A human activity recognition framework using max-min features and key poses with differential evolution random forests classifier

Urbano Miguel Nunes, Diego R. Faria, Paulo Peixoto

PII: S0167-8655(17)30141-1
DOI: 10.1016/j.patrec.2017.05.004
Reference: PATREC 6810

To appear in: *Pattern Recognition Letters*

Received date: 20 October 2016
Revised date: 25 March 2017
Accepted date: 1 May 2017

Please cite this article as: Urbano Miguel Nunes, Diego R. Faria, Paulo Peixoto, A human activity recognition framework using max-min features and key poses with differential evolution random forests classifier, *Pattern Recognition Letters* (2017), doi: 10.1016/j.patrec.2017.05.004

Highlights

- Simple and effective approach that extracts skeleton-based max-min features;
- Fast training times requiring few training examples;
- Random forest classifier with no thresholds to tune;
- Differential evolution as base in seeking for the best splitting node condition;
- Code of the proposed random forest classifier in C++ freely available.

Pattern Recognition Letters
journal homepage: www.elsevier.com

A human activity recognition framework using max-min features and key poses with differential evolution random forests classifier

Urbano Miguel Nunes[a,**], Diego R. Faria[b], Paulo Peixoto[a]

[a] *Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, Portugal*
[b] *System Analytics Research Institute, School of Engineering and Applied Science, Aston University, Birmingham, UK*

ABSTRACT

This paper presents a novel framework for human daily activity recognition that is intended to rely on few training examples evidencing fast training times, making it suitable for real-time applications. The proposed framework starts with a feature extraction stage, where the division of each activity into actions of variable-size, based on key poses, is performed. Each action window is delimited by two consecutive and automatically identified key poses, where static (i.e. geometrical) and max-min dynamic (i.e. temporal) features are extracted. These features are first used to train a random forest (RF) classifier which was tested using the CAD-60 dataset, obtaining relevant overall average results. Then in a second stage, an extension of the RF is proposed, where the differential evolution meta-heuristic algorithm is used, as splitting node methodology. The main advantage of its inclusion is the fact that the differential evolution random forest has no thresholds to tune, but rather a few adjustable parameters with well-defined behavior.

1. Introduction

Robot perception is a very active research area, which combines research endeavors from many fields, such as computer vision, machine learning and pattern recognition. It is a very challenging field due to the dynamic nature of the environment in real-world application scenarios. This is specially true when there is the need for robots to interact with humans, like in the case of assistive robots. Not only a human activity recognition system is necessary, but it also has to be very fast, capable of adapting rapidly to different actions performed by users. For instance, in the case of assistance to elderly people, robots could be used to recognize activities/actions, in order to improve the quality of life of those people by, not only assisting them, but also identifying life-risk situations (e.g. falling) (Parisi and Wermter, 2016). In recent years, this field of research has received the attention of researchers from all around the world, specially with the introduction of RGB-D sensors (Vieira et al., 2012). These sensors provide depth images and 3D point clouds with important attributes, such as robustness to illumination's variation, scaling and rotation. Also, a 3D human skeleton is possible to be acquired in real-time (Shotton et al., 2013), with affordable equipment, such as the Microsoft Kinect RGB-D cameras. 3D skeleton-based representation has the potential to describe a human body and motion, with a relatively small amount of information, such as joint positions, as demonstrated by Johansson (1973). Also, it is possible to extract additional meaningful information from them, such as joint velocities and accelerations (i.e. skeleton-based features). In addition, not only 3D skeleton-based representations are robust to illumination's variability and camera's perspective view, but they are also not significantly affected by skeleton's rotation or motion speed. Such human representation method is particularly suited for real-time applications, since it provides a compact representation of the human body, requiring less computational power to process it. In this context, this paper contributes with the proposal of a novel approach for human activity recognition, for real-time application scenarios, where characteristics like the number of training samples and the time needed for training play an important role. Also, a variation of the Random Forest (RF) classifier is proposed and used for the classification. It

**Corresponding author:
e-mail: urbanomiguel.g.nunes@ieee.org (Urbano Miguel Nunes), d.faria@aston.ac.uk (Diego R. Faria), peixoto@isr.uc.pt (Paulo Peixoto)

integrates a new method for finding the best splitting node condition in decision trees (DT), based on the differential evolution (DE) meta-heuristic algorithm. This paper is an extension of the work proposed by Nunes et al. (2016) and its main contributions are the following:

1. A simple and effective approach to extract extremal skeleton information (e.g. max-min dynamic features), based on variable-sized actions delimited by key poses;
2. Very fast training times, requiring few training examples, making it suitable for real-time applications;
3. Random forest (RF) classifier with no thresholds to tune, where the best splitting node condition in each decision tree is found, based on the DE algorithm.

For evaluation purposes, the CAD-60 (Sung et al., 2011) was used and the relevant performance obtained may serve as a solid human activity recognition framework for real-time applications. In Fig. 1, an overview of the global framework proposed is presented. The remainder of the paper is organized as follows: in section 2, the relevant related work is briefly described, highlighting the contributions that the proposed approach provide; in section 3, the approach that is proposed for features extraction is explained; in section 4, the proposed RF classifier with the inclusion of the DE algorithm to find the best splitting node condition is discussed; in section 5, the results of the experimental procedure used to evaluate the proposed method are shown and analyzed; in section 6, the key ideas proposed are summarized and some new related lines of research for the near future are introduced.

2. Related Work

Human activity recognition has become a very important research area, due to its future possible real-world applications, such as surveillance (Jun et al., 2013), assistive living (Okada et al., 2005) and human-machine interaction (Song et al., 2012). Some relevant related work is presented concerning: human activity recognition systems; random forests applications and improvements; and finally differential evolution applications.

2.1. Human Activity Recognition

As mentioned previously, 3D skeleton-based representations can be easily obtained with affordable equipment, making them suitable for real-time applications, due to the relatively small amount of information extracted. This representation is very compact and allows the extraction of meaningful information, based on the skeleton joints. Such representation has been the base for several human activity recognition frameworks. An approach of action segmentation using key poses, based on kinetic energy, is employed by Shan and Akella (2014), so that it becomes insensitive to nonlinear stretching. This is a very useful property, since human activities may be performed at distinct rates. Faria et al. (2015) proposed a probabilistic ensemble of classifiers, using skeleton-based features, where temporal information is used, accounting for uncertainty measures. The authors showed that the use of an ensemble of classifiers is advantageous, when comparing its performance with the performance of each of its individual constituents classifiers. Using a clustering algorithm, Cippitelli et al. (2016) propose the extraction of sequences of distinct postures, defined as key poses[1], without the need of a learning procedure. However, it has some difficulty distinguishing between similar activities. Zhu et al. (2016) propose a human action recognition system based on sequences of poses and atomic motions. These sequences are encoded into multi-layer codebooks, which are used to classify each activity. Koppula et al. (2013) proposed a framework, where, not only information about a human activity sequence is extracted, but also the interaction of the human with the surrounding objects, in terms of associated affordances. This approach has the merit to extract some environmental context, which can be very useful when recognizing human activities. Besides the high performances in terms of precision and recall, other performance indicators, such as training/testing time, number of examples in the training set or computational memory usage are not easily accessed nor available. In this sense, having in view possible real-time applications for the robotic domain, there is a clear opportunity for the development of classification strategies for human activity recognition that are fast to train and rely on few training samples, possible allowing the learning of new unknown activities on the fly.

2.2. Random Forests

Random forests (RF) were first introduced by Breiman (2001). RF consist on an ensemble of classifiers (e.g. decision trees (DT)) with low bias and variance performances. Inheriting some advantageous properties of DT and, at the same time, bridging some of their disadvantageous, such as the fact that they are very sensitive to noise, RF have become popular for solving classification and regression problems. These classifiers have proved to be very accurate, simple and fast, comparing to other machine learning techniques (Hastie et al., 2008). A comprehensive work about the properties of RF has been developed by Louppe (2014), which served as a base for the proposed RF implementation. Several improvements were proposed, in order to increase the overall performance of the RF: Robnik-Šikonja (2004) proposed attribute evaluation measures and voting weighted, in order to increase strength or decrease correlation of individual DT in the ensemble; Tsymbal et al. (2006) proposed an improvement in the prediction performance of RF, by replacing the majority voting with dynamic integration; Segal (2004) demonstrated that small performance gains can be made by limiting the size of each DT in the ensemble and thus reducing the problem of overfiting. Although these approaches achieve better overall performances, the introduced overhead can make them inappropriate for time demanding applications. In this sense, the proposed RF algorithm is in line with the ones proposed by Breiman (2001) and Hall et al. (2009). Also, RF have been used in the context of human pose recognition (Shotton et al., 2013), human action recognition (Gan and Chen, 2013), human gesture recognition (Miranda et al., 2012) and recognizing temporal events (Demirdjian

[1] In the referenced work, the definition of key pose is the one proposed by Baysal et al. (2010), while in the present work, it follows the definition of Shan and Akella (2014).

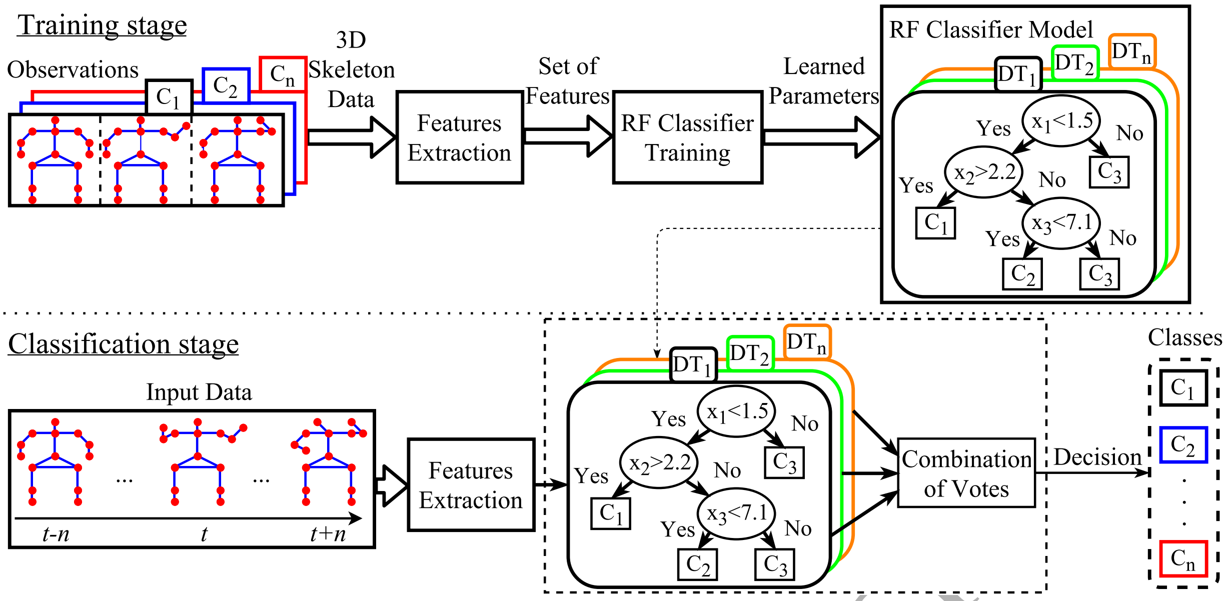


Fig. 1: Overview of the global architecture of the proposed framework. **Training stage**: a RF classifier is trained from each class (i.e. activity), considering observations of humans performing labeled activities; for each observation, 3D skeleton data is extracted, as well as discriminative information (i.e. features) and a RF model is built. **Classification stage**: given a set of observations of a human performing an activity, features are extracted and selected; the previously trained RF classifier makes a decision, based on the considered features, classifying the human activity performed.

and Varri, 2009). Two interesting characteristics of the RF, that are transversal to each one of the mentioned applications, are its training/testing speed and its high accuracy prediction.

2.3. *Differential Evolution*

The Differential Evolution algorithm (DE) (Storn and Price, 1997) is a very competitive optimization algorithm. Based on the survey of Das and Suganthan (2011), DE is the only evolutionary optimizer to secure competitive rankings in all International Conference on Evolutionary Computation (CEC) editions. It is widely used in several fields of research, due to its simplicity, high performance, convergence speed, low number of control parameters that are well-studied and low space complexity. In the classical approach, three parameters must be adjusted: mutation factor F, recombination factor C and the number of individuals of the population N_{pop}. Several studies have been conducted to evaluate the DE's performance, depending on the choice of these parameters (Ronkkonen et al., 2005). In this article, the authors point to the fact that $F = 0.9$ is a relevant first choice and $0.9 \leq C \leq 1$, if the function's parameters are dependent. More recently, self-adaptation techniques, concerning the optimal F and C parameter values, have been studied (Liu and Lampinen, 2005), (Brest et al., 2006). Although these approaches provide better convergence speed and accuracy, in the context of this work, the overhead of such procedures possibly does not compensate its incorporation. This is due to the fact that it is not expected to be provided the best solution by the DE algorithm, since even in such case, the optimal global solution in the scope of this work is not guaranteed, as explained in Sect. 4. Due to its properties, DE is applied in many research areas, such as aircraft control (Menon et al., 2006), robot control (Neri and Mininno, 2010), clustering data (Das et al., 2008), digital filtering design (Storn, 2005) and molecular configuration (Moloi and Ali, 2005). Also it has been applied in machine learning, such as on the training of artificial neural networks (Subudhi and Jena, 2008), (Chauhan et al., 2009), where it has proven its merits. Therefore, in this work, the DE algorithm is used to find the best possible splitting node condition in the context of growing trees, so that an ensemble of classifiers (e.g. random forests) can be formed. To the best of our knowledge, no such procedure has been proposed nor considered.

3. Proposed Approach

Considering the coordinates system defined as (x, y, z) corresponding to the width, height and depth, respectively, relatively to the camera, a dataset containing 3D coordinates of skeleton's joints, describing relevant information of a person performing an activity, is assumed to be provided. Each 3D joint's position is given by $P^t_j = \left(p^t_{jx}, p^t_{jy}, p^t_{jz}\right)$, where p^t_{dj} is the value of the coordinate $d \in \{x, y, z\}$ of the joint $j \in \{1, ..., m\}$, at frame t and m is the number of body's joints. Based on the provided dataset, a set of static and dynamic features $\mathbf{F}$ is extracted. Fig. 2 presents an overview of the proposed features extraction approach.

3.1. *Preprocessing of 3D Skeleton Data*

A preprocessing step is applied to the 3D skeleton data in order, not only to attenuate noise introduced by the sensor, but also to normalize the data to accommodate for different user's height, limb length, orientation and position. This preprocessing stage consists on the following steps:

1. **Translation**, to guarantee the same origin of the coordinates system for all acquired frames; the reference was set to the *torso* of the human skeleton;

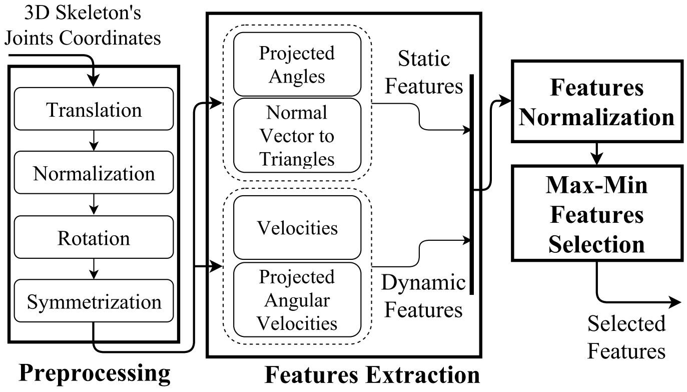


Fig. 2: Overview of the feature extraction process. The skeleton's joints coordinates are first preprocessed, undergoing a set of transformations: translation, normalization, rotation and symmetrization. Then, relevant information is extracted, such as velocities of joints, projected angles formed by three joints, etc. These features are normalized and finally, max-min skeleton-based features are selected.

2. **Normalization**, to reduce the influence of different user's height and limb lengths; first, the height of the subject is determined; then all skeleton 3D coordinates are normalized according to this value;
3. **Rotation**, to guarantee that the activity is always observed from the same point of view, independently of the initial pose of the subject with respect to the camera. This procedure is based on the work of Wang et al. (2014) and consists on the rotation of the skeleton in the y axis, considering the plane formed by the *torso*, *right* and *left hips* to make it fronto-parallel in relation to the camera plane;
4. **Symmetrization**, to disambiguate between mirrored versions of the same activity (e.g. gestures performed by right and left-handed people); since the skeleton is already in the same fronto-parallel pose with respect to the camera, it is just necessary to consider a new sample based on a mirrored version of the original 3D skeleton data.

3.2. Spatio-Temporal Features

After the preprocessing stage, the relevant and discriminative information (i.e. features) is extracted in the next step. Before enumerating the considered discriminative features, one should notice that they are divided into two distinct categories: *static* (e.g. geometrical) and *dynamic* (e.g. temporal) features. Static features are relevant to represent extremal positions of the skeleton, defined as *key poses*, where the skeleton pose has zero kinetic energy (Shan and Akella, 2014) (i.e. at a given frame, each joint of the skeleton has zero kinetic energy; therefore the body has no movement). On the other hand, dynamic features are intended to encode information about the skeleton's motion, described in terms of joint movements between key poses.

These features are combined along time to form a set of features $\mathbf{F}'$ represented in a matrix form, where each row contains the features that are computed at a given frame and each column corresponds to the variation of each feature along time.

3.2.1. Static Features

1. **Projected distances** between two joints (a and b) as

$$\delta^t_{ab} = \sqrt{\sum_{d'} \left(p^t_{ad'} - p^t_{bd'}\right)^2}, \tag{1}$$

where d' belongs to one of the following sets, for each projection considered: $\{x, y\}$, $\{y, z\}$, $\{z, x\}$;

2. **Projected angles** based on three joints (a, b and c) as

$$\theta^t_{id_p} = \arccos\left(\frac{(\delta^t_{ab})^2 + (\delta^t_{bc})^2 - (\delta^t_{ac})^2}{2 \cdot \delta^t_{ab} \cdot \delta^t_{bc}}\right), \tag{2}$$

where δ is the distance between two joints, given by Eq. (1) and $d_p \in \{xy, yz, zx\}$, for each projection considered;

3. **Normal vector to triangles** formed by three joints (a, b and c) as

$$\Delta^t_k = \frac{(P^t_a - P^t_b) \times (P^t_a - P^t_c)}{\|(P^t_a - P^t_b) \times (P^t_a - P^t_c)\|}; \tag{3}$$

3.2.2. Dynamic Features

1. **Velocities of joints coordinates** as

$$v^t_{jd} = \left(p^t_{jd} - p^{t-1}_{jd}\right) \cdot f_r, \tag{4}$$

where f_r is the frame rate;

2. **Projected angular velocities** as

$$\omega^t_{id_p} = \left(\theta^t_{id_p} - \theta^{t-1}_{id_p}\right) \cdot f_r. \tag{5}$$

3.3. Feature Normalization

Feature normalization is a recurrent practice in several machine learning domains (Chapelle and Keerthi, 2008), (Forman et al., 2009), (Faria et al., 2015). In this sense, the data normalization is done according to:

$$f_{ij} = \frac{f'_{ij} - \min(\mathbf{F}'_{\mathrm{tr}\cdot j})}{\max(\mathbf{F}'_{\mathrm{tr}\cdot j}) - \min(\mathbf{F}'_{\mathrm{tr}\cdot j})}, \tag{6}$$

where f'_{ij} is the current value being normalized, f_{ij} is its respective value normalized and $\mathbf{F}'_{\mathrm{tr}\cdot j}$ refers to the column j of the matrix $\mathbf{F}'_{\mathrm{tr}}$, representing the consecutive occurrences of the same feature. This process is done for both training and testing sets, resulting in $\mathbf{F}_{\mathrm{tr}}$ and $\mathbf{F}_{\mathrm{te}}$ matrices, respectively. From this point on, these sets are generically referred as $\mathbf{F}$, since the following steps are applied to both of them. $\mathbf{F}$ is a set in matrix form containing all examples of all activities performed of the form:

$$\mathbf{F} = \left[\left(\mathbf{F}^1\right)^T \quad \left(\mathbf{F}^2\right)^T \quad \ldots \quad (\mathbf{F}^a)^T \quad \ldots\right]^T, \tag{7}$$

where $\mathbf{F}^a$ is a sub-matrix relatively to activity a.

3.4. Selection of Max-Min Skeleton-based Features based on Key Poses

Many activities consist of repetitive action sequences. In this sense, it is possible to assume that each action may be discriminated just by considering extreme movements (given by dynamic features) and extreme poses (given by static features). Based on this assumption, the proposed approach extracts, for each activity, the maximum and minimum local values of the considered dynamic features, within a variable-size action window. This window is automatically determined by frames where the skeleton has zero kinetic energy. These poses are defined as *key poses* (Shan and Akella, 2014). Key poses represent extreme points in the motion path of each joint, where most of the discriminative properties of each action are encoded. In this sense, an activity may be represented by a sequence of distinct body actions, and key poses may be used to determine/delimit their respective window size. In other words, each action is determined by considering two consecutive key poses, which delimit a variable-size window. Based on Shan and Akella (2014), the pose kinetic energy is defined as

$$E^t = \frac{1}{2} \sum_{j=1}^{m} \sum_{d} \left(v_{jd}^t \right)^2 , \tag{8}$$

where $d \in \{x, y, z\}$ and the key poses must satisfy

$$E^t < E_{\text{min}}, \tag{9}$$

where E_{min} is a tuned threshold, which is close to zero to accommodate noise in the feature space. This method has low computational cost and is relatively fast to compute, since just max-min local values need to be computed.

In order to avoid consecutive frames to be considered as key poses, an upper bound E_u is introduced after the detection of the first key pose, to guarantee that the kinetic energy value evidences a new motion being performed by the skeleton. The next key pose is only determined if the value of the kinetic energy rises again above this upper bound.

Considering that each sub-matrix $\mathbf{F}^a$ describing an activity a is divided into variable-size sample windows as

$$\mathbf{F}^a = \left[\left(\mathbf{F}_1^a\right)^T \quad \left(\mathbf{F}_2^a\right)^T \quad \ldots \quad (\mathbf{F}_w^a)^T \quad \ldots \right]^T , \tag{10}$$

the following set of features is considered, since it reached the better overall performance in several experimental tests:

$$\mathbf{F}_w^a = \left[v_{jd}^t \quad \omega_{id_p}^t \quad \theta_{id_p}^t \quad \Delta_k^t \right], \tag{11}$$

with a size given by $n_a \times [3\,((m - m_{\text{extd}}) + n_\theta + n_\Delta) + n_\omega]$, where n_a is the size of the activity sample window, m_{extd} is the number of joints not considered for the velocity feature (e.g. *torso*), n_θ is the number of projected angles between joints, n_Δ is the number of considered normals to triangles formed by three joints and n_ω is the number of projected angular velocities considered.

The notion of static and dynamic features is crucial from this point on. Key poses are determined by frames where the skeleton joints have no movement or, in other words, near zero velocity. In this sense, for these poses, dynamic features (e.g. joint velocities) are not considered, since they become irrelevant. On the contrary, static features are very discriminative in key poses and must be considered as such. Dynamic features however become very important in between key poses, since the body's motion occurs in those frames, while information provided by static features loses significance.

For the case of the dynamic features, a max-min approach is followed, where only the maximum and minimum values of each considered feature (i.e. v and ω) in the window are used. For the static features (i.e. θ and Δ) only the ones that correspond to the key poses used to define the analysis window (i.e. the first w_{i_f} and the last w_{i_l}) are considered. This means, not only a set of interpretable features may be extracted, but also the set itself has its own meaning and intuition appealing, in the context of discriminating an activity. Therefore, the **main contribution** of this approach is not centered in the selected features themselves, but in the way they are combined and used to discriminate each activity. It may be used with other (possibly more discriminative) features, as long as they can be categorized as being static or dynamic features.

From each activity sample $\mathbf{F}_w^a$, an instance vector is constructed:

$$f_{w_i}^a = \left[f_{w_{i_f}}^{\text{static}} \quad f_{w_i}^{\text{dynamic}} \quad f_{w_{i_l}}^{\text{static}} \right], \tag{12}$$

where $f_{w_{i_f}}^{\text{static}}$ and $f_{w_{i_l}}^{\text{static}}$ represent the static features that are selected of the first and last key poses identified of the window, given generically by

$$f_t^{\text{static}} = \left[\theta_{id_p}^t \quad \Delta_k^t \right], \tag{13}$$

and $f_{w_i}^{\text{dynamic}}$ represents the max-min dynamic features that are selected as

$$f_{w_i}^{\text{dynamic}} = \left[(v_{jd}^{\text{max}})_{w_i} \quad (\omega_{id_p}^{\text{max}})_{w_i} \quad (v_{jd}^{\text{min}})_{w_i} \quad (\omega_{id_p}^{\text{min}})_{w_i} \right]. \tag{14}$$

Each considered instance vector has a length given by $[2 \cdot 3(n_\theta + n_\Delta) + (n_{\text{max}} + n_{\text{min}}) \cdot (3(m - m_{\text{extd}}) + n_\omega)]$, where n_{max} and n_{min} are the number of maximum and minimum values for each considered feature. An example of a sequence of skeleton poses of a human waving with an arm is illustrated in Fig. 3. A plot of the corresponding kinetic energies is represented as well.

3.5. Practical Considerations on a Real-Time Implementation

Based on the description of the proposed approach and considering a practical real-time implementation, the computation of the described features and kinetic energy could be done at the frame rate at which 3D skeleton's joints coordinates are captured. Since each sample must be in the form of Eq. (12), a first key pose must be detected and, for that frame, f_1^{static} features are computed. Following that frame, max-min dynamic features are computed and updated, forming f_1^{dynamic}. Once a second key pose is detected (i.e. forming f_2^{static}), a sample vector is formed and tested in order to discriminate the respective activity.

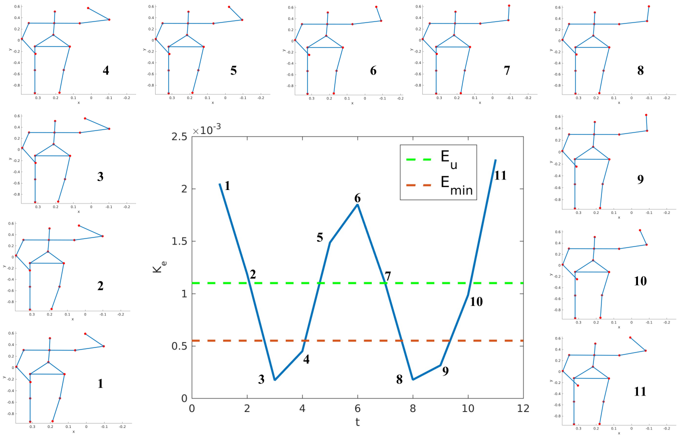


Fig. 3: Example of a sequence of human poses and the plot of their respective kinetic energies. In this example, based on the thresholds that are considered ($E_{\min}$ and E_u), pose 3 and 8 are identified as key poses. Static features are considered in these extreme poses, which will also delimit a variable-size window (in this case, the size of the window is 6 frames). From pose 4 to 7, max-min dynamic features are selected. Note that, after pose 3 being classified as key pose, only from pose 5 a new key pose may be identified, since the kinetic energy of this pose is higher than the upper threshold (i.e. pose 4 can not be identified as key pose, even though its kinetic energy is lower that $E_{\min}$).

4. Classifier - Random Forests

RF classifier consist on an ensemble of DT and each one is trained from a bootstrap sample of the original training set, so that each sample is independent identically distributed (i.i.d.), minimizing the correlation between the predictors. Another source of randomness is that, in the case of tree predictors, such as DT, which are used in the majority of the literature and in this work, each node of each DT is divided considering random selections of subsets of the input variables. If the effects of these sources of randomness are strong enough, it yields that the variance of the RF's generalization error is low. Also, each DT is grown to the largest extent possible, without pruning[2], keeping the generalization error rate low. Another property is that, using the Strong Law of Large Numbers, Breiman (2001) demonstrated that RF do not overfit, as more classifiers are added. In short, RF have the following characteristics (Breiman, 2001), (Hastie et al., 2008): 1) low error rates reported in multiple application domains (Svetnik et al., 2003), (Shotton et al., 2013), (Díaz-Uriarte and De Andres, 2006); 2) always converge (no overfiting); 3) fast to train/compute, comparing to other ensembles; 4) robust to outliers/noise; 5) simple and easily parallelized.

In the proposed framework, a variant of the CART algorithm Breiman et al. (1984) is employed, to train each DT. There are several methods proposed in the literature to find the best strategy to split a node on the tree, based on several metrics. Metrics based on impurity[3] of a node are widely used (e.g. Gini impurity, entropy impurity, etc). Based on this measure, a good heuristic to find the best local split at a node m is to find the split that maximizes the drop in impurity, defined as:

$$\Delta\beth_m = \beth_m - \sum_{j=1}^{b} p_m^j \beth_m^j, \tag{15}$$

where $\beth_m$ is the impurity value of the node m, p_m^j is the fraction of instances going for branch j, b is the number of branches the node is split into and $\beth_m^j$ is the impurity value of the descendant node corresponding to branch j. However, this is a *greedy local method*, not guaranteeing the best global solution, in terms of obtaining the smaller and simplest DT model (and consequently the smaller and simplest RF model). In this sense, the idea of searching for the best local split may not be advantageous, considering also the computational effort required, with implications on the time taken to train the model. This is the main motivation that lead to the introduction of an alternative search for the best split based on a meta-heuristic algorithm: the differential evolution algorithm. The idea is that the best splitting node condition may not be found, but a good one is, in a desirable amount of time/iterations. Another advantage is that it allows multivariate DT, with no significant increase of computational complexity nor time consumption. This means, each split may take any direction along the feature space, not constrained to parallel splits along the feature axes. Nevertheless, considering these properties, it is of most importance that the performance of the classifier (i.e. in terms of precision, recall, etc) must be at least equivalent to other RF's implementations (e.g. Breiman (2001), Hall et al. (2009)).

4.1. Differential Evolution - Seeking the Best Splitting Node Condition

Differential Evolution (DE) is a meta-heuristic algorithm proposed by Storn and Price (1997). It is a very efficient yet simple general optimization algorithm, which aims to solve non-linear, non-differentiable, non-continuous and real-parameters problems. It employs similar computational steps as standard evolutionary algorithms. In this sense, from a randomly generated *population* containing solutions to the given problem (i.e. in this context, denominated *individuals*), the main idea of DE is to iteratively select the best one through a set of rules. Besides this, it has few parameters to adjust, whose effects are very well studied and documented: mutation factor $F \in [0, 2]$; recombination factor $C \in [0, 1]$; number of individuals of the population N_{pop}.

Apart from the *initialization* step, which is performed only once in the beginning, in a simplistic formulation, it consists in a loop of three distinct steps: *mutation*, *recombination*, *selection*. In the context of the present work, the implementation of the algorithm is as follows. In the *initialization* step, a randomly population of size N_{pop} is generated:

$$X_t = \{x_{t,1}, x_{t,2}, ..., x_{t,N_{\text{pop}}}\}, \tag{16}$$

where $x_{t,i}$ is an individual (i.e. a possible solution to the problem)

$$x_{t,i} = \begin{pmatrix} x_{t,i,1} & x_{t,i,2} & \dots & x_{t,i,n} \end{pmatrix}, \tag{17}$$

[2] Procedure to make smaller/simpler trees, if they become to large.

[3] An impurity measure is the quantity of the goodness of a split. A split is pure if, after the split, all the instances reaching a node belong to the same class. In this case, the impurity of that node is 0.

with $x_{t,i,j}$ being the variable j of the problem, of the individual i, at iteration t (in the initial state $t = 0$). In the *mutation* step, each individual is mutated according to

$$v_{t,i} = x_{t,r_1} + F \cdot (x_{t,r_2} - x_{t,r_3}), \tag{18}$$

where $r_1, r_2, r_3 \in \{1, 2, \ldots, N_{\text{pop}}\}$ are mutually exclusive random indexes of individuals distinct from i. In the *recombination* step, a trial population U_t is generated, according to:

$$u_{t,i,j} = \begin{cases} v_{t,i,j} & \text{if rand} < C \vee j = \delta \\ x_{t,i,j} & \text{otherwise} \end{cases}, \tag{19}$$

where rand $\in [0, 1]$ is a random value and $\delta \in \{1, \ldots, n\}$ is a random index for recombination to ensure that $u_{t,i} \neq x_{t,i}$. In the *selection* step, the trial individuals are tested with the individuals of the population with the same index:

$$x_{t+1,i} = \begin{cases} u_{t,i} & \text{if } f(u_{t,i}) \leq f(x_{t,i}) \\ x_{t,i} & \text{otherwise} \end{cases}, \tag{20}$$

where $f(\cdot)$ is the objective function. If the trial individual yields a lower or equal value of the objective function, it replaces the corresponding target individual; otherwise the individual remains in the population.

In the context of this work, the objective function is to find the best linear coefficients a_j, $j \in \{1, \ldots, n\}$, such that the splitting condition

$$a_1 y_1 + a_2 y_2 + \cdots + a_n y_n < 1, \tag{21}$$

where y_j corresponds to the variable with index j of the randomly selected subset from the training set, minimizes the symmetrical of the drop in impurity, given by Eq. (15). Note that n is the number of randomly selected variables to split each node and is adjustable. In other words, the objective function to minimize may be formulated in the following way, considering $a = \begin{pmatrix} a_1 & a_2 & \ldots & a_n \end{pmatrix}$:

$$-\Delta\Im_m(a) = -\Im_m + \sum_{j=1}^{b} p_m^j(a)\Im_m^j(a). \tag{22}$$

Naturally, the drop in impurity depends on the splitting condition, given by Eq. (21). The splitting condition depends on its linear coefficients a_j. Therefore, the drop in impurity depends on the linear coefficients a_j. However, their relation is not analytically easy to find, since the relation between the splitting condition, given by Eq. (21), and how the instance space is divided is not clear and may depend on the training set. This means, no assumption about their mathematical relation (e.g. linear, quadratic, etc) can be made. This is another reason for the inclusion of the DE algorithm, since the objective function may not be linear, continuous nor differentiable.

4.2. Implementation - Differential Evolution Random Forests

One of the main reasons that inspired the proposed RF implementation was that, if possible, it should not depend on tunable thresholds. In this context, thresholds are assumed to be variables considered on an algorithm, whose effects are not well behaved and/or may depend on other variables and/or input/output data. Nevertheless, another interesting characteristic arises mainly from the fact that the effects of the adjustable parameters of the DE are very well studied and documented, since the effects of these parameters in the overall performance of the classifier are very well defined. Some of these effects are shown and discussed subsequently in Sect. 5.2.

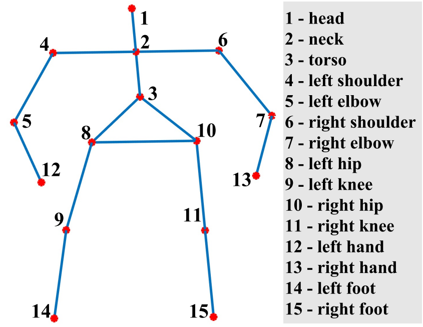


Fig. 4: Base skeleton and respective joints.

Another interesting approach of the proposed RF implementation concerns the stopping criterion of the iterative process to split a node, which is based on the *hypothesis testing* (Duda et al., 2012). The objective is to determine if a given splitting node condition differs from a random one, based on statistical significance (i.e. desirable confidence level). Considering the chi-squared statistics, this deviation may be quantified, for two branches, as:

$$\chi_m^2 = \sum_{i=1}^{K} \frac{\left(N_{m,i}^L - N_{m,i} p_m^L\right)^2}{N_{m,i} p_m^L}, \tag{23}$$

where $N_{m,i}^L$ is the number of instances of class C_i sent to the left branch, $N_{m,i} p_m^L$ is the expected number of instances of class C_i sent to the left branch by the random rule and K is the number of classes considered. When χ_m^2 is greater than some value given by the desired confidence level associated to the chi-squared statistics, it means that the candidate split differs from the random one and the null hypothesis is rejected, continuing the node splitting. On the other hand, if χ_m^2 is smaller, the splitting is stopped, since the candidate split does not differ significantly from a random one.

5. Experimental Results

In order to assess the proposed method, the Cornell Activity Dataset (CAD-60) (Sung et al., 2011) was used. Two different implementations of the RF classifier were used: RF provided by Weka Version 3-6-13 software (Hall et al., 2009) (denominated as WRF); and the proposed RF implementation, using DE algorithm as splitting node methodology (denominated as DERF). The experimental results were obtained in a 2.60 GHz Intel Core i5 CPU machine.

Figure 4 exemplifies the skeleton's data provided, as well as the indexes considered for each joint. The angles considered are defined in Table 1 and the normal to the triangles formed by groups of three joints are described in Table 2. These features

Table 1: Considered Angles

Angle	Joints Triplet	Angle	Joints Triplet
i	(a,b,c)	i	(a,b,c)
1	(6,7,13)	2	(4,5,12)
3	(6,10,11)	4	(4,8,9)
5	(10,11,15)	6	(8,9,14)
7	(6,10,13)	8	(4,8,12)
9	(1,7,13)	10	(1,5,12)
11	(3,12,13)	12	(3,14,15)

Table 2: Considered Normal to Triangles Formed by Three Joints

Normal	Joints Triplet	Normal	Joints Triplet
k	(a,b,c)	k	(a,b,c)
1	(4,5,12)	2	(6,7,13)
3	(8,9,14)	4	(10,11,15)
5	(1,12,13)	6	(3,12,13)
7	(5,8,12)	8	(7,10,13)

Table 3: Performance of the Proposed Approach on the CAD-60

Location	Activity	WRF		DERF	
		Prec (%)	Rec (%)	Prec (%)	Rec (%)
Bathroom	random+still	95.87	96.60	95.98	99.35
	rinsing water	81.57	68.32	93.65	65.40
	brushing teeth	96.05	92.97	98.55	96.14
	wearing lens	84.05	85.35	84.37	95.14
	average	89.38	85.81	93.14	89.01
Bedroom	random+still	97.87	99.62	97.28	99.62
	talking on phone	56.05	66.50	57.58	76.64
	drinking water	57.15	36.85	55.58	29.08
	opening container	100	94.35	99.24	94.35
	average	77.77	74.33	77.42	74.92
Kitchen	random+still	93.00	98.47	91.70	98.46
	drinking water	99.00	95.82	90.67	94.10
	chopping	83.77	92.50	85.03	90.98
	stirring	73.07	64.80	71.84	59.86
	opening container	100	86.32	100	88.10
	average	89.77	87.58	87.85	86.30
Living room	random+still	96.70	99.62	97.63	100
	talking on phone	59.82	75.07	56.96	75.03
	drinking water	58.15	33.42	56.92	28.39
	talking on couch	81.25	85.72	82.43	95.10
	relaxing on couch	75.00	62.50	75.00	68.92
	average	74.18	71.27	73.79	73.49
Office	random+still	94.60	96.90	90.56	98.96
	talking on phone	49.72	71.02	55.49	74.47
	writing on board	92.17	90.87	87.81	84.17
	drinking water	51.32	21.55	51.02	24.48
	working on computer	100	100	100	100
	average	77.56	76.07	76.98	76.42
	Overall Average	**81.73**	**79.01**	**81.83**	**80.02**

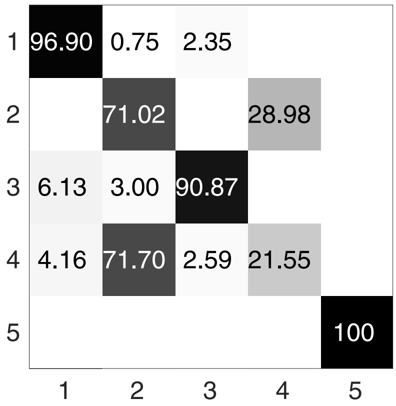
aim to provide a good discrimination between activities, since they provided better overall results in several preliminary experiments. Different combination of features were evaluated, but due to space limitations they are not presented here. In the particular case of the angles considered, the work of Faria et al. (2015) was very influential, due to the supporting rationale and the results obtained. The performance indicators in terms of precision (Prec) and recall (Rec) are presented for each scenario, adopting the same strategy described in Sung et al. (2012). A leave-one-out cross validation procedure was employed. This procedure is important to check the classifier's generalization capability.

5.1. Cornell Activity Dataset

The CAD-60 consists of 3D skeleton's coordinates joints, acquired by a RGB-D sensor at a frame rate of 30 Hz. The dataset contains 12 human distinct activities plus 1 random action and 1 still posture, categorized into 5 environments (bathroom, bedroom, kitchen, living room and office), performed by 4 different subjects. The experimental results obtained are presented in Table 3. Both WRF classifier, with overall average for precision and recall of 81.73% and 79.01%, respectively, and DERF classifier, with overall average for precision and recall of 81.83% and 80.02%, respectively, were used. The following features extraction parameters were implemented for both tests: $m_{\text{extd}} = 3$ (the corresponding velocities of *head*, *neck* and *torso* are not considered); $n_\theta = 12$ (the considered angles are presented in Table 1); $n_\omega = 12$ (the considered angular velocities are obtained based on their respective angles); $n_\Delta = 8$ (the considered normals to triangles formed by triplets of joints are shown in Table 2); $n_{\max} = n_{\min} = 1$ (only the most extreme values for each feature on the respective analysis window are considered); $E_{\min} = 0.0028$ (this value was tuned empirically, based on experimental tests on the training data); $E_u^a = 2 \times \text{mean}(E^a)$ (mean$(\cdot)$ is the mean function and $E^a = \left\{(E^1)^a, (E^2)^a, \ldots, (E^t)^a, \ldots\right\}$ is the set of kinetic energy values of the activity a, for all its corresponding frames; this value was also tuned empirically, based on experimental tests on the training data). For the DERF classifier, the implemented values of the adjustable parameters are as follows: F takes a different random value in $[0.5, 1]$ for each individual (e.g. *dither*); $C = 0.9$; $N_{\text{pop}} = 20$; $N_{\text{iter}} = 6$ (number of iterations of the DE algorithm, i.e., stopping criterion); $N_{\text{tree}} = 100$ (number of trees in the ensemble); $N_{\text{rand}} = \lfloor \log_2(N_{\text{var}}) + 1 \rfloor$ (number of randomly selected variables to split a node from the N_{var} of the training set); $\text{conf}_{\text{lvl}} = 99\%$ (confidence level of the stopping criterion to split a node). For the WRF classifier, the default parameters were used: ensemble of 100 trees; $\lfloor \log_2(N_{\text{var}}) + 1 \rfloor$ (number of randomly selected variables to split a node from the N_{var} of the training set).

Based on these parameters, the classifiers were trained with an average number of 630.30 training samples, each one with 264 features, requiring an average training time of 1 sec. In terms of precision and recall, they have similar performances.

In Fig. 5, a confusion matrix is presented, with respect to the WRF classifier performance, classifying the activities related to the office environment of the CAD-60. As shown, there is a lot of misclassification between the *talking on phone* and *drinking water* activities. This is due to the fact that these activities are very similar, regarding the skeleton-based features considered. Also, it is a recurrent evidence, concerning the obtained results for other environments. Nevertheless, this approach can discriminate between not similar activities very effectively (e.g. in the kitchen environment, since the *talking on phone* and *drinking water* activities are not simultaneously present, the proposed framework identifies very well the *drinking water* activity, with performance indicators above 90%, for both considered classifiers). Considering it requires few training examples and training time, using static and just max-min features, the



	1	2	3	4	5
1	96.90	0.75	2.35		
2		71.02		28.98	
3	6.13	3.00	90.87		
4	4.16	71.70	2.59	21.55	
5					100

Fig. 5: Confusion matrix of the proposed approach for the office environment of the CAD-60, using the WRF (1-*random+still*; 2-*talking on phone*; 3-*writing on board*; 4-*drinking water*; 5-*working on computer*). As shown, *drinking water* and *talking on phone* activities are not very well discriminated between themselves.

Table 4: Comparison of the Proposed Approach with Other Methods

Method	Prec (%)	Rec (%)
Faria et al. (2015)	94.8	94.7
Cippitelli et al. (2016)	93.9	93.5
Shan and Akella (2014)	93.8	94.5
Zhu et al. (2014)	93.2	84.6
Parisi et al. (2015)	91.9	90.2
Zhang and Tian (2012)	86.0	84.0
Proposed approach, using DERF	**81.83**	**80.02**
Koppula et al. (2013)	80.8	71.4
Gupta et al. (2013)	78.1	75.4
Gaglio et al. (2015)	77.3	76.7
Ni et al. (2013)	75.9	69.5
Yang and Tian (2014)	71.9	66.6
Piyathilaka and Kodagoda (2013)	70.0	78.0
Sung et al. (2011)	67.9	55.5

assumption that each activity may be discriminated just by considering extreme movements and poses is corroborated.

In Table 4, a comparison of the overall results of the proposed approach, in relation to other state-of-the-art methods, is presented. The comparison that was made is only based on precision and recall indicators, since no performance indicators such as training examples, number of features or training time are provided by the other considered methods. In this sense, the approach that is proposed shows relevant overall performance, particularly considering that the number of training examples and required training time are very small.

5.2. *Influences of Some Parameters of DERF in the Overall Performance*

A claim was made that the proposed RF classifier (DERF) has no *thresholds* to tune, but has some *parameters* that can be adjusted, which were mentioned previously. This means, the influence of each and every parameter considered is known and controllable, not depending on other parameters or input/output data. Due to space constraints, the influence of just two parameters will be discussed, in terms of overall precision and recall, training/testing time and size of the classifier's model obtained: number of individuals of the DE algorithm population N_{pop}; confidence level of the stopping criterion to split a node conf_{lvl}. Besides CAD-60, the influence of the mentioned parameters was tested on other datasets from the UCI repository (Lichman, 2013). The results obtained on two additional datasets (Iris and Wine) will also be shown. Although these datasets are not very challenging, they are useful for testing classifiers with newly implemented ideas. Also, the experimental results obtained, based on these three datasets corroborate the claim that was made about DERF not having thresholds to tune, since the same tendency of each parameter's influence is observable. The default experimental parameters are as follows: $F = 0.8$; $C = 0.6$; $N_{\text{pop}} = 15$; $N_{\text{iter}} = 6$; $N_{\text{tree}} = 100$; $N_{\text{rand}} = \lfloor \log_2(N_{\text{var}}) + 1 \rfloor = 9$; $\text{conf}_{\text{lvl}} = 90\%$.

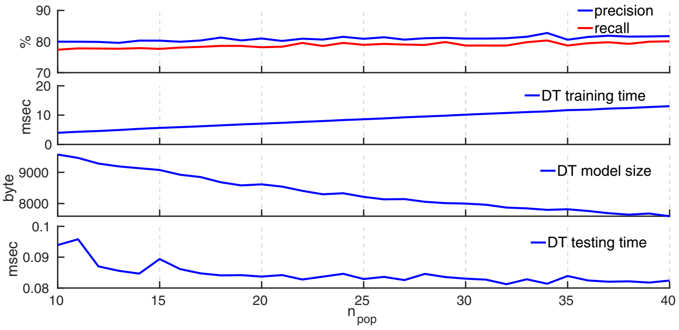


Fig. 6: Influence of the number of individuals of the DE algorithm population (CAD-60).

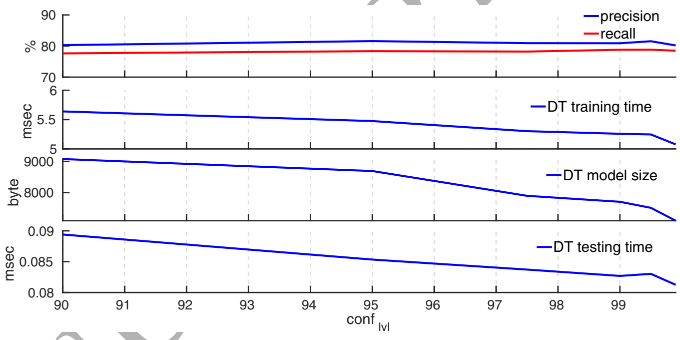
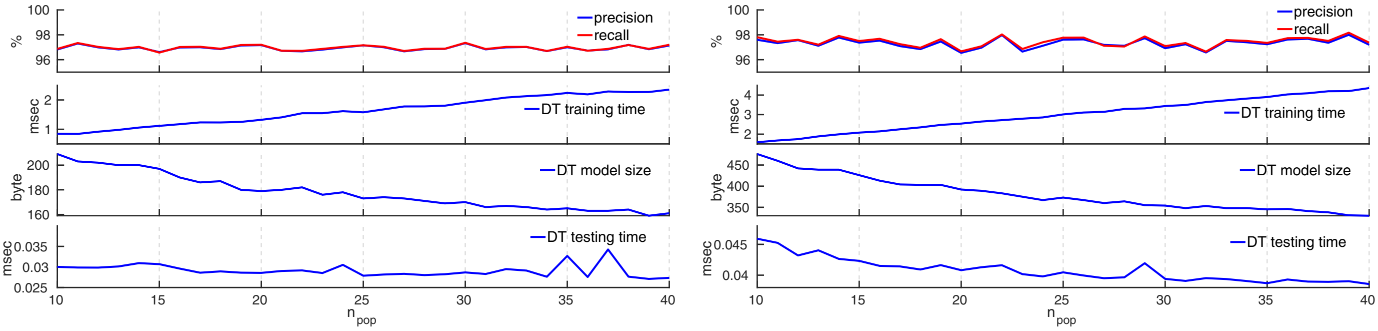


Fig. 7: Influence of the confidence level of the stopping criterion to split a node (CAD-60).

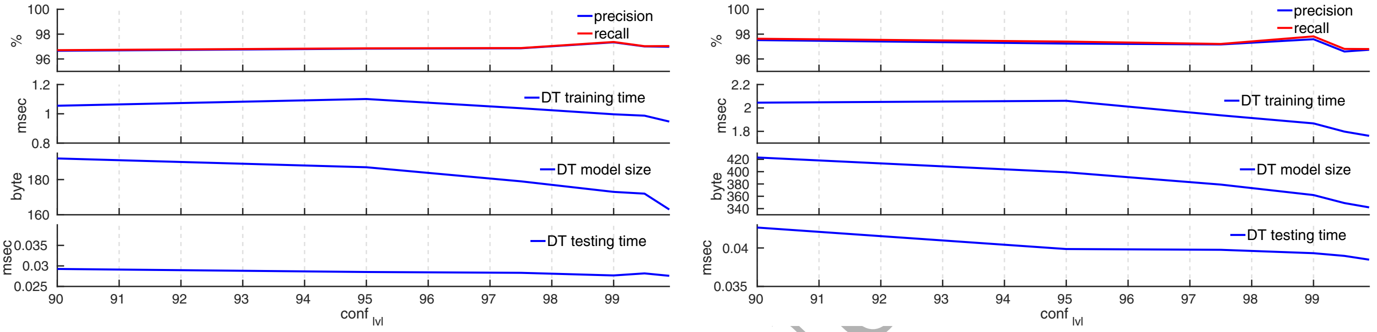
Figure 6 shows the influence of the number of individuals of the DE population, when searching for the best splitting condition at each node, for CAD-60. First, in terms of precision and recall, the classifier's performance do not differ significantly. Second, the training time, increases as the number of individuals increases. Third, the size of the resulting model decreases, as the number of individuals increases, and consequently, since the model becomes simpler and more compact, the time required for the classifier to make a decision decreases. The exact analysis can be made, based on the results shown in Fig. 8, on Iris and Wine datasets. In this sense, when designing this parameter, there must be a compromise between training time and model size. In Fig. 7, the effect of the confidence level, based on which each node is considered or not a leaf, is shown, for



(a) Iris Dataset.

(b) Wine Dataset.

Fig. 8: Influence of the number of individuals of the DE algorithm population (Iris and Wine Datasets).



(a) Iris Dataset.

(b) Wine Dataset.

Fig. 9: Influence of the confidence level of the stopping criterion to split a node (Iris and Wine Datasets).

CAD-60. The overall performance, in terms of precision and recall, is not significantly influenced, nor the training time required. However, the model size is greatly reduced, as the confidence level is increased and consequently the time required to make a decision. In other words, this parameter does not impose a compromise. Therefore, higher values for the confidence level are better, since the overall model complexity is reduced, not affecting the overall performance of the classifier, in terms of the rest of considered performance indicators. A similar conclusion is corroborated, based on the experimental results illustrated in Fig. 9.

6. Conclusion and Future Work

The work presented may be separated into two core components: the skeleton-based features extraction approach proposed, which is based on compressing a sequence of consecutive temporal frames into training samples segmented by automatically identified key poses, comprising only static and max-min dynamic features, and the inclusion of the DE algorithm within the RF classifier, which was built and implemented from scratch. The main objective of this work was to develop a very fast training framework, requiring just a few training examples, with relevant performance, comparing to other state-of-the-art methods. Nonetheless, a major limitation of the proposed method is recognized to be the determination of key poses, which is done considering all body joints. Such scheme enables the possible misidentification of key poses, since each and every body joint is considered to equally important, during the process of human activity recognition. In order to overcome this, one idea may be segmenting the human body into parts, consisting of subsets of joints (e.g. limbs). A similar approach is explored by Zhang et al. (2016). Also, a more careful analysis could be made, performing a pre-feature extraction step (e.g. PCA), in order to find better sets of discriminative static and dynamic features. Based on the developed work, new research directions are summarized and highlighted:

1. Divide the human skeleton into several parts, differentiating parts with more/less relevant information;
2. Train specialized classifiers for each part, having a higher-level classifier discriminating between activities;
3. Consider key poses as activities transitions;
4. Test other sets of static and dynamic features;
5. Implementation and validation of the method in real-life scenarios (e.g. falling) (Parisi and Wermter, 2016).

References

Baysal, S., Kurt, M.C., Duygulu, P., 2010. Recognizing human actions using key poses, in: Pattern Recognition (ICPR), 2010 20th International Conference on, IEEE. pp. 1727–1730.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A., 1984. Classification and Regression Trees. CRC press.

Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V., 2006. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE transactions on evolutionary computation 10, 646–657.

Chapelle, O., Keerthi, S.S., 2008. Multi-class feature selection with support vector machines, in: Proceedings of the American statistical association.

Chauhan, N., Ravi, V., Chandra, D.K., 2009. Differential evolution trained wavelet neural networks: Application to bankruptcy prediction in banks. Expert Systems with Applications 36, 7659–7665.

Cippitelli, E., Gasparrini, S., Gambi, E., Spinsante, S., 2016. A human activity recognition system using skeleton data from rgbd sensors. Computational Intelligence and Neuroscience 2016.

Das, S., Abraham, A., Konar, A., 2008. Automatic clustering using an improved differential evolution algorithm. IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans 38, 218–237.

Das, S., Suganthan, P.N., 2011. Differential evolution: a survey of the state-of-the-art. IEEE transactions on evolutionary computation 15, 4–31.

Demirdjian, D., Varri, C., 2009. Recognizing events with temporal random forests, in: Proceedings of the 2009 international conference on Multimodal interfaces, ACM. pp. 293–296.

Díaz-Uriarte, R., De Andres, S.A., 2006. Gene selection and classification of microarray data using random forest. BMC bioinformatics 7, 1.

Duda, R.O., Hart, P.E., Stork, D.G., 2012. Pattern Classification. John Wiley & Sons.

Faria, D.R., Vieira, M., Premebida, C., Nunes, U., 2015. Probabilistic human daily activity recognition towards robot-assisted living, in: Robot and Human Interactive Communication (RO-MAN), 2015 24th IEEE International Symposium on, IEEE. pp. 582–587.

Forman, G., Scholz, M., Rajaram, S., 2009. Feature shaping for linear svm classifiers, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM. pp. 299–308.

Gaglio, S., Re, G.L., Morana, M., 2015. Human activity recognition process using 3-d posture data. Human-Machine Systems, IEEE Transactions on 45, 586–597.

Gan, L., Chen, F., 2013. Human action recognition using apj3d and random forests. Journal of Software 8, 2238–2245.

Gupta, R., Chia, A.Y.S., Rajan, D., 2013. Human activities recognition using depth images, in: Proceedings of the 21st ACM international conference on Multimedia, ACM. pp. 283–292.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The weka data mining software: an update. ACM SIGKDD Explorations Newsletter 11, 10–18.

Hastie, T., Tibshirani, R., Friedman, J., 2008. The Elements of Statistical Learning. volume 2. Springer series in statistics Springer, Berlin.

Johansson, G., 1973. Visual perception of biological motion and a model for its analysis. Perception & Psychophysics 14, 201–211.

Jun, B., Choi, I., Kim, D., 2013. Local transform features and hybridization for accurate face and human detection. IEEE transactions on pattern analysis and machine intelligence 35, 1423–1436.

Koppula, H.S., Gupta, R., Saxena, A., 2013. Learning human activities and object affordances from rgb-d videos. The International Journal of Robotics Research 32, 951–970.

Lichman, M., 2013. UCI machine learning repository. Http://archive.ics.uci.edu/ml.

Liu, J., Lampinen, J., 2005. A fuzzy adaptive differential evolution algorithm. Soft Computing 9, 448–462.

Louppe, G., 2014. Understanding random forests: From theory to practice. arXiv preprint arXiv:1407.7502 .

Menon, P.P., Kim, J., Bates, D.G., Postlethwaite, I., 2006. Clearance of nonlinear flight control laws using hybrid evolutionary optimization. IEEE transactions on evolutionary computation 10, 689–699.

Miranda, L., Vieira, T., Martinez, D., Lewiner, T., Vieira, A.W., Campos, M.F., 2012. Real-time gesture recognition from depth data through key poses learning and decision forests, in: 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images, IEEE. pp. 268–275.

Moloi, N., Ali, M., 2005. An iterative global optimization algorithm for potential energy minimization. Computational Optimization and Applications 30, 119–132.

Neri, F., Mininno, E., 2010. Memetic compact differential evolution for cartesian robot control. IEEE Computational Intelligence Magazine 5, 54–65.

Ni, B., Pei, Y., Moulin, P., Yan, S., 2013. Multilevel depth and image fusion for human activity detection. IEEE Transactions on Cybernetics 43, 1383–1394.

Nunes, U.M., Faria, D.R., Peixoto, P., 2016. Human activity recognition using max-min skeleton-based features and key poses, in: Workshop on Behavior Adaptation, Interaction and Learning for Assistive Robotics (BAILAR) on Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on.

Okada, K., Ogura, T., Haneda, A., Fujimoto, J., Gravot, F., Inaba, M., 2005. Humanoid motion generation system on hrp2-jsk for daily life environment, in: IEEE International Conference Mechatronics and Automation, 2005, IEEE. pp. 1772–1777.

Parisi, G.I., Weber, C., Wermter, S., 2015. Self-organizing neural integration of pose-motion features for human action recognition. Frontiers in Neurorobotics 9.

Parisi, G.I., Wermter, S., 2016. A neurocognitive robot assistant for robust event detection, in: Trends in Ambient Intelligent Systems. Springer, pp. 1–27.

Piyathilaka, L., Kodagoda, S., 2013. Gaussian mixture based hmm for human daily activity recognition using 3d skeleton features, in: 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA), IEEE. pp. 567–572.

Robnik-Šikonja, M., 2004. Improving random forests, in: European Conference on Machine Learning, Springer. pp. 359–370.

Ronkkonen, J., Kukkonen, S., Price, K.V., 2005. Real-parameter optimization with differential evolution, in: Proc. IEEE CEC, pp. 506–513.

Segal, M.R., 2004. Machine learning benchmarks and random forest regression. Center for Bioinformatics & Molecular Biostatistics .

Shan, J., Akella, S., 2014. 3d human action segmentation and recognition using pose kinetic energy, in: Advanced Robotics and its Social Impacts (ARSO), 2014 IEEE Workshop on, IEEE. pp. 69–75.

Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R., 2013. Real-time human pose recognition in parts from single depth images. Communications of the ACM 56, 116–124.

Song, Y., Demirdjian, D., Davis, R., 2012. Continuous body and hand gesture recognition for natural human-computer interaction. ACM Transactions on Interactive Intelligent Systems (TiiS) 2, 5.

Storn, R., 2005. Designing nonstandard filters with differential evolution. IEEE Signal Processing Magazine 22, 103–106.

Storn, R., Price, K., 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359.

Subudhi, B., Jena, D., 2008. Differential evolution and levenberg marquardt trained neural network scheme for nonlinear system identification. Neural Processing Letters 27, 285–296.

Sung, J., Ponce, C., Selman, B., Saxena, A., 2011. Human activity detection from rgbd images. Plan, Activity, and Intent Recognition 64.

Sung, J., Ponce, C., Selman, B., Saxena, A., 2012. Unstructured human activity detection from rgbd images, in: Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE. pp. 842–849.

Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P., Feuston, B.P., 2003. Random forest: a classification and regression tool for compound classification and qsar modeling. Journal of chemical information and computer sciences 43, 1947–1958.

Tsymbal, A., Pechenizkiy, M., Cunningham, P., 2006. Dynamic integration with random forests, in: European Conference on Machine Learning, Springer. pp. 801–808.

Vieira, A.W., Nascimento, E.R., Oliveira, G.L., Liu, Z., Campos, M.F., 2012. Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences, in: Iberoamerican Congress on Pattern Recognition, Springer. pp. 252–259.

Wang, J., Liu, Z., Wu, Y., Yuan, J., 2014. Learning actionlet ensemble for 3d human action recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on 36, 914–927.

Yang, X., Tian, Y., 2014. Effective 3d action recognition using eigenjoints. Journal of Visual Communication and Image Representation 25, 2–11.

Zhang, C., Tian, Y., 2012. Rgb-d camera-based daily living activity recognition. Journal of Computer Vision and Image Processing 2, 12.

Zhang, L., Yang, W., Zhu, G., Shen, P., Song, J., 2016. Human activity recognition based on weighted limb features, in: Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on, IEEE. pp. 4404–4409.

Zhu, G., Zhang, L., Shen, P., Song, J., 2016. Human action recognition using multi-layer codebooks of key poses and atomic motions. Signal Processing: Image Communication .

Zhu, Y., Chen, W., Guo, G., 2014. Evaluating spatiotemporal interest point features for depth-based action recognition. Image and Vision Computing 32, 453–464.