Hindawi
Journal of Sensors
Volume 2019, Article ID 7060491, 12 pages
https://doi.org/10.1155/2019/7060491

Research Article

The Influence of Speed and Position in Dynamic Gesture Recognition for Human-Robot Interaction

José Carlos Castillo, Fernando Alonso-Martín, David Cáceres-Domínguez, María Malfaz, and Miguel A. Salichs

Departamento de Sistemas y Automática, Universidad Carlos III de Madrid, 28911 Madrid, Spain

Correspondence should be addressed to José Carlos Castillo; jocastil@ing.uc3m.es

Received 15 April 2018; Revised 31 October 2018; Accepted 7 November 2018; Published 12 February 2019

Academic Editor: Abdellah Touhafi

Human communication relies on several aspects beyond the speech. One of them is gestures as they express intentions, interests, feelings, or ideas and complement the speech. Social robots need to interpret these messages to allow a more natural Human-Robot Interaction. In this sense, our aim is to study the effect of position and speed features in dynamic gesture recognition. We use 3D information to extract the user's skeleton and calculate the normalized position for all of its joints, and using the temporal variation of such positions, we calculate their speeds. Our three datasets are composed of 1355 samples from 30 users. We consider 14 common gestures in HRI involving upper body movements. A set of classification techniques is evaluated testing these three datasets to find what features perform better. Results indicate that the union of both speed and position achieves the best results among the three possibilities, 0.999 of *F*-score. The combination that performs better to detect dynamic gestures in real time is finally integrated in our social robot with a simple HRI application to run a proof of concept test to check how the proposal behaves in a realistic scenario.

1. Introduction

Body gestures are a continuous (and frequently unconscious) source of information that humans use to provide insights about intentions, interests, feelings, and ideas [1]. These gestures can be defined as actions produced with the intent to communicate and are typically expressed using body motion or facial features [2]. This constitutes a form of interaction in which body movements and actions can provide information on their own, regardless of verbal information. Moreover, interaction between humans depends on those gestures produced by the speakers since expressions pay an essential role in the communicative process, complementing it in different ways as gestures are able to (i) reflect speakers' thoughts, even unspoken ones; (ii) gestures have the ability of changing the speaker's thoughts; and finally, (iii) gestures provide building blocks that can be used to construct a language [3]. These actions are inherent to humans regardless of culture and age of the individuals. In fact, children gesture even before starting to speak and continue using gestures as they grow up.

These ideas motivated that developments in gesture recognition for Human-Robot Interaction (HRI) gain importance in recent years as robots need to interpret human gestures in order to accomplish natural interaction. Initial research focused on hand gestures, sign language, and command gesture recognition while more recent approaches are tackling with the problem of full-body static gesture analysis [4, 5].

The main contribution of this work is to explore this latest trend, proposing a new gesture detection approach that extends the current ones by exploiting the evolution of the human body joints along their trajectories. That is, we consider dynamic gestures instead of just analysing static body poses. More specifically, we explore a set of upper body common gestures in HRI, although the proposal could be easily extended to full-body dynamic gestures. Our goal is therefore to compare the performance of using just static features

(joint positions along the time) against just dynamic ones (joint speeds) and finally a combination of both, using machine learning to assess which approach recognises human gestures with higher accuracy. We tested these combinations of features with cross-validation and then took the ones with the best performance and tested them with untrained data to get more realistic results.

This proposal requires first an offline analysis to assess the best-performing machine learning method over an initial set of training data. The selected technique is next integrated in an online system that runs in a social robot where we developed a simple HRI application as a proof of concept to study the feasibility of dynamic gesture recognition. We are also aware that currently there are other works proposing gesture recognition for Human-Computer Interaction (a recent survey can be found in [6]) but few dealing with the challenges posed by dynamic gesture recognition.

The rest of the manuscript is structured as follows: Section 2 reviews some relevant approaches for gesture recognition and classifies those techniques depending on what representation of the human body they use. Next, Section 3 introduces the proposed approach for dynamic gesture recognition, along with the metrics and the integration in the social robot. Section 4 describes the robotic platform used in this work and introduces the set of gestures to be recognised as well as the data collection procedure. After that, Section 5 analyses the results obtained from the tests. Section 6 presents the main limitations of our approach and some possible ways for overcoming them, and finally, Section 7 extracts the main conclusions from this work.

2. Related Work

2.1. Gesture Recognition Approaches. There are several approaches dealing with the challenges of human activity and pose recognition. For instance, the work of Castillo et al. [7] intended to understand the dynamics of the actions performed by people when performing different tasks using several sensory sources such as cameras, GPS, and accelerometers. Fernández-Caballero [8] presented a state machine-based technique incorporating domain knowledge where motion-based image features are linked to a symbolic notion of hierarchical activity. Successful research focuses on recognising rather simple human activities or patterns [9] even proposing frameworks for monitoring and activity interpretation [10]. These patterns are interesting and related to our proposal in the sense that they involve detections along the time to encode a full activity.

Pose recognition is intended to recover the pose of a body constituted by still joins or rigid body parts using sensory observations. Several works approached this task obtaining good results in terms of accuracy and low computational requirements. As an example, the work of Shotton et al. [11] offered a pose recognition approach using 3D information. This system relied on object recognition and proposed randomized decision forests to recognise body parts. Then, a classifier recognised these parts, so that at run time a single input depth image was segmented into a dense probabilistic body part labelling. Jalal et al. [12] also proposed a real-time tracking system for human pose recognition utilizing ridge body parts' features from depth information. Recent works propose deep learning to tackle with this matter [13, 14] where approaches such as convolutional neural networks or stacked auto encoders demonstrated good performance.

In contrast, there are few works that deal with the idea of dynamic gesture recognition. Wu et al. [15] presented a work in this line that proposes dynamic time warping applied to Kinect's skeletal data for user access control, studying the differences in users' gestures to identify them. The work of Morency et al. [16] proposed a Latent-Dynamic Conditional Random Field for dynamic gesture recognition to discover the latent structure that best differentiates motion patterns. This approach was applied to head and eye gestures while interacting with a robot. Another work applied gesture recognition during HRI for service robotics (interactive clean-up tasks) using neural networks and a template-based approach. Both techniques were combined with the Viterbi algorithm for arm motion gesture recognition [17]. Santos et al. [18] presented a system for dynamic hand gesture recognition based on depth maps and a hybrid classifier that integrates dynamic time warping and Hidden Markov Models (HMMs). Milazzo et al. [19] proposed a modular middleware to ease the development of gesture-based applications focused on Human-Computer Interaction. Authors faced this problem from a software perspective and reported limitations regarding bandwidth, number of instructions for recognition, or the CPU load.

2.2. Techniques for Gesture Recognition. There is an important aspect when recognising gestures from RGB-D information, which is the representation of the human body. Traditionally, there are two categories starting with *body part-based methods*, which consider the human body as a set of connected segments, ranging from few parts [20], to more complex ones with several segments [21, 22]. Usually, body movements are detected by looking at the horizontal and vertical translations of the parts or in plane rotations where gestures can be represented as combinations of sequences of movements. As an example, Jalal et al. [21] proposed a twenty-three-part division of the body and a Hidden Markov Model (HMM) to recognise six human activities with a classification rate of 97%. Other body part-based methods use joint angles [23], measuring the geometry between connected pairs of body parts that allows modelling linear dynamical systems. Here, the evolution of the angles is calculated using dynamic time warping, an algorithm for measuring similarity of two temporal sequences. Oh et al. [24] presented an approach for upper body gesture recognition based on key poses and Markov chain models, which represents the relationship between gesture states and pose events.

The second category corresponds to *joint-based methods*, which consider the skeleton as a set of individual points. Celebi et al. [25] proposed a representation of the body using twenty 3D points and HMM for classification. Gu et al. [26] followed a similar approach, using 15 skeleton joints provided by a Kinect and HMM for gesture modelling and

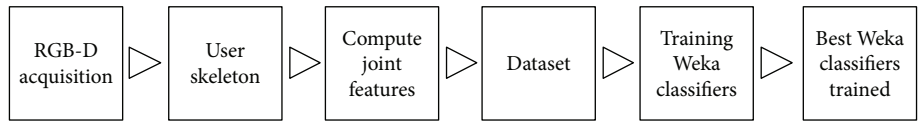

Figure 1: Offline pipeline: assessing the best classifier.

classification. Recently, Ofli et al. [27] presented a system that used few informative skeletal joints automatically selected based on measures such as the maximum velocity of the joints or the variance or mean of the joint angles. Next, a Support Vector Machine (SVM) classified the gestures. Zhu et al. [28] also proposed SVMs in order to analyse the position and relative displacement of human skeleton joints. Mangera et al. [29] presented an approach that selected key-frames in a gesture sequence and then a cascading neural network determined whether a gesture was performed by the left or right side. With this information, a second neural network classified the gesture.

It is also important to decide the features that could be relevant on the machine learning process. Not all features collected and included in a dataset provide useful data for the training process. In this regard, Fong et al. [30] proposed an approach for gesture recognition from 3D information coupled with a series of data mining techniques, particularly 14 classifiers. Authors studied of how these techniques performed with and without feature selection. The feature selection method applied was particle swarm optimization.

3. Our Approach for Dynamic Gesture Recognition

This proposal builds on 3D information extracted from the user skeletal joints since this kind of representation eases the extraction of information related to positions and speeds of the joints. This information eases the application of machine learning techniques, in our case for dynamic gesture recognition. Next, we intend to assess the effect of using position and speed information and, after training several classification techniques, to analyse which one provides the best performance with these input data. Our first step is to conduct the first part of the study in an offline fashion, and after analysing the results, the best combination will be integrated in a social robot to implement dynamic gesture recognition in real time. Figure 1 shows the pipeline for assessing the classifier that works best with the features extracted from dynamic gestures.

3.1. Extracting Features from 3D Joints. Our proposal uses information acquired from a Kinect device and extracts the user skeleton with the software provided by PrimeSense NiTE (NiTe website: http://openni.ru/files/nite/index.html) that discretises the human body into a set, J, of 15 joints with their 3D positions in the space with respect to the camera origin of coordinates. We wanted to make this proposal as realistic as possible in terms of user-robot dynamics. Therefore, and since users can move freely in front of the robot, the acquired information needs to be homogenized regardless of users' position and orientation with respect to the sensor to enhance the classification results. The torso joint, J^t, is used as a reference, and all other joints are normalized as shown in equation 1. Since the gestures we propose in this work are all performed standing, we do not consider here rotations of the whole body.

$$\bar{j} = \left(j_x - J_x^t, j_y - J_y^t, j_z - J_z^t\right), \quad \forall j \in J. \tag{1}$$

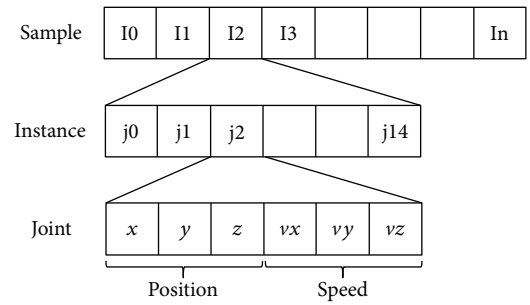

Figure 2: Dataset containing both position and speed information. Each joint is characterized by its position and speed. A set of 15 joints forms an instance, that is, the whole skeleton information at an instant of time. Instances along the time encode a whole gesture or sample that is fed to the classifiers.

Since the acquisition rate of the Kinect is fixed (10 frames per second), we can also calculate, in addition to the position of each joint, its speed ($\bar{j}^s$) as the difference in joint positions detected between two consecutive frames acquired with a difference of $t_1 - t_0$ seconds:

$$\bar{j}^s = \left(\frac{\bar{j}_x^{t_1} - \bar{j}_x^{t_0}}{t_1 - t_0}, \frac{\bar{j}_y^{t_1} - \bar{j}_y^{t_0}}{t_1 - t_0}, \frac{\bar{j}_z^{t_1} - \bar{j}_z^{t_0}}{t_1 - t_0}\right), \quad \forall \bar{j} \in \bar{J}. \tag{2}$$

As a result, each joint is encoded through six features, three for its last detected position (x, y, z) and three for the speeds in 3D (vx, vy, vz). The information from the 15 joints corresponding to a single measure is grouped in an instance. Finally, as a gesture evolves along the time (e.g., waving a hand), instances are collected into samples (see Figure 2), which are the inputs for the classification stage. Samples from different dynamic gestures constitute our dataset that will be used to train different classification algorithms in order to assess which one performs best in dynamic gesture recognition.

3.2. Classification Techniques in Gesture Recognition. In this study, we seek to find what the best parameters for dynamic gesture recognition are. For this reason, our system extracts speed and position features from skeletal joints as described before. This information is organized in three datasets: the first one containing only speed information, the second one

with positions only, and the third one combining all features as shown in Figure 2. Besides, we need to find the classifier that performs best with each dataset and compare them to finally develop an online approach that will be integrated in the social robot. In our case, we decided to try a series of classifiers implemented in Weka [31]. Among these 82 classification algorithms, we can find decision trees, random forests, Bayesian, SVM, nearest-neighbors, rule-based, or stacking, among others. (A complete list of classifiers available in Weka can be found here: http://weka.sourceforge.net/doc.dev/weka/classifiers/Classifier.html.) To complement the study, we decided to include 44 other third-party algorithms, which may lead to finding better classification approaches for our data. Table 1 includes a complete list of the third-party classifiers tested.

As a metric to evaluate classifier performance, we decided to consider precision and recall but combined using the F-score as shown in equation 3. Since both measures are important, it is usual to use the F-score as the harmonic mean of recall and precision. In our specific case, we used the weighted F-score since it takes into account not only the F-score of each class (in this case the kind of gesture) but also the number of instances of each one (see equation 4).

$$F\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{3}$$

$$\text{Weighted } F\text{-score} = \frac{\sum_{i=0}^{\#\text{classes}} (F\text{-score}_i \times \text{instances}_i)}{\text{total instances in dataset}}. \tag{4}$$

When using machine learning techniques, an additional problem arises. This is related to selecting the best parameters for the different classifiers [32]. One option to overcome this issue would be to manually adjust the configuration of the different algorithms, but that would be time consuming and would limit the number of techniques that could be tested. Alternatively, AutoWeka [33] automates this task, finding the best configuration for a set of classifiers. Note that currently AutoWeka only compares the 82 classifiers integrated by default in Weka. Therefore, we used AutoWeka for tuning the integrated algorithms, while the 44 third-party ones were manually adjusted.

3.3. Integrating the Gesture Recognition Approach in a Social Robot. The next step is to integrate the best combination of classifier and features in our social robot to detect gestures online from data acquired from the onboard 3D camera. As shown in Figure 3, the three first steps are similar as in the offline method but with one limitation: when operating in real-time it is not easy to assess when a gesture begins and ends as users freely perform the gestures. For this reason, a temporal aggregation through a sliding window is applied to the input data, continuously gathering instances coming from the 10 last frames into a sample that is then processed by the classifier.

In this online mode, the best classifier was integrated in the social robot through the machine learning libraries provided within scikit-learn (scikit-learn homepage: http://scikit-learn.org). Additionally, we developed a simple HRI application using the robot's Text-To-Speech system (TTS) in order to provide feedback when a gesture was detected. Finally, every second the algorithm gathers the last 10 dynamic gestures recognised, and the one with more detections is considered as the final detection. Then, the application sends a command to the TTS to provide feedback.

4. Experiment Description

All data has been collected from our social robot. First, just using the 3D camera to record the datasets, those were processed offline to assess the best classifier and finally as an online proof of concept with simple interaction. This section describes the main features of the robot as well as the experimental procedure for acquiring data.

4.1. Robotic Platform. Users interacted with the robot Mini designed and built at RoboticsLab research group from Carlos III University of Madrid (see Figure 4). This desktop robot has been employed in stimulation sessions for mild cognitive impaired elderly people [34]. The platform offers multiple interaction interfaces such as automatic speech recognition [35], voice activity detection [36], user recognition [37], user localization [38], user identification [39], emotion detection [40], and TTS capabilities as well as a 3D camera. This device was a Kinect for Xbox One with a colour resolution of 1920×1080 pixels at 30 frames per second (limited in our study to 10 fps) and a depth resolution of 512×424 points at the same frame rate. The operational depth range is 0.5 to 8 meters with a horizontal field of view of 70. The software architecture of the robot builds on the *ROS* framework [41] as the backbone that connects all components.

4.2. Set of Gestures. We consider 14 common gestures in HRI involving upper body movements. Since we want to apply gesture recognition to one-to-one interaction tasks, each user stood in front of the robot, and although we recorded information from the whole body, our gestures involve mainly arms and torso movements. These gestures include standing, come towards, crossing arms, move hands to the head, point front, stop sign, greetings, and pointing left and pointing right, as shown in Figure 5.

Other works also consider upper body gestures that are included within our proposed set. For example, the work of Oh et al. [24] recognises four upper body gestures: idle, hands to the head, and greet with the left and the right hands. Mangera et al. [29] use a total of eight gestures, namely, left hand push up, right hand push up, left hand pull down, right hand pull down, left hand swipe right, right hand swipe left, left hand wave, and right hand wave. Hasanuzzaman et al. [42] reduce the set of gestures applied in HRI to only eight hand static gestures and two dynamic gestures (move face up and down for affirmations and move face left and right for negations).

4.3. Data Collection Procedure. Data collection followed a thorough process. The experimenter welcomed the users to the experimentation area and started by explaining the main steps that would take place. After clarifying any doubts or concerns, the users willing to participate signed consent

Table 1: Third-party classifiers tested in this work (table from [50]).

Name	Developed by	Available on
EBMC	A. Lopez Pineda	https://github.com/arturolp/ebmc-weka
Discriminant analysis	Eibe Frank	http://weka.sourceforge.net/doc.packages/discriminantAnalysis
Complement naive Bayes	Ashraf M. Kibriya	http://weka.sourceforge.net/doc.packages/complementNaiveBayes
IBKLG	S. Sreenivasamurthy	https://github.com/sheshas/IBkLG
Alternating decision trees	R. Kirkby et al.	http://weka.sourceforge.net/doc.packages/alternatingDecisionTrees
HMM	Marco Gillies	http://www.doc.gold.ac.uk/mas02mg/software/hmmweka/index.html
Multilayer perceptrons	Eibe Frank	http://weka.sourceforge.net/doc.packages/multiLayerPerceptrons
CHIRP	Leland Wilkinson	http://www.myweb.ttu.edu/tnhondan/file/CHIRP-KDD.pdf
AnDE	Nayyar Zaidi	http://weka.sourceforge.net/packageMetaData/AnDE/index.html
Ordinal learning method	TriDat Tran	http://weka.sourceforge.net/doc.packages/ordinalLearningMethod
Grid search	B. Pfahringer et al.	http://weka.sourceforge.net/doc.packages/gridSearch
AutoWeka	Lars Kotthoff et al.	https://github.com/automl/autoweka
Ridor	Xin Xu	http://weka.sourceforge.net/doc.packages/ridor
Threshold selector	Eibe Frank	http://weka.sourceforge.net/doc.packages/thresholdSelector
ExtraTrees	Eibe Frank	http://weka.sourceforge.net/doc.packages/extraTrees
LibLinear	B. Waldvogel	https://liblinear.bwaldvogel.de/
SPegasos	Mark Hall	http://weka.sourceforge.net/doc.packages/SPegasos
Clojure classifier	Mark Hall	http://weka.sourceforge.net/doc.packages/clojureClassifier
SimpleCART	Haijian Shi	http://weka.sourceforge.net/doc.packages/simpleCART
Conjunctive rule	Xin Xu	http://weka.sourceforge.net/doc.packages/conjunctiveRule
DTNB	Mark Hall et al.	http://weka.sourceforge.net/doc.packages/DTNB
J48 consolidated	J. M. Perez	http://www.aldapa.eus
Lazy associative classifier	Gesse Dafe et al.	https://code.google.com/archive/p/machine-learning-dcc-ufmg/wikis/LACLazyAssociativeAlgorithmCpp.wiki
DeepLearning4J	C. Beckham et al.	http://weka.sourceforge.net/doc.packages/wekaDeeplearning4j
HyperPipes	Len Trigg et al.	http://weka.sourceforge.net/doc.packages/hyperPipes
J48Graft	J. Boughton	http://weka.sourceforge.net/doc.packages/J48graft
Lazy Bayesian rules classifier	Zhihai Wang	http://weka.sourceforge.net/doc.stable/weka/classifiers/lazy/LBR.html
Hidden naive Bayes classifier	H. Zhang	http://weka.sourceforge.net/doc.packages/hiddenNaiveBayes
Dagging meta-classifier	B. Pfahringer et al.	http://weka.sourceforge.net/doc.packages/dagging
MultilayerPerceptronCS	Ben Fowler	http://weka.sourceforge.net/doc.packages/multilayerPerceptronCS
Winnow and balanced winnow classifier	J. Lindgren	http://weka.sourceforge.net/doc.packages/winnow
Nearest-neighbor-like classifier	Brent Martin	http://weka.sourceforge.net/doc.packages/NNge
Naive Bayes tree	Mark Hall	http://weka.sourceforge.net/doc.packages/naiveBayesTree
Kernel logistic regression	Eibe Frank	http://weka.sourceforge.net/doc.packages/kernelLogisticRegression
LibSVM	FracPete	https://www.csie.ntu.edu.tw/cjlin/libsvm/
Fuzzy unordered rule induction	J. C. Hhn	http://weka.sourceforge.net/doc.packages/fuzzyUnorderedRuleInduction
Best first tree	Haijian Shi	http://weka.sourceforge.net/doc.packages/bestFirstTree
MetaCost meta-classifier	Len Trigg	http://weka.sourceforge.net/doc.packages/metaCost
Voting feature intervals classifier	Mark Hall	http://weka.sourceforge.net/doc.packages/votingFeatureIntervals
Ordinal stochastic dominance learner	Stijn Lievens	http://weka.sourceforge.net/doc.packages/ordinalStochasticDominance
RBFNetwork	Eibe Frank	http://weka.sourceforge.net/doc.packages/RBFNetwork
MODLEM rule algorithm	S. Wojciechowski	https://sourceforge.net/projects/modlem/
The fuzzy lattice reasoning classifier	I. N. Athanasiadis	http://weka.sourceforge.net/doc.packages/fuzzyLaticeReasoning
Functional trees	C. Ferreira	http://weka.sourceforge.net/doc.packages/functionalTrees

forms for participation in the experiments and to authorize video recordings.

In all cases, the data collection started with each user standing alone in front of the robot Mini as shown in Figure 6. Then, for each gesture, the users watched a short video of an actor performing the gesture. After that, users performed that gesture as many times as they wanted and repeated the process with the remaining ones. 30 users

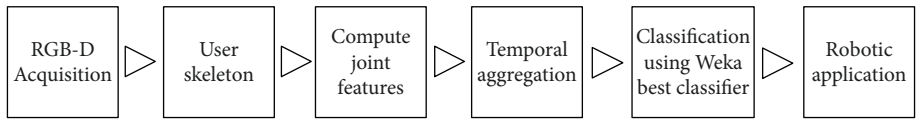

Figure 3: Online pipeline. This operation mode is integrated in the social robot. The output of the best classifier is used in the high-level application to provide feedback.

Figure 4: Mini, the social robot involved in these experiments.

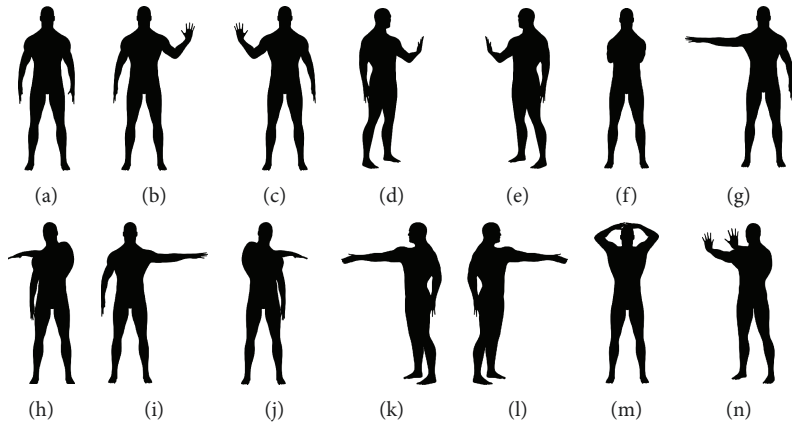

Figure 5: Static representation of the dynamic gestures employed: (a) standing; (b) greeting left hand; (c) greeting right hand; (d) come toward left hand; (e) come toward right hand; (f) cross arms; (g) pointing right; (h) pointing right crossing arm; (i) pointing left; (j) pointing left crossing arm; (k) pointing front left hand; (l) pointing front right hand; (m) hand to head; and (n) stop.

participated in the data collection stage, providing a total of 1355 valid samples (of 1 second duration). These samples were filtered to generate the speed dataset, that contains only joint speed features, and the position dataset, that contains only position information for each joint. Additionally, we also considered the whole set of samples containing both speed and position features. In the case of this third dataset, 900 input features were collected per sample (6 features per joint $\times$ 15 joints $\times$ 10 frames per second). The number of samples per gesture was common for the three datasets: 133 for standing, 94 for greeting with the left hand, 99 for greeting with the right hand, 104 for come towards with the left hand, 103 for come towards with the right hand, 92 for crossing arms, 91 for pointing right, 91 for pointing right crossing the left arm, 92 for pointing left, 92 for

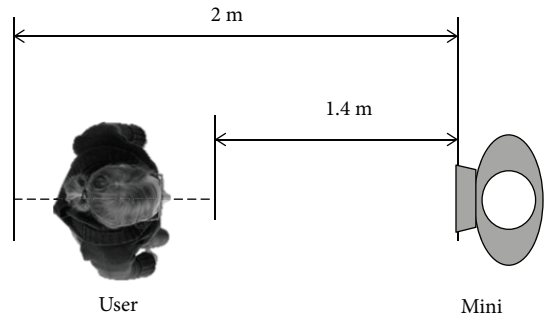

Figure 6: Schematic view of the interaction distances.

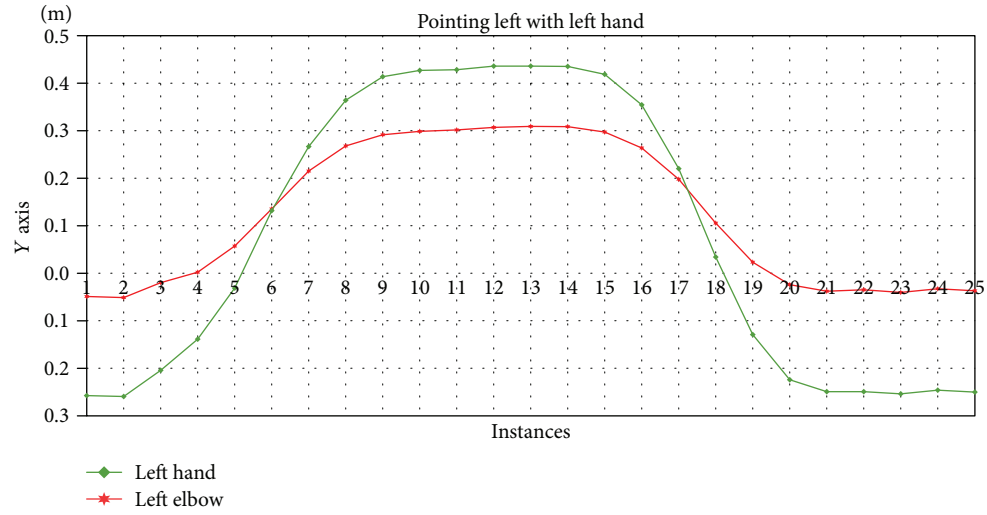

FIGURE 7: Temporal evolution of the *pointing left* dynamic gesture. The red line corresponds to the evolution on the Y axis of the left elbow joint. The green line corresponds to the evolution on the Y axis of the left hand joint. Each dot corresponds to the time in which an instance was recorded.

pointing left crossing the right arm, 90 for pointing front with the left hand, 90 for pointing front with the right hand, 92 for hands to the head, and 92 for stop gesture. (The datasets can be downloaded following this link: https://github.com/jccmontoya/dynamic_gesture_dataset.)

As an example, Figure 7 shows the temporal evolution of a dynamic gesture, pointing left with the left hand. For the sake of clarity, the plot only includes the spatial evolution on the Y axis (the most representative for this gesture) for the two most relevant joints (left arm and left elbow). In this case, the sample for classification would consist of the 10 first plot instances as we are considering just the first second to define each gesture. In the figure, it is easy to see that for the first 10 instances the user moves the arm on the axis to the left side, and then, between instances 14 and 21 it goes back to the initial position.

5. Results

After collecting data for our datasets, we proposed three cases of analysis. The first one only takes into account the position of the skeleton joints, the second one considers only information related to the speed of the joints, and the third one considers both kinds of features together to define dynamic gestures. The reason to train the classifiers with these three sets of features was to assess whether information about the velocity of each joint provides some additional value. If this hypothesis was true, it should be reflected in the F-score obtained during the evaluation process. Note that all classification techniques have been evaluated using *cross-validation*, and these results are presented on Section 5.1.

In the next phase of our analysis, once the performance of the three datasets was ascertained, we evaluated the one that achieved the best result in a more realistic scenario. In this phase, usually known as the *test phase* (see Section 5.2), we split the dataset into two parts, the first one with the 70% of samples and the second one with the remaining 30%. The first set then was used for training, having the second (the untrained one) for validating the performance of the classifiers. Usually, this test phase gets worse results than the previous made based on cross-validation; however, the accuracy obtained is closer than the one achieved in real interactions.

Apart from these quantitative tests, we tested the online approach, integrated in the social robot, in the controlled settings of our lab (see Section 5.3).

5.1. Evaluating Features and Classifiers with Cross-Validation. In this first set of tests, we trained the classifiers using tenfold cross-validation for the three datasets. This method provides a good compromise between variance and bias estimating the error [43, 44]. Our aim here was to assess how the classification results vary with the sets of features. The results showed that the performance achieved was high, with at least one classifier providing an average F-score 0.96 or higher in all cases. Figure 8 shows the best F-score achieved by the set of classifiers.

More specifically, the classification of the position features provided good accuracy (see Figure 8(a)). We can highlight here that 8 classifiers obtained a weighted F-score above 0.90. The best performance, nevertheless, was achieved by the neural network classifier, with a weighted F-score of 0.974. Similarly, random forest obtained a competitive performance with an F-score of 0.973. The tests over the speed dataset, depicted in Figure 8(b), showed lower performance. In this case, six classifiers still managed to obtain an F-score above 0.90, but the highest performance was 0.968, not as good as in the previous test. In the last set of cross-validation tests, we assessed how combining position and velocity features relate to the accuracy of the classifiers. It is worth remembering that each dataset sample was composed of 900 input features (see Figure 8(c)).

Analysing the results in detail, we found that the top-scored classifiers (among the 126 tested) coincide with those showing good performance in traditional machine

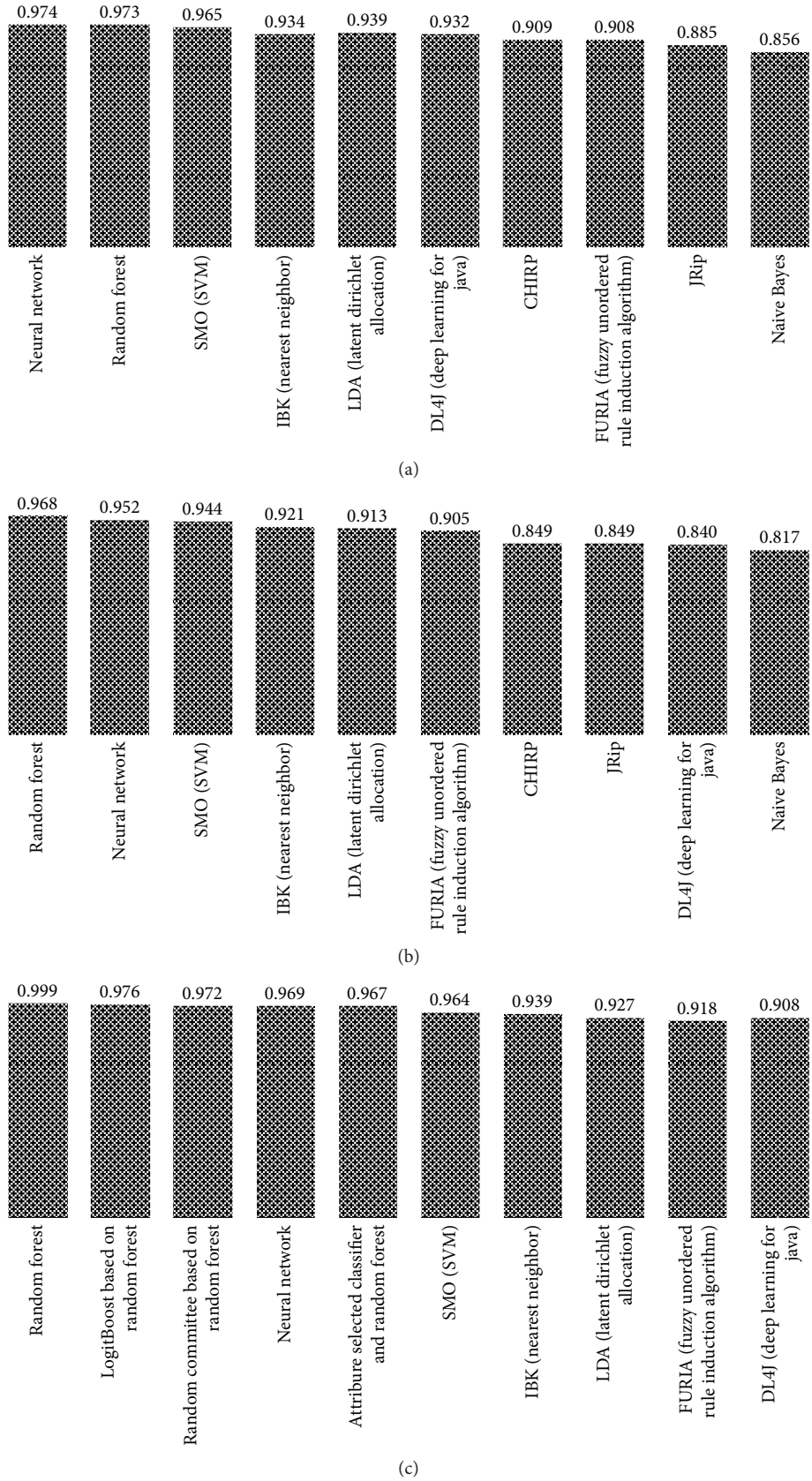

Figure 8: Best weighted F-score using cross-validation for the datasets. (a) Results of the position dataset with cross-validation. (b) Results of the velocity dataset with cross-validation. (c) Results of the full dataset (position+velocity).

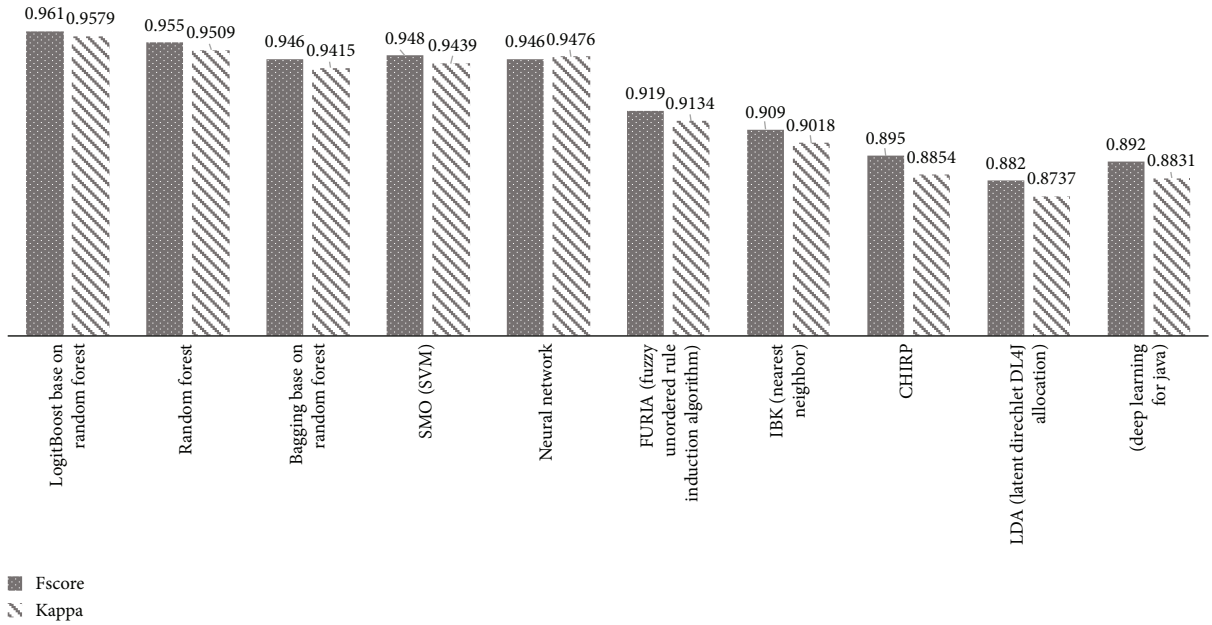

Figure 9: Best weighted F-scores and Cohen's kappa coefficients for the test phase: results of the full dataset (*position* and *velocity*) with test.

learning works [45, 46]. In our case, random forest reached the highest accuracy in this validation phase.

In the case of deep learning (DL4J classifier), we tested different network configurations. The best configuration for this problem included the following layers: dense, convolutional, subsampling (pooling), convolutional, subsampling, and output (fully connected or perceptron). Although the deep learning-based algorithm integrated performed acceptably, it does not improve on the results achieved by traditional classifiers such as random forest. According to the literature, these algorithms reach their best performance in high-dimensional problems (working with raw data instead of just a set of features) with thousands of samples [47].

5.2. Testing the Best Features. In the previous testing phase, results indicated that the combination of position and speed features provide the best performance. Additionally, we wanted to test the performance of the classifiers with untrained data, avoiding the overfitting problem that could appear in cross-validation. Therefore, we split the dataset into two: one of them to train the classifiers with the 70% of the instances and the rest of the set is used for testing. According to the Pareto Principle [48], a training set is about 70%–80% and test set about 20%–30% of the total amount of samples.

Although the performance in this last tests was not as good as in the previous cases, the F-score remains high, finding 7 classifiers able to achieve 0.90 or more. Again, random forest provided the best performance. More precisely, one of its variants (LogitBoost based on random forest) achieved a competitive performance of 0.961, which still is close to the results found with cross-validation. To complement the results, we have included an additional popular metric, the Cohen's kappa, in the results of the full dataset with test data. Figure 9 shows that the results yielded by both metrics are consistent. We believe this is important as both metrics could be biased (e.g., true negatives are ignored in the calculation of F-score or Cohen's kappa has limitations for skewed data). In any case, since those limitations are independent, we believe that these results offer realistic information about the performance of our system.

5.3. Operation in the Real Robot. The last test of this work was a qualitative proof of concept to ascertain whether or not the online pipeline could be used in gesture recognition applications on a social robot. For this reason, we developed a simple HRI application and the research team performed some test rounds to assess the performance of the online pipeline (see Figure 10). The performance in these tests dropped if compared to the previous ones as in this case the beginning and end of the gestures were not limited, but the sliding-window mechanism managed to take care of this limitation. Therefore, more noisy samples of intermediate movements were classified, but still, since the system also implements a polling mechanism, the final result was promising. Nevertheless, still some gestures were confused. For instance, the gestures *come towards with the left hand* and *pointing front with the left hand* could present similar features depending on what part of each gesture is considered. An example of the performance of the system can be found in the following video https://youtu.be/AgTvaNtnZFs.

6. Discussion

The results show good performance under the experimental conditions. For the offline mode, the dynamic gestures were limited to 1-second (10 instances) duration. When recording the datasets, we considered only the first second per instance recorded. This could have been a problem in the online mode when the system was integrated into the social robot, but this limitation was overcome using a sliding-window approach with a window size of 1 second. In order to improve accuracy in the real setup, a polling mechanism was developed on

(a)

(b)

(c)

Figure 10: Still frames of the user performing dynamic gestures when interacting with the real robot.

top of the sliding window, considering the majority out of ten detections as the detected gesture. One possible improvement to this work is related to detecting when a gesture starts and ends.

Another limitation could be related to the method we used to show the users the possible gestures. We acknowledge that using videos could induce some bias to the tests, but this is also a way of homogenizing the information that users receive from the experimenters. In fact, it was demonstrated that a more significant bias lies in the fact that when human experimenters present a test they know exactly what condition will be tested and consequently, it is possible that objects are presented differently in possible versus impossible trials, which may also alter the experiments [49]. In our case, we believe that the bias effect was mitigated as users freely performed as many gestures as they wanted without further intervention from the research team.

Also, we have considered just upper body dynamic gestures since in our applications the users tend to interact in front of a desktop robot. Nevertheless, the literature demonstrates that these gestures are common between different works. However, since we are using all joints of the user's skeleton, the proposal could be easily extended to full body gestures.

7. Conclusion

This work proposed the study of how different combination of features affected the recognition of dynamic gestures. A series of machine learning techniques were tested in order to find the one able to classify better those gestures with different sets of features. Therefore, we trained 126 classifiers with three datasets containing 1355 samples each. These were acquired from 30 users performing 14 upper body dynamic gestures. For each gesture, features related to position and speed were collected.

Results seem to indicate that both speed and position matter when detecting dynamic gestures. Combining them, the classifiers achieved a weighted F-score of 0.999 versus the performance obtained by just using position or speed separately (0.978 and 0.968, respectively). Additionally, we performed an extra test to check the performance of the classifiers in a more realistic scenario, that is, with untrained data. In that case, the performance was lower, as expected, but still providing competitive results since the weighted F-score reached 0.961. In all cases, the random forest classifier provided either the best performance or the second best one, close to the best classifier.

After finding the best classifier and the set of parameters more adequate for upper body dynamic gesture recognition, we integrate the approach into a social robot coupled to an HRI application to provide feedback, when gestures were detected, using the TTS capabilities of the robot. This approach was tested in a controlled scenario, our research lab. Since the aim of this work was to study and assess the feasibility of the approach, we believe that we are ready to move forward, integrating the system in our HRI architecture and starting the tests with real users.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The research leading to these results has received funding from the ROBSEN project (Desarrollo de robots sociales para ayuda a mayores con deterioro cognitivo; DPI2014-57684-R) funded by the Spanish Ministry of Economy and Competitiveness and from the RoboCity2030-III-CM project (Robótica aplicada a la mejora de la calidad de vida de los ciudadanos. Fase III; S2013/MIT-2748) funded by the Programas de Actividades I+D en la Comunidad de Madrid and cofunded by Structural Funds of the EU.

References

[1] A. Kendon, *Gesture: Visible Action as Utterance*, Cambridge University Press, 2004.

[2] J. Iverson, D. Thal, A. Wetherby, S. Warren, and J. Reichle, "Communicative transitions: there more to the hand than meets the eye," *Transitions in Prelinguistic Communication*, vol. 7, pp. 59–86, 1998.

[3] S. Goldin-Meadow and M. W. Alibali, "Gesture's role in speaking, learning, and creating language," *Annual Review of Psychology*, vol. 64, no. 1, pp. 257–283, 2013.

[4] S. W. Lee, "Automatic gesture recognition for intelligent human-robot interaction," in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pp. 645–650, Southampton, UK, 2006.

[5] A. Ramey, V. González-Pacheco, and M. A. Salichs, "Integration of a low-cost RGB-D sensor in a social robot for gesture recognition," in *Proceedings of the 6th international conference on Human-robot interaction - HRI '11*, pp. 229-230, Lausanne, Switzerland, 2011.

[6] S. S. Rautaray and A. Agrawal, *Vision Based Hand Gesture Recognition for Human Computer Interaction: A Survey*, Artificial Intelligence Review, 2015.

[7] J. C. Castillo, D. Carneiro, J. Serrano-Cuerda, P. Novais, A. Fernández-Caballero, and J. Neves, "A multi-modal approach for activity classification and fall detection," *International Journal of Systems Science*, vol. 45, no. 4, pp. 810–824, 2013.

[8] A. Fernández-Caballero, J. C. Castillo, and J. M. Rodríguez--Sánchez, "Human activity monitoring by local and global finite state machines," *Expert Systems with Applications*, vol. 39, no. 8, pp. 6982–6993, 2012.

[9] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *IEEE Pervasive Computing*, vol. 9, no. 1, pp. 48–53, 2010.

[10] A. Fernández-Caballero, J. C. Castillo, M. T. López, J. Serrano-Cuerda, and M. V. Sokolova, "INT3-Horus framework for multispectrum activity interpretation in intelligent environments," *Expert Systems with Applications*, vol. 40, no. 17, pp. 6715–6727, 2013.

[11] J. Shotton, T. Sharp, A. Kipman et al., "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[12] A. Jalal, Y. Kim, and D. Kim, "Ridge body parts features for human pose estimation and recognition from RGB-D video data," in *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pp. 1–6, Hefei, China, July 2014.

[13] A. Jain, J. Tompson, Y. LeCun, and C. Bregler, "MoDeep: a deep learning framework using motion features for human pose estimation," in *Computer Vision – ACCV 2014*, D. Cremers, I. Reid, H. Saito, and M. H. Yang, Eds., vol. 9004, Springer, 2015.

[14] A. Mohanty, A. Ahmed, T. Goswami, A. Das, P. Vaishnavi, and R. R. Sahay, *Robust Pose Recognition Using Deep Learning*, Springer, Singapore, 2017.

[15] J. Wu, J. Konrad, and P. Ishwar, "Dynamic time warping for gesture-based user identification and authentication with Kinect," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2371–2375, Vancouver, Canada, May 2013.

[16] L. P. Morency, A. Quattoni, and T. Darrell, "Latent-dynamic discriminative models for continuous gesture recognition," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis, MN, USA, June 2007.

[17] S. Waldherr, R. Romero, and S. Thrun, "A gesture based interface for human-robot interaction," *Autonomous Robots*, vol. 9, no. 2, pp. 151–173, 2000.

[18] D. Santos, B. Fernandes, and B. Bezerra, "HAGR-D: a novel approach for gesture recognition with depth maps," *Sensors*, vol. 15, no. 11, pp. 28646–28664, 2015.

[19] F. Milazzo, V. Gentile, G. Vitello, A. Gentile, and S. Sorce, "Modular middleware for gestural data and devices management," *Journal of Sensors*, vol. 2017, 13 pages, 2017.

[20] Y. Yacoob and M. Black, "Parameterized modeling and recognition of activities," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pp. 1–28, Bombay, India, 1998.

[21] A. Jalal, S. Lee, J. T. Kim, and T.-S. Kim, "Human activity recognition via the features of labeled depth body parts," in *Impact Analysis of Solutions for Chronic Disease Prevention and Management: 10th International Conference on Smart Homes and Health Telematics*, M. Donnelly, C. Paggetti, C. Nugent, and M. Mokhtari, Eds., pp. 246–249, ICOST 2012, Artiminio, Italy, 2012.

[22] S. Zuffi and M. J. Black, "The stitched puppet: a graphical model of 3D human shape and pose," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3537–3546, Boston, MA, USA, June 2015.

[23] D. Gavrila and L. Davis, "3-D model-based tracking of humans in action: a multi-view approach," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 73–80, San Francisco, CA, USA, 1996.

[24] C.-M. Oh, M. Z. Islam, J.-S. Lee, C.-W. Lee, and I.-S. Kweon, "Upper body gesture recognition for human-robot interaction," in *Human-Computer Interaction. Interaction Techniques and Environments*, J. A. Jacko, Ed., pp. 294–303, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[25] S. Celebi, A. S. Aydin, T. T. Temiz, and T. Arici, "Gesture recognition using skeleton data with weighted dynamic time warping," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, p. 79, Barcelona, Spain, 2013.

[26] Y. Gu, H. Do, Y. Ou, and W. Sheng, "Human gesture recognition through a Kinect sensor," in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1379–1384, Guangzhou, China, 2012.

[27] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (SMIJ): a new representation for human skeletal action recognition," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 24–38, 2014.

[28] Y. Zhu, W. Chen, and G. Guo, "Evaluating spatiotemporal interest point features for depth-based action recognition," *Image and Vision Computing*, vol. 32, no. 8, pp. 453–464, 2014.

[29] R. Mangera, F. Senekal, and F. Nicolls, "Cascading neural networks for upper-body gesture recognition," in *Proceedings of the International Conference on Machine Vision and Machine Learning*, pp. 5–9, Prague, Czech Republic, 2014.

[30] S. Fong, J. Liang, I. Fister, I. Fister, and S. Mohammed, "Gesture recognition from data streams of human motion sensor using accelerated PSO swarm search feature selection algorithm," *Journal of Sensors*, vol. 2015, Article ID 205707, 16 pages, 2015.

[31] G. Holmes, A. Donkin, and I. Witten, "WEKA: a machine learning workbench," in *Proceedings of ANZIIS '94 -*

Australian New Zealnd Intelligent Information Systems Conference, pp. 357–361, Brisbane, Australia, 1994.

[32] R. Z. Marques, L. R. Coutinho, T. B. Borchartt, S. B. Vale, and F. J. Silva, "An experimental evaluation of data mining algorithms using hyperparameter optimization," in *2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI)*, pp. 152–156, Cuernavaca, Mexico, October 2015.

[33] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-WEKA," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*, p. 847, New York, USA, 2013.

[34] M. A. Salichs, I. P. Encinar, E. Salichs, Á. Castro-González, and M. Malfaz, "Study of scenarios and technical requirements of a social assistive robot for Alzheimer's disease patients and their caregivers," *International Journal of Social Robotics*, vol. 8, no. 1, pp. 85–102, 2016.

[35] F. Alonso-Martín and M. A. Salichs, "Integration of a voice recognition system in a social robot," *Cybernetics and Systems*, vol. 42, no. 4, pp. 215–245, 2011.

[36] F. Alonso-Martin, Á. Castro-González, J. Gorostiza, and M. A. Salichs, "Multidomain voice activity detection during human-robot interaction," in *International Conference on Social Robotics (ICSR 2013)*, pp. 64–73, Springer International Publishing, Bristol, 2013.

[37] A. Ramey, Á. Castro-González, M. Malfaz, F. Alonso-Martín, and M. A. Salichs, "Vision-based people detection using depth information for social robots," *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, article 172988141770592, 2017.

[38] F. Alonso-Martín, J. F. Gorostiza, M. Malfaz, and M. A. Salichs, "User localization during human-robot interaction," *Sensors*, vol. 12, no. 7, pp. 9913–9935, 2012.

[39] F. Alonso-Martin, A. Ramey, and M. Á. Salichs, "Speaker identification using three signal voice domains during human-robot interaction, pp. 114-115, Bielefeld, Germany, 2014.

[40] F. Alonso-Martín, M. Malfaz, J. Sequeira, J. Gorostiza, and M. Salichs, "A multimodal emotion detection system during human-robot interaction," *Sensors*, vol. 13, no. 11, pp. 15549–15581, 2013.

[41] M. Quigley, B. Gerkey, K. Conley et al., "ROS: an open-source robot operating system," *ICRA workshop on open source software*, vol. 3, no. 3.2, p. 5, 2009.

[42] M. Hasanuzzaman, V. Ampornaramveth, M. Bhuiyan, Y. Shirai, and H. Ueno, "Real-time vision-based gesture recognition for human robot interaction," in *2004 IEEE International Conference on Robotics and Biomimetics*, pp. 413–418, Shenyang, China, 2004.

[43] L. Breiman, "The little bootstrap and other methods for dimensionality selection in regression: X-fixed prediction error," *Journal of the American Statistical Association*, vol. 87, no. 419, pp. 738–754, 1992.

[44] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, New York, NY, 2009.

[45] R. Caruana, N. Karampatziakis, and A. Yessenalina, "An empirical evaluation of supervised learning in high dimensions," in *Proceedings of the 25th international conference on Machine learning - ICML '08*, pp. 96–103, Helsinki, Finland, 2008.

[46] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, and D. Amorim Fernández-Delgado, "Do we need hundreds of classifiers to solve real world classification problems?," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3133–3181, 2014.

[47] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, p. 1, 2015.

[48] F. J. Reh, *Pareto's Principle-the 80-20 Rule*, vol. 107, no. 7, 2005, Business credit, New York, 2005.

[49] G. R. Bock and G. Cardew, *Characterizing Human Psychological Adaptations*, vol. 208, John Wiley & Sons, 2008.

[50] F. Alonso-Martín, J. Gamboa-Montero, J. Castillo, Á. Castro-González, and M. Salichs, "Detecting and classifying human touches in a social robot through acoustic sensing and machine learning," *Sensors*, vol. 17, no. 5, p. 1138, 2017.