A Novel Time Series Kernel for Sequences Generated by LTI Systems

Liliana Lo Presti, Marco La Cascia

V.le delle Scienze Ed.6, DIID, Universitá degli studi di Palermo, Italy

Abstract. The recent introduction of Hankelets to describe time series relies on the assumption that the time series has been generated by a vector autoregressive model (VAR) of order p. The success of Hankelet-based time series representations prevalently in nearest neighbor classifiers poses questions about if and how this representation can be used in kernel machines without the usual adoption of mid-level representations (such as codebook-based representations). It is also of interest to investigate how this representation relates to probabilistic approaches for time series modeling, and which characteristics of the VAR model a Hankelet can capture. This paper aims at filling these gaps by: deriving a time series kernel function for Hankelets (TSK4H), demonstrating the relations between the derived TSK4H and former dissimilarity/similarity scores, highlighting an alternative probabilistic interpretation of Hankelets.
Experiments with an off-the-shelf SVM implementation and extensive validation in action classification and emotion recognition on several feature representations, show that the proposed TSK4H allows achieving state-of-the-art or even superior accuracy values in classification with respect to past work. In contrast to state-of-the-art time series kernel functions that suffer of numerical issues and tend to provide diagonally dominant kernel matrices, empirical results suggest that the TSK4H has limited numerical issues in high-dimensional spaces. On three widely used public benchmarks, TSK4H consistently outperforms other time series kernel functions despite its simplicity and limited time complexity.

1 Introduction

Time series arise naturally in several computer vision applications including tracking [42, 56, 25], action/motion modeling and classification [30, 37, 43, 23], event causality [41, 18, 61], face emotion recognition [31], affective behavior [34, 7], gait recognition [14, 60], sequence alignment [64].

When dealing with time series, there is the need of formulating suitable kernel functions for adopting kernel methods [19] such as Support Vector Machine (SVM) [15]. Formerly proposed time series kernels are the Dynamic Time Warping (DTW) kernel [36] and the Global Alignment (GA) kernel [9, 11]. While the DTW kernel considers data similarities along the optimal alignment path of the two time series, the GA kernel function takes into account all the possible alignments between two time series. The resulting kernel matrix is guaranteed to be positive definite. GA kernel has shown promising results in face emotion classification given the non-rigid 2D deformations of facial landmarks [31]. With a focus on time series alignment, [40] proposes the temporal matching (TM) kernel to align videos efficiently. The TM kernel generalizes the circulant temporal encoding (CTE) [44] to consider the cross-correlation of two series of vectors in the Fourier domain.

In this paper, we assume that each time series is generated by a vector autoregressive model (VAR) of order p and unknown parameters, and formulate a time series kernel function to compare the generating VAR models without any costly system parameter identification [38] or sophisticated data embeddings [44].

The VAR(p) model assumption for data generation is not novel and has been adopted in several former works. In particular, in [52, 47, 4] VAR model parameters of each time series are explicitly estimated and used to discriminate between different classes within a SVM framework [52, 4] or a NN classifier [47].

In contrast to these works, the autoregressive kernel (AR kernel) in [10] does not require of any system identification to compare time series. Under the VAR model assumption, the AR kernel is defined as the product kernel [21] of data posterior probability density functions of the two time series. However, the AR kernel suffers of numerical issues when dealing with high-dimensional time series and tends to produce diagonally dominant kernel matrices, which in turn may yield to serious difficulties during the learning stage of kernel machines.

In recent works [24, 30, 28], the observed time series are described by means of Hankel matrix-based representations denoted Hankelets. The main motivation behind the adoption of this representation is that Hankel matrices embed the system parameters and also represent the subspace where the trajectories lie [24]. The dissimilarity score in [24] is used to compare two Hankelets by approximating the cosine of the principal angles of the two different subspaces. Despite the dissimilarity score is not a distance, it has been successfully used for face emotion recognition within NN classifiers [28], and for action classification in [24, 30] within a SVM framework and a discriminative HMM respectively. In order to adopt Hankelet-based representation of time series with SVM, bag-of-Hankelets and codebook-based representations are used in [24] and [28] respectively. At the best of our knowledge, Hankelets have never been used directly within a kernel machine due to the lack of a proper kernel function. This paper aims at filling this gap by defining a kernel function for Hankelets and, hence, a kernel for time series. We will discuss the relation between our proposed kernel and the dissimilarity score in [24], and the relation between our time series kernel function and the Matrix Cosine Similarity (MCS) in [50].

This paper will discuss, in order, the following main contributions:

- the interpretation of a (unnormalized) Hankelet in terms of precision matrix of the parameter posterior when assuming a Gaussian Autoregressive model for data generation;
- a time series kernel function for Hankelets (TSK4H). We formally show the relation between our kernel and formerly proposed scores;
- extensive validation of our approach in different settings to empirically show the generality of our approach.

In our experiments we focus on action classification and emotion recognition, and test our kernel within a standard SVM framework on publicly available benchmarks. In both application domains, we considered two different kinds of input data: (1) trajectories of 2D/3D landmarks, and (2) trajectories of visual features extracted from RGB videos. In our experiments, the adoption of our kernel function with SVM yields to comparable or superior performance with respect to other works at the-state-of-the-art, but consistently outperforms GA and AR kernels.

2 Representing Time Series by means of Hankelets

In recent years, there has been a growing interest into the representation of time series dynamics by Hankel matrices [24, 30, 28, 27]. In [24], a truncated block-Hankel matrix H represents the time series $Y = [y_1, \ldots, y_\tau]$ as follows:

$$H = \begin{bmatrix} y_1, & y_2, & y_3, & \ldots, & y_m \\ y_2, & y_3, & y_4, & \ldots, & y_{m+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y_p, & y_{p+1}, & y_{p+2}, & \ldots, & y_\tau \end{bmatrix}. \tag{1}$$

The implicit assumption is that the time series Y has been generated by a linear time invariant (LTI) system

$$\begin{aligned} x_k &= A \cdot x_{k-1} + \epsilon_{k-1}; \\ y_k &= C \cdot x_k, \end{aligned} \tag{2}$$

where x_k represents the internal state of the system, the matrices A and C are the system and output matrices respectively, and ϵ_k is uncorrelated zero mean Gaussian noise. While the time series Y can be observed, x_k and ϵ_k are not, and A and C are unknown.

The main justification about the use of Hankel matrices as dynamics representation is that each Hankel matrix embeds the observability matrix $\Gamma = [CA^\tau, \ldots, CA, C]$ of the LTI system that has generated the time series. Indeed, $H = \Gamma \cdot X$, where $X = [x_1, \cdots, x_\tau]$ is the matrix formed by the sequence of internal states of the LTI system [30]. Former works such as [24, 30, 47] normalize the Hankel matrix H as follows:

$$\hat{H} = \frac{H}{\sqrt{||H \cdot H^T||_F}}. \tag{3}$$

Finally, we note that HH^T (which is denoted Hankelet in [24]) plays a central role into the least square estimation of the AR model parameters [54].

2.1 Probabilistic Interpretation of Hankelets

Let us consider a vector autoregressive model of order p defined as follows:

$$y_k = \sum_{i=1}^{p} A_{p-i+1} \cdot y_{k-i} + \epsilon_k \tag{4}$$

where $y_k \in \mathbb{R}^d$, $A_i \in \mathbb{R}^{d \times p}$ for $1 < i < p$, and $\epsilon_k \sim \mathcal{N}(0, V)$. Another equivalent formulation of the VAR model is

$$y_k = A \cdot x_k + \epsilon_k \tag{5}$$

with $A = [A_1, \dots A_p] \in \mathbb{R}^{d \times dp}$ and $x_k^T = [y_{k-p}^T, \dots, y_{k-1}^T]$.

Due to the Gaussian VAR hypothesis, y_k follows a normal distribution, i.e. $p(y_k|A, x_k, V) = \mathcal{N}(A \cdot x_k, V)$. Let us denote the set of $m-1$ vectors in a temporal window $Y = [y_{p+1}, \dots, y_\tau]$ ($Y \in \mathbb{R}^{d \times m-1}$) and $X = [x_{p+1}, \dots, x_\tau]$ ($X \in \mathbb{R}^{dp \times m-1}$). Due to the Markov property, the joint density of Y is

$$p(Y|A, X, V) = \frac{1}{(2\pi)^{\frac{md^2}{2}} |V|^{\frac{C}{2}}} e^{-\frac{1}{2} \operatorname{Trace}((Y-AX)^T V^{-1} (Y-AX))}. \tag{6}$$

By taking a closer look at X and comparing it with Eq. 1, we find that X is the (unnormalized) Hankel matrix H of the sequence of predictors (i.e. past values), i.e. $[y_1, \cdots, y_{\tau-1}]$.

Furthermore, if we consider a normal matrix distribution prior for A with zero mean and V equals to the noise covariance matrix [10], $A \sim \mathcal{NM}_{d,dp}(0, \Sigma, V)$, then it is possible to express the posterior distribution over A as a normal matrix distribution, namely:

$$A|Y, X \sim \mathcal{NM}_{d,dp}(M, U, W); \tag{7}$$

$$M = YX^T U; \tag{8}$$

$$W = V; \tag{9}$$

$$U = (XX^T + \Sigma^{-1})^{-1}. \tag{10}$$

As a consequence, the precision matrix (or inverse covariance matrix) of the parameter posterior can be rewritten as follows:

$$U^{-1} = (HH^T + \Sigma^{-1}). \tag{11}$$

If the prior for A has a precision matrix equals to $\Sigma^{-1} = \alpha I$ with $0 < \alpha << 1$ (i.e., the prior is mostly uninformative), U^{-1} can be interpreted as a regularized version of the matrix HH^T.

In this sense, the comparison of (unnormalized) Hankelets entails the comparison of precision matrices of the two parameter posterior densities of the underlying Gaussian Processes. It is well known that elements of a precision matrix represent partial covariances of pairs of variables, that is they measure how two variables covariate conditionally on the remaining ones.

In our application, each precision matrix refers to the parameter posterior and its elements reflect statistical links on the parameters of the VAR(p) model referring to different time lags and components of the time series vectors. Hence, given two time series, comparison of the corresponding Hankelets allows comparison of how the model parameters conditionally covariate in the two underlying Gaussian Processes.

3 Time Series Kernel for Hankelets

The comparison of two Hankelets aims to establish if the corresponding time series might have been generated by similar or even the same VAR model. The lack of a suitable kernel function has limited the adoption of Hankelets within kernel machine frameworks. This paper aims at filling this gap by deriving a suitable kernel function for Hankelets.

In a nutshell, we propose to adopt a very popular kernel function, the cosine similarity kernel [46] that, as we will show in Secs. 3.1 and 3.2, assumes a special meaning for Hankelets. Given two vectors u and v, the cosine similarity kernel is defined as follows:

$$K(u,v) = \frac{< u, v >}{\sqrt{< u, u >}\sqrt{< v, v >}}. \tag{12}$$

We now rely on a well-known relation between the vectorization operator of a matrix, $\mathrm{vec}(A) : \mathbb{R}^{n\times m} \rightarrow \mathbb{R}^{nm\times 1}$, and the Frobenius dot product of matrices, $< \cdot, \cdot >_F$. For a matrix A, it holds that

$$< \mathrm{vec}(A), \mathrm{vec}(A) >= \mathrm{Trace}(A^T A) =< A, A >_F= ||A||_F^2. \tag{13}$$

Given two matrices A and B both in $\mathbb{R}^{n\times m}$, their Frobenius dot product is defined as $< A, B >_F= \mathrm{Trace}(A^T B)$.

Let us assume that $A = H_p H_p^T$ and $B = H_q H_q^T$ are two (unnormalized) Hankelets for the time series Y_p and Y_q respectively. In this special case:

$$< H_p H_p^T, H_q H_q^T >_F= \mathrm{Trace}(H_p H_p^T H_q H_q^T) = ||H_p^T H_q||_F^2 \tag{14}$$

and it turns out that the cosine similarity kernel of two Hankelets is

$$K(H_p H_p^T, H_q H_q^T) = \frac{<H_p H_p^T, H_q H_q^T>_F}{\sqrt{<H_p H_p^T, H_p H_p^T>_F}\sqrt{<H_q H_q^T, H_q H_q^T>_F}} = \tag{15}$$

$$=< \hat{H}_p \hat{H}_p^T, \hat{H}_q \hat{H}_q^T >_F= ||\hat{H}_p^T \hat{H}_q||_F^2 \tag{16}$$

which is a valid, separable, positive definite kernel function for Hankelets.

In the statistics literature, the measurement in Eq. 16 is known as RV-coefficient or vector correlation [50, 1, 53]. The RV-coefficient was proposed as a measure of similarity between positive semi-definite matrices and as a theoretical tool to analyze multivariate techniques [1]. The RV-coefficient measures the alignment of the subspaces represented by positive semi-definite matrices and is invariant to rotation transformations [53]. Given a data matrix X, by denoting with $\Sigma_{X_p X_q} = X_p^T X_q$, the RV-coefficient coincides with our kernel function when $X = H$, that is:

$$RV(X_p, X_q) = \frac{\mathrm{Trace}(\Sigma_{X_p X_q} \Sigma_{X_q X_p})}{\sqrt{\mathrm{Trace}(\Sigma_{X_p X_p} \Sigma_{X_p X_p})}\sqrt{\mathrm{Trace}(\Sigma_{X_q X_q} \Sigma_{X_q X_q})}} = K(H_p H_p^T, H_q H_q^T).$$

Finally, our kernel relates to the one proposed in [16] where data points are compared on the Grassmannian manifold. In [16], it is also proposed to embed points X to XX^T, which is similar to the unnormalized Hankelet. This embedding suffers of numerical issues when high-dimensional time series are considered. In our kernel definition, each Hankelet is normalized by means of its Frobenius norm, and a different rescaling is applied to each pair of time series.

3.1 Relation with Dissimilarity and Similarity Scores

In [24], two normalized Hankelets $\hat{H}_p\hat{H}_p{}^T$ and $\hat{H}_q\hat{H}_q{}^T$ are compared by the dissimilarity score defined as follows:

$$d(\hat{H}_p\hat{H}_p{}^T, \hat{H}_q\hat{H}_q{}^T) = 2 - ||\hat{H}_p\hat{H}_p{}^T + \hat{H}_q\hat{H}_q{}^T||_F. \tag{17}$$

An equivalent form of the dissimilarity score that takes advantage of the normalization in Eq. 3 (see [24]) is:

$$d(\hat{H}_p\hat{H}_p{}^T, \hat{H}_q\hat{H}_q{}^T) = 2 - \sqrt{2 + 2||\hat{H}_p{}^T\hat{H}_q||_F^2} \tag{18}$$

and, by considering the SVDs of the two Hankelets, this score can be regarded as an approximation of the cosine of the principle angles between the two subspaces [24, 30]. Based on Eq. 18, the work in [27] proposes a similarity score:

$$s(\hat{H}_p\hat{H}_p{}^T, \hat{H}_q\hat{H}_q{}^T) = ||\hat{H}_p{}^T\hat{H}_q||_F. \tag{19}$$

By comparing Eq. 19 to our kernel formulation in Eq. 16, we get that

$$K(H_pH_p^T, H_qH_q^T) = s(\hat{H}_p\hat{H}_p{}^T, \hat{H}_q\hat{H}_q{}^T)^2. \tag{20}$$

In practice, the cosine similarity kernel of two Hankelets is the squared similarity score. It measures the cosine of the angle between the two vectorized Hankelets and, since the same considerations in [24] hold also in our case, the cosine similarity kernel of two Hankelets might be regarded as an approximation of the cosine of the principal angles of the two subspaces.

3.2 Relation with the Matrix Cosine Similarity

The work in [50] proposes to compare matrices of features by means of the matrix cosine similarity (MCS) score. Features f_i are extracted from local neighborhood and stacked into columns of a feature matrix F. Two feature matrices F and Q (of same size) are compared by means of the MCS defined as

$$MCS(F, Q) = \frac{< F, Q >_F}{\sqrt{< F, F >_F}\sqrt{< Q, Q >_F}}. \tag{21}$$

MCS is a generalized version of the vector cosine similarity designed to compare matrices of same size.

Our time series kernel function differs from the MCS in several respects. (1) In contrast to [50], which deals with objects detection and stacks in a unique matrix a set of visual features, our kernel function applies to Hankelets computed upon vector time series (or any ordered sequence of features); (2) In [50], there is not an underlying model for the generation of the feature matrices F and Q. In contrast, in our formulation we assume that the two time series are generated by VAR(p) models, and the kernel function aims at measuring model similarities rather than feature similarities. (3) In [50], the MCS is a RV-coefficient if F and Q are positive semi-definite matrices, which might not be true in general. Hankelets are positive semi-definite symmetric matrices and our kernel estimates exactly the RV-coefficient of the two matrices. This is of interest because, in this case, the RV-coefficient measures communality between the two subspaces even in high-dimensional data [53]. (4) Finally, the MSC score can only compare sets of feature descriptors of same size (same number of feature vectors). Our TSK4H can compare vector time series of different lengths.

3.3 Time and Space Complexity

A Hankel matrix H of maximal order p of a d-dimensional vector time series is built by a simple reordering of the time series elements as shown in Eq. 1. Computing the Frobenius norm of HH^T when H has size $dp \times m$ with $dp >> m$, considering that $||HH^T||_F = ||H^TH||_F$, has a cost of $O(m^2(dp+1))$. The time complexity of evaluating our kernel function is of about $O(m^2(dp+1))$, that is linear in the dimension of the vector time series and quadratic in the number of columns of H. Finally, storing a single Hankel matrix H has a space complexity of $O(dpm)$ and is more convenient than storing HH^T $(O(\frac{d^2p^2}{2}))$.

4 Applications

Our kernel function may be adopted in any domain where time series (or any ordered sequence of features) arise. Here, we detail how our TSK4H can be used in two challenging applications: face emotion recognition and action classification. In each of these applications, we will extract a time series of per-frame feature vector to represent a video depicting a face emotion or an action respectively. A set of these time series will be used to train a standard SVM[1] by employing our TSK4H. To perform multi-class classification, we will consider a 1-vs-all classification schema. Due to the VAR(p) model assumption, each time series has to be made zero mean before evaluating our kernel. Whenever the Hankel matrices have a dimensionality higher than the training set size, each SVM can be trained in the dual space. In the following, we briefly describe both the adopted visual features extracted to represent RGB videos, and features that are more application domain dependent.

Visual Features We adopt two widely used per-frame descriptors: Haar-like features [57] and HoG features [12].

We extract the 6 basic Haar-like features[2] used in [27] from 13 × 13 non-overlapping regions of same size from each image. The extracted features are stacked into one vector of dimension 1014. A video of N frames is represented by a 1014-dimensional time series of length N.

We extract HoG features[3] from blocks of size 32 pixels, cells of size 16 pixels and a block stride (shift) of 16 pixels. Before extracting the HoG features, images are resized to a multiple of the block size. A video of N frames is represented by a M-dimensional time series of length N, with M dependent on the number of blocks.

[1] In our experiments, we used the publicly available library LIBSVM [3]

[2] We used the public implementation available within the Struck tracking method [17], which is the one suggested in [27]

[3] We used the implementation available with the OpenCV library [2]

Face Emotion Recognition deals with the problem of inferring the emotion (i.e., fear, anger, surprise, etc.) given a sequence of face images, and suffers of strong inter-subject variations, illumination changes, biometric differences, head pose changes, etc.. Useful literature reviews on the topic are [62, 48].

A common approach is that of representing a face expression through 2D face landmark coordinates estimated, for instance, by an active appearance model [8]. Following [28], we represent a sequence of face expressions by means of vector time series of: (1) concatenated 2D facial landmark coordinates (L); (2) pairwise landmark distances (D); (3) concatenation of pairwise landmark distances and landmark coordinates (L+D).

Since face landmark detection is still an open problem, it is appealing the adoption of visual features extracted from face regions, such as Haar-like and HoG features. In contrast to [27], which extracts Haar-like features from different spatial windows within the face image and builds a Hankelet for each kind of Haar-like template, we concatenate all Haar-like features in a single vector. When adopting HoG features, each face image is resized to 160×128 and the per-frame descriptor has a size of 2268.

Action Classification entails the problem of assigning an action label to a sequence of per-frame feature descriptors and, in general, it suffers of the following issues: difficulties in reliably describing human poses, biometric differences, subjective velocities and characteristics (i.e. different human gaits), illumination changes and clothing variation (especially in RGB videos), etc.. More details on these challenges can be found in popular literature reviews [39, 55, 5].

In recent years, there has been a proliferation of works about action classification based on sequences of skeletons obtained from MoCap data [33] or estimated from depth data [51]. Our kernel function applies to this kind of application as well. We will consider the case in which action samples are described as sequences of 3D body joints, and the case in which action samples are described as time series of per-frame descriptors (Haar-like or HoG features) extracted from bounding boxes of the detected persons. Before extracting HoG features, each bounding box is resized to 128×64 to better preserve its aspect ratio. The resulting HoG descriptor is of 756 features.

5 Experimental Results

In the following, we will refer to our method as to time series kernel for Hankelets $TSK4H(f)$, where f is the feature type that has been considered.

We have compared our kernel function to the GA and the AR kernels [10][4] on equal terms of features. In our experiments, the AR kernel performed very poorly compared with both our kernel and the GA kernel and we decided to drop these results. We believe the poor performance of the AR kernel might be ascribable to the high data dimensionality that raises serious numerical issues. As for the GA kernel, we have found benefits in adopting the normalized GA kernel (NGAK) [9]. NGAK has two parameters: the bandwidth σ of the exponential

[4] Both the code of the AR Kernel and of the (normalized) GA kernel are publicly available at Dr. Cuturi's website

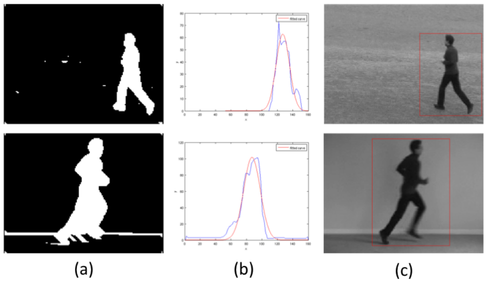

Fig. 1. Examples of bounding boxes extracted for two frames in the KTH dataset: (a) the mask obtained by applying a threshold on the gradient magnitude, (b) the Gaussian curve fitted on the sum of binary values along the columns, (c) the final bounding box centered on the mean of the Gaussian curve.

kernel, and T that regulates the triangular weighting function within the kernel. This weighting function is necessary to take into account the level of warping needed to align the two time series. In our experiments we used brute force to set these parameters by testing various parameter combinations. Similarly to [31], we set $\sigma = 2^s$ and let s varying in $[0, 20]$ with step 2. Moreover, we let T varying in $[0, 14]$ with step 2 ($T = 0$ indicates that no triangular weighting function is used). As a result, a total number of 88 parameter combinations have been tested. For each experiment, we report the parameters and the accuracy values in classification corresponding to the best parameter combination. We also report the accuracy values achieved with NGAK combined with a NN classifier. We will use the notation $NGAK(f, \sigma, T)$ to indicate that the NGA kernel was computed on sequences of features f with parameters (σ, T).

We have further implemented a baseline method that, given a pair of time series, aligns them with DTW by maximizing the cosine similarity of pairs of vectors. The similarity value of the best alignment was normalized by the length of the aligned sequences. As explained in [11], a DTW kernel is not guaranteed to be positive definite. Therefore we adopt the NN classifier over this similarity score. We refer to this baseline method as to DTW-S(f), where f is the considered feature type.

We will use $HH^T(f) + NN$ to indicate that Hankelets have been computed on the feature f and classified by the NN classifier. During training we set the parameter C of our SVM to 1000. In all of our experiments, when applying PCA, we skipped the first (less discriminative) component and retained 99% of the total variance. Coefficients of the PCA were estimated on the training set, then the test set was transformed accordingly.

Datasets In experiments for face emotion recognition, we adopted the widely used Extended Cohn-Kanade dataset (CK+) [32], which provides 327 video sequences of 118 different individuals displaying 7 emotions: *angry (A), contempt (C), disgust (D), fear (F), happy (H), sadness (Sa), surprise (Su).* The number of frames of these sequences ranges in $[6, 71]$ with an average value of about

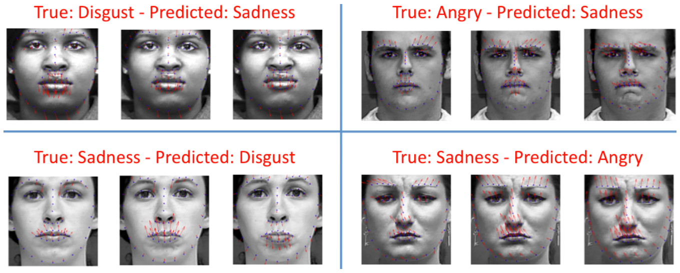
True: Disgust - Predicted: Sadness
True: Angry - Predicted: Sadness
True: Sadness - Predicted: Disgust
True: Sadness - Predicted: Angry

Fig. 2. Misclassified samples from the CK+ dataset. Blu dots represent the 2D facial landmarks. Red arrows show the velocities of each landmark.

18 ± 8.6. Considering that some sequences have very few frames, to guarantee a fair comparison with other works and to use all the sequences in the dataset, the order p of the Hankelets was set to 3 (with $p = 3$, at least $2 \cdot p - 1 = 5$ frames are needed to build a Hankelet). The adopted validation protocol is leave-one-subject-out cross-validation. The CK+ dataset provides landmark tracking results estimated by an active appearance model, which we use in our experiments as 2D face landmarks as also proposed in [28, 32].

In experiments for action classification based on 3D skeletons, we have adopted the UCF dataset [13], which provides skeletons of 15 joints for 16 actions performed 5 times by 16 subjects. It comprises 1280 action sequences of length in [27,229], (on average 6634 frames). The actions are: *balance (B), climbladder (CR), climbup (CP), duck (D), hop (H), kick (K), leap (L), punch (P), run (R), stepback (SB), stepfront (SF), stepleft (SL), stepright (SR), twistleft (TL), twistright (TR), and vault (V)*. The adopted protocol on this dataset is 4-fold cross-subject validation: subjects are split in 4 subsets; at each run, 3 subsets are used for training the models, the remaining subset is used in test.

Finally, in experiments for action classification based on visual features extracted from RGB videos, we have adopted the KTH dataset [49], which contains six types of human actions, *boxing (B), hand clapping (HC), hand waving (HW), jogging (J), walking (W), running*, performed several times by 25 subjects in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes, and indoors. The dataset contains 2391 sequences with a spatial resolution of 160×120 pixels. The length of the sequences (after people detection) ranges in $[12, 362]$, with an average length of about 81.9 ± 43.1 frames. Considering the minimum length of the sequences, the order of the Hankel matrix was set to $p = 6$. The adopted protocol is leave-one-subject-out cross-validation. Bounding boxes for person detection (shown in Fig. 1) were computed with a simple detector that: computes the gradient magnitude at each frame, performs a morphological closing operation with a line structuring element (to highlight vertical edges), applies a threshold to the resulting image, computes the sum of the binary pixels across the rows, fits a Gaussian and takes the bounding box centered on the Gaussian mean and with a width proportional to the standard deviation (which corresponds to a Gaussian-based peak detector).

Emotions:	A	C	D	F	H	Sa	Su	Avr.
$HH^T(L)$+NN [28]	82.2	77.8	94.9	80	**100**	64.3	97.6	85.3
$HH^T(D)$+NN [28]	88.9	**83.3**	**96.6**	84	**100**	67.9	98.8	88.5
$HH^T(L+D)$+NN [28]	91.1	**83.3**	94.9	84	**100**	71.4	98.8	89.1
$HH^T(L)$+CSVM [28]	86	75	92	85.6	98.3	74.3	95.9	86.7
$HH^T(D)$+CSVM [28]	89.1	72.8	92.4	**89.6**	97	80.7	97.2	88.4
$HH^T(L+D)$+CSVM [28]	89.8	73.9	90.8	89.2	97.4	81.8	97.7	88.7
NGAK(L, 2^6, 10) + NN	62.2	77.8	83	56	97.1	50	94	74.3
NGAK(D, 2^8, 10) + NN	73.3	88.9	88.1	68	97.1	75	94	83.5
NGAK(L, 2^8, 14) + SVM	86.7	72.2	94.9	64	95.6	78.6	96.4	84.1
NGAK(D, 2^{10}, 0) + SVM	88.9	77.8	91.5	72	97.1	85.7	95.2	86.9
DTW-S(L) + NN	86.7	72.2	93.2	68	**100**	53.6	97.6	81.6
DTW-S(D) + NN	88.9	83.3	96.6	68	**100**	60.7	97.6	85
TSK4H(L)+SVM [ours]	86.7	**83.3**	94.9	88	98.5	**82.1**	97.6	90.2
TSK4H(D)+SVM [ours]	91.1	**83.3**	**96.6**	88	**100**	78.6	98.8	**90.9**
CK+ [32]	35	25	68.4	21.7	98.4	4	**100**	50.4
CLM-based [6]	70.1	52.4	92.5	72.1	94.2	45.9	93.6	74.4
LRBM [35]	**97.8**	72.2	89.8	84	**100**	78.6	97.6	88.6
ITBN [58]	91.1	78.6	94	83.3	89.8	76	91.3	86.3

Table 1. Accuracy values (in %) for the Emotion Recognition task on the CK+ dataset when using 2D facial landmarks. In bold font the highest accuracy values.

Face Emotion Recognition On the CK+ dataset, when adopting 2D face landmarks, PCA was used to filter out noise and reduce data redundancy, and provided per-frame vectors of average size 88, 123, 127 for L, D, and L+D features respectively. We report our results and comparison with former works in Table 1. The first column describes the type of features used to represent the time series and the classifier. The other columns indicate the emotion labels, and the last column reports the average per-class accuracy value.

The first and second parts of the table reports the results described in [28] when adopting a NN classifier and a codebook-based SVM (CSVM). The third part of the table reports the results achieved when using the NGAK in SVM and NN classifiers. The fourth part of the table reports the accuracy values achieved with our baseline method DTW-S while the fifth part of the table shows the accuracy values achieved by adopting our time series kernel for Hankelets

Emotions:	**A**	**C**	**D**	**F**	**H**	**Sa**	**Su**	**Avr.**
Ens.of H(Haar-like) + NN [27]	86.7	83.3	96.6	52	**100**	71.4	97.6	83.9
HH^T(Haar-like) + NN	60	77.8	93.2	56	**100**	64.3	96.4	78.2
HH^T(HoG) + NN	57.8	61.1	**100**	72	97.1	64.3	98.8	78.7
NGAK(Haar-like, 1, 12) + NN	55.6	88.9	79.7	28	87	60.7	83.1	69
NGAK(HoG, 2^2, 12) + NN	26.7	88.9	54.2	12	49.3	85.7	59	53.7
NGAK(Haar-like, 2^2, 0) + SVM	80	88.9	88.1	56	95.7	75	97.6	83
NGAK(HoG, 2^4, 0)+ SVM	84.4	72.2	94.9	64	98.6	85.7	98.8	85.5
DTW-S(Haar-like) + NN	71.1	77.8	96.6	52	**100**	71.4	98.8	81.1
DTW-S(HoG) + NN	71.1	50	**100**	64	98.5	53.6	98.8	76.6
TSK4H(Haar-like)+SVM [ours]	91.1	83.3	98.3	72	**100**	85.7	97.6	89.7
TSK4H(HoG)+SVM [ours]	86.7	83.3	98.3	88	**100**	75	98.8	90.01
CAPP + SVM [32]	70	21.9	94.7	21.7	**100**	60	98.7	66.7
LDN + RBF-SVM [45]*	71.7	73.7	93.4	**90.5**	95.8	78.9	97.6	85.9
LBP + CC + SVM [20]	**93**	**89**	98	80	**100**	**86**	**100**	**92.3**

Table 2. Accuracy values in Emotion Recognition (CK+) with visual features. Bold font highlights the highest accuracy values. *10-fold cross validation

(TSK4H). Finally, the bottom part of the table reports the accuracy values of other works at the state-of-the-art that uses 2D facial landmark time series.

In contrast to [28], in our experiments we did not notice any significant difference when using the features D or $L + D$ and we dropped the results obtained with $L+D$. As the table shows, the adoption of a discriminative method such as SVM over Hankelets by means of our TSK4H allows us to get an increase in the emotion classification accuracy values.

By comparing the results achieved with the CSVM [28] and with our approach, the gain in accuracy value is of about 3.4% (average gain over L and D). We stress that our approach does not require a codebook learning stage. The gain in accuracy value is even higher when comparing to the NGAK and DTW-S (5.9% and 8.7% respectively – average gains over L and D).

Irrespectively of the adopted representations (L, D, L+D), most of the confusion in our experiments was between the classes *sadness* and *disgust*, and the classes *angry* and *sadness*. Some misclassified samples are shown in Fig.2. This kind of mistakes was formerly observed in [30]. In practice, 2D landmark trajectories corresponding to raising and lowering eyebrows/lips tend to be mirrored versions of each other, and yield to similar Hankelet representations.

We report in Table 2 the experimental results obtained when adopting Haar-like and HoG features as per-frame face expression descriptors. Adoption of PCA resulted in 94 and 807 (average) dimensional per-frame vectors for Haar-like and HoG features respectively.

As Table 2 shows, we achieve state-of-the-art results on the CK+ dataset when adopting visual features. When using Haar-like features, the adoption of SVM over Hankelets allows us to achieve a higher accuracy value with respect to the work in [27], and to NN classifier (the gain in accuracy value is of about 6.9% and 14.7%, respectively). With respect to the NGAK, the gain in accuracy values of our TSK4H is of 8.1% and 5.3% on Haar-like and HoG features respectively. Overall, the average accuracy values reached with our TSK4H on Haar-like features and HoG features are close each other.

Action Classification On the UCF dataset, application of PCA over skeletal data resulted in 28-dimensional per-frame descriptors (on average). For compar-

Actions:	B	CR	CP	D	H	K	L	P	R	SB	SF	SL	SR	TL	TR	V	Avr.
HH^T+DHMM [30]	99.9	98.7	96.9	98.6	96.9	98.5	95	98.4	98.5	97.9	99.3	98.1	97.5	93.6	92.9	97.8	97.4
HH^T+NN(p=4)	**100**	98.8	98.8	**100**	96.3	**100**	**100**	**100**	96.3	91.3	78.8	85	93.8	98.8	**100**	96.3	95.9
HH^T+NN(p=14)	**100**	98.8	**100**	**100**	97.5	**100**	**100**	**100**	93.8	98.8	98.8	97.5	98.8	98.7	**100**	**98.8**	98.8
NGAK(1,0)+NN	98.7	98.7	**100**	**100**	**100**	**100**	97.5	**100**	95	**100**	**100**	98.7	98.7	**100**	98.7	98.7	99
NGAK(4,0)+SVM	83.7	98.7	96.2	**100**	98.7	**100**	97.5	97.5	88.7	**100**	**100**	98.7	97.5	96.3	97.5	96.3	96.7
DTW-S+NN	**100**	**100**	98.8	**100**	95	**100**	**100**	98.8	98.8	**100**	**100**	97.5	98.8	**100**	**100**	**98.8**	98.5
TSK4H(p=4)	**100**	98.8	**100**	**100**	**100**	98.8	98.8	**100**	97.5	98.8	95	96.3	97.5	97.5	98.8	97.5	98.4
TSK4H(p=14)	**100**	98.8	**100**	**100**	**100**	**100**	98.8	**100**	97.5	**100**	98.8	98.8	97.5	**100**	**100**	97.5	**99.2**
Log. Reg. [13]	97.5	93.8	98.8	**100**	96.2	98.8	**100**	95	97.5	97.5	97.5	96.2	98.8	88.8	86.2	92.5	95.9
LTBSVM [52]*	**100**	**100**	93.3	**100**	93.3	**100**	96.7	**100**	**100**	93.3	**100**	**100**	**100**	**100**	93.3	96.7	97.9

Table 3. Accuracy values in Action Recognition (UCF dataset) with 3D body joints trajectories. Bold font highlights the highest values. *70-30% cross validation protocol.

ison purposes, we test our approach with order $p = 4$ as used in [30], and further report results achieved with the highest possible value of $p = 14$ considering the minimum length of the sequences.

Table 3 compares the results of different approaches. The method in [30] adopts discriminative HMMs over small temporal windows described in terms of Hankelets of order 4. With respect to this method, our approach achieves a gain in accuracy values of 1% on equal terms of order. We stress here that, in contrast to [30], we calculate Hankelets over entire sequences. The gain in accuracy values when $p = 14$ is of about 1.8%. With respect to NGAK, on equal terms of classification framework (SVM), the gain in accuracy value of our TSK4H is of about 2.6%. Interesting, in this experiment, NGAK+NN performs better than NGAK+SVM. Differently than the experiments presented for emotion recognition, on the UCF dataset NGAK performs similarly to our kernel. We note that the vector dimensionality in this experiment is of about 28, and it is much lower than the feature representation dimensionality of the other experiments we present. This suggests that high-dimensionality may have a negative impact on NGAK while our kernel function seems to be less affected by the vector dimensionality. To verify this, we performed a further experiment on the KTH dataset.

On the KTH dataset, application of PCA resulted in 460 and 517 dimensional per-frame vectors for Haar-like and HoG features respectively. Table 4 reports the achieved accuracy values in classification. Overall, our method achieves the same accuracy value of [59]. The work in [59] proposes a shape-motion prototype-based approach. An action is represented as a sequence of prototypes. Prototypes are trained via K-means clustering and, at test time, are inferred by maximizing a model conditional probability. Given the sequence of prototypes, classification is performed by applying DTW and K-NN classifier. We note that the whole framework in [59], which is a composition of methods, has a higher time complexity than our approach due to the need of performing DTW and comparing with the sequences in the training set for applying KNN. With respect to the NGAK on equal terms of classification framework (SVM) and feature representa-

Actions:	**B**	**HC**	**HW**	**J**	**R**	**W**	**Avg.**
HH^T(Haar-like) + NN	90	90	93	61	53	89	79
HH^T(HoG) + NN	93	99	98	84	73	98	91
NGAK(Haar-like, 2^2,0) + NN	24	73	50	70	73	74	61
NGAK(HoG, 2^2,8) + NN	26	81	69	62	55	52	57
NGAK(Haar-like,2^8, 0) +SVM	47	72	76	81	77	90	74
NGAK(HoG,2^8,0) +SVM	65	72	81	82	81	95	79
DTW-S(Haar-like) +NN	82	87	95	57	53	83	76
DTW-S(HoG) +NN	87	97	**99**	87	78	97	91
TSK4H(Haar-like)+SVM [ours]	97	95	89	77	84	94	89
TSK4H(HoG)+SVM [ours]	99	98	98	**95**	91	99	**97**
Descriptor-based [22]	96	99	86	91	85	92	92
MSRR [63]	98	97	**99**	90	90	**100**	96
SMP + DTW + KNN [59]	**100**	**100**	**99**	90	**93**	**100**	**97**

Table 4. Accuracy values in Action Classification on the KTH dataset when using all the scenarios in leave-one-subject-out cross-validation.

tions, our method allows to obtain a gain in the accuracy values of about 20.3% and 22.8% on Haar-like and HoG features respectively. The table also shows that, on this dataset, our method works better on HoG features rather than Haar-like features. By inspecting the confusion matrices, when adopting Haar-like features, most of the confusion is between the classes *jogging* and *running*: 14.75% of sequences in the class *jogging* are recognized as *running*, while 16.25% of sequences in the class *running* are classified as *jogging*. Such percentages reduce to 5.25% and 8.5% respectively when adopting HoG features. These results suggest that Haar-like features might not be suitable to represent fine-grained differences in the body poses of these two actions.

6 Conclusion

Recent works [24, 30, 27, 28] have successfully adopted Hankelets as a time series dynamics representation especially in NN classifiers. This paper discusses a probabilistic interpretation of Hankelets in terms of precision matrix of the Gaussian process that generates the time series. Based on this interpretation, comparison of Hankelets turns into the comparison of partial covariances of the VAR(p) model parameters.

Furthermore, this paper proposes a time series kernel function for Hankelets, which is the cosine similarity kernel function of the vectorized Hankelets. This paper shows that: (1) the proposed TSK4H measures the angle of the vectorized Hankelets, and hence of the vectors of partial covariances of the model parameters; (2) the proposed kernel coincides with the RV-coefficient used in statistics to measure the similarity between positive semi-definite matrices; (3) TSK4K has a relation with the dissimilarity score proposed in [24] but, in contrast to it, our TSK4H defines a valid positive definite kernel that allows the use of kernel machines directly over Hankelets; this offers the advantage of skipping the codebook generation step that was necessary in [24, 28] in order to adopt SVM. Finally, similarly to the score in [24], TSK4H approximates the cosine of the principal angles of the two subspaces.

Our extensive validation in action and emotion classification suggests that TSK4H is robust to numerical issues in high-dimensional spaces, and provides high accuracy values irrespectively of the adopted feature representation. In our experiments, TSK4H consistently outperforms other time series kernels such as GA and AR kernels. Time complexity of the GA kernel is claimed to be of about $O(n_p n_q d)$ with n_i indicating the length of the i-th time series [9], and d is the vector dimension. Time complexity of the AR kernel is $O((p+1)dN^2 + N^3)$ with $N = \max(n_p, n_q)$, and p the order of the assumed VAR model [10]. Our TSK4H and the GA kernel have comparable time complexity (see Sec. 3.3 for the complexity of TSK4H). However, in practice we have noticed that computation of the TSK4H (implemented in Matlab) seems faster than the computation of the GAK (publicly available C++ implementation). The main reason might be that the claimed complexity for the GA kernel does not consider the local kernel computation that, in high dimensions, may greatly affect the complexity.

References

1. Abdi, H.: RV coefficient and congruence coefficient. Encyclopedia of measurement and statistics, pp. 849–853, Sage Thousand Oaksˆ eCA CA, (2007)
2. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools, (2000)
3. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), **2**(3), pp. 27- 37, ACM, (2011)
4. Chaudhry, R., Ofli, F., Kurillo, G., Bajcsy, R., Vidal, R.: Bio-inspired dynamic 3d discriminative skeletal features for human action recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2013), pp. 471-478, IEEE, (2013)
5. Chen, L., Wei, H., Ferryman, J.: A survey of human motion analysis using depth imagery. Pattern Recognition Letters, **34**(15), pp. 1995-2006, Elsevier, (2013)
6. Chew, S., Lucey, P., Lucey, S., Saragih, J., Cohn, J., Sridharan, S.: Person-independent facial expression detection using constrained local models Proc. of Conf. and Workshop on Automatic Face & Gesture Recognition (FG), pp. 915–920, IEEE, (2011)
7. Cohn, J., Schmidt, K.: The timing of facial motion in posed and spontaneous smiles. International Journal of Wavelets, Multiresolution and Information Processing, **2**(2), pp. 121–132, World Scientific, (2004)
8. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), **23**(6), pp. 681–685, IEEE, (2001)
9. Cuturi, M.: Fast global alignment kernels. Proc. of Int. Conf. on Machine Learning (ICML), pp. 929-936, (2011)
10. Cuturi, M., Doucet, A.: Autoregressive kernels for time series. arXiv preprint arXiv:1101.0673, (2011)
11. Cuturi, M., Vert, J., Birkenes, O., Matsui, T.: A kernel for time series based on global alignments. Proc. of Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), **2**, pp. 413-420, IEEE, (2007)
12. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. Proc. of Conference on Computer Vision and Pattern Recognition (CVPR 2005), **1**, pp. 886-893, IEEE, (2005)
13. Ellis, C., Masood, S. Z., Tappen, M. F., Laviola Jr, J. J., Sukthankar, R.: Exploring the trade-off between accuracy and observational latency in action recognition. International Journal of Computer Vision, **101**, (3), pp. 420-436, Springer, (2013)
14. Frank, J., Mannor, S., Precup, D.: Activity and Gait Recognition with Time-Delay Embeddings. Conference on Artificial Intelligence (AAAI), (2010)
15. Gehler, P.V.: Kernel learning approaches for image classification. PhD Thesis, Universitat des Saarlandes, (2009)
16. Harandi, M. T., Salzmann, M., Jayasumana, S., Hartley, R., Li, H.: Expanding the family of Grassmannian kernels: An embedding perspective. Proc. of European Conference on Computer Vision (ECCV 2014), pp. 408-423, Springer International Publishing. (2014)
17. Hare, S., Saffari, A., Torr, P. H. S. : Struck: Structured output tracking with kernels. Proc. of International Conference on Computer Vision (ICCV 2011) pp. 263-270, IEEE, (2011)
18. Haufe, S., Nolte, G., Mueller, K., Krämer, N.: Sparse causal discovery in multivariate time series. arXiv preprint arXiv:0901.2234, (2009)

19. Hofmann, T., Schölkopf, B., Smola, A.: Kernel methods in machine learning. The annals of statistics, pp. 1171–1220, JSTOR, (2008)
20. Huang, X., Zhao, G., Pietikainen, M., Zheng, Wenming.: Robust facial expression recognition using revised canonical correlation. Proc. of International Conference on Pattern Recognition (ICPR), pp. 1734–1739, IEEE, (2014)
21. Jebara, T., Kondor, R., Howard, A.: Probability product kernels, The Journal of Machine Learning Research, **5**, pp. 819-844, JMLR. org, (2004)
22. Jiang, Z., Lin, Z., Davis, L. S.: Recognizing human actions by learning and matching shape-motion prototype trees. Trans. on Pattern Analysis and Machine Intelligence, **34**(3), pp. 533-547, IEEE, (2012)
23. Lehrmann, A., Gehler, P., Nowozin, S.: Efficient nonlinear Markov models for human motion. Proc. of Conference on Computer Vision and Pattern Recognition (CVPR 2014), pp. 1314–1321, IEEE, (2014)
24. Li, B., Camps, O., Sznaier, M.: Cross-view activity recognition using Hankelets. Proc. of Conf. on Computer Vision and Pattern Recognition (CVPR 2012), pp. 1362–1369, IEEE, (2012)
25. Lin, R., Liu, C.B, Yang, M.H., Ahuja, N., Levinson, S.: Learning nonlinear manifolds from time series. Proc. of European Conference on Computer Vision (ECCV 2006), pp. 245–256, Springer, (2006)
26. Lo Presti, L., La Cascia M.: An on-line learning method for face association in personal photo collection. Image and Vision Computing, **30**(4), pp.306-316, Elsevier, (2012)
27. Lo Presti, L., La Cascia M.: Ensemble of Hankel Matrices for Face Emotion Recognition. Image Analysis and Processing (ICIAP 2015), pp. 586-597, Springer International Publishing, (2015)
28. Lo Presti, L., La Cascia M.: Using Hankel matrices for dynamics-based facial emotion recognition and pain detection. Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2015), pp. 26-33, IEEE, (2015)
29. Lo Presti, L., La Cascia M., Sclaroff S., Camps O.: Gesture modeling by Hanklet-based hidden Markov model. Asian Conference on Computer Vision (ACCV 2014), pp. 529-546. Springer International Publishing, (2014)
30. Lo Presti, L., La Cascia M., Sclaroff S., Camps O.: Hankelet-based Dynamical Systems Modeling for 3D Action Recognition. Image and Vision Computing, **40**, pp. 1–53, Elsevier, (2015)
31. Lorincz, A., Jeni, L., Szabó, Z., Cohn, J., Kanade, T.: Emotional expression classification using time-series kernels. Proc. of Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 889–895, IEEE, (2013)
32. Lucey, P., Cohn, J., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. Proc. of Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 94–101, IEEE, (2010)
33. Moeslund, T., Granum, E.: A survey of computer vision-based human motion capture, Computer vision and image understanding, **81**,(3), pp. 231–268, Elsevier, (2001)
34. Nicolaou, M., Pavlovic, V., Pantic, M.: Dynamic probabilistic CCA for analysis of affective behaviour. Proc. of European Conf. on Computer Vision (ECCV 2012), pp. 98–111, Springer, (2012)
35. Nie, S., Wang, Z., Ji, Q.: A generative restricted Boltzmann machine based method for high-dimensional motion data modeling. Computer Vision and Image Understanding, **136**, pp. 14–22, Elsevier, (2015)

36. Noma, H., Shimodaira, K.: Dynamic time-alignment kernel in support vector machine. Advances in neural information processing systems, **14**, pp. 921 - 930, (2002)
37. Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., Bajcsy, R.: Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition. Journal of Visual Communication and Image Representation, **25**(1), pp. 24–38, Elsevier, (2014)
38. Paoletti, S., Juloski, A., Ferrari-Trecate, G., Vidal, R.: Identification of hybrid systems a tutorial. European journal of control, **13**(2), pp. 242-260, Elsevier, (2007)
39. Poppe, R.: A survey on vision-based human action recognition. Image and vision computing, **28**,(6), pp. 976-990, Elsevier, (2010)
40. Poullot, S., Tsukatani, S., Phuong Nguyen, A., Jégou, H., Satoh, S.: Temporal Matching Kernel with Explicit Feature Maps. Proc. of Conference on Multimedia Conference, pp. 381–390, ACM, (2015)
41. Prabhakar, K., Oh, S., Wang, P., Abowd, G., Rehg, J. M.: Temporal causality for the analysis of visual events. Proc. on Computer Vision and Pattern Recognition (CVPR 2010), pp. 1967-1974, IEEE, (2010)
42. Rahimi, A., Recht, B., Darrell, T.: Learning to transform time series with a few examples, Trans. on Pattern Analysis and Machine Intelligence, **29**(10), pp. 1759-1775, IEEE, (2007)
43. Raptis, M., Kokkinos, I., Soatto, S.: Discovering discriminative action parts from mid-level video representations. Proc. of Conf. on Computer Vision and Pattern Recognition (CVPR 2012), pp. 1242–1249, IEEE, (2012)
44. Revaud, J., Douze, M., Schmid, C., Jégou, H.: Event retrieval in large video collections with circulant temporal encoding. Proc of Conference on Computer Vision and Pattern Recognition (CVPR 2013), pp. 2459–2466, IEEE, (2013)
45. Ramirez Rivera, A., Castillo, R., Chae, O.: Local directional number pattern for face analysis: Face and expression recognition. Transactions on Image Processing (TIP), **22**(5), pp. 1740–1752, IEEE, (2013)
46. Sahami, M., Heilman, T. D.: A web-based kernel function for measuring the similarity of short text snippets. Proc. of International conference on World Wide Web, pp. 377-386, ACM, (2006)
47. Sankaranarayanan, A., Turaga, P., Baraniuk, R., Chellappa, R.: Compressive acquisition of dynamic scenes. Proc. of European Conf. on Computer Vision (ECCV 2010), pp. 129-142, Springer, (2010)
48. Sariyanidi, E., Gunes, H., Cavallaro, A.: Automatic analysis of facial affect: A survey of registration, representation and recognition. Trans. on Pattern Analysis and Machine Intelligence (PAMI), **37**(6), pp. 1113-1133, IEEE, (2014)
49. Schüldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. Proc. of International Conference on Pattern Recognition (ICPR 2004), **3**, pp. 32-36, IEEE, (2004)
50. Seo, H. J., Milanfar, P.: Training-free, generic object detection using locally adaptive regression kernels. Trans. on Pattern Analysis and Machine Intelligence, **32**(9), pp. 1688–1704, IEEE, (2010)
51. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM, **56**(1), pp. 116–124, ACM, (2013)
52. Slama, R., Wannous, H., Daoudi, M., Srivastava, A.: Accurate 3D action recognition using learning on the Grassmann manifold. Pattern Recognition (PR), **48**, (2), pp. 556-567, Elsevier, (2015)

53. Smilde, A. K., Kiers, H. A.L., Bijlsma, S., Rubingh, C.M., Van Erk, M.J.: Matrix correlations for high-dimensional data: the modified RV-coefficient. Bioinformatics, **25**(3), pp. 401-405, Oxford Univ Press, (2009)
54. Songsiri, J., Dahl, J., Vandenberghe, L.: Graphical models of autoregressive processes. Convex Optimization in Signal Processing and Communications, pp. 89-116, Cambridge, UK: Cambridge Univ. Press, (2010)
55. Turaga, P., Chellappa, R., Subrahmanian, V. S., Udrea, O.: Machine recognition of human activities: A survey. Trans. on Circuits and Systems for Video Technology, **18**(11), pp. 1473-1488, IEEE, (2008)
56. Urtasun, R., Fleet, D. J., Fua, P.: 3D people tracking with Gaussian process dynamical models. Proc. of Conference on Computer Vision and Pattern Recognition (CVPR 2006), **1**, pp. 238-245, IEEE,(2006)
57. Viola, P., Jones, M. J.: Robust real-time face detection. International journal of computer vision, **57**(2), pp. 137-154, Springer, (2004)
58. Wang, Z., Wang, S., Ji, Q.: Capturing complex spatio-temporal relations among facial muscles for facial expression recognition. Proc. of Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 3422–3429, IEEE, (2013)
59. Wu, B.,Yuan, C., Hu, W.: Human action recognition based on context-dependent graph kernels. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), pp. 2609-2616, IEEE, (2014)
60. Xu, D., Yan, S., Tao, D., Zhang, L., Li, X., Zhang, H.: Human gait recognition with matrix representation. Trans. on Circuits and Systems for Video Technology, **16**(7), pp. 896–903,IEEE, (2006)
61. Yang, M.H., Ahuja, N., Tabb, M.: Extraction of 2D motion trajectories and its application to hand gesture recognition. Trans. on Pattern Analysis and Machine Intelligence, **24**(8), pp. 1061–1074, IEEE, (2002)
62. Zeng, Z., Pantic, M., Roisman, G.I., Huang, T. S.: A survey of affect recognition methods: Audio, visual, and spontaneous expressions. Transactions on Pattern Analysis and Machine Intelligence, **31**(1), pp. 39-58, IEEE, (2009)
63. Zhang, X., Yang, Y., Jiao, L.C., Dong, F.: Manifold-constrained coding and sparse representation for human action recognition. Pattern Recognition, **46**(7), pp. 1819-1831, Elsevier, (2013)
64. Zhou, F., De la Torre, F.: Generalized Canonical Time Warping. Transactions on Pattern Analysis and Machine Intelligence (PAMI), **38**(2), pp. 279 - 294, IEEE, (2016)