University of Groningen

Learning skeleton representations for human action recognition

Saggese, Alessia; Strisciuglio, Nicola; Vento, Mario; Petkov, Nicolai

Published in:
Pattern Recognition Letters

DOI:
[10.1016/j.patrec.2018.03.005](https://doi.org/10.1016/j.patrec.2018.03.005)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2019

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Saggese, A., Strisciuglio, N., Vento, M., & Petkov, N. (2019). Learning skeleton representations for human action recognition. *Pattern Recognition Letters*, 118, 23-31. <https://doi.org/10.1016/j.patrec.2018.03.005>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Learning skeleton representations for human action recognition

Alessia Saggese^{a,*}, Nicola Strisciuglio^{b,*}, Mario Vento^a, Nicolai Petkov^b

^a Department of Information and Electrical Engineering and Applied Mathematics, University of Salerno, via Giovanni Paolo II, Fisciano, SA 84084, Italy

^b Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, Nijenborgh 9, Groningen, AG 9747, The Netherlands

ARTICLE INFO

Article history:

Available online 6 March 2018

Keywords:

41A05

41A10

65D05

65D17

ABSTRACT

Automatic interpretation of human actions gained strong interest among researchers in pattern recognition and computer vision because of its wide range of applications, such as in social and home robotics, elderly people health care, surveillance, among others. In this paper, we propose a method for recognition of human actions by analysis of skeleton poses. The method that we propose is based on novel trainable feature extractors, which can learn the representation of prototype skeleton examples and can be employed to recognize skeleton poses of interest. We combine the proposed feature extractors with an approach for classification of pose sequences based on string kernels. We carried out experiments on three benchmark data sets (MIVIA-S, MSRSDA and MHAD) and the results that we achieved are comparable or higher than the ones obtained by other existing methods. A further important contribution of this work is the MIVIA-S dataset, that we collected and made publicly available.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Automatic recognition of human actions received large interest in the last years by researchers in computer vision and pattern recognition because of its wide range of applications. For instance, in intelligent surveillance systems for the detection of potentially abnormal events of interest, in business intelligence for the analysis of customer behavior in a market area, or in social robotics for facilitating interactions between a person and one or more robots.

In this paper, we focus on the recognition of human actions seen by a frontal-view for which several survey paper were published [1,9,17,20,26,36,38]. The term *actions* indicates a class of movements that people can perform. It is important to distinguish it from *gestures* and *activities*. Gestures mainly involve a single part of the body, such as the head, a hand or the face. An activity may involve two or more persons and/or objects and takes place over a longer period of time. *Two persons are fighting* is an example of activity. An action is defined as the movement of a part of the human body (for instance, *drinking*, which mainly involves the upper part of the body) or of the whole body (e.g., *jumping*). An action can be considered as composed by a sequence of body poses. The concept of pose is strongly related with the one of skeleton. The skeleton is a set of rigid segments representing the bones, connected each other by joints that correspond to limb articulations.

A particular configuration of the skeleton, expressed in terms of spatial arrangement of joints, can be considered a *pose*.

The work of Johansson [18] is a milestone of the research on skeleton analysis for action recognition. It was demonstrated that several human actions can be recognized by looking at the position of limb joints only. In the experiments, they marked the limb joints of a number of subjects with light-point sources and observers were asked to recognize the actions without any additional information. Human observers were indeed able to correctly recognize many actions. These experiments confirmed that the information contained in the skeleton is sufficient in many situations to recognize the poses and the actions performed by a human subject. Nowadays, there are several devices and techniques for the automatic and reliable extraction of skeletons by visual sensors: e.g. the Microsoft Kinect depth sensor and the MOCAP system. Yao et al. [43] carried out an extensive experimentation with the aim of evaluating the discriminant power of different representations based on skeletons and more traditional low-level appearance features extracted from the silhouette of the subjects moving in the scene. They concluded that pose-based features outperform low-level appearance features in many cases, even when skeletons are highly noisy.

Existing methods for action recognition based on skeleton analysis can be divided into two categories: joint-based and body-part-based approaches.

The methods belonging to the first category consider the skeleton as a set of points, and the motion is estimated by analyzing the temporal evolution of single joints or small groups of joints.

* Corresponding author.

E-mail addresses: asaggese@unisa.it (A. Saggese), n.strisciuglio@rug.nl (N. Strisciuglio).

Chaudhry et al. [10] constructed a 3-dimensional hierarchical representation of the actions with skeleton features (the third dimension is the time) and the dynamics of these features are learned using Linear Dynamical Systems (LDSs). In more details, the skeleton is partitioned in joints and each joint is separately analyzed in terms of temporal motion. Yang and Tian [41] proposed a joint spatio-temporal representation of the difference in the positions of skeleton joints. In this way, both redundancy and noise are reduced. Principal Component Analysis (PCA) was applied to the spatio-temporal joint differences in order to obtain eigen-joint representations and, successively, action classification was performed with a Naive-Bayes-Nearest-Neighbor classifier.

The methods in the second group consider the entire skeleton to be composed of body parts, which are a connected set of rigid segments linking the joints. The movements of body parts are analyzed together and cast to the movements of the entire skeleton. Vemulapalli et al. [37] represented the skeleton in terms of 3-dimensional geometric relationships between different body parts using rotation and translation transformations in a 3-D space, called *Lie space*, where dynamic time warping, Fourier temporal pyramid representation and linear SVM are employed for classification. Laban movement analysis (LMA) framework, typically adopted for describing, annotating and interpreting human movement in the field of choreography, was used to describe the skeleton and its motion by Roudposhti et al. [27,28]. The proposed representation was combined with a classifier based on Bayesian Programming to perform action classification.

The existing approaches are mainly based on hand-crafted features that measure particular properties of the patterns of interest. The sets of features are usually designed by a person that has knowledge of the particular application domain. Extensions of already configured systems to recognize other patterns of interest may require the design of extended feature sets, which are meant to measure important characteristics of the new categories of patterns as well. Automatic learning of features is a desirable property of pattern recognition systems, also in cases where a limited amount of training samples is available.

Recent advances in *learning representations* of input data were made by using deep neural networks (DNNs) and convolutional neural networks (CNNs). Although powerful and able to learn effective data representations, deep and convolutional networks require very large amount of data to be trained, and the tuning of millions of parameters. Their high computational demand makes their implementation in low-power embedded systems a challenge.

In this paper, we propose a method for the analysis of human actions based on trainable feature extractors, which are learned directly from single prototype samples without requiring either a feature engineering process or a large amount of data to be effective. We start from the assumption that the action recognition problems can be decomposed into two sub-problems: a) given a skeleton pose, to extract a reliable representation b) given two sequences of skeleton poses, to encode them and compute their similarity.

In a preliminary work on skeleton pose feature extractors [29], we stated that a pose can be described by a particular spatial arrangement of joint positions, and we propose a trainable pose feature extractor that can be automatically configured on prototype skeleton patterns of interest. A configuration algorithm models the position of the joints with respect to a reference point in a given prototype skeleton, so configuring a new feature extractor. In the training process, we propose a feature selection procedure to optimize the number of configured feature extractors. Successively, in the application phase, the proposed feature extractor is used to detect the same skeleton pose used for configuration and also similar versions of it. The concept of trainable feature extractors was introduced by Azzopardi and Petkov [3], that proposed COSFIRE filters

for visual information processing. The main advantage of trainable feature extractors is their automatic configuration. Trainable feature extractors can be considered as a technique for *representation learning*, where the important characteristics of the patterns of interest are directly learned from training examples.

We take into account the temporal sequence of skeleton poses by using a string-based representation, and classify an action by means of a classifier based on string kernels.

We carried out experiments on three benchmark data sets, namely the MIVIA-Skeleton (MIVIA-S), MSR Daily Activity 3D (MSRDA), and the Berkeley Multimodal Human Action Database (MHAD) data sets. We recorded and made the MIVIA-S data set publicly available. The three considered data sets allow to test the proposed approach with data recorded by different kinds of acquisition devices, namely a very robust sensor suit (MHAD data set) and a less precise, but non-expensive depth sensor (MIVIA-S and MSRDA data sets). The performance results that we achieved are comparable or higher than the ones reported in existing published works.

The paper is organized as follows. In Section 2 we present the proposed trainable feature extractors and how we employ them for action classification. In Section 3 we present the data sets that we uses, and report the experimental results that we achieved together with a comparative analysis with other existing methods. We provide a discussion in Section 4 and draw conclusions in Section 5.

2. Method

Our basic idea is that human actions can be represented as temporal sequences of skeleton poses. We propose a method for action recognition based on a novel trainable feature extractor that can be configured to detect any preferred pose of human skeletons. For the design of the feature extractor, we assumed that the skeleton is provided by an external system, software or hardware (e.g. a kinect sensor). We designed an action classification method based on temporal analysis of the responses of the proposed feature extractors, by employing a string-like representation of the skeleton movement and a classifier based on *string kernels*. In Fig. 1, we show the architectural overview of the proposed action classification system.

2.1. Skeleton pose feature extractor

The structure of the proposed feature extractor is learned in an automatic configuration process performed on a training prototype skeleton of interest, rather than fixed in a-priori implemented skeleton models. This step can be considered a kind of *representation learning*, where a model of the pattern of interest is automatically constructed. In the application phase, the feature extractor achieves high response when applied to the same skeleton used for the configuration and also to similar skeleton configurations, accounting for generalization to new patterns. In the following of the section, we explain in details the configuration and application steps of the proposed feature extractor.

2.1.1. Configuration

Let us consider a prototype skeleton $\Upsilon = \{\zeta_i \mid i = 1, \dots, n_\Upsilon\}$, with n_Υ joints, and a reference point $\zeta_r = (x_r, y_r)$, which we choose to be the barycenter of the body. We represent the i -th joint point with a tuple $(a_i, \rho_i, \phi_i, \omega_i)$: a_i is the joint identifier, while ρ_i and ϕ_i are the polar coordinates of the position of the i -th joint with respect to the reference point (x_r, y_r) :

$$\rho_i = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2} \quad (1)$$

Fig. 1. Overview of the architecture of the proposed action classification method. In the training phase (dashed lines and boxes) the feature extractors are trained from prototype skeleton samples S_i , and successively their responses are used to construct feature vectors v_i for the description of input skeleton poses. The feature vector are mapped to a finite set of prototype poses (learned during the training phase by means of a k -Means clustering algorithm) for the construction of a pose sequence representation based on strings. Action classification is finally performed with a classifier base on string kernels.

Fig. 2. Examples of skeleton pose feature extractors configured on four skeletons that compose the action 'waving hand'. The (green) shaded areas indicate tolerance regions for the position of the joints, when the feature extractors are applied to new skeleton patterns. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\phi_i = \tan^{-1} \left(\frac{y_i - y_r}{x_i - x_r} \right) \quad (2)$$

where the values (x_i, y_i) indicate the location of the joint in the image reference system (the top-left corner is the point $(0, 0)$). The parameter ω_i is a weight assigned to the joint as a measure of its importance in the concerned action. In the configuration process we include *a-priori* knowledge about certain actions and divide the skeleton into two parts, corresponding to the upper and lower parts of the body. This allows to increase the selectivity of the configured feature extractors for actions that involve only specific parts of the body. In Fig. 2, we show examples of feature extractors configured on poses of the action 'waving hand', which involves only the upper part of the body, disregarding whether the subject is sitting or standing up. In such case, the weights ω_i of the joints in the lower skeleton are 0, so that they do not contribute to the configuration phase and to the computation of the skeleton pose feature. This procedure helps to limit the intra-class variations and to reduce the total number of configured feature extractors (e.g. a feature extractor configured on a skeleton of a sat

person that waves his hands will have strong response also on the skeleton of a person that waves hands while standing up).

2.1.2. Feature extraction

Given an input skeleton S , in the application phase, the value of a configured feature is computed by combining a score s_i for each of its joint points. The score of a joint is a measure of similarity of its position with respect to the expected position of the corresponding joint in the feature extractor model. In practice, the score value follows the profile of a Gaussian function, and it is defined as:

$$s_i(\Upsilon, S) = \exp \left\{ -\frac{\bar{d}(\zeta_i, \iota_i)^2}{2(\sigma'_i)^2} \right\} \quad (3)$$

where $\zeta_i \in \Upsilon$ and $\iota_i \in S$ are the i -th joints in the model and test skeletons, respectively. The function $\bar{d}(\zeta_i, \iota_i)$ is the relative euclidean distance between the position of corresponding joints of the two skeletons w.r.t. the reference joint point. It indicates the amount of drift of the i -th joint in the test skeleton w.r.t. its expected position in the model. The value σ'_i is the standard deviation of a Gaussian weighting function, which determines the amount of tolerance to the position of the i -th joint with respect to the position of its corresponding joint in the model Υ . It is a function of the distance $d^*(\zeta_r, \iota_i)$ between the position of the reference point ζ_r and the one of the joint ι_i , computed along the skeleton segments:

$$\sigma'_i = \sigma_0 + \alpha \cdot d^*(\zeta_r, \iota_i) \quad (4)$$

where σ_0 and α are parameters that regulate the amount of tolerance. More tolerance is given to terminal joints, such as hands and feet, since they have wider movement capabilities. In Fig. 3, we show an example of application of one feature extractor previously configured. The two joints of the arm on the left of the figure are not in the expected position according to the feature model. However, their score s_i is not null, since their location determined by the distance $\bar{d}(\zeta_i, \iota_i)$ (blue lines) is inside the tolerance region determined by the Gaussian weighting function. The use of a Gaussian weighting function accounts for some tolerance in the relative

Fig. 3. Example of application of a pre-configured feature extractor on a test skeleton. The (green) shaded circles refer to the tolerance regions around the joints, while the blue lines indicate the displacement $\bar{d}(\zeta_i, \iota_i)$ from the expected joint position according to the feature model. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

position of joints, which provides a certain degree of generalization to similar poses.

We compute the value of the proposed feature extractor by combining the scores of the joints by a weighted geometric mean operation:

$$r_{\Upsilon}(S) = \left[\prod_{i=1}^{|\Upsilon|} s_i(\Upsilon, S)^{w_i} \right]^{1/\sum_{i=1}^{|\Upsilon|} w_i}. \quad (5)$$

2.1.3. Tolerance to scale and reflection

In order to achieve robustness w.r.t. scale transformations of the skeletons, due to varying distance between the subjects and the video acquisition device, one can manipulate the parameters in the feature extractor and create a new set Υ_{λ} from the original model Υ , as follows:

$$\Upsilon_{\lambda} = \{(a_i, \lambda \rho_i, \phi_i, \omega_i) | i = 1, \dots, n_{\Upsilon}\} \quad (6)$$

where the parameter $\lambda > 0$ is a scale factor.

Furthermore, certain actions are symmetrical with respect to the vertical axis of the body. For example, *waving one hand* can be done with the right or the left arm. In such cases, we can achieve co-variance w.r.t. reflection of the poses of interest by modifying the model Υ , so obtaining a new set of tuples $\hat{\Upsilon}$ as follows:

$$\hat{\Upsilon} = \{(a_i, \rho_i, \pi - \phi_i, \omega_i) | i = 1, \dots, n_{\Upsilon}\} \quad (7)$$

We compute the value of the proposed feature, configured to be tolerant to scale and reflection transformations, as:

$$\hat{r}(S) = \max\{r_{\Upsilon}(S), r_{\Upsilon_{\lambda}}(S), r_{\hat{\Upsilon}}(S)\} \quad (8)$$

Fig. 4. Set of feature extractors selected with the proposed configuration algorithm, that represent the action 'jumping jack'.

2.2. A bank of feature extractors

We designed an iterative procedure to automatically train a near-optimal set of M feature extractors, with the aim of reducing the number of features to configure in order to effectively represent the training action sequences. The value M is a parameter. At each iteration, given an action (a sequence of poses), we train a feature extractor on the skeleton pose in the first frame. On the successive frames we compute the response of the previously trained feature extractors. The frame at which the responses of all the feature extractors are lower than a given threshold is used as key-frame for the configuration of a new feature extractor. We repeat this process on all the training actions. At the end of an iteration, if the total number of configured feature extractors is higher than M , the selection algorithm is repeated with an higher σ'_i value. This increases the tolerance of the configured feature extractors to variation of the skeleton poses, and reduces the number of features needed to represent the training actions. The update rule of σ'_i is defined as follows:

$$\begin{cases} \sigma'_i = \sigma_0^{(t)} + \alpha^{(t)} \cdot d^*(\zeta_r, \iota_i) \\ \sigma_0^{(t)} = \hat{\sigma} \cdot \sigma_0^{(t-1)} \\ \alpha^{(t)} = \hat{\alpha} \cdot \alpha^{(t-1)} \end{cases} \quad (9)$$

where the values in the t -th iteration depend on the ones in the $(t-1)$ -th iteration, and the parameters $\hat{\sigma}$ and $\hat{\alpha}$ are update factors. In Fig. 4, we show the selected features that represent the basic skeleton poses of the action *jumping jack*.

The selection process results in a bank of M feature extractors. We use the responses $\hat{r}_i$ (with $i = 1, \dots, M$) of the configured feature extractors to construct a feature vector $v = [\hat{r}_1, \hat{r}_2, \dots, \hat{r}_M]$ for the description of skeleton poses Fig. 5.

2.3. Action classification

We use the feature vector v in combination with a classification system to determine the class of a given pose skeleton, among N classes of interest. We explore two approaches to classify a given action, which is composed of a sequence of skeleton poses. The former is based on majority voting on the classification of single poses, while the latter exploits a representation of the pose sequence as a string and a classification strategy based on string kernels, as in [6].

2.3.1. Majority voting

Let us consider an action composed of a sequence of F poses. For each pose, we compute a feature vector v_i , which we use in combination with a multi-class Support Vector Machine (SVM) to distinguish between the N classes of interest $\{c_1, c_2, \dots, c_N\}$. We classify a given sequence as belonging to the class that is the most frequent among the frame-level classifications fc_i . The class C of the action under test is decided as follows:

$$C = \underset{j \in \{1, \dots, N\}}{\operatorname{argmax}} \left(\sum_i^F h_{ij} \right) \quad (10)$$

Fig. 5. Example of video frames from the (a) MIVIA-S, (c) MSRDA and (e) MHAD data sets, together with (b,d,f) their corresponding skeleton poses.

where

$$h_{ij} = \begin{cases} 1 & \text{if } f_{c_i} = c_j \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

It is important to point out that in the majority voting scheme, the temporal information about the sequence of skeleton poses is not taken into account for classification. Indeed, it has the limitation that similar actions in terms of poses, like stand-up and sit-down, might generate inter-class confusion.

2.3.2. Sequence classification with string kernels

We encode the sequence of feature vectors associated to an action into a string of symbols, each of them being a prototype pose. In order to determine the prototype pose symbols, we use the k -Means clustering algorithm on the frame-level feature vector. We consider the similarity of two actions as the similarity between their string representation and compute it with a string kernel based approach.

From feature vector to symbol: In the classification process we transform the feature vectors v_i into pose symbols according to a dictionary D (of finite size $|D|$) which we obtain during the training phase of the system. In details, we construct the dictionary of pose symbols by quantization of the feature space with the K -means clustering algorithm. The dictionary $D = \{d_1, \dots, d_{|D|}\}$ is the set of cluster centroids, which correspond to the learned pose symbols.

In the application phase, the dictionary D is used to convert the feature vector v_i into the corresponding pose symbol z_i , by associating it to the closest cluster centroid d_j :

$$z_i = \arg \min_j \|v_i - d_j\| \text{ for } j \in \{1, \dots, |D|\}. \quad (12)$$

From action to string: For a given time instant t , the sequence of the last L symbols is concatenated so as to obtain the string Z^t :

$$Z^t = \langle z_{t-L}, \dots, z_t \rangle \quad (13)$$

The value L is the average length of the action strings contained in the training set.

Similarity between strings: We compute the similarity between two strings in a kernel space. In more details, we exploited the *fast global alignment kernel* proposed by Cuturi [11], which computes a soft-minimum of all string alignment scores. In this way, we are able to take into account not only the best possible alignments, but also the contributions coming from all the possible alignments. As shown by Brun et al. [6], this approach allows to deal with strings of different lengths and time scales.

More formally, given the following two string of length n and m respectively: $X = \langle x_1, \dots, x_n \rangle$ and $Y = \langle y_1, \dots, y_m \rangle$, an alignment between X and Y is a pair of increasing integral vectors (π_1, π_2) of length $p < n + m$, such that $1 = \pi_1(1) \leq \dots \leq \pi_1(p) = n$ and $1 = \pi_2(1) \leq \dots \leq \pi_2(p) = m$, with unary increments and no simultaneous repetitions. Let $A(n, m)$ be the set of all possible alignments between X and Y . The global alignment kernel k_{GA} , which measures the similarity between two strings X and Y , is defined as:

$$k_{GA}(X, Y) = \sum_{\pi \in A(n, m)} \prod_{i=1}^{|\pi|} k(x_{\pi_1(i)}, y_{\pi_2(i)}, \pi_1(i), \pi_2(i)). \quad (14)$$

The kernel k is a combination of a triangular kernel k_t , which makes the global alignment kernel *fast*, and a weighted kernel k_w :

$$k(x_i, y_j, i, j) = \frac{k_t(i, j) \cdot k_w(x_i, y_j)}{1 - k_t(i, j) \cdot k_w(x_i, y_j)}, \quad (15)$$

where x_i and y_j represent two generic symbols, while i and j encode the position of these symbols inside the strings X and Y , respectively. The triangular kernel k_t is defined as follows:

$$k_t(i, j) = \left(1 - \frac{|i - j|}{T}\right)_+, \quad (16)$$

where T is the order of the kernel and $+$ refers to the fact that $k_t(i, j) = 0$ if $|i - j| \geq T$. The idea is that if the indices of two symbols differ by more than T , their kernel value is equal to 0.

The weighted kernel k_w , introduced by Brun et al. [6], allows to perform a soft assignment by evaluating the distance $f(x_i, y_i)$ between two symbols x_i and y_i associated to the respective centroids x_i and d_{y_i} :

$$f(x_i, y_i) = \|d_i - d_{y_i}\|^2. \quad (17)$$

The kernel $k_w(x_i, y_i)$ is computed as follows:

$$k_w(x_i, y_i) = e^{-\phi_\sigma(x_i, y_i)}, \quad (18)$$

where:

$$\phi_\sigma(x_i, y_i) = \frac{1}{2\sigma^2} d(x_i, y_i) + \log \left(2 - e^{-\frac{d(x_i, y_i)}{2\sigma^2}}\right). \quad (19)$$

Classification: Finally, we employ a k -NN classifier, using the kernel $k_{GA}(\cdot, \cdot)$ as distance metric between actions, to perform action classification.

3. Experimental analysis

3.1. Datasets

We carried out experiments on three benchmark data sets for human action recognition, namely the MSR Daily Activity 3D dataset (MSRDA) [39], the Berkeley Multimodal Human Action Database (MHAD) [23], and the MIVIA-Skeleton data set (MIVIA-S).

Table 1

Details of the composition of the MIVIA-S data set.

ID	Action	Avg. length	#Sequences	#Frames
A1	Standing still	4 secs	30	1793
A2	Sitting still	4 secs	30	1603
A3	Waving one hand	5 secs	30	2010
A4	Waving two hands	5 secs	30	1997
A5	Clapping	4 secs	30	1869
A6	Drinking	5 secs	30	2142
A7	Punching	4 secs	30	1896
A8	Walking	5 secs	30	2103
A9	Sit down	4 secs	30	1738
A10	Stand up	4 secs	30	1703

MIVIA-S. We collected and made publicly available for benchmark purposes the MIVIA-S data set.¹ It is composed of 10 actions performed by 10 different subjects (8 males and 2 females). Each subject performs a single action three times, except for the *waving hand* action, where we collected two repetitions for each hand, and for the *punching* action, where subjects were asked to repeat four times the action. The dataset was acquired using a Kinect depth sensor and contains, for each action, the RGB images, the depth images and the skeleton data (containing 20 joints). In Table 1, we report detailed information on the composition of the data set and the average length of action sequences.

MSRDA. This data set is composed of 16 classes of daily actions: drink (B1), eat (B2), read book (B3), call cellphone (B4), write on a paper (B5), use laptop (B6), use vacuum cleaner (B7), cheer up (B8), sit still (B9), toss paper (B10), play game (B11), lay down on sofa (B12), walk (B13), play guitar (B14), stand up (B15), sit down (B16). Each action was performed by ten different subjects, the first time while sitting and the other time while standing up, for a total of 320 action sequences. The data set was recorded with a Kinect sensor and contains highly noisy samples, i.e. the position of skeleton joints is not always reliable, which makes the classification problem very challenging.

MHAD. This data set contains 11 actions performed by 12 different persons (7 males and 5 females): Jumping in place (C1), Jumping jacks (C2), Bending - hands up all the way down (C3), Punching (boxing) (C4), Waving - two hands (C5), Waving - one hand (right) (C6), Clapping hands (C7), Throwing a ball (C8), Sit down then stand up (C9), Sit down (C10), Stand up (C11). Each subject performed 5 repetitions of each action, so resulting in 660 sequences, corresponding to about 82 minutes of video data. Differently from the MSRDA data set, the skeleton information was acquired with an optical motion capture system, which records the 3D position of active LED markers with a frequency of 480 Hz. The subjects were asked to wear a tight-fitting suit with 43 leds and the authors recorded the positions of the markers with eight motion-capture cameras arranged in a circular configuration. The skeleton data available in this data set is very precise and reliable.

3.2. Performance evaluation

We evaluated the performance results of the proposed method by computing the recognition rate (RR), error rate (ER) and miss rate (MR) on the classification of entire action sequences. The considered metrics are widely used in the literature on action recognition. The recognition rate is the ratio of correctly classified actions on the total number of action sequences of a given class. The error rate refers to those actions that are classified to wrong classes,

while the miss rate refers to actions classified as belonging to the background class and is computed as $MR = 1 - RR - ER$.

We carried out leave-one-out cross-validation experiments on the three considered data sets. For each action, we in turn used $M - 1$ sequences to train the proposed classifiers and the remaining one for testing. In Table 2, we report the configuration parameters of the proposed method that we determined in the cross-validation experiments.

3.3. Results and comparison

In Table 3, we report the results that we obtained on the MIVIA-S data set. We achieved a total average recognition rate of 0.83 using majority voting and of 0.95 using string kernel classifiers. It is noticeable that employing an approach that takes into account the temporal sequence of poses increases the reliability of classification. The classification errors made when using the action classification approach based on majority voting are related to actions which share similar poses. For instance, the actions *Sit down* (A9) and *Stand up* (A10) contains the same poses, but in inverse sequence. When taking into account the temporal sequence of skeleton pose features, the accuracy of classification increases from 0.50 and 0.60 to 0.90 and 0.97 for A9 and A10, respectively. Similar considerations can be made for the actions *Sitting still* (A2) and *Standing still* (A1), which can be seen as part of the actions *Sit down* (A9) and *Stand up* (A10).

We report the performance results that we achieved on the MSRDA data set in Table 4. We obtained average recognition rate equal to 0.64, error rate equal to 0.34 and miss rate equal to 0.02 when using the classification approach based on majority voting on frame-level classifications. Classification errors are mostly made on samples of the following classes: *read book* (B3), *write a paper* (B5), *use laptop* (B6) and *play game* (B11). Similarly to what we observed in the MIVIA-S data set, the most of errors are attributable to the fact that certain actions have many skeleton poses in common, resulting in inter-class confusion when using static skeleton information only. We observed errors also on the classification of the following actions: *sit down* (B15), *stand up* (B16), *sit still* (B9). In such cases, the majority voting scheme does not retain information about the temporal sequence of skeleton poses, which is fundamental to distinguish these kinds of action. We achieved higher performance results ($RR = 0.68$) by employing the classification approach based on string kernel, which takes into account the information on the sequence of poses. We observed an average improvement of the recognition rate of 0.4 for the actions B9, B15 and B16.

Similar observations can be done on the MHAD data set, for which we report detailed results in Table 5. We achieved average recognition rate equal to 0.85 and error rate equal to 0.15 by using majority voting classification on the frame-level decision. In a similar way to what we observed on the MIVIA-S and MSRDA data sets, the classification confusions are mostly obtained for the actions *Sit down then stand up* (C9), *Sit down* (C10) and *Stand up* (C11), that can not be correctly distinguished without taking into account the temporal information on skeleton pose sequences. The use of string kernels solves the errors on classes C9, C10 and C11, and contributes to an improvement of the average recognition rate up to 0.97.

We compared the results achieved by the proposed methods with the ones reported in the literature on the MHAD and MSRDA benchmark data sets. Experiments on these data sets contribute to evaluate and compare the performance results of the proposed approach used in combination with different data acquisition devices.

In Table 6, we report the results achieved by existing methods on the MSRDA data set and compare them with the ones obtained by the proposed methods based on trainable feature extractors. A

¹ The data set is available at the following url: <http://mivia.unisa.it>.

Table 2

Configuration parameters of the proposed method for the three considered data sets, which we determined from leave-one-out cross-validation experiments.

Configuration parameters for the proposed method				
Parameter	Description	MSRDA	MHAD	MIVIA-S
λ	Scale factor	{0.75, 1, 1.25}	{0.75, 1, 1.25}	{0.75, 1, 1.25}
σ_0	Regulate the size of the Gaussian weighting function for tolerance of joint positions	10	25	10
α		0.01	0.03	0.01
$\hat{\sigma}_0$	Update factors for the tolerance function σ'_i , used in the iterative feature configuration and selection process.	0.1	0.1	0.1
$\hat{\alpha}$		0.5	0.2	0.2
M	Number of configured feature extractors	400	1500	500
$ D $	Number of clusters for the vector quantization process for dictionary learning	256	64	64
T	Order of the triangular kernel used in string comparison	85	125	55
σ_K	Tolerance of Soft Weighted Kernel	1	5	1

Table 3

Results achieved by the proposed method for each class in the MIVIA-S data set.

Results on the MIVIA-S data set				
Class	Majority Voting		String Kernel	
	RR	ER	RR	ER
Standing still (A1)	0.77	0.20	0.83	0.07
Sitting still (A2)	0.60	0.40	0.97	0.03
Waving one hand (A3)	1.00	–	0.95	0.05
Waving two hands (A4)	1.00	–	1.00	–
Clapping (A5)	0.93	0.07	0.93	0.07
Drinking (A6)	0.83	0.17	0.93	0.07
Punching (A7)	1.00	–	1.00	–
Walking (A8)	1.00	–	1.00	–
Sit down (A9)	0.50	0.50	0.90	0.10
Stand up (A10)	0.60	0.40	0.97	0.03
Average	0.83	0.17	0.95	0.05

Table 4

Results achieved by the proposed method for each class in the MSRDA data set.

Results on the MSRDA data set				
Class	Majority Voting		String Kernel	
	RR	ER	RR	ER
Drink (B1)	0.70	0.20	0.90	0.10
Eat (B2)	0.70	0.20	0.30	0.70
Read book (B3)	0.60	0.20	0.50	0.50
Call cellphone(B4)	0.60	0.40	0.30	0.70
Write on paper (B5)	0.10	0.90	0.20	0.80
Use laptop (B6)	0.50	0.50	0.50	0.50
Use vacuum cleaner (B7)	1.00	–	1.00	–
Cheer up (B8)	1.00	–	1.00	–
Sit still (B9)	0.30	0.70	0.70	0.30
Toss paper (B10)	0.60	0.40	0.90	0.10
Play game (B11)	0.50	0.40	0.60	0.40
Lay on sofa (B12)	1.00	–	0.80	0.20
Walk (B13)	1.00	–	0.90	0.10
Play guitar (B14)	0.6	0.40	0.50	0.50
Stand up (B15)	0.6	0.40	1.00	–
Sit down (B16)	0.4	0.6	0.80	0.20
Average	0.64	0.34	0.68	0.32

direct comparison of performance is possible with the approaches that are based on the analysis of the skeleton information only. The average recognition rate that we achieved ($RR = 0.68$) is comparable with the results obtained by the method proposed by Wang et al. [39], which are the highest results achieved by methods that employ only skeleton information. Better results on the MSRDA data set are obtained by methods that take into account other information, such as color images or depth maps, to improve the classification accuracy.

In Table 7, we report the results that we achieved on the MHDA data set in comparison with the ones obtained by other exist-

Table 5

Results achieved by the proposed method for each class in the MHAD data set.

Results on the MHAD data set				
Class	Majority Voting		String Kernel	
	RR	ER	RR	ER
Jumping in place (C1)	0.82	0.05	1.00	–
Jumping jacks (C2)	0.93	0.07	0.95	0.05
Bending (C3)	1.00	–	0.92	0.08
Bozing (C4)	0.97	–	1.00	–
Waving - two hands (C5)	1.00	–	0.95	0.05
Waving - one hand (C6)	1.00	–	0.98	0.02
Clapping hands (C7)	0.93	0.07	0.92	0.08
Throwing a ball (C8)	1.00	–	1.00	–
Sit down and stand up (C9)	0.77	0.23	1.00	–
Sit down (C10)	0.42	0.58	1.00	–
Stand up (C11)	0.49	0.51	1.00	–
Average	0.85	0.14	0.97	0.03

Table 6

Comparison of the results achieved by the proposed approach with the ones achieved by existing methods on the MSRDA data set.

MSRDA data set - Result comparison		
Method	Reference	Accuracy
LOP	[39]	0.42
Depth Motion Maps	[42]	0.43
Dynamic Temporal Warping	[22]	0.54
3DTS	[19]	0.55
Eigen Joints	[40]	0.58
TSD	[19]	0.58
TSD+3DTS	[19]	0.63
Proposed (majority-voting)	–	0.64
TSD+RTD	[19]	0.65
Joint Position	[39]	0.68
Proposed (string-kernel)	–	0.68
Hierarchical Classification	[19]	0.72
HON4D	[25]	0.80
Actionlet Ensemble	[39]	0.85

ing methods. The recognition rate that we achieved by applying the proposed method with string kernel based classification ($RR = 0.97$) is comparable with the one obtained by state-of-the-art approaches. Is it worth noting that the results obtained by the classification method based on majority voting of frame-level decisions are comparable with the ones of many other methods, also based on deep network classifiers [14].

The high performance achieved by the proposed approach on different data sets, also compared with methods based on skeleton analysis, are attributable to the trainable characteristic of the proposed trainable feature extractors. The results reported in Tables 6 and 7 demonstrate that the proposed features can be used

Table 7

Comparison of the results achieved by the proposed approach with the ones achieved by existing methods on the MHAD data set.

MHAD data set - Result comparison		
Method	Reference	Accuracy
Reject Option	[8]	0.57
Cuboids	[12]	0.66
BOW	[13]	0.73
HMW	[24]	0.81
LDSP	[24]	0.82
HMIJ	[24]	0.83
Proposed (majority-voting)	–	0.85
Deep Learning	[14]	0.86
HARED	[5]	0.87
SMIJ	[24]	0.94
Aclets	[7]	0.95
HAcK	[6]	0.97
Proposed (string-kernel)	–	0.97

to recognize different kinds of actions, without requiring the design of feature sets to specific properties of the patterns of interest.

4. Discussion

One of the points of strength of the method that we propose is the *trainability* of the skeleton pose feature extractors, which can be automatically configured directly from prototype samples. The automatic configuration of feature extractors from training samples, that is among the main contributions of this work, can be considered a kind of *representation learning*. In the design of traditional pattern recognition systems, a set of suitable features has to be engineered to describe the salient characteristics of the data for the problem at hand. This process often requires extensive knowledge of the application domain and can be time-consuming. The use of trainable features is meant to avoid the feature engineering step and to automatically construct a set of suitable features from training data. The results that we achieved on different data sets show that the proposed trainable feature extractors provide the action recognition system with flexibility and adaptability to different recording devices and actions of interest. The concept of trainable feature extractors was introduced by Azzopardi and Petkov [3], that proposed COSFIRE filters for visual pattern recognition. Successively, COSFIRE filters have been employed in various applications of visual data processing such as contour detection [2], retinal vessel segmentation [4,32], curvilinear structure delineation [30,34], color object recognition [15] and adapted for audio event detection and classification [35].

Recently, deep learning and convolutional neural networks (CNNs) raised a great interest in the pattern recognition and computer vision communities since they are employed for learning effective representations of the patterns of interest directly from large amount of training data. Similarly to deep learning approaches, the proposed trainable feature extractors can construct suitable representations of the patterns of interest from training samples. The main advantage with respect to deep learning methodologies lies in the fact that the proposed trainable pose feature extractors do not required large amount of data for model training, since only one prototype example is necessary to train a new feature.

The proposed trainable feature extractors can effectively learn representations of skeleton poses. However, they carry one limitation that consists in describing only static information: they can be applied to extract features from skeletons in a given frame, i.e. at a fixed time instant. Hence, as an extension w.r.t. to our previous work [29], we employ the proposed feature extractors into an architecture that takes into account the temporal evolution of pose

sequences in order to increase the reliability and accuracy of classification. We represented the action as a sequence of symbols, i.e. a string, where each pose is a particular symbol, and used an action classifier based on string kernels. The combination of effective skeleton pose features and a mechanism for temporal integration of frame-level decisions provides reliability and allows to distinguish those actions that are composed of similar poses. Actions like *stand up* and *sit down*, which can likely be confused if analyzed at fixed time instants only, are robustly distinguished when the time dimension is taken into account. In principle, the proposed trainable feature extractors can be coupled with any other technique for classification of time series, such as Hidden Markov Models [21] or temporal bag of words [16].

A considerable property of the proposed feature extractors is that they account for tolerance w.r.t. the expected position of skeleton joints. This contributes to achieve robustness to deformations of the patterns of interest, and provides generalization capabilities to the proposed pose feature extractors that strongly responds to the same skeleton pattern used for configuration but also to modified versions of it.

A key requirement for action recognition systems, especially in robotics, is real-time response. The computational load of the proposed features is very low, since it consists of only shifting and multiplication of joint weighting function and real-time implementation for a the recognition of a reduces set of action was successfully realized.² However, in order to build a system that recognizes a larger number of actions, it is required to optimize the set of learned feature extractors that has to be compact for efficiency and effective for accuracy. In that respect a feature selection and optimization technique, like the ones presented for selection of relevant trainable features for the application at hand can be applied [31,33], on top of the feature set construction algorithms that we proposed in this work.

5. Summary and conclusion

In summary, the contributions of this work are the following: a) Novel trainable feature extractors for skeleton pose description whose structure is determined in an automatic configuration process on a single training skeleton, b) a selection method to configure a nearly-optimal set of feature extractors, c) a method for the recognition of human actions based on string kernels for the temporal integration of skeleton poses, d) the MIVIA-S data set of RGB-D videos and corresponding skeleton information of human actions recorded with a Kinect sensor, which we made publicly available for benchmark purpose.

The results that we achieved on three benchmark data sets, namely the MIVIA-S, MSRDA and MHAD data sets (recognition rate equal to 0.95, 0.68 and 0.97, respectively) demonstrate the effectiveness of the proposed trainable feature extractors and their temporal integration with string kernels.

References

- [1] J. Aggarwal, M. Ryoo, Human activity analysis: a review, *ACM Comput. Surv.* 43 (3) (2011) 16:1–16:43, doi:10.1145/1922649.1922653.
- [2] G. Azzopardi, N. Petkov, A CORF computational model of a simple cell that relies on lgn input outperforms the gabor function model, *Biol. Cybern.* 106 (2012) 177–189.
- [3] G. Azzopardi, N. Petkov, Trainable cosfire filters for keypoint detection and pattern recognition, *Pattern Anal. Mach. Intell. IEEE Trans.* 35 (2) (2013) 490–503, doi:10.1109/TPAMI.2012.106.
- [4] G. Azzopardi, N. Strisciuglio, M. Vento, N. Petkov, Trainable COSFIRE filters for vessel delineation with application to retinal images, *Med. Image Anal.* 19 (1) (2015) 46–57, doi:10.1016/j.media.2014.08.002.

² A demo video is available at: <http://www.mivia.unisa.it/>.

- [5] L. Brun, P. Foggia, A. Saggese, M. Vento, Recognition of human actions using edit distance on aclet strings, in: VISAPP 2015, 2015, pp. 97–103, doi:[10.5220/0005304700970103](https://doi.org/10.5220/0005304700970103).
- [6] L. Brun, G. Percannella, A. Saggese, M. Vento, Hack: A system for the recognition of human actions by kernels of visual strings, in: IEEE AVSS, 2014, pp. 142–147.
- [7] L. Brun, G. Percannella, A. Saggese, M. Vento, Action recognition by using kernels on aclets sequences, *Comput. Vis. Image Understanding* 144 (Supplement C) (2016) 3–13, doi:[10.1016/j.cviu.2015.09.003](https://doi.org/10.1016/j.cviu.2015.09.003).
- [8] V. Carletti, P. Foggia, G. Percannella, A. Saggese, M. Vento, Recognition of Human Actions from Rgb-d Videos Using a Reject Option, in: ICIAP 2013, 2013, pp. 436–445.
- [9] J.M. Chaquet, E.J. Carmona, A. Fernandez-Caballero, A survey of video datasets for human action and activity recognition, *Comput. Vis. Image Understanding* 117 (6) (2013) 633–659, doi:[10.1016/j.cviu.2013.01.013](https://doi.org/10.1016/j.cviu.2013.01.013).
- [10] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, R. Vidal, Bio-inspired dynamic 3d discriminative skeletal features for human action recognition, in: IEEE CVPRW, 2013, pp. 471–478, doi:[10.1109/CVPRW.2013.153](https://doi.org/10.1109/CVPRW.2013.153).
- [11] M. Cuturi, Fast global alignment kernels, in: ICML 2011, 2011, pp. 929–936.
- [12] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on, 2005, pp. 65–72.
- [13] P. Foggia, G. Percannella, A. Saggese, M. Vento, Recognizing human actions by a bag of visual words, in: IEEE SMC, 2013, pp. 2910–2915.
- [14] P. Foggia, A. Saggese, N. Strisciuglio, M. Vento, Exploiting the deep learning paradigm for recognizing human actions, in: IEEE AVSS, 2014, pp. 93–98.
- [15] B. Gecer, G. Azzopardi, N. Petkov, Color-blob-based COSFIRE filters for object recognition, *Image Vis. Comput.* 57 (2017) 165–174.
- [16] Z.-W. Gui, Y.-R. Yeh, Time series classification with temporal bag-of-words model, pp. 145–153. doi:[10.1007/978-3-319-13987-6_14](https://doi.org/10.1007/978-3-319-13987-6_14).
- [17] W. Hu, T. Tan, L. Wang, S. Maybank, A survey on visual surveillance of object motion and behaviors, *IEEE Trans. Syst. Man, Cybern. C* 34 (3) (2004) 334–352, doi:[10.1109/TSMCC.2004.829274](https://doi.org/10.1109/TSMCC.2004.829274).
- [18] G. Johansson, Visual perception of biological motion and a model for its analysis, *Percept. Psychophys.* 14 (2) (1973) 201–211, doi:[10.3758/BF03212378](https://doi.org/10.3758/BF03212378).
- [19] M. Koperski, P. Bilinski, F. Bremond, 3D Trajectories for Action Recognition, IEEE ICIP, 2014. Paris, France.
- [20] H. Lee, V. Morariu, L. Davis, Robust pose features for action recognition, in: IEEE CVPRW, 2014, pp. 365–372, doi:[10.1109/CVPRW.2014.60](https://doi.org/10.1109/CVPRW.2014.60).
- [21] Z. Li, Z. Wei, Y. Yue, H. Wang, W. Jia, L.E. Burke, T. Baranowski, M. Sun, An adaptive hidden Markov model for activity recognition based on a wearable multi-sensor device, *J. Med. Syst.* 39 (5) (2015). 57 doi: [10.1007/s10916-015-0239-x](https://doi.org/10.1007/s10916-015-0239-x).
- [22] M. Müller, T. Röder, Motion templates for automatic classification and retrieval of motion capture data, in: ACM SIGGRAPH, 2006, pp. 137–146.
- [23] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy, Berkeley mhad: a comprehensive multimodal human action database, in: IEEE WACV, 2013, pp. 53–60, doi:[10.1109/WACV.2013.6474999](https://doi.org/10.1109/WACV.2013.6474999).
- [24] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy, Sequence of the most informative joints (smij): a new representation for human skeletal action recognition, *J. Vis. Commun. Image Represent.* 25 (1) (2014) 24–38.
- [25] O. Oreifej, Z. Liu, Hon4d: histogram of oriented 4d normals for activity recognition from depth sequences, in: IEEE CVPR, 2013, pp. 716–723, doi:[10.1109/CVPR.2013.98](https://doi.org/10.1109/CVPR.2013.98).
- [26] R. Poppe, A survey on vision-based human action recognition, *Image Vis. Comput.* 28 (6) (2010) 976–990, doi:[10.1016/j.imavis.2009.11.014](https://doi.org/10.1016/j.imavis.2009.11.014).
- [27] K.K. Roudposhti, J. Dias, P. Peixoto, V. Metsis, U. Nunes, A multilevel body motion-based human activity analysis methodology, *IEEE Trans. Cognitive Dev. Syst.* 9 (1) (2017) 16–29.
- [28] K.K. Roudposhti, U. Nunes, J. Dias, Probabilistic social behavior analysis by exploring body motion-based patterns, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (8) (2016) 1679–1691.
- [29] A. Saggese, N. Strisciuglio, M. Vento, N. Petkov, Action recognition by learning pose representations, *CoRR abs/1708.00672* (2017).
- [30] N. Strisciuglio, G. Azzopardi, N. Petkov, Detection of curved lines with b-cosfire filters: a case study on crack delineation, in: CAIP, 2017, pp. 108–120, doi:[10.1007/978-3-319-64689-3_9](https://doi.org/10.1007/978-3-319-64689-3_9).
- [31] N. Strisciuglio, G. Azzopardi, M. Vento, N. Petkov, Multiscale Blood Vessel Delineation Using B-COSFIRE Filters, in: CAIP 2015, in: LNCS, 9257, 2015, pp. 300–312, doi:[10.1007/978-3-319-23117-426](https://doi.org/10.1007/978-3-319-23117-426).
- [32] N. Strisciuglio, G. Azzopardi, M. Vento, N. Petkov, Unsupervised Delineation of the Vessel Tree in Retinal Fundus Images, in: VIPIMAGE, 2015, pp. 149–155.
- [33] N. Strisciuglio, G. Azzopardi, M. Vento, N. Petkov, Supervised vessel delineation in retinal fundus images with the automatic selection of B-COSFIRE filters, *Mach. Vis. Appl.* (2016). 113.
- [34] N. Strisciuglio, N. Petkov, Delineation of line patterns in images using b-cosfire filters, in: IEEE IWOB, 2017, pp. 1–6, doi:[10.1109/IWOB.2017.7985538](https://doi.org/10.1109/IWOB.2017.7985538).
- [35] N. Strisciuglio, M. Vento, N. Petkov, Bio-inspired Filters for Audio Analysis, in: BrainComp 2015, Revised Selected Papers, in: LNCS, 10087, 2016, pp. 101–115, doi:[10.1007/978-3-319-50862-7_8](https://doi.org/10.1007/978-3-319-50862-7_8).
- [36] P. Turaga, R. Chellappa, V. Subrahmanian, O. Udrea, Machine recognition of human activities: a survey, *IEEE Trans. Circuits Syst. Video Technol.* 18 (11) (2008) 1473–1488, doi:[10.1109/TCSVT.2008.2005594](https://doi.org/10.1109/TCSVT.2008.2005594).
- [37] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3d skeletons as points in a lie group, in: IEEE CVPR, 2014, pp. 588–595, doi:[10.1109/CVPR.2014.82](https://doi.org/10.1109/CVPR.2014.82).
- [38] S. Vishwakarma, A. Agrawal, A survey on activity recognition and behavior understanding in video surveillance, *Vis. Comput.* 29 (10) (2013) 983–1009, doi:[10.1007/s00371-012-0752-6](https://doi.org/10.1007/s00371-012-0752-6).
- [39] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: CVPR'12, 2012, pp. 1290–1297.
- [40] X. Yang, Y. Tian, Eigenjoints-based action recognition using Naive-Bayes-nearest-neighbor, in: IEEE CVPRW, 2012, pp. 14–19, doi:[10.1109/CVPRW.2012.6239232](https://doi.org/10.1109/CVPRW.2012.6239232).
- [41] X. Yang, Y. Tian, Effective 3d action recognition using eigenjoints, *J. Vis. Commun. Image Represent.* 25 (1) (2014) 2–11. Visual Understanding and Applications with RGB-D Cameras. doi: [10.1016/j.jvcir.2013.03.001](https://doi.org/10.1016/j.jvcir.2013.03.001).
- [42] X. Yang, C. Zhang, Y. Tian, Recognizing actions using depth motion maps-based histograms of oriented gradients, in: ACM International Conference on Multimedia, 2012, pp. 1057–1060, doi:[10.1145/2393347.2396382](https://doi.org/10.1145/2393347.2396382).
- [43] A. Yao, J. Gall, G. Fanelli, L.J. Van Gool, Does human action recognition benefit from pose estimation? BMVC, 2011.