_2} \left(\mathbf{X}_{(j)i:} - \overline{\mathbf{X}^{\mathcal{Q}_2}_{(j)}}\right)^T \left(\mathbf{X}_{(j)i:} - \overline{\mathbf{X}^{\mathcal{Q}_2}_{(j)}}\right). \end{aligned} \tag{15}$$

The equivalent version of the total scatter matrix when the vectorized forms of the matrices are used is given as:

$$\mathbf{\Sigma} = \frac{1}{M} \sum_{i=1}^{M} (\mathbf{x}_i - \overline{\mathbf{x}})^T (\mathbf{x}_i - \overline{\mathbf{x}}) \tag{16}$$

where $\mathbf{x}_i = \text{vec}(\mathbf{X}_{(1)i:})$ and $\overline{\mathbf{x}}$ is the mean of tensor samples. The within-class scatter matrix is similarly defined as:

$$\begin{aligned} \mathbf{\Sigma}_w &= \sum_{\mathbf{x}_i \in \mathcal{Q}_1} (\mathbf{x}_i - \widetilde{\mathbf{x}}^{\mathcal{Q}_1})(\mathbf{x}_i - \widetilde{\mathbf{x}}^{\mathcal{Q}_1})^T \\ &\quad + \sum_{\mathbf{x}_i \in \mathcal{Q}_2} (\mathbf{x}_i - \widetilde{\mathbf{x}}^{\mathcal{Q}_2})(\mathbf{x}_i - \widetilde{\mathbf{x}}^{\mathcal{Q}_2})^T \end{aligned} \tag{17}$$

where $\mathbf{x}_i = \text{vec}(\mathbf{X}_{(1)i:})$, $\widetilde{\mathbf{x}}^{\mathcal{Q}_1} = \frac{1}{M_{\mathcal{Q}_1}} \sum_{\mathbf{x}_i \in \mathcal{Q}_1} \mathbf{x}_i$ and $\widetilde{\mathbf{x}}^{\mathcal{Q}_2} = \frac{1}{M_{\mathcal{Q}_2}} \sum_{\mathbf{x}_i \in \mathcal{Q}_2} \mathbf{x}_i$.

3. Support Tucker Machines

Let us consider a dataset consisting of L samples as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{M+1}}$. Every sample is a tensor of order M denoted by $\mathcal{X}_{i_n:} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$, $i_n = 1, 2, \ldots, L$, indexed by M indices $(i_1, i_2, \ldots, i_M)$ and belonging to one of the two classes, $\mathcal{Q}_1$ and $\mathcal{Q}_2$. This dataset can be built of action image sequences, forming in that way a 4-th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$, where I_1 and I_2 refer to the image dimensions (height and width), respectively, I_3 corresponds to the number of images in every tensor sample and I_4 is the number of action samples in the database. In the remainder of the paper, with a slight abuse of notation we will use $\mathcal{Q}_1$ and $\mathcal{Q}_2$ to refer not only to the first and second class, but also to the tensor samples they include.

3.1. Support Tucker Machines

In this Section we will present the novel Support Tucker Machines (STuMs) in which the weights parameters form a tensor $\mathcal{W}$ that is obtained using the Tucker tensor decomposition. STuMs aim at finding a multilinear decision function $g : \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M} \rightarrow [-1, 1]$, that during testing, classifies a test tensor $\mathcal{Y}\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ and is of the following form:

$$g(\mathcal{Y}) = \text{sign}\left(\langle \mathcal{Y}, \mathcal{W} \rangle + b\right). \tag{18}$$

The weights parameters form a tensor $\mathcal{W}$ that is estimated by solving the following problem:

$$\min_{\mathcal{W}, b, \boldsymbol{\xi} \geq 0} \frac{1}{2} \langle \mathcal{W}, \mathcal{W} \rangle + C \sum_{i=1}^{I_{M+1}} \xi_i \tag{19}$$

$$\text{s.t.} \quad y_i \left[\langle \mathcal{W}, \mathcal{X}_{i:} \rangle + b \right] \geq 1 - \xi_i, \quad 1 \leq i \leq I_{M+1}, \ \xi_i \geq 0 \tag{20}$$

where b is the bias term, $\boldsymbol{\xi} = [\xi_i, \ldots, \xi_w]$ is the slack variable vector and C is the term that controls the relative importance of penalizing the training errors. It should be noted here that the above problem is not convex with respect to all parameters in $\mathcal{W}$. Therefore we adopt an iterative scheme in which at each iteration we solve only for the parameters that are associated with the j-th mode (i.e $\mathbf{W}^{(j)}$) of the parameters tensor $\mathcal{W}$, while keeping all the other parameters fixed. More specifically, the optimization problem at the iteration for the j-th mode is given by:

$$\min_{\mathbf{W}_{(j)}, b, \boldsymbol{\xi} \geq 0} \frac{1}{2} \text{Tr}\left[\mathbf{W}_{(j)} \mathbf{W}_{(j)}^T \right] + C \sum_{i=1}^{I_{M+1}} \xi_i \tag{21}$$

$$\text{s.t.} \quad y_i \left[\text{Tr}(\mathbf{W}_{(j)}, \mathbf{X}_{(j)i:}^T) + b \right] \geq 1 - \xi_i, \quad 1 \leq i \leq I_{M+1}, \ \xi_i \geq 0. \tag{22}$$

Since the weights tensor $\mathcal{W}$ is obtained using the Tucker decomposition (6), the minimization problem defined in (21) is rewritten as:

$$\min_{\mathbf{A}^{(j)}, b, \boldsymbol{\xi} \geq 0} \frac{1}{2} \text{Tr}\left[\mathbf{A}^{(j)} \mathbf{G}_{(j)} {\mathbf{A}_{\otimes}^{(j)}}^T \mathbf{A}_{\otimes}^{(j)} {\mathbf{G}_{(j)}}^T (\mathbf{A}^{(j)})^T \right] + C \sum_{i=1}^{I_{M+1}} \xi_i \tag{23}$$

$$\text{s.t.} \quad \frac{1}{2} y_i \left[\text{Tr}\left(\mathbf{A}^{(j)} \mathbf{G}_{(j)} {\mathbf{A}_{\otimes}^{(j)}}^T \mathbf{X}_{(j)i:}^T \right) + b \right] \geq 1 - \xi_i, \tag{24}$$

$$1 \leq i \leq I_{M+1}, \ \xi_i \geq 0.$$

We will now proceed in reformulating the problem defined in (23) such that it can be solved using a classic vector-based SVM implementation. To this extend, we set $\mathbf{P}^{(j)} = \mathbf{A}_{\otimes}^{(j)} \mathbf{G}^{(j)}$. Then (23) is rewritten as

$$\min_{\mathbf{A}^{(j)}, b, \boldsymbol{\xi} \geq 0} \frac{1}{2} \text{Tr}\left[\mathbf{A}^{(j)} \mathbf{P}^{(j)} {\mathbf{P}^{(j)}}^T (\mathbf{A}^{(j)})^T \right] + C \sum_{i=1}^{I_{M+1}} \xi_i \tag{25}$$

$$\text{s.t.} \quad \frac{1}{2} y_i \left[\text{Tr}\left(\mathbf{A}^{(j)} \mathbf{P}^{(j)} \mathbf{X}_{(j)i:}^T \right) + b \right] \geq 1 - \xi_i, \tag{26}$$

$$1 \leq i \leq I_{M+1}, \ \xi_i \geq 0.$$

Let us define $\mathbf{K} = \mathbf{P}^{(j)} {\mathbf{P}^{(j)}}^T$. By definition, $\mathbf{K}$ is a positive definite matrix. Also, let $\widetilde{\mathbf{A}}^{(j)} = \mathbf{A}^{(j)} \mathbf{K}^{\frac{1}{2}}$. Then,

$$\begin{aligned} \text{Tr}\left[\mathbf{A}^{(j)} \mathbf{P}^{(j)} {\mathbf{P}^{(j)}}^T (\mathbf{A}^{(j)})^T \right] &= \text{Tr}\left[\widetilde{\mathbf{A}}^{(j)} (\widetilde{\mathbf{A}}^{(j)})^T \right] \\ &= \text{vec}(\widetilde{\mathbf{A}}^{(j)})^T \text{vec}(\widetilde{\mathbf{A}}^{(j)}). \end{aligned} \tag{27}$$

By letting $\widetilde{\mathbf{X}}_{(j)i:} = \mathbf{X}_{(j)i:} \mathbf{P}^{(j)} \mathbf{K}^{-\frac{1}{2}}$ we have

$$\begin{aligned} \text{Tr}\left[\mathbf{A}^{(j)} \mathbf{P}^{(j)} \mathbf{X}_{(j)i:}^T \right] &= \text{Tr}\left[\widetilde{\mathbf{A}}^{(j)} \widetilde{\mathbf{X}}_{(j)i:}^T \right] \\ &= \text{vec}(\widetilde{\mathbf{A}}^{(j)})^T \text{vec}(\widetilde{\mathbf{X}}_{(j)i:}). \end{aligned} \tag{28}$$

Then, (25) is written as

$$\min_{\text{vec}(\widetilde{\mathbf{A}}^{(j)}), b, \boldsymbol{\xi} \geq 0} \frac{1}{2} \text{vec}(\widetilde{\mathbf{A}}^{(j)})^T \text{vec}(\widetilde{\mathbf{A}}^{(j)}) + C \sum_{i=1}^{I_{M+1}} \xi_i \tag{29}$$

$$\text{s.t.} \quad y_i \left[\text{vec}(\widetilde{\mathbf{A}}^{(j)})^T \text{vec}(\widetilde{\mathbf{X}}_{(j)i:}) + b \right] \geq 1 - \xi_i,$$

$$1 \leq i \leq I_{M+1}, \ \xi_i \geq 0.$$

Thus, the optimization problem for the j-th mode in (23) is formulated as a SVM problem (29) with respect to $\widetilde{\mathbf{A}}^{(j)}$.

After solving for $\mathbf{A}^{(1)}, \ldots \mathbf{A}^{(N)}$ we solve for $\mathcal{G}$. $\mathcal{G}$ is found by solving the minimization problem defined below:

$$\min_{\mathbf{G}_{(1)}, b, \boldsymbol{\xi} \geq 0} \frac{1}{2} \left(\mathbf{A}_{\otimes} \text{vec}(\mathbf{G}_{(1)}) \right)^T \left(\mathbf{A}_{\otimes} \text{vec}(\mathbf{G}_{(1)}) \right) + C \sum_{i=1}^{I_{M+1}} \xi_i \tag{30}$$

$$\text{s.t.} \quad y_i \left[\left(\mathbf{A}_{\otimes} \text{vec}(\mathbf{G}_{(1)}) \right)^T \text{vec}(\mathbf{X}_{(1)i:}) + b \right] \geq 1 - \xi_i, \tag{31}$$

$$1 \leq i \leq I_{M+1}, \ \xi_i \geq 0,$$

where with $\text{vec}(\mathbf{G}_{(1)})$ we denote the vectorized form of the matricization of $\mathcal{G}$ for the first dimension. We can also use another matricization of the tensor $\mathcal{G}$ taking under consideration that we will then have to change the multiplication order of $\mathbf{A}_{\otimes}$, accordingly. Then (30) is rewritten as

$$\min_{\mathbf{G}_{(1)}, b, \boldsymbol{\xi} \geq 0} \text{vec}(\mathbf{G}_{(1)})^T {\mathbf{A}_{\otimes}}^T \mathbf{A}_{\otimes} \text{vec}(\mathbf{G}_{(1)}) + C \sum_{i=1}^{I_{M+1}} \xi_i \tag{32}$$

$$\text{s.t.} \quad y_i \left[\text{vec}(\mathbf{G}_{(1)})^T \mathbf{A}_{\otimes} \text{vec}(\mathbf{X}_{(j)i:}) + b \right] \geq 1 - \xi_i, \tag{33}$$

$$1 \leq i \leq I_{M+1}, \ \xi_i \geq 0.$$

Let us define as $\mathbf{Q} = \mathbf{A}_{\otimes}{}^T\mathbf{A}_{\otimes}$, $\widetilde{\mathbf{V}} = \mathbf{Q}^{\frac{1}{2}}\text{vec}(\mathbf{G}_{(1)})$ and $\widetilde{\mathbf{X}}_{(j)i:} = \mathbf{Q}^{-\frac{1}{2}}\mathbf{X}_{(j)i:}$. Then (32) is rewritten as

$$\min_{\mathbf{G}^{(j)},b,\boldsymbol{\xi}\geq 0} \text{vec}(\widetilde{\mathbf{V}})^T\text{vec}(\widetilde{\mathbf{V}}) + C\sum_{i=1}^{I_{M+1}}\xi_i \tag{34}$$

$$\text{s.t.} \quad y_i\left[\text{vec}(\widetilde{\mathbf{V}}^T)\text{vec}(\widetilde{\mathbf{X}}_{(j)i:}) + b\right] \geq 1-\xi_i, \tag{35}$$
$$1 \leq i \leq I_{M+1},\ \xi_i \geq 0.$$

For completeness, we present the derivations that optimize (23) directly with respect to $\mathbf{A}^{(j)}$. The optimal $\mathbf{A}^{(j)}$ can be found at the saddle point of the Lagrangian

$$L^{(j)}_{STMs}(\mathbf{A}^{(j)}, b, \boldsymbol{\xi}) = \frac{1}{2}\text{Tr}\left[\mathbf{A}^{(j)}\mathbf{P}^{(j)}\mathbf{P}^{(j)T}\mathbf{A}^{(j)T}\right] + C\sum_{i=1}^{I_{M+1}}\xi_i - \sum_{i=1}^{I_{M+1}}\alpha_i^j\left(y_i\left[\text{Tr}\left(\mathbf{A}^{(j)}\mathbf{P}^{(j)}\mathbf{X}^T_{(j)i:}\right) + b\right] - 1 + \xi_i\right) - \sum_{i=1}^{I_{M+1}}\kappa_i\xi_i \tag{36}$$

that is by solving

$$\nabla_{\mathbf{A}^{(j)}}L^{(j)}_{STMs} = 0 \Rightarrow$$
$$\mathbf{A}^{(j)} = \textstyle\sum_{i=1}^{I_{M+1}}\alpha_i^j y_i\mathbf{X}_{(j)i:}\mathbf{P}^{(j)}\left[\mathbf{P}^{(j)}\mathbf{P}^{(j)T}\right]^{-1}. \tag{37}$$

The Lagrangian multipliers α_i^j can be found by solving the dual problem of (36), that is

$$\max_{0\leq\alpha_i^j\leq C} -\frac{1}{2}\sum_{i=1}^{I_{M+1}}\sum_{k=1}^{I_{M+1}}\alpha_i^j\alpha_k^j y_i y_k \text{Tr}\left(\mathbf{P}^{(j)T}\mathbf{P}^{(j)}\mathbf{P}^{(j)T}\mathbf{P}^{(j)}\mathbf{X}^T_{(j)i:}\mathbf{X}_{(j)i:}\right) + \sum_{i=1}^{I_{M+1}}\alpha_i^j$$
$$\text{s.t.} \quad \boldsymbol{\alpha}^{jT}\mathbf{y} = 0, \tag{38}$$

where $\boldsymbol{\alpha}^j = [\alpha_1^j, \alpha_2^j, \ldots, \alpha^j_{I_{M+1}}]$. It should be mentioned here that the whole procedure is repeated iteratively for every mode, so as to find $\mathbf{A}^{(j)}$, $j = 1\ldots M$.

We also present the equivalent derivations (Lagrangian and dual problems) that optimize (30) directly with respect to $\mathcal{G}$. The optimal tensor $\mathcal{G}$ can be found at the saddle point of the Lagrangian

$$L_{STMs}(\mathcal{G}, b, \boldsymbol{\xi}) = \frac{1}{2}\left[\text{vec}(\mathbf{G}_{(1)})^T\mathbf{A}_{\otimes}{}^T\mathbf{A}_{\otimes}\text{vec}(\mathbf{G}_{(1)})\right] + C\sum_{i=1}^{I_{M+1}}\xi_i - \sum_{i=1}^{I_{M+1}}\alpha_i\left(y_i\left[(\text{vec}(\mathbf{G}_{(1)})^T\mathbf{A}_{\otimes}\text{vec}(\mathbf{X}_{(1)i:}) + b\right] - 1 + \xi_i\right) - \sum_{i=1}^{I_{M+1}}\kappa_i\xi_i \tag{39}$$

that is by solving

$$\nabla_{\mathbf{G}}L_{STMs} = 0 \Rightarrow$$
$$\text{vec}(\mathbf{G}_{(1)}) = \tfrac{1}{2}\textstyle\sum_{i=1}^{I_{M+1}}\alpha_i y_i\left[\mathbf{A}_{\otimes}^T\mathbf{A}_{\otimes}\right]^{(-1)}\mathbf{X}_{(j)i:}. \tag{40}$$

The Lagrangian multipliers α_i for that problem can be found by solving the dual problem of (39), that is

$$\max_{0\leq\alpha_i\leq C} -\frac{1}{2}\sum_{i=1}^{I_{M+1}}\sum_{k=1}^{I_{M+1}}\alpha_i\alpha_k y_i y_k \text{vec}(\mathbf{G}_{(1)})^T\left[\mathbf{A}_{\otimes}^T\mathbf{A}_{\otimes}\right]^{(-1)}\text{vec}(\mathbf{G}_{(1)})$$
$$\text{s.t.} \quad \boldsymbol{\alpha}^T\mathbf{y} = 0, \tag{41}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_{I_{M+1}}]$.

3.2. $\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_w$ Support Tucker Machines

In this Section we will present the novel $\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_w$ STuMs, that are an extension of the STuMs proposed in Section 3.1. More precisely, the $\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_w$ STuMs consider the data spread and whiten the data using a covariance matrix. In that way, a transformed space is created containing data that are closer to the corresponding class mean and classes that are further apart from each other, thus being more separable.

Let us use $\mathbf{S}_{(j)}$ to refer either to the total scatter matrix $\boldsymbol{\Sigma}_{(j)}$ or to the within-class scatter matrix $\boldsymbol{\Sigma}_{w_{(j)}}$ for the j-th mode, respectively. Then, the formulation of $\mathbf{S}-$ STuMs for the j-th mode is given by

$$\min_{\mathbf{G}^{(j)},b,\boldsymbol{\xi}\geq 0} \frac{1-D}{2}\text{Tr}\left[\mathbf{A}^{(j)}\mathbf{P}^{(j)}\mathbf{P}^{(j)T}\mathbf{A}^{(j)T}\right] + \frac{D}{2}\text{Tr}\left[\mathbf{A}^{(j)}\mathbf{P}^{(j)}\mathbf{S}_{(j)}\mathbf{P}^{(j)T}(\mathbf{A}^{(j)})^T\right] + C\sum_{i=1}^{I_{M+1}}\xi_i \tag{42}$$

$$\text{s.t.} \quad y_i\text{Tr}\left[\mathbf{A}^{(j)}(\mathbf{P}^{(j)})(\mathbf{X}_{(j)i:} - \overline{\mathbf{X}_{(j)}})^T + b\right] \geq 1 - \xi_i$$
$$1 \leq i \leq I_{M+1}, \xi_i \geq 0 \tag{43}$$

where $0 \leq D \leq 1$ is the parameter that balances the two regularization terms $\mathbf{A}^{(j)}\mathbf{P}^{(j)}\mathbf{P}^{(j)T}\mathbf{A}^{(j)T}$ and $\mathbf{A}^{(j)}\mathbf{P}^{(j)}\mathbf{S}_{(j)}\mathbf{P}^{(j)T}(\mathbf{A}^{(j)})^T$.

Following derivations that are similar to the ones presented in (27)-(29), we can arrive at the classic vector-based SVM formulation in (29) with a slightly different definition for $\mathbf{K}$. More precisely, we now define as $\mathbf{K} = \mathbf{P}^{(j)}\left[(1-D)\mathbf{I} + D\mathbf{S}_{(j)}\right]^{\frac{1}{2}}\mathbf{P}^{(j)^T}$. The definition for $\widetilde{\mathbf{A}}^{(j)}$ still remains as $\widetilde{\mathbf{A}}^{(j)} = \mathbf{A}^{(j)}\mathbf{K}^{\frac{1}{2}}$. Then,

$$\begin{aligned}&\mathrm{Tr}\left[\mathbf{A}^{(j)}\mathbf{P}^{(j)}\left[(1-D)\mathbf{I} + D\mathbf{S}_{(j)}\right]\mathbf{P}^{(j)^T}\mathbf{A}^{(j)^T}\right] \\ &= \mathrm{Tr}\left[\widetilde{\mathbf{A}}^{(j)}(\widetilde{\mathbf{A}}^{(j)})^T\right] = \mathrm{vec}(\widetilde{\mathbf{A}}^{(j)})^T\mathrm{vec}(\widetilde{\mathbf{A}}^{(j)}). \quad (44)\end{aligned}$$

By letting $\widetilde{\mathbf{X}}_{(j)i:} = \mathbf{X}_{(j)i:}\mathbf{P}^{(j)}\mathbf{K}^{-\frac{1}{2}}$ we have

$$\begin{aligned}\mathrm{Tr}\left[\mathbf{A}^{(j)}\mathbf{P}^{(j)}\mathbf{X}^T_{(j)i:}\right] &= \mathrm{Tr}\left[\widetilde{\mathbf{A}}^{(j)}\widetilde{\mathbf{X}}^T_{(j)i:}\right] \\ &= \mathrm{vec}(\widetilde{\mathbf{A}}^{(j)})^T\mathrm{vec}(\widetilde{\mathbf{X}}_{(j)i:}).\end{aligned} \quad (45)$$

Then, (44) is written as (29). After finding $\widetilde{\mathbf{A}}^{(j)}$, we solve for $\mathcal{G}$. More precisely, $\mathcal{G}$ is found by solving the minimization problem defined below:

$$\begin{aligned}\min_{\mathbf{G}_{(1)},b,\boldsymbol{\xi}\geq 0} \; & \frac{1-D}{2}\left(\mathbf{A}_{\otimes}\mathrm{vec}(\mathbf{G}_{(1)})\right)^T\left(\mathbf{A}_{\otimes}\mathrm{vec}(\mathbf{G}_{(1)})\right) \\ & + \frac{D}{2}\left(\mathbf{A}_{\otimes}\mathrm{vec}(\mathbf{G}_{(1)})\mathbf{S}^{\frac{1}{2}}\right)^T\left(\mathbf{A}_{\otimes}\mathrm{vec}(\mathbf{G}_{(1)})\mathbf{S}^{\frac{1}{2}}\right) \\ & + C\sum_{i=1}^{I_{M+1}}\xi_i \quad (46)\end{aligned}$$

$$\begin{aligned}\text{s.t.} \quad & y_i\left[\left(\mathbf{A}_{\otimes}\mathrm{vec}(\mathbf{G}_{(1)})\right)^T \right. \\ & \left.\left(\mathrm{vec}(\mathbf{X}_{(1)i:}) - \mathrm{vec}(\overline{\mathbf{X}_{(1)i:}})\right) + b\right] \geq 1 - \xi_i, \quad (47) \\ & 1 \leq i \leq I_{M+1}, \; \xi_i \geq 0,\end{aligned}$$

where with $\mathrm{vec}(\mathbf{G}_{(1)})$ refers to the vectorized form of the matricization of $\mathcal{G}$ for the first dimension and $\mathbf{S}$ refers to the vectorized versions of the total or the within-class scatter matrices, defined as in (16) and (17), respectively. Then (46) is rewritten as

$$\begin{aligned}\min_{\mathbf{G}_{(1)},b,\boldsymbol{\xi}\geq 0} \; & \mathrm{vec}(\mathbf{G}_{(1)})^T\mathbf{A}_{\otimes}{}^T\mathbf{A}_{\otimes}\mathrm{vec}(\mathbf{G}_{(1)}) \\ & + \mathrm{vec}(\mathbf{G}_{(1)})^T\mathbf{A}_{\otimes}{}^T\mathbf{S}\mathbf{A}_{\otimes}\mathrm{vec}(\mathbf{G}_{(1)}) + C\sum_{i=1}^{I_{M+1}}\xi_i \quad (48)\end{aligned}$$

$$\begin{aligned}\text{s.t.} \quad & y_i\left[\mathrm{vec}(\mathbf{G}_{(1)})^T\mathbf{A}_{\otimes}\right. \\ & \left.\left(\mathrm{vec}(\mathbf{X}_{(1)i:}) - \mathrm{vec}(\overline{\mathbf{X}_{(1)i:}})\right) + b\right] \geq 1 - \xi_i, \quad (49) \\ & 1 \leq i \leq I_{M+1}, \; \xi_i \geq 0.\end{aligned}$$

Let us define as $\mathbf{Q} = \mathbf{A}_{\otimes}{}^T\left[(1-D)\mathbf{I} + D\mathbf{S}\right]^{\frac{1}{2}}\mathbf{A}_{\otimes}$, $\widetilde{\mathbf{V}} = \mathbf{Q}^{\frac{1}{2}}\mathrm{vec}(\mathbf{G}_{(1)})$ and $\widetilde{\mathbf{X}}_{(j)i:} = \mathbf{Q}^{-\frac{1}{2}}\mathbf{X}_{(j)i:}$. Then (48) is rewritten as

$$\min_{\mathbf{G}^{(j)},b,\boldsymbol{\xi}\geq 0} \; \mathrm{vec}(\widetilde{\mathbf{V}})^T\mathrm{vec}(\widetilde{\mathbf{V}}) + C\sum_{i=1}^{I_{M+1}}\xi_i \quad (50)$$

$$\begin{aligned}\text{s.t.} \quad & y_i\left[\mathrm{vec}(\widetilde{\mathbf{V}}^T)\mathrm{vec}(\widetilde{\mathbf{X}}_{(j)i:}) + b\right] \geq 1 - \xi_i, \quad (51) \\ & 1 \leq i \leq I_{M+1}, \; \xi_i \geq 0.\end{aligned}$$

We should note here that if we set $D = 0$ in (42) we obtain the proposed STuMs. Similarly, if we set $D = 1$ and consider $\mathbf{S}_{(j)}$ to be the within-class covariance matrix $\mathbf{\Sigma}_{w_{(j)}}$, we obtain the Minimum Class Variance STuMs. The latter can be regarded as an extension of the Minimum Class Variance SVMs that were proposed in [20].

4. Experimental Results

In this section, we will present the experiments conducted in order to prove the superiority of the proposed STuMs and $\mathbf{\Sigma}/\mathbf{\Sigma}_w$ STMs over vector-based methods and classical STMs, in terms of classification accuracy. To this extend, we addressed the gait and human action recognition using publicly available databases.

We should mention here, that both the gait and human action recognition problems require multiclass classification methods. The classical STMs, however, as well as the proposed STuMs and $\mathbf{\Sigma}/\mathbf{\Sigma}_w$ STMs, deal with two-class classification problems. Therefore, in order to deal with multiclass problems, we have to combine several two-class classifiers, trained in a one-against-all manner. The final decision regarding the class the test sample belongs to is taken either using a voting scheme or by assigning each test sample to a class k, such that the distance of the test sample to the separating tensorplane of the class k is minimum over the distances to the tensorplanes of all other classes.

4.1. Gait recognition

The database used for the gait recognition experiments is the USF HumanID Gait Challenge data set version 1.7, as provided in [9], so as to allow easy comparison to previous works. The database consists of 452 sequences of 74 subjects walking in elliptical paths in front of the camera. Three variations are available: viewpoint (left/right), shoe type (two different types) and surface type (grass/concrete). Seven Probe Sets are provided, each one consisting of 71 people, of increasing complexity/difficulty with probe set A being the easiest and probe set G being the most difficult one. There are no common sequences between the gallery sets and any of the probe sets. Each probe set is unique.

The input features for our algorithms consist of the silhouettes extracted from the gait sequences, like the one depicted in Fig. 1. The classifiers were trained using the

Gal dataset and tested using each one of the A, B, C, D, E, F and G datasets, following the protocol described in [13]. In order to calculate the accuracy of gait recognition for a probe sequence $probe$ consisting of N_{probe} samples (gait periods) against a gallery sequence $gallery$ consisting of $N_{gallery}$ samples, we calculate the distance of that particular probe sample from each of the separating tensorplanes of each class of the gallery samples. For the calculation of a symmetric dissimilarity measure between a gallery sequence and a probe sequence (class) we also calculate the distance of each gallery sample to the separating tensorplane of each class of each probe sample. The dissimilarity measure between a probe sequence $probe$ and a gallery sequence $gallery$ is chosen to be the minimum of the calculated distances. Finally, the probe sequence $probe$ is assigned to the gallery sequence class to which is has the smallest dissimilarity.

In Table 1 we present the results acquired for the proposed methods as well as for various methods presented in the literature. We also present the equivalent results when the vectorized version of the tensorial input was used as input to SVMs, for comparison reasons. The results are evaluated using the Cumulative Scores 1 and 5 (CS-1 / CS-5, respectively) similarly to [3][9]. The results reported are for the core tensor being of dimension $18 \times 18 \times 18$. As we can see in Table 1 tensors largely outperform vectors, something that was highly expected due to the high dimensionality of the data and the presence of few available examples for every class. We can also see that the proposed STuMs outperform the classical STMs in terms of recognition accuracy, while at the same time the proposed $\mathbf{\Sigma}/\mathbf{\Sigma}_w$ STMs achieve the best recognition results. The best accuracies achieved for each Probe Set are highlighted in bold.

4.2. Human action recognition

For the action recognition experiments conducted we used the commonly used KTH dataset [14]. The KTH Action Dataset depicts 25 subjects performing 6 different activities namely: “boxing”, “handclapping”, “handwaving”, “jogging”, “running” and “walking”. The data were collected in four different settings corresponding to indoors recordings, outdoor recordings with scale variations (camera zoom in and out) and outdoor recordings with the subjects wearing different clothes. Since the background remains the same through the entire video, for each case Scenario we have extracted features after performing background modelling and subtraction. On the resulting images the magnitude of the gradient was calculated as shown in Fig. 2. The leave-one-person-out cross-validation approach was used to test the performance of the classifiers.

The results reported are for the core tensor being of dimension $6 \times 6 \times 6$. The recognition accuracies achieved when the proposed STuMs and $\mathbf{\Sigma}/\mathbf{\Sigma}_w$ STuMs were used are

Figure 2. An example of a person walking from the KTH dataset.

equal to 92.3% and 95.3%, respectively. The equivalent results when the vectorized version of the tensorial input was used as input to SVMs and when classical STMs were used, were significantly lower (88.5% and 89.3%, respectively). In Table 2, previous results as reported in the literature for the KTH database are presented, for comparison reasons.

In order to give some further insight on where the misclassifications take place, we report the confusion matrices. The confusion matrices calculated for for classical STMs, the proposed STuMs and the proposed $\mathbf{\Sigma}/\mathbf{\Sigma}_w$ STuMs are shown in Table 3. For brevity, when all methods achieve the same performance we report a single number. As can be seen, the use of STuMs and $\mathbf{\Sigma}/\mathbf{\Sigma}_w$ STuMs improves the results by 3.0% and 6.0%, respectively.

Table 3. Confusion matrices of STMs, STuMs /($\mathbf{\Sigma}/\mathbf{\Sigma}_w$) STuMs for the KTH action database.

$Ac_{ac} \backslash Ac_{cl}$	Box	Clap	Wave	Jog	Run	Walk
Box	84/86/92	8/4/0	8/6/4	0	0	0
Clap	12/10//8	76/84/92	12/6/4	0	0	0
Wave	4/4/0	16/12/8	80/88/92	0	0	0
Jog	0	0	0	96	0	0
Run	0	0	0	0	100	0
Walk	0	0	0	4	0	100

5. Conclusions

In this work, we addressed the two-class classification problem within the tensor-based framework by proposing the Support Tucker Machines, in which the weights parameters are obtained using the Tucker tensor decomposition. We also proposed the $\mathbf{\Sigma}/\mathbf{\Sigma}_w$ STuMs that use the total of within-class scatter matrix in order to exploit class information and provide invariance to affine transformations. For both of the proposed classifiers, the corresponding optimization problems were solved in an iterative way, such that a typical SVM formulation could be employed. The efficiency of the proposed methods was illustrated on the problems of gait and action recognition.

6. Acknowledgement

This work was supported by the EPSRC grant ’Recognition and Localization of Human Actions in Image Sequences’

Table 1. Comparison of proposed methods with state of the art (CS-1 / CS-5).

Probe Set	Baseline [13]	HMM [6]	LTN [3]	GEI [5]	ETG [9]	ETGLDA [9]	SVMs	STMs	STuMs	Σ/Σ_w STuMs
A	79/96	99/**100**	94/99	**100**/**100**	92/96	99/**100**	80/97	92/**100**	99/**100**	**100**/**100**
B	66/81	**89**/90	83/85	85/85	85/90	88/93	79/93	81/90	85/93	87/**95**
C	56/76	78/90	78/83	80/88	76/81	**83**/88	68/85	73/88	79/90	81/**91**
D	29/61	35/65	33/65	30/55	39/55	36/71	30/54	47/67	53/71	**55**/**74**
E	24/55	29/65	24/67	33/55	29/52	29/60	23/46	48/79	63/86	**65**/**90**
F	30/46	18/60	17/58	21/41	21/58	21/59	24/49	29/49	42/63	**44**/**66**
G	10/33	24/50	21/48	29/48	21/50	21/60	12/37	31/71	52/87	**54**/**90**
Mean	42/64	53/74	50/72	54/67	52/69	54/76	45/62	57/68	68/84	**69**/**87**

Table 2. Accuracies achieved by various methods for the KTH action database.

Method	[4]	[12]	[1]	[7]	SVMs	STMs	STuMs	Σ/Σ_w STuMs
Accuracy	90.5%	91.7%	87.7%	95.3%	88.5 %	89.3%	92.3 %	95.3%

(EP/G033935/1).

References

[1] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1, 2010. 8

[2] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997. 1

[3] N. V. Boulgouris, K. N. Plataniotis, and D. Hatzinakos. Gait recognition using linear time normalization. *Pattern Recognition*, 39(5):969–979, February 2006. 7, 8

[4] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. *IEEE Conference on Computer Vision and Pattern Recognition*, 41:1–8, 2008. 8

[5] J. Han and B. Bhanu. Individual recognition using gait energy image. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(2):316–322, February 2006. 8

[6] A. Kale, A. N. Rajagopalan, A. Sunderesan, A. R.-C. N. Cuntoor, V. Krueger, and R. Chellappa. Identification of humans using gait. *IEEE Trans. on Image Processing*, pages 1163–1173, September 2004. 8

[7] T.-K. Kim and R. Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(8):1415–1428, August 2009. 1, 8

[8] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, September 2009. 2

[9] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Mpca: Multilinear principal component analysis of tensor objects. *IEEE Trans. on Neural Networks*, 19(1):18–39, 2008. 1, 6, 7, 8

[10] S. Mika, R. G., J. Weston, B. Scholkopf, A. Smola, and K.-R. Muller. Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5):623–628, 2003. 1

[11] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Bilinear classifiers for visual recognition. In *Neural Information Processing Systems*, December 2009. 1, 2

[12] K. Rapantzikos, Y. Avrithis, and S. Kollias. Dense saliency-based spatiotemporal feature points for action recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 8

[13] S. Sarkar, P. J. Phillips, Z. Liu, I. Robledo, P. Grother, and K. W. Bowyer. The human id gait challenge problem: Data sets, performance, and analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(2):162–177, 2005. 7, 8

[14] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. *International Conference on Pattern Recognition*, 2004. 7

[15] P. Shivaswamy and T. Jebara. Ellipsoidal kernel machines. In M. Meila and X. Shen, editors, *11th International Conference on Artificial Intelligence and Statistics*, volume 2 of JMLR: W&CP, March 2007. 1

[16] D. Tao, X. Li, X. Wu, W. Hu, and S. J. Maybank. Supervised tensor learning. *Knowledge and Information Systems*, 13(1):1–42, 2007. 1

[17] L. Wolf, H. Jhuang, and T. Hazan. Modeling appearances with low-rank svm. *IEEE Conf. Computer Vision and Pattern Recognition*, 2007. 1

[18] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. J. Zhang. Multilinear discriminant analysis for face recognition. *IEEE Trans. on Image Processing*, 16(1):212–220, 2007. 1

[19] S. Zafeiriou. Discriminant nonnegative tensor factorization algorithms. *IEEE Trans. on Neural Networks*, 20(2):217–235, 2009. 1

[20] S. Zafeiriou, A. Tefas, and I. Pitas. Minimum class variance support vector machines. *IEEE Trans. on Image Processing*, 16(10):2551–2564, 2007. 6