Cognitive Systems Research 71 (2022) 1–8

Contents lists available at ScienceDirect

Cognitive Systems Research

journal homepage: www.elsevier.com/locate/cogsys

Action recognition for educational proposals applying concepts of Social Assistive Robotics

Kevin Braathen de Carvalho *, Vitor Thinassi Basílio, Alexandre Santos Brandão

Núcleo de Especialização em Robótica, Universidade Federal de Viçosa, Viçosa, Brazil

ARTICLE INFO

Keywords:
Motion sensing
Skeleton joints
Eigenvalues
Artificial Neural Networks
Robotics application

ABSTRACT

Action recognition has been gaining interest in research due to its great number of applications. So, the main contribution of this manuscript is a human–robot interaction framework, relying on dimension reduction of the system's inputs in order to require a smaller dataset for training of an Artificial Neural Network. Our motivation is the development of a Social Assistive Robotics application. In summary, we choose nine standard actions to guide a robot, and two neutral ones to represent stand-by or resting cases. The dataset is created by people with different body shape, for robustness purposes, using only 5 to 10 samples of each class per person. Offline and online tests validate the method's accuracy and confusion matrices clarify the results. A TicTacToe game using a ground robot exemplify a real world application, where each action represents a desired spot in the game. The results confirm a high accuracy, above 96.7%, in all the tests. Based on this, we can conclude our preprocessing strategy and classifier are capable of identifying the action patterns, even for a tiny dataset; thus, it is recommended for educational proposals due to its simplicity.

1. Introduction

Social Assistive Robotics (SAR) can be defined as the intersection between Social Interactive Robotics (SIR) and Assistive Robotics (AR). SIR is inspired by the communication between the robot and the environment where it is in, or even among robots themselves (Feil-Seifer & Mataric, 2005; Fong, Nourbakhsh, & Dautenhahn, 2003). In other words, whenever there is a type of interaction, SIR concepts are present and can be applied. In turn, AR can be defined as a machine that assists a human in any way (Feil-Seifer & Mataric, 2005).

Many works have SAR as field of study and they have been applied to the aid of the elderly (Tapus, Maja, & Scassellatti, 2007; Tsardoulias, Kintsakis, Panayiotou, Thallas, Reppou, Karagiannis, Iturburu, Arampatzis, Zielinski, Prunet, et al., 2017), individuals in physical rehabilitation (Kahn, Averbuch, Rymer, & Reinkensmeyer, 2001), support of daily activities (Wilson, Pereyda, Raghunath, de la Cruz, Goel, Nesaei, Minor, Schmitter-Edgecombe, Taylor, & Cook, 2019) or educational robotics (Benitti, 2012). The interface between human and robot can be done in several different manners, such as:

Gestures or actions, when body language can be useful in situations where both human and robot interact with the environment. Furthermore, when used for physical therapy recognizing position and orientation of bodies is vital and can reassure the communication between the user and the agent (Chen, Hua, Dai, He, & Han, 2019; Kanda, Ishiguro, Imai, & Ono, 2004).

Voice commands, used on a daily human communication, can also be applied when talking to a robot, specially those who use a synthetic speech generator or pre-recorded human voice (Eriksson, 2004; Liu & Zhang, 2019).

Peripherals, such as mouse, keyboards and touchscreens, can be useful in situations where gesture or voice recognition could be rather tedious such as pointing a location on a map. In this way using a peripheral could be more efficient and natural for most users (Huttenrauch & Eklundh, 2002; Montemerlo, Pineau, Roy, Thrun, & Verma, 2002).

When it comes to using robotics as an academic tool, several obstacles arise in this knowledge transfer mission. However, teaching professionals indicate that it is more didactic and, many times, it can be said that it is more stimulating, when the teaching and learning process takes place through a problem-based approach (Ackovska & Kirandziska, 2017; Hansen, Hees, & Jeschke, 2013; Jara, Candelas, Puente, & Torres, 2011). Therefore, there is a clear motivation for the development of easy-to-implement platforms for robotics applications, especially those that take into account the human–robot interaction. In this context, the present work presents the solution of a real problem, which can be taken to other more complex scenarios, as long as it

* Corresponding author.
E-mail address: kevinbdc@gmail.com (K.B. de Carvalho).

https://doi.org/10.1016/j.cogsys.2021.09.002
Received 21 September 2020; Received in revised form 20 August 2021; Accepted 14 September 2021
Available online 2 October 2021

preserves its attributes and the proposal to be an application of Socially Assistive Robotics.

This paper main contribution is a human–robot interaction framework, having as motivation the development of a Social Assistive Robotics application. The system has easy-to-follow implementation and provides a clear path to be used as a more hands-on approach. Its topics facilitate the teaching tasks, for instance, the description and the response of an action classifier for a real-world agent. The Artificial Neural Networks (ANN) based classifier relies on dimension reduction to the system's inputs in order to require a smaller dataset for training. To achieve it, initially, an external depth sensor provides the skeleton joints features, and then eigenvalues decomposition reduces the input's dimension. The major insight of the work was the requirement of a tiny dataset, being flexible for several applications. In this work, our proposal is validated through offline and online tests, and in a real world application, where the proposed algorithm is used as a communication tool in a TicTacToe game, i.e., a didactic and ludic application.

The rest of this paper is divided as follows. Action Recognition section contains a broader review on how it has been seen in the literature. Next, Related Works examples different similar applications seen in the literature. The Methodology section details the implementation of the proposed method as well as the dataset creation and the online tests. Further, offline and online tests are explained and their results are exposed and discussed on Experiments and Results section. Finally, concluding remarks are detailed, and future works are exposed as motivation for the readers.

2. Action recognition

Due to its vast array of applications, action recognition has been gaining more room in the academic field in the last decades, such as detection of suspicious activities in public environment, medical monitoring, athletics performance, and advanced human–robot interactions in SAR (Aggarwal & Ryoo, 2011; Aggarwal & Xia, 2014; Moeslund, Hilton, & Krüger, 2006; Sempena, Maulidevi, & Aryan, 2011).

According to Agahian, Negin, and Köse (2019), human action recognition can be divided in two parts: features extraction (using tracking sensors, such as radio frequency identification, InfraRed, RGB-D), and classification of them (using for instance, Dynamic Time Warping, Hidden Markov Model, Artificial Neural Networks, Vector Space).

Tracking sensors usually are either Vision Based or Motion Caption (MoCap). The first consists in using images which can have color (RGB) or depth information, to process the situation and to do the action classification. The second extracts specific features such as skeleton 3D joint position and/or velocities that may be done by processing the image data or by having markers on the body to directly extract its features (Mitra & Acharya, 2007).

As action recognition relies on its practical implementation, it demands the usage of different image and tracking devices (Thobbi & Sheng, 2010). Nowadays, low cost RGB-D sensors such as Intel RealSense and Microsoft Kinect are gaining more room due to their cost–benefit ratio. So this is moving research efforts in actions or gestures classification using 3D skeleton data (Patrona, Chatzitofis, Zarpalas, & Daras, 2018; Yang, Shen, Lui, Lee, Chen, Ding, Liu, Lu, Duan, Wang, et al., 2017). In our project, we use Kinect v2.0 sensor to get body information and extract their features.

As a next step, these features must be labeled, tracked and finally classified. In the universe of all classifiers, we highlight some of them:

Dynamic Time Warping (DTW) which compares two time series to find their similarity level. These series can be depicted as a feature vector, which represents the posture of the skeleton joints, or by depth images. This technique also finds home in fields such as data mining and voice recognition (Barnachon, Bouakaz, Boufama, & Guillou, 2014; Celebi, Aydin, Temiz, & Arici, 2013; Hang, Zhang, Chen, Li, & Li, 2017; Raheja, Minhas, Prashanth, Shah, & Chaudhary, 2015);

Hidden Markov Model (HMM), which uses a quantification of a system's configuration through a finite number of discrete states, whose values stored represent an approximation of the system's dynamics. These states can be a feature vector which consists on the spatial position of each of the objects characteristics bubble obtained using a self-calibrating stereo blob tracker (Kumar, Gauba, Roy, & Dogra, 2017; Zhang, Wang, Wang, & Ma, 2016);

Artificial Neural Networks (ANN), a type of supervised learning technique, which uses a training dataset in order to tune its parameters and then is able to correlate its inputs with its output labels. There are numerous different structures for these networks, such as Recurrent Neural Networks, more used when the inputs are a temporal sequence (Du, Wang, & Wang, 2015; Ng & Ranganath, 2002; Veeriah, Zhuang, & Qi, 2015), Deep Neural Networks, where several neuron layers are employed and requires a large training dataset, but this gives the network the feature extraction capability, without further need of human supervision in this task (Ordóñez & Roggen, 2016; Wang, Li, Gao, Zhang, Tang, & Ogunbona, 2015); the Convolutional Neural Networks, where subsequent layers are not fully connected, allowing the network to be deep but having fewer parameters (Akula, Shah, & Ghosh, 2018; Wang, Li, Gao, Tang, & Ogunbona, 2018; Yan, Xiong, & Lin, 2018).

3. Related works

This section presents some works regarding techniques used for action and gesture recognition as well as some more focused applications using gestures or actions as communication method on Human–Robot interaction.

In Li, Zhong, Xie and Pu (2017), a Convolutional Neural Network (CNN) approach is proposed based on skeleton joint information. The authors apply a linear transformation on the skeletal data before inserting it in the ANN in order to remove less relevant joints for the training process. A two stream CNN is used where one of the streams input is the skeletal 3D information for each captured frame and the other stream input is the movement of each joint from one frame to the last one. The two stream outputs are concatenated and inserted in a deeper layer further in the network.

In Li, Hou, Wang and Li (2017) the authors also used a CNN approach. Instead of using a two stream network, with movement and 3D joint information as inputs, this work proposes a Joint Distance Map as network input. The euclidean distance from one joint to the others is calculated and separated in different maps from using views from the planes xy, xz, yz and also the full 3D map. Each of these four maps are used as input for different CNNs and their outputs are fused in order to classify the action.

The work Gharaee (2020) proposes a novel architecture for action recognition using layers of growing grid neural networks, allowing the system to automatically arrange the representational structure. It consists of two growing grid where the first receives the input data and generates the action pattern vector and the second categorizes the input vector to the corresponding action clusters and finally a one layer ANN labels the clusters with action labels.

An approach integrating information of both RGB and Depth camera is proposed in Buonamente, Dindo, and Johnsson (2016). It uses Salient Information Map where Sign, Magnitude and Center descriptors are extracted to represent the Complete Local Binary Pattern. A Support Vector Machine is used to classify the features into the action classes.

As for gestures or actions used for human robot interaction, the paper Xu, Wu, Chen, and Xu (2015) presents a RGB-D recognition system for hand gesture recognition. It uses two self-built datasets containing dynamic gestures to train the system, which uses multi-feature extraction and quantization of the hand trajectory and Hidden Markov Models to recognize each of the 16 possible classes.

A multi modal fusion between speech and gesture communication for human–robot interaction is proposed in Burger, Ferrané, Lerasle, and Infantes (2012), where an Interactively Distributed Multiple Object Tracking is used to track the user's both hands and head and classifies the two handed gestures using HMM as well having a total of twelve gesture classes with more practical commands associated to them, such as "greetings", "stop", "come to me","go away" and others. In such a case, an embedded speech recognition system is paired with a HMM, to fuse the gesture and speech interpretation and get the classification of seven distinct commands.

The authors of Nagi, Ducatelle, Di Caro, Cireşan, Meier, Giusti, Nagi, Schmidhuber, and Gambardella (2011) proposed a max-pooling CNN to recognize up to six different hand gestures to control, being possible to extend to eleven classes using both hands. The authors use a 6000 self-built image dataset to train the network. A specific color glove was used during image capture and storage. These images were taken from cameras installed in a small robot.

In general terms, the proposal presented in this manuscript is similar to those referenced. However, comparatively speaking, the biggest and most significant advantage lies in the use of a very small database to achieve the desired action recognition functionality. Therefore, such feature leads the proposal being well suited for activities that required fast prototyping and validation, such as problem based learning, that are currently present in education institutions.

4. Methodology

For action or gesture recognition using machine learning techniques, one has often to rely on open datasets, which can have two implications. First, they can have really large amount of data to process, making it unfeasible in some situations. Second, they provide little flexibility if we want to use it for a specific proposal. In other words, the dataset might not have the actions or gestures that are interesting for certain applications. To attest to this statement, the datasets used in the works Li, Zhang and Liao (2017), Shahroudy, Liu, Ng, and Wang (2016) and Liu, Shahroudy, Perez, Wang, Duan, and Kot (2020) are available online and offer plenty of data to train a classifier. However, the user cannot control how many samples were provided per person for each action, neither control the time duration of each sample, neither define what specific types of actions will be used for an application, as some of them might be inappropriate for it. Therefore, databases are excellent for testing and validating classifiers; however, maybe they are not feasible to real robot command situations, as it is the case in this work.

So, in order to have control of sample size, time duration and the action themselves, we created our own dataset[1] with classes suited for moving a robot in such a way it can cooperate with a human. This section details our dataset and how it was conceived. As well, we present the proposed algorithm and stress some particularities for online application.

4.1. DataSet

Kinect v2.0 sensor extracts the depth image and the skeleton joints information. Nine classes of standard actions were chosen to integrate the dataset, as one can be seen on Fig. 1. All the actions start from stand up resting position and are described as follows: a goodbye gesture with the right hand (A), raising your arm and drawing a circle around your head with your hand (B), raising the right arm until it forms an approximate 45 degrees angle with the torso (C), a goodbye gesture with the left hand (D), crossing both arms in front of your torso forming an X (E), raising the left arm until it forms an approximate 45 degrees angle with the torso(F), raising the right arm in front of you on a "stop sign" (G), putting both hands together in front of the waist (H), raising the arm with the right palm facing upwards doing as a "come closer" gesture (I).

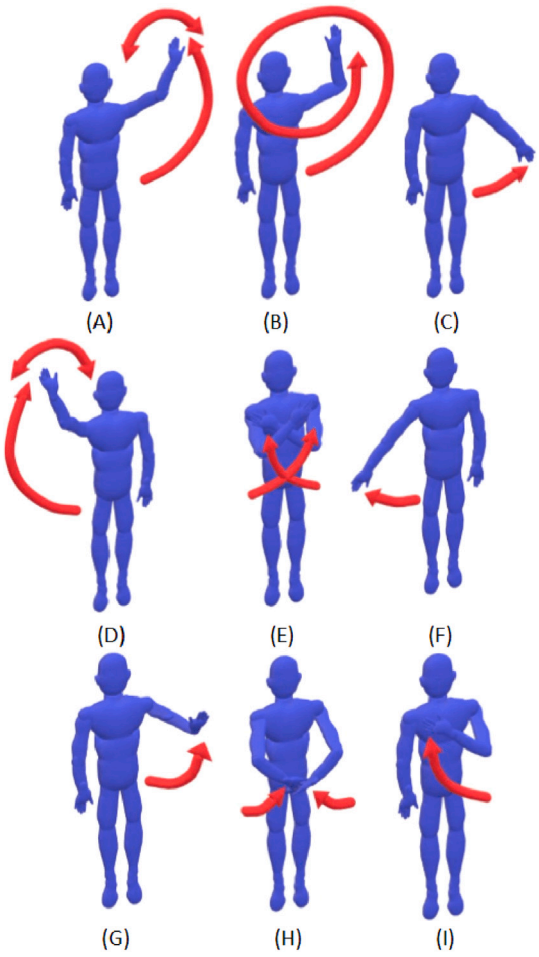

Fig. 1. Standard action classes on the dataset.

Furthermore, two neutral action classes are added, which means that when they are identified, the robot should execute no actions. The first neutral action has samples of the user standing still and the second one has samples of the user walking. In this way the agent will be able to distinguish these neutral actions with natural user movements from the standard action classes, not taking any commands when one of them is identified.

It is important to stress that the capture sensor should be seeing the user constantly, as expected in Social Assistive Robotics, to ensure the existence of an interaction.

The classes A, D, G and I are fairly intuitive, two goodbye signs, a "stop" sign and a clear "come closer" one. The classes C and F may not seem intuitive at first sight, but they were chosen due to the fact of being a base command for Xbox users. So the user may have done them or, at least, seen them before, giving them a more familiar look. Classes E and H are not that present on one's daily life, but are simple to do, therefore they were chosen to integrate the dataset. Class B is the most elaborate and dynamic of all, but not enough to make its execution uncomfortable. This last class was added to test if the classifier would be capable of identifying a more complex action.

The Kinect v2.0 sensor can operate at 30 FPS (Frames Per Second), but it was chosen a 15 FPS capture rate, not pushing the sensor to work on its limit, to reduce the total processing time of the action recognition algorithm, allowing it to be used in systems with lower processing capabilities.

[1] Dataset available at https://github.com/RoboticaUFV/SAR-AuRoRA-Dataset.

The capture window used was of 1,66 s. This was obtained empirically, testing towards finding a big enough window to perform the actions in a comfortable fashion, specially action B (taken as the most complex one).

Given the capture rate and window, each action is represented by a total of 25 frames. Each frame consists in a discrete group of features, which are stored in a matrix:

$$\mathbf{F_k} = \begin{bmatrix} x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_i & y_i & z_i \end{bmatrix}, \quad (1)$$

where F_k is the feature matrix of the k-th frame and the columns represent the 3-D Cartesian coordinates of the i-th skeleton joint. With k and i going up to 25 (25 total frames and 25 skeleton joints).

In the eyes of an observer, an action done in different places in a room would look the same. In contrast, this does not occur when those actions are captured by the Kinect v2.0 sensor, since they will have totally different coordinates for each skeleton joint, even though these actions could belong to the same class. To avoid such Cartesian displacement, all the captured skeleton data is centralized in reference to the coordinates of the left shoulder (adopted as a body reference for our research group, without leading to loss of generality). After being centralized, the complete action is stored as the subsequent concatenation of the feature matrix of each frame, given by

$$\mathbf{A}_{m,n} = [\mathbf{F}_1 \ \mathbf{F}_2 \ \dots \ \mathbf{F}_k] \quad (2)$$

where $\mathbf{A}_{m,n}$ is the action matrix containing the features from all frames, m is the class label and n is the sample's number, and $k = 1, 2, \dots, 25$ (25 total frames).

The dataset created contains 10 action samples of each action class collected by 5 different people, summing a total of 50 action samples for each action class and a total of 550 samples (9 standard actions plus 2 neutral actions).

4.2. Action recognition algorithm

This work proposes a simple neural network that with the proper preprocessing of the inputs can learn how to classify different action classes with high precision rates and tiny dataset.

The used for classification were extracted by the Kinect v2.0 sensor extracts the skeleton shown in Figure Fig. 2 left. However only 8 joints were used for classification, in order to start reducing the input's dimension, as highlighted in green in Fig. 2 right. They are the shoulder, elbow, wrist and hand joints. It is worthy mentioning these joints were chosen because they belong to the body parts that move the most in the actions, as shown in Fig. 1.

The next step is to reduce the action matrix from 8×75 to a set of few features. In other words, our approach consists in getting the principal components of $A \cdot A^T \in R^{8\times8}$, given by the eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_8$.

The vector $\Lambda = [\lambda_1 \ \lambda_2 \ \dots \lambda_8]^T$ is the input of our ANN, whose configuration is one hidden layer only with 30 neurons with all layers fully connected, and 11 outputs (one for each class). The activation function was Sigmoid tangent for the hidden layer and softmax function for the classification layer. Finally the training method used is Bayesian Regularization with Levenberg–Marquadt Optimization.

Fig. 3 illustrates the framework used summarizing our classification proposal, where depth images are captured and converted to skeleton data via Kinect, then this information is converted to eighth eigenvalues, used as inputs for our Neural Network, which classifies the frame sequence.

4.3. Online classification

For training the neural network with action samples, all of them were recorded starting and ending approximately at the same time-window. Thus, there is a little room for time shifting between two different action samples, but it was not relevant to the point of invalidating the extraction of characteristics and classification of them.

In order to avoid a situation where the algorithm would classify actions all the time, a trigger starts storing frames before classifying. In such a case, it happens whenever the user opens one of his hands, because Kinect SDK detects if the skeleton hands are opened or closed.

5. Experiments and results

In order to validate the proposed algorithm, five tests were performed. In this section, they are divided in three offline tests (simulations) and two online tests (experiments), as following description:

Offline Test 1: showcases the algorithm's performance using only two different people in the training dataset. The goal is to evaluate the algorithm's performance using few different people for training.

Offline Test 2: showcases the algorithm's performance using the full dataset, with 10 samples of each action class, taken from 5 different people, for training. The goal is to evaluate how well the algorithm can perform when there are more people on the training dataset.

Offline Test 3: showcases the algorithm's performance using only 5 samples of each action class, taken from 5 different people, for training. The goal of this test is to evaluate how well the algorithm can perform having such a tiny dataset for training.

Online Test 1: was done with two users performing each action class 30 times in a random order. The classification took place in real time. The goal of this test if to guarantee that the dataset is not biased, by analyzing the algorithm's performance with two users, which provided samples for the training dataset, performing real actions.

Online Test 2: showcases, in a playful way, the algorithm's performance on a real world application. This is done by having each position in a TicTacToe game associated to a standard action. A total of three different TicTacToe matches were played by two different users.

6. Offline tests

All the simulation results are shown in a confusion matrix showing the occurrence percentage. The last line of the tables show the false positive (FP) occurrence percentage and the last column shows the false negative (FN) occurrence percentage. The columns are the desired classes and the lines are the predicted classes by the action recognition system. The training and testings were done using MATLAB on an Intel i7-4810MQ CPU on a 16 GB RAM computer. The average training time using the whole dataset was 10.5 s for the whole training and the average time to get the input's classification through the network after 1000 repetitions was of 6.8 ms.

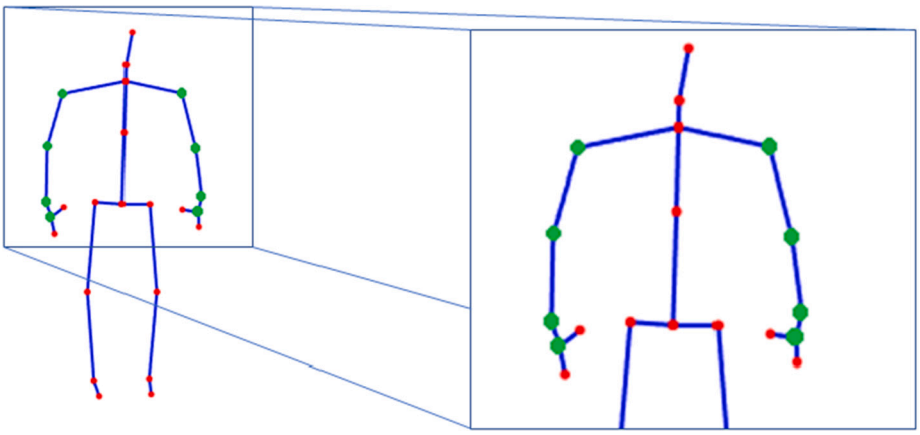

Fig. 2. Skeleton joints with highlights on the ones used for classification.

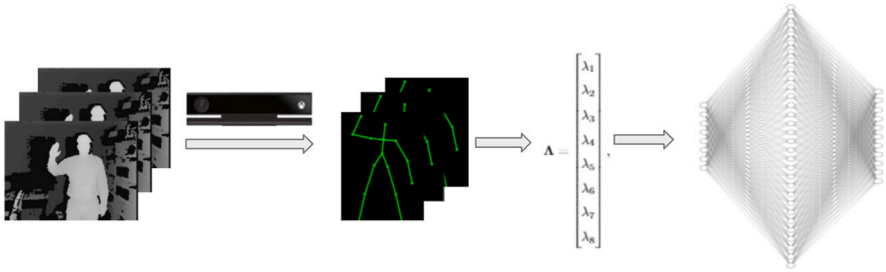

Fig. 3. The proposed Framework. First, Kinect v2 sensor captures the depth information frames and converts to skeleton data, which is a 3D spatial information. These Cartesian position descriptions are transformed into a set of features, which is later used as inputs on the Neural Network.

Table 1
Results for Simulation 1, 98.3% overall accuracy.

		Desired Classes											
		A	B	C	D	E	F	G	H	I	J	K	FN
Predicted Classes	A	100	–	–	–	–	–	–	–	–	2	–	2
	B	–	100	–	–	–	–	–	–	–	–	–	–
	C	–	–	100	–	–	–	–	–	–	–	–	–
	D	–	–	–	100	–	–	–	–	–	–	–	–
	E	–	–	–	–	100	–	–	–	–	–	1	1
	F	–	–	–	–	–	94	–	–	2	–	2	4
	G	–	–	–	–	–	–	96	–	–	–	–	–
	H	–	–	–	–	–	–	–	100	–	–	–	–
	I	–	–	–	–	–	2	–	–	98	–	2	4
	J	–	–	–	–	–	–	–	–	–	98	–	–
	K	–	–	–	–	–	4	4	–	–	–	95	7
	FP	-	-	-	-	-	4	2	-	2	-	5	98,3

6.1. Simulation 1

In this simulation 10 samples for each action class provided by two different people were used, for a total of 220 samples. This was done in order to evaluate the system's performance with few people on the dataset.

To obtain the ANN's results, 10-fold cross validation was used and the results are on Table 1.

First, it is important to highlight the application of this work. It proposes an action recognition method to be used in a social robotics application, where the user's actions will serve as commands to the agent. Neutral actions must have a higher accuracy, given that if a neutral action is mistaken by a standard action, the robot would do something when it should do nothing, rendering this application unfeasible.

When it comes to standard actions, having false negatives is something highly undesirable, because it means the user asked for a standard action and the agent executed a different one, or that the user signalized a neutral action and the system interpreted as a standard action. For neutral actions, false positives are not that bad for the system's performance, because it means that the user did a neutral action and the classifier identified as another neutral action. In this case the agent would continue not performing any tasks, or it would mean that a standard action was mistaken by a neutral action, which is not desirable, but not as damaging as the contrary.

As seen in Table 1, the proposed method had a low amount of false negatives, being most of them with class K. This is not that bad for the application since this implies that very few times the user will issue a command and the system will perform no task. Then the user will just have to repeat the command.

The overall accuracy rate was as high as 98.3%, which shows that, for two different people, the proposed method has great performance.

6.2. Simulation 2

In this simulation 10 samples for each action class provided by five different people were used, for a total of 550 samples. This was done in

Table 2
Results for Simulation 2, 99.1% overall accuracy.

		Desired Classes											
		A	B	C	D	E	F	G	H	I	J	K	FN
	A	96	–	–	–	–	–	–	–	–	–	–	–
	B	2	100	–	–	–	–	–	–	–	–	–	2
	C	–	–	100	–	–	–	–	–	–	–	–	–
	D	–	–	–	96	–	–	–	–	–	–	–	–
	E	–	–	–	–	100	–	–	–	–	–	–	–
Predicted	F	–	–	–	–	–	100	–	–	–	–	–	–
Classes	G	2	–	–	4	–	–	100	–	2	–	–	5,7
	H	–	–	–	–	–	–	–	100	–	–	–	–
	I	–	–	–	–	–	–	–	–	98	–	–	–
	J	–	–	–	–	–	–	–	–	–	100	–	–
	K	–	–	–	–	–	–	–	–	–	–	100	–
	FP	4	–	–	4	–	–	–	–	2	–	–	99,1

Table 3
Results for Simulation 3, 96.7% overall accuracy.

		Desired Classes											
		A	B	C	D	E	F	G	H	I	J	K	FN
	A	96	–	–	–	–	–	–	–	–	–	–	–
	B	4	100	–	4	–	–	–	–	–	–	–	7,5
	C	–	–	96	–	–	–	–	–	–	–	–	–
	D	–	–	–	96	4	–	8	–	–	–	–	11,2
	E	–	–	–	–	96	–	–	–	–	–	–	–
Predicted	F	–	–	–	–	–	96	–	–	–	–	–	–
Classes	G	–	–	–	–	–	–	92	–	–	–	–	–
	H	–	–	–	–	–	–	–	96	–	–	–	–
	I	–	–	–	–	–	–	–	–	96	–	–	–
	J	–	–	–	–	–	–	–	–	4	100	–	3,9
	K	–	–	4	–	–	4	–	4	–	–	100	10,8
	FP	4	-	4	4	4	4	8	4	4	-	1	96,7

order to evaluate how the system's performance would scale for more people in the dataset.

To obtain the ANN's results, 10-fold cross validation was used. The results are on Table 2

Comparing Tables 1 and 2 it is possible to see a decrease in overall false negatives and a minor increase in standard actions false positives, not high enough to hinder the system's performance, achieving an overall accuracy rate of 99.1%, which suggests that increasing from 2 to 5 people in the dataset improved the system's generalization capabilities.

6.3. Simulation 3

In this simulation 5 samples for each action class provided by five different people were used, for a total of 225 samples and the resulting ANN had its performance tested on the other 5 samples that were not used for training. This was done to evaluate the system's accuracy training with such a tiny dataset.

To obtain the ANN's results, 10-fold cross validation was used. The results are on Table 3

Analyzing Tables 2 e 3 it is possible to see that the system's performance using the whole dataset and just half of it was only 2.4% apart. The results on Table 3 show that there was a low amount of false positive and negatives and a significant part of the false positives were on the neutral class K, implying that these mistakes will not lead to the agent performing an incorrect or not issued command; it will, at worse, not perform the desired task on the first command. With an accuracy as high as 96.7%, the proposed method shows that it has great performance with a tiny dataset.

7. Online tests

For the online experiments, two important factors must be highlighted. First, the experiments were done using the ANN trained on the full dataset. Second, a Human–Robot Interaction scenario requires that the agent must be able to identify the users actions as soon as they are done. With that in mind, a manual trigger was implemented intending to not allow the action recognition system to execute uninterrupted, leading to possibly wrong classifications. So in order to signalize to the system that the user is going to begin an action, he must first open one of his hands. This way, the system understands that it must start storing features for classification and it does so for the capture window of 1.66 s.

7.1. Experiment 1

This experiment intends to guarantee that the used dataset is not biased in any way, invalidating the great simulation results obtained.

This was done by taking two people that are part of the dataset samples used for training, and having them performing each of the action classes 30 times. In order to avoid that, during the experiment, a user could learn how to correct the mistakes in recent repetitions in case of having to do all 30 of them in sequence, a random order lists was generated. This way the users did not know what was the order they would have to perform the actions beforehand. The results are on Table 4.

With the results in Table 4, is clear that the proposed method is reliable for practical applications given the overall accuracy rate of 98%. With few false negatives, where most of them were regarded to the neutral classes J and K, which will lead to a situation where the user issues a command and it will mistake with a neutral action, so the system will not perform an action.

7.2. Experiment 2

In order to showcase in a playful way a simple application of the proposed algorithm, three TicTacToe games were played between two people where nine standard actions represent the nine available spots. To represent the end of each match, the winner must do the same action twice more to reset the robots position.

Fig. 4. Action Zone, where the user issues his actions; Game Zone, where the TicTacToe game occurs; and the standard action classes for each of the nine spots in the game.

Table 4
Results for Experiment 1, 98% overall accuracy.

		Desired Classes											
		A	B	C	D	E	F	G	H	I	J	K	FN
Predicted Classes	A	100	5	–	–	–	–	6,7	–	–	–	–	10
	B	–	91,6	–	–	–	–	–	–	–	–	–	–
	C	–	–	96,7	–	–	–	–	–	–	–	1,7	1,7
	D	–	–	–	100	–	–	–	–	–	–	–	–
	E	–	–	–	–	98,3	–	–	–	–	–	–	–
	F	–	–	–	–	–	100	–	–	–	–	–	–
	G	–	–	–	–	–	–	93,3	–	–	–	–	–
	H	–	–	–	–	1,7	–	–	100	–	–	–	1,7
	I	–	1,6	–	–	–	–	–	–	100	–	–	1,6
	J	–	–	–	–	–	–	–	–	–	100	–	–
	K	–	1,8	3,3	–	–	–	–	–	–	–	98,3	5
	FP	–	8,4	3,3	–	1,7	–	6,7	–	–	–	40	98

The actions for each spot are shown on Fig. 4 and the experimental video can be seen at https://youtu.be/NVXGA58JhtM. In the video, it is possible to observe a flawless use of the classification algorithm, with two different people, on an uncut footage of three TicTacToe games. For better view experience, a description is done to show which match is currently starting and several parts of the video were hastened. None of the actions were mistaken even when one had to repeat them to signal the end of a game. This shows high reliability on the algorithm's accuracy.

8. Concluding remarks and future works

In this paper it is proposed an easy to implement action classification method using ANN based on human skeleton features for social assistive robotics application. The method relies on a high dimension reduction in order to require a smaller dataset to have the ANN learn the patterns of each action. In our case, we demonstrate that only 5 to 10 samples of each class for each person is enough for getting accuracy ranging from 96.7% to 99.1%. The results show a high training accuracy also for unseen actions, furthermore it shows that up to five different skeletons in the dataset does not dim its performance, making the proposed algorithm reliable, requiring a small size dataset to be effective.

The developed algorithm could lead to a flexible and easy implementation of a human–robot interface using action recognition due to its simplicity and the fact that it does not rely on big external datasets for its training. The small size of the used neural network favored a quick application time for the action classification

Furthermore, the Social Assisitve Robotics concepts used as starting point in this paper were playfully demonstrated in a TicTacToe game, showing that it room for robotics educational applications.

For future works, we intent to validate our proposal in a more challenge scenario of Social Assistive Robotics. Specifically, we aim to help common users in monotony or dangerous activities, where they can interact with robot using actions. Some of these tasks are carrying books in a library, taking medicines in a hospital, transporting a radioactive object from a contaminated environment, and so on. All of them involves sub-tasks that require the robot to follow, stop close to, and carry weight for the user, as well be capable of going to predetermined places and return to a home spot.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, a Brazilian agency that supports scientific and technological development, through a Master's program scholarship to the first author.

References

Ackovska, N., & Kirandziska, V. (2017). The importance of hands-on experiences in robotics courses. In *IEEE EUROCON 2017-17th international conference on smart technologies* (pp. 56–61). IEEE.

Agahian, S., Negin, F., & Köse, C. (2019). An efficient human action recognition framework with pose-based spatiotemporal features. *Engineering Science and Technology, An International Journal*, http://dx.doi.org/10.1016/j.jestch.2019.04.014.

Aggarwal, J. K., & Ryoo, M. S. (2011). Human activity analysis: A review. *ACM Computing Surveys*, *43*(3), 16. http://dx.doi.org/10.1145/1922649.1922653.

Aggarwal, J. K., & Xia, L. (2014). Human activity recognition from 3d data: A review. *Pattern Recognition Letters*, *48*, 70–80. http://dx.doi.org/10.1016/j.patrec.2014.04.011.

Akula, A., Shah, A. K., & Ghosh, R. (2018). Deep learning approach for human action recognition in infrared images. *Cognitive Systems Research*, *50*, 146–154.

Barnachon, M., Bouakaz, S., Boufama, B., & Guillou, E. (2014). Ongoing human action recognition with motion capture. *Pattern Recognition*, *47*(1), 238–247. http://dx.doi.org/10.1016/j.patcog.2013.06.020.

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, *58*(3), 978–988. http://dx.doi.org/10.1016/j.compedu.2011.10.006.

Buonamente, M., Dindo, H., & Johnsson, M. (2016). Hierarchies of self-organizing maps for action recognition. *Cognitive Systems Research*, *39*, 33–41.

Burger, B., Ferrané, I., Lerasle, F., & Infantes, G. (2012). Two-handed gesture recognition and fusion with speech to command a robot. *Autonomous Robots*, *32*(2), 129–147.

Celebi, S., Aydin, A. S., Temiz, T. T., & Arici, T. (2013). Gesture recognition using skeleton data with weighted dynamic time warping. In *VISAPP (1)* (pp. 620–625). http://dx.doi.org/10.5220/0004217606200625.

Chen, B., Hua, C., Dai, B., He, Y., & Han, J. (2019). Online control programming algorithm for human–robot interaction system with a novel real-time human gesture recognition method. *International Journal of Advanced Robotic Systems*, *16*(4), 1–18. http://dx.doi.org/10.1177/1729881419861764.

Du, Y., Wang, W., & Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1110–1118). http://dx.doi.org/10.1109/CVPR.2015.7298714.

Eriksson, J. (2004). Hands-off robotics for post-stroke arm rehabilitation. *Technical Report*.

Feil-Seifer, D., & Mataric, M. J. (2005). Defining socially assistive robotics. In *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.* (pp. 465–468). IEEE, http://dx.doi.org/10.1109/ICORR.2005.1501143.

Fong, T., Nourbakhsh, I., & Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and Autonomous Systems*, *42*(3–4), 143–166. http://dx.doi.org/10.1016/S0921-8890(02)00372-X.

Gharaee, Z. (2020). Hierarchical growing grid networks for skeleton based action recognition. *Cognitive Systems Research*.

Hang, C., Zhang, R., Chen, Z., Li, C., & Li, Z. (2017). Dynamic gesture recognition method based on improved dtw algorithm. In *2017 international conference on industrial informatics-computing technology, intelligent technology, industrial information integration (ICIICII)* (pp. 71–74). IEEE, http://dx.doi.org/10.1109/ICIICII.2017.17.

Hansen, A., Hees, F., & Jeschke, S. (2013). Hands on robotics-concept of a student laboratory on the basis of an experience-oriented learning model. In *Automation, communication and cybernetics in science and engineering 2011/2012* (pp. 325–339). Springer.

Huttenrauch, H., & Eklundh, K. S. (2002). Fetch-and-carry with CERO: Observations from a long-term user study with a service robot. In *Proceedings. 11th ieee international workshop on robot and human interactive communication* (pp. 158–163). IEEE, http://dx.doi.org/10.1109/ROMAN.2002.1045615.

Jara, C. A., Candelas, F. A., Puente, S. T., & Torres, F. (2011). Hands-on experiences of undergraduate students in automatics and robotics using a virtual and remote laboratory. *Computers & Education*, *57*(4), 2451–2461.

Kahn, L. E., Averbuch, M., Rymer, W. Z., & Reinkensmeyer, D. J. (2001). Comparison of robot-assisted reaching to free reaching in promoting recovery from chronic stroke. In *Proceedings of the international conference on rehabilitation robotics* (pp. 39–44). IOS Press.

Kanda, T., Ishiguro, H., Imai, M., & Ono, T. (2004). Development and evaluation of interactive humanoid robots. *Proceedings of the IEEE*, *92*(11), 1839–1850. http://dx.doi.org/10.1109/JPROC.2004.835359.

Kumar, P., Gauba, H., Roy, P. P., & Dogra, D. P. (2017). Coupled HMM-based multi-sensor data fusion for sign language recognition. *Pattern Recognition Letters*, *86*, 1–8. http://dx.doi.org/10.1016/j.patrec.2016.12.004.

Li, C., Hou, Y., Wang, P., & Li, W. (2017). Joint distance maps based action recognition with convolutional neural networks. *IEEE Signal Processing Letters*, *24*(5), 624–628. http://dx.doi.org/10.1109/LSP.2017.2678539.

Li, X., Zhang, Y., & Liao, D. (2017). Mining key skeleton poses with latent svm for action recognition. *Applied Computational Intelligence and Soft Computing*, *2017*, http://dx.doi.org/10.1155/2017/5861435.

Li, C., Zhong, Q., Xie, D., & Pu, S. (2017). Skeleton-based action recognition with convolutional neural networks. In *2017 IEEE international conference on multimedia & expo workshops* (pp. 597–600). IEEE.

Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.-Y., & Kot, A. C. (2020). Ntu rgb+d 120: A large-scale benchmark for 3D human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(10), 2684–2701.

Liu, R., & Zhang, X. (2019). A review of methodologies for natural-language-facilitated human–robot cooperation. *International Journal of Advanced Robotic Systems*, *16*(3), 1–17. http://dx.doi.org/10.1177/1729881419851402.

Mitra, S., & Acharya, T. (2007). Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *37*(3), 311–324. http://dx.doi.org/10.1109/TSMCC.2007.893280.

Moeslund, T. B., Hilton, A., & Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, *104*(2–3), 90–126. http://dx.doi.org/10.1016/j.cviu.2006.08.002.

Montemerlo, M., Pineau, J., Roy, N., Thrun, S., & Verma, V. (2002). Experiences with a mobile robotic guide for the elderly. *AAAI/IAAI*, *2002*, 587–592.

Nagi, J., Ducatelle, F., Di Caro, G. A., Cireşan, D., Meier, U., Giusti, A., et al. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *2011 IEEE international conference on signal and image processing applications* (pp. 342–347). IEEE.

Ng, C. W., & Ranganath, S. (2002). Real-time gesture recognition system and application. *Image and Vision Computing*, *20*(13–14), 993–1007. http://dx.doi.org/10.1016/S0262-8856(02)00113-0.

Ordóñez, F., & Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, *16*(1), 115. http://dx.doi.org/10.3390/s16010115.

Patrona, F., Chatzitofis, A., Zarpalas, D., & Daras, P. (2018). Motion analysis: Action detection, recognition and evaluation based on motion capture data. *Pattern Recognition*, *76*, 612–622. http://dx.doi.org/10.1016/j.patcog.2017.12.007.

Raheja, J., Minhas, M., Prashanth, D., Shah, T., & Chaudhary, A. (2015). Robust gesture recognition using kinect: A comparison between DTW and HMM. *Optik*, *126*(11–12), 1098–1104. http://dx.doi.org/10.1016/j.ijleo.2015.02.043.

Sempena, S., Maulidevi, N. U., & Aryan, P. R. (2011). Human action recognition using dynamic time warping. In *Proceedings of the 2011 international conference on electrical engineering and informatics* (pp. 1–5). IEEE, http://dx.doi.org/10.1109/ICEEI.2011.6021605.

Shahroudy, A., Liu, J., Ng, T.-T., & Wang, G. (2016). Ntu rgb+d: A large scale dataset for 3D human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1010–1019).

Tapus, A., Maja, M., & Scassellatti, B. (2007). The grand challenges in socially assistive robotics. *IEEE Robotics & Automation Magazine*, *14*(1), N–A.

Thobbi, A., & Sheng, W. (2010). Imitation learning of hand gestures and its evaluation for humanoid robots. In *The 2010 IEEE international conference on information and automation* (pp. 60–65). IEEE, http://dx.doi.org/10.1109/ICINFA.2010.5512333.

Tsardoulias, E. G., Kintsakis, A. M., Panayiotou, K., Thallas, A. G., Reppou, S. E., Karagiannis, G. G., et al. (2017). Towards an integrated robotics architecture for social inclusion–the RAPP paradigm. *Cognitive Systems Research*, *43*, 157–173.

Veeriah, V., Zhuang, N., & Qi, G.-J. (2015). Differential recurrent neural networks for action recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 4041–4049). http://dx.doi.org/10.1109/ICCV.2015.460.

Wang, P., Li, W., Gao, Z., Tang, C., & Ogunbona, P. O. (2018). Depth pooling based large-scale 3-d action recognition with convolutional neural networks. *IEEE Transactions on Multimedia*, *20*(5), 1051–1061. http://dx.doi.org/10.1109/TMM.2018.2818329.

Wang, P., Li, W., Gao, Z., Zhang, J., Tang, C., & Ogunbona, P. O. (2015). Action recognition from depth maps using deep convolutional neural networks. *IEEE Transactions on Human-Machine Systems*, *46*(4), 498–509. http://dx.doi.org/10.1109/THMS.2015.2504550.

Wilson, G., Pereyda, C., Raghunath, N., de la Cruz, G., Goel, S., Nesaei, S., et al. (2019). Robot-enabled support of daily activities in smart home environments. *Cognitive Systems Research*, *54*, 258–272.

Xu, D., Wu, X., Chen, Y.-L., & Xu, Y. (2015). Online dynamic gesture recognition for human robot interaction. *Journal of Intelligent and Robotic Systems*, *77*(3), 583–596.

Yan, S., Xiong, Y., & Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence* (pp. 7444–7452).

Yang, Y., Shen, S., Lui, K., Lee, K., Chen, J., Ding, H., et al. (2017). Ultrasonic robotic system for noncontact small object manipulation based on kinect gesture control. *International Journal of Advanced Robotic Systems*, *14*(6), 1–7. http://dx.doi.org/10.1177/1729881417738739.

Zhang, X.-H., Wang, J.-J., Wang, X., & Ma, X.-L. (2016). Improvement of dynamic hand gesture recognition based on HMM algorithm. In *2016 international conference on information system and artificial intelligence* (pp. 401–406). IEEE, http://dx.doi.org/10.1109/ISAI.2016.0091.