


Article

Computer Vision Intelligent Approaches to Extract Human Pose and Its Activity from Image Sequences

Paulo J. S. Gonçalves ^{1,*} , Bernardo Lourenço ¹, Samuel Santos ¹, Rodolphe Barlogis ² and Alexandre Misson ²

¹ IDMEC, Instituto Politécnico de Castelo Branco, Av Empresário, 6000-767 Castelo branco, Portugal; bernardolourenco@yahoo.com (B.L.); sam358@live.com (S.S.)

² IMT Mines Alès, 6 Avenue de Clavières, 30100 Alès, France; rodolphe.barlogis@mines-ales.org (R.B.); alexandre.misson@mines-ales.org (A.M.)

* Correspondence: paulo.goncalves@ipcb.pt

Received: 13 December 2019; Accepted: 11 January 2020; Published: 15 January 2020



Abstract: The purpose of this work is to develop computational intelligence models based on neural networks (NN), fuzzy models (FM), support vector machines (SVM) and long short-term memory networks (LSTM) to predict human pose and activity from image sequences, based on computer vision approaches to gather the required features. To obtain the human pose semantics (output classes), based on a set of 3D points that describe the human body model (the input variables of the predictive model), prediction models were obtained from the acquired data, for example, video images. In the same way, to predict the semantics of the atomic activities that compose an activity, based again in the human body model extracted at each video frame, prediction models were learned using LSTM networks. In both cases the best learned models were implemented in an application to test the systems. The SVM model obtained 95.97% of correct classification of the six different human poses tackled in this work, during tests in different situations from the training phase. The implemented LSTM learned model achieved an overall accuracy of 88%, during tests in different situations from the training phase. These results demonstrate the validity of both approaches to predict human pose and activity from image sequences. Moreover, the system is capable of obtaining the atomic activities and quantifying the time interval in which each activity takes place.

Keywords: computer vision; human pose estimation; human activity estimation; deep learning; neural network; fuzzy modelling; support vector machines

1. Introduction

Humans and machines are deemed to work and live together in the years to come. Technology has advanced significantly in past years, and machines are part of our lives at home, with transport and also in the work environment. Examples of such technologies are included in smart homes [1], in autonomous vehicles [2] and, for example, in the use of collaborative robots in industry [3] and in office buildings [4] or homes [5]. In this sense, human activity monitoring in these technological environments is crucial to ensure safety in these human robot interactions, as described in Reference [6] for the industrial case. Related works can be found in the review paper in Reference [7] on relevant environments for safe human robot interaction. From the four safety pillars highlighted?control, motion planning, prediction, psychological consideration, the approaches proposed in this paper are related to prediction. The prediction aims to ensure a proper safe interaction with machines, that is, by allowing robots and other agents to infer and predict human pose and activity.

Human Pose and Activity Monitoring Systems are crucial to achieve a proper integration of machines in the human environment and moreover to ensure that external agents can access in real

time the state of the human. This is important to check, for example, if an older adult passed out or fell at home. Other application scenarios are, for example, to monitor a physical exercise that a medical doctor prescribed to his/her patient. This example is depicted further in the paper as a use case application of the proposed approaches. To evaluate the performance or even to verify whether the human is performing an activity, it is necessary to collect data that allow a system—in the case of this paper, an intelligent system—to measure the type of pose or activity of the human.

Several types of sensors can be used to obtain data to infer pose or human activity. 3D information can be obtained from the environment to clearly obtain real 3D metric values for pose. Other sensors, for example, wearable sensors, smartphone embedded sensors and so forth, give information about events that can occur after post-processing of accelerometers or physiological sensors, like ECG or respiratory. Considering the smartphone in another level, information on its use on social networks can allow the detection of events and detect anomalies or disasters based on multimedia contents [8]. In the following, an analysis will be performed on the state-of-the-art of such systems, relating them to the proposed approaches in this paper.

Starting with 3D information, in the review paper in Reference [9] are presented different methods to obtain 3D Data for human activity recognition, using Mocaps <http://mocap.cs.cmu.edu/>, using multiple views obtain the 3D information [10] and in recent years RGB-D cameras. The first approach based in Mocaps have the disadvantage of the apparatus that need to be attached to the human, for example, inertial sensors and/or markers for optical measurements. The advantage of such systems is that they have a very good accuracy, where the optical mocap can be applied to high accuracy demanding systems, such as medical surgery [11]. Such millimetre accuracy is not needed for monitoring humans in its daily life activities at home, as tackled in this paper. Moreover, such systems are very expensive and not adequate to attach to humans in its common daily life activities. The second approach estimates the 3D information using computer vision techniques, based on multiple views of the target object, in this case the human. This method estimates the depth information from those views and relies on the solution of a very non-linear system of equations, that can hamper the global solution in local minima. In recent years and with the advent of depth cameras, the depth information can be calculated from this piece of hardware, avoiding some complex computational frameworks and the attachment of inertial sensors or markers to the human body. As such, in this research work, an RGB-D camera is used to obtain the required 3D data from image sequences, making the system more stable than the 3D estimation from multiple views and cheaper than the mocap solutions.

From RGB-D cameras, 3D data of the human pose can be obtained, along with the 3D occupancy map, as presented in Reference [9]. Moreover, Kinect and other RGB-D cameras have software libraries that allow to obtain directly the human 3D model, based on some key 3D points of the skeleton. Recently, developments by the research community allowed the obtaining of the 3D pose (body model) of a human, using the OpenPose [12] software library. This is of special interest because, from 2D data from a single camera, over a sequence of frames, the 3D human body model can be obtained.

From the set of points obtained, no semantic meaning is associated to each pose. To the best of our knowledge there is no intelligent system that can estimate the semantic meaning of the pose, that is, in natural language, only based on visual features. That is one contribution of the paper to classify each human pose directly into its semantic meaning, for example, lying down, sitting. For this purpose, computational intelligence techniques, for example, fuzzy models, support vector machines, neural networks, will be compared to obtain the best model to be implemented into a software application to run at a robot. This robot is set to monitor a human at home and the pose of the human is important to infer its health status and to detect potentially dangerous poses of the human. Other approaches rely on wearable sensors, smartphone embedded sensors, that directly can estimate the human activity.

Human activity recognition has been an active field of research, where several sensors are used. In Reference [13] a survey is presented on the use of wearable sensors, such as accelerometer, GPS, heart monitor, electrocardiogram (ECG), light sensor, thermometer and so forth, integrated in smartphones and similar electronic devices. Those sensors when combined with proper intelligent systems (support

vector machines) to obtain accuracy close to 97% for simple activities, for example, ambulation. In Reference [14] results are presented for video data but a sequence of states must be known a-priori to detect the activity.

Smartphones and its embedded sensors are also used in this type of research work, due to its generalised availability and ease of programming, as, for example, the work in References [15,16], which present an overview of different approaches. Those approaches also rely on intelligent systems to classify the human activity and several algorithms were used, being support vector machines, Bayesian approaches and neural networks the most applied. However, it is also shown in the overview paper in Reference [16] that the use of recurrent neural networks with LSTM performs better than traditional solutions.

Deep learning was applied with Smartphones in Reference [17,18], which obtained again better results than Support Vector Machines. Other approaches that use deep learning techniques were recently proposed [19], which evaluate Support Vector Machine models, Convolutional Neural Networks, LSTMs and Semisupervised Recurrent Convolutional Network, with the use multimodal wearable sensors. The later model obtained the best accuracy. However, no image data was used in this work. The use of video images is important because it is not an intrusive way for the human, that is, he/she does not have to wear any type of sensors in the body or cloth. In all the described works, the activity recognition is based on the main activity and not on atomic activities, with the exception of the work in Reference [17], which uses smartphone sensors.

In the literature, several examples do exist for activity recognition that use probability-based algorithms to build activity models, for example, in Reference [14] and references therein. Hidden Markov model and the conditional random field have been extensively studied by the research community. Reference [14] proposed probabilistic based activity models from video data, although without specification the numerical quality of the models. Moreover, models for each atomic activity are to be learned. This same issue also appears in the work that combines Convolutional Neural Networks and the LSTM model presented in Reference [20], which uses a 3D point model of the human body obtained from the Kinect video sensor, from Microsoft.

In our approach to activity recognition, an LSTM architecture was proposed with a unique model to be learned to obtain the atomic activity predictions, with the great advantage that no a-priori knowledge of the number of states is needed. Moreover, quantitative metrics are presented to validate the learned models. The approach uses video images and no other types of information such as wearable sensors. It has the advantage of the person not having to use wearables, but a camera is needed to look at the person's activity. This is a limitation of the proposed system, because it can only be used when the human is within the field of view of a video camera.

Existing methods only detail the type of activity and cannot measure, based on video images, the time of different phases of the activity. This allows to monitor the performance of humans physical activity. The system is also appropriate to measure the performance of some test exercises prescribed by a Medical Doctor, such as the one presented in Section 3.2. To validate the second system a physical activity exercise that the robotic system can monitor was used. This monitoring is used to test the quality of the performance of the human while doing the exercise.

Recently, novel approaches based on multimedia data [21] were proposed that can detect important relations between objects and defined topics of interest within online social networks, while persons perform activities. This approach can then summarize such relations and be used to detect and manage events [8] such as emergencies and disasters in real life applications. This new trend, associated with the massive usage by persons of online social networks, can be used to track the behavior of persons while performing activities and spread alerts over social networks, for example.

In the proposed approaches for pose and activity recognition, an RGB-D camera is to be used to obtain the required 3D data from image sequences, making the system more stable than the 3D estimation from multiple views and cheaper than the mocap solutions, where the sensors should be attached to the human. By using cameras, the sensor does not need to be attached to the human body,

which is an advantage of the proposed approach, although the human should be visible by the camera. The proposed system for pose recognition have the following advantage related to the state-of-the-art: is able to estimate the semantic meaning of the pose, that is, in natural language, only based on visual features. Within the paper, will be presented the intelligent model that will be more robust for the application tackled in this research work, for example, to monitor the human pose within its home, along with human activity. For the activity recognition, a LSTM architecture will be proposed with a unique learned model to obtain the atomic activity predictions, with the great advantage that no a-priori knowledge of the number of states is needed. Moreover, all the system is based on visual features. The proposed methods, will be validated in real scenarios, where the data was captured from the cameras. The learned models will be implemented in a robotic system to monitor human pose and activity in a home environment.

The paper is organized as follows. Section 2 presents the theoretical framework behind the models that will be learned, compared and then implemented in test scenarios. In Section 3 are presented the developed applications for human pose and activity detection and recognition. Next section presents the results for training and testing, in real scenarios, along with its discussion. The paper ends with conclusions and future work proposals.

2. Theoretical Framework

2.1. Classical Computational Intelligence Based Models

In the general case, data is organized in inputs and outputs of the system to be modelled. The input data is defined by a vector, $x = [x_1 \ x_2 \ \dots \ x_n]^T$, where n , is the number of inputs. The output data is defined by a vector $y = [y_1 \ y_2 \ \dots \ y_m]^T$, where m , is the number of outputs. From the learning approaches depicted in the following sub-sections, several models, $y = F(x)$, will be obtained to solve the stated classification problem. Depending on the learning approaches, Multiple Input Multiple Output (MIMO) models or several Multiple Input Single Output (MISO) models, will be obtained for the classification of human poses and activity, the latter enhancing the atomic activities that compose the general activity.

2.1.1. Support Vector Machines

The Support Vector Machine (SVM) [22] maps an input vector into a high-dimensional descriptor space through some nonlinear mapping, chosen a priori. In this space, an optimal separating hyperplane is constructed. SVM classification was applied for the human pose estimation case and the cubic epsilon loss insensitive optimization technique was used, where the Gaussian kernel's bandwidth have to be set, along with the Lagrangian multipliers [22]. The cubic function was chosen, in this work, as the best SVM contribution to classifiers comparison, based on preliminary results obtained from the gathered data. The purpose of the method is to obtain the cubic function,

$$f(x) = (x^T \cdot \beta + b)^3 \quad (1)$$

This function is obtained by formulating a convex optimization problem to find $f(x)$ with minimal value for $\|\beta^T \cdot \beta\|$, with the following stopping criteria for the residuals: $\forall_n : |y_n - (x^T \cdot \beta + b)^3| \leq \epsilon$. Following Reference [22], the goal is to minimize the following Lagrangian equation, where nonnegative multipliers α_k , and α_k^* , were introduced for each data sample k , from the N available.

$$L(\alpha) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) x_i^T \cdot x_j + \epsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) x_i^T \cdot x_j + \sum_{i=1}^N y_i (\alpha_i^* - \alpha_i) \quad (2)$$

Constrained with:

$$\begin{aligned} \sum_{i=1}^N (\alpha_n - \alpha_n^*) &= 0 \\ \forall_n : 0 &\leq \alpha_k \leq C \\ \forall_n : 0 &\leq \alpha_k^* \leq C \end{aligned} \quad (3)$$

2.1.2. Fuzzy Models

From the modelling techniques based on soft computing, fuzzy modelling [23] is one of the most appealing. If no a priori knowledge is available, the rules and membership functions can be directly extracted from process measurement. Fuzzy models can provide a transparent description of the system, which can reflect the nonlinearities of the system. This paper uses Takagi-Sugeno fuzzy models [24] where the consequents are crisp functions of the antecedent variables.

Different classes of fuzzy clustering algorithms can be used to approximate a set of data by local models. From the available clustering algorithms, subtractive clustering (SC) [25] was used in this work.

Fuzzy Inference systems are if-then rule based and each rule, which is associated with the number of data clusters, K . Rules, R_i , have antecedents, associated to fuzzy sets, A_{ij} and consequents, B_i , Equation (4):

$$\begin{aligned} R_i : \text{ If } x_1 \text{ is } A_{i1} \text{ and } \dots \text{ and } x_n \text{ is } A_{in} \\ \text{ then } y_i = B_i, i = 1, 2, \dots, K. \end{aligned} \quad (4)$$

The number of rules, K , and the antecedent fuzzy sets A_{ij} are determined by means of fuzzy clustering in the product space of the inputs and the outputs [26]. To obtain each estimated output, \hat{y} , Equations (5) and (6) were used to average the contribution of each rule, where β_i is the degree of activation of each rule and $\mu_{A_{ij}}$ is the membership function of each fuzzy set A_{ij} .

$$\hat{y} = \frac{\sum_{i=1}^K \beta_i y_i}{\sum_{i=1}^K w_i \beta_i}, \quad (5)$$

$$\beta_i = \prod_{j=1}^n \mu_{A_{ij}}(x_j), \quad i = 1, 2, \dots, K, \quad (6)$$

2.1.3. Neural Networks

Neural Networks (NN) [27] are based on the interconnection between a set of simple processing units (neurons). Each of these neurons contains a linear or nonlinear transformation function. The connections between the neurons have an associated weight that must be trained in order to adjust the performance of the network to the purpose of its use.

NN are flexible classification systems that are easily trained using backpropagation. By repeatedly showing a neural network inputs classified into groups, the network can be trained to discern the criteria used to classify and it can do so in a generalized manner allowing successful classification of new inputs not used during training [28]. Each NN is characterized by the number of its layers, the units that compose each one of these layers, the interconnections between the units composing each layer and the ones that compose the following layer and all the associated weights.

The probabilistic neural network was first presented by Reference [29] and is a type of feedforward neural network, obtained from the Bayes Net and the Kernel Fisher discriminant analysis, as presented in the seminal paper. This NN is specially designed for classification problems. The network architecture have three layers: the input, the radial basis and the competitive layers. The input layer calculates the distances, by a hidden neuron, from the input data to the training data set. These values are then sent to the radial basis functions, with a given spread, and a vector of probabilities is obtained. In the competitive layer, is chosen the large probability to define the main plant genus class. Figure 1 depicts the PNN architecture, previously presented, and the main mathematical equations, showing the number of inputs, outputs and radial based functions obtained for the network trained with the data studied in this paper, for human pose detection.

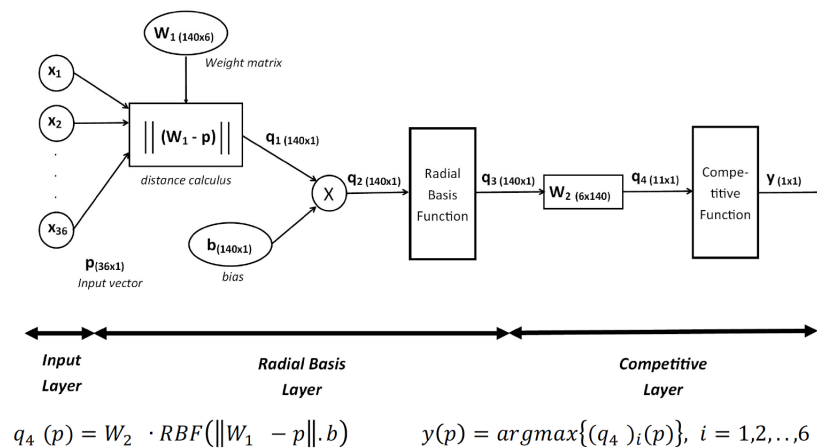


Figure 1. Probabilistic Neural Network architecture, depicting 140 neurons in the radial basis layer, the weights W_1 and W_2 , the bias b , for 36 inputs (the 2D coordinates of the 18 nodes human body model) and the output class (the 6 possible classes of human poses).

2.2. Deep Learning Based Model-LSTM

In this sub-section is presented the deep learning based model LSTM and the architecture developed to obtain the prediction of the human activities. The architecture is sequential and comprises two LSTM layers and a final feedforward neural network, depicted in Figure 2.

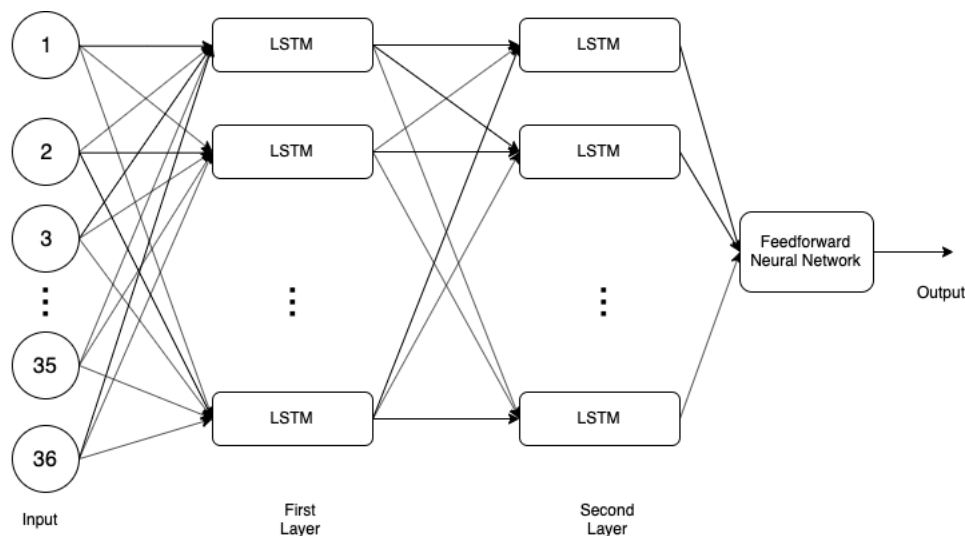


Figure 2. Deep learning architecture, depicting two LSTM layers with a dropout of 0.2, 100 and 50 neurons in the first and the second respectively and a feedforward network with 30 and 10 neurons in each layer, for 36 inputs (the 2D coordinates of the 18 nodes human body model) and the output (a 3×1 vector for the 3 possible classifications of human poses).

The LSTM [30] has complex dynamics that allow it to memorize information for several timesteps, which is essential to model activities. The *long term* memory is stored in a vector of *memory cells* $c_t \in \mathbb{R}^n$, where n is the number of cells/neurons. Several LSTM architectures have been proposed over the years to tackle different problems. A review was recently published in Reference [31]. Many differ on the architecture, for example, the connectivity structure and activation functions. However, all LSTM architectures have explicit memory cells for storing information for long periods of time, which is the main essence of this deep learning model and one of the basis of the large scope of its applications.

The LSTM can decide to overwrite the memory cell, retrieve it, or keep it for the next time step, depending on the learned activation vector forget gate. The LSTM architecture used in the work presented in this paper follows the LSTM cell Equations (7), that are also depicted in Figure 3.

$$\begin{aligned}
 i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
 f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
 c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
 o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
 h_t &= o_t \tanh(c_t)
 \end{aligned}
 \tag{7}$$

where:

$x_t \in \mathbb{R}^n$: input vector to the LSTM unit

$f_t \in \mathbb{R}^h$: forget gate's activation vector

$i_t \in \mathbb{R}^h$: input/update gate's activation vector

$o_t \in \mathbb{R}^h$: output gate's activation vector

$h_t \in \mathbb{R}^h$: hidden state vector also known as output vector of the LSTM unit

$c_t \in \mathbb{R}^h$: cell state vector

$W \in \mathbb{R}^{h \times n}$, $U \in \mathbb{R}^{h \times h}$ and $b \in \mathbb{R}^h$ are weight matrices and bias vector parameters which need to be learned during training where the superscripts n and h refer to the number of input features and number of cells/neurons, respectively.

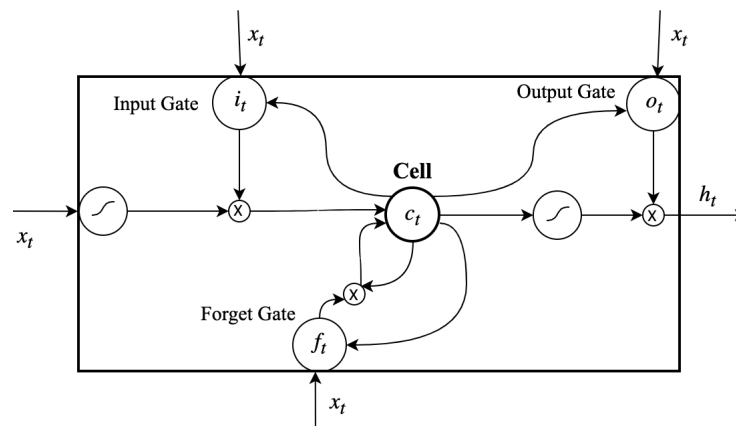


Figure 3. Graphical representation of a long short-term memory cell.

2.3. Advantages and Disadvantages of the Computational Intelligence Models

In this sub-section are presented the advantages and disadvantages of using the computational methods presented in Sections 2.1 and 2.2.

Support vector machines is a method that delivers a unique solution, since the problem is convex. This an important advantage related to neural networks, that can have several solutions related to local minima. SVM also produce robust models with a proper chose of the kernel, in this paper a polynomial, that can generalize even if the data is biased. Moreover, to avoid over-fitting the regularization kernel can be used. Since it is a non-parametric method, SVM have a lack on the transparency of results. Moreover, the theory covers the determination of the parameters for a given kernel.

The inferred results from fuzzy models can be easily interpreted because the nature of the learned fuzzy rules, in the case of the paper, Sugeno type. Another advantage of such models is that they can handle inherent uncertainties of the domain knowledge and are also robust to possible disturbance on the input/output data. As drawbacks, these learned models are highly dependent on the trained data, from which the fuzzy model rules are obtained. As such, the presented fuzzy models cannot generalize and new rules need to be added, needing a new learning phase.

Neural Networks have the disadvantage of the results interpretation, for the same reasons as the SVM models. Moreover, it is not straightforward to identify the proper number of layers and neurons for each dataset. As advantages, neural networks can generalize very well, are robust do disturbances on the data and changes in the modelled process.

For sequential datasets, classical neural networks have the drawback of not taking into account memory. LSTM models are specially developed to tackle this issue by introducing short term and long term memory modules. Moreover, are capable to tackle long time lags, and noisy data. As a special feature, related to state automata or hidden markov models, LSTM does not need a-priori knowledge of the number of states. As a drawback, they require a large amount of data, in the training phase, where existing data is often inflated. Also LSTM requires more memory and training time, than the classical methods in Section 2.1, due to the memory feature.

3. Developed Applications

In this section will be presented the developed applications that implement the pose classification and the human activity recognition systems. The first application implements the classical computational intelligence based models, presented in Section 2.1. The second application implements the deep learning Long Short-Term Memory (LSTM) artificial neural network architecture, presented in Section 2.2.

The human poses are obtained directly from the images captured from an image capturing device. In this work, a RGB-D camera (Intel RealSense D415) was used, depicted in Figure 4. From this device, the images are sent to the OpenPose [12] to obtain a set of points that model the human pose [12]. The 18 nodes body model was chosen and used, after preliminary tests, because it outputs the needed features to classify the human daily life pose, at home. In Figure 5 are presented the points (large ones with several colours) of the human model for a human pose.



Figure 4. The RGB-D camera used in image acquisition located at a mobile robot manipulator.

The human pose was obtained using an open-source computational framework widely spread in the research community with results that suits the developed applications, it is OpenPose [12]. Openpose is the first real-time pose detection system for one or more people and detects a set of key points (nodes) in the human body, face and hands. In the paper was used the model of the body and not the model for hands and face that OpenPose also delivers, because it was not needed for the proposed solutions. In Figure 5, are depicted the constituent nodes of the human body containing a total of 18 nodes, being the same ones that are part of the construction of the Human Pose and Activity Monitoring Systems, proposed in this paper. The pose model for each image is delivered as a json file with the nodes for each person detected. In all the captured images, used in this work, only a person was the object of study and, as such, only one model is considered for each image.

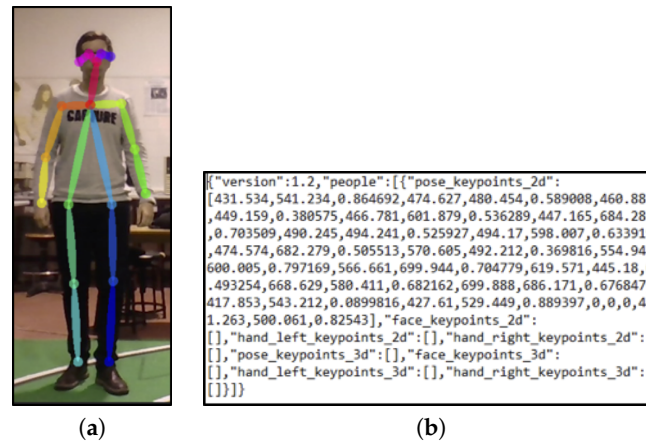


Figure 5. The human pose model: (a) The feature points, in circles with different colours, of the OpenPose human body 18 nodes model; (b) a sample of the json file, only with the 2D node points.

The machine that runs OpenPose is an Ubuntu 16.04 desktop, with an GPU graphic card NVIDIA Quadro P2000, an i7 processor @ 3.60 GHz and 64 Gb of RAM. The images are captured from an Intel Intellisense camera D415, setup to capture images at 60Hz at the resolution of 640×480 pixels. Within the CPU and GPU specifications, the OpenPose algorithm runs at an average of two images per second, that is, can obtain the human body model two times per second.

3.1. Human Pose Detection and Recognition in Homes

For the purpose of human pose analysis, a set of poses were identified that represent the typical human poses in home daily life. They are: lying down; standing; sitting; slanted right; slanted left; crouching. In Figure 6 are presented the images of the set of poses.

The application to obtain the learned models was developed taking into account previous works from the authors, when building computational intelligence platforms and applications to other domains, such as health [32] and agriculture [33]. This stage of the process was performed using Matlab r2018, running in a desktop computer, with i7 processor @ 3.60 GHz and 64 Gb of RAM.

The second stage of human pose estimation, is to implement the learned model in a machine that can operate autonomously, for example, a robot. As such, the learned model from the previous stage, coded in Matlab, should be coded in C++. For that, the Matlab Coder toolbox was used and a C++ program was developed as a Robot Operating System (ROS <https://www.ros.org/>) node to run in the ROS environment and be able to communicate seemingly to other ROS nodes that controls the robotic system. This node was called */classifyX*, given as output the name of the pose of the human, at a time rate of 0.5 seconds.

To control the robot, several other nodes were implemented, that are responsible to capture the image, run the OpenPose algorithm, navigate the robot and various to start a dialogue with an human. Despite the pose estimation being refreshed two times at each second, the robot navigation control is done at an higher frame rate, based on the robot encoders and laser range finder. The behavior of the robot was implemented under the ROS environment using the state machine framework SMACH (<http://wiki.ros.org/smach>), which, depending on the state of the human: can start a dialogue with the person, or call for help in the case of a person lying on the floor.

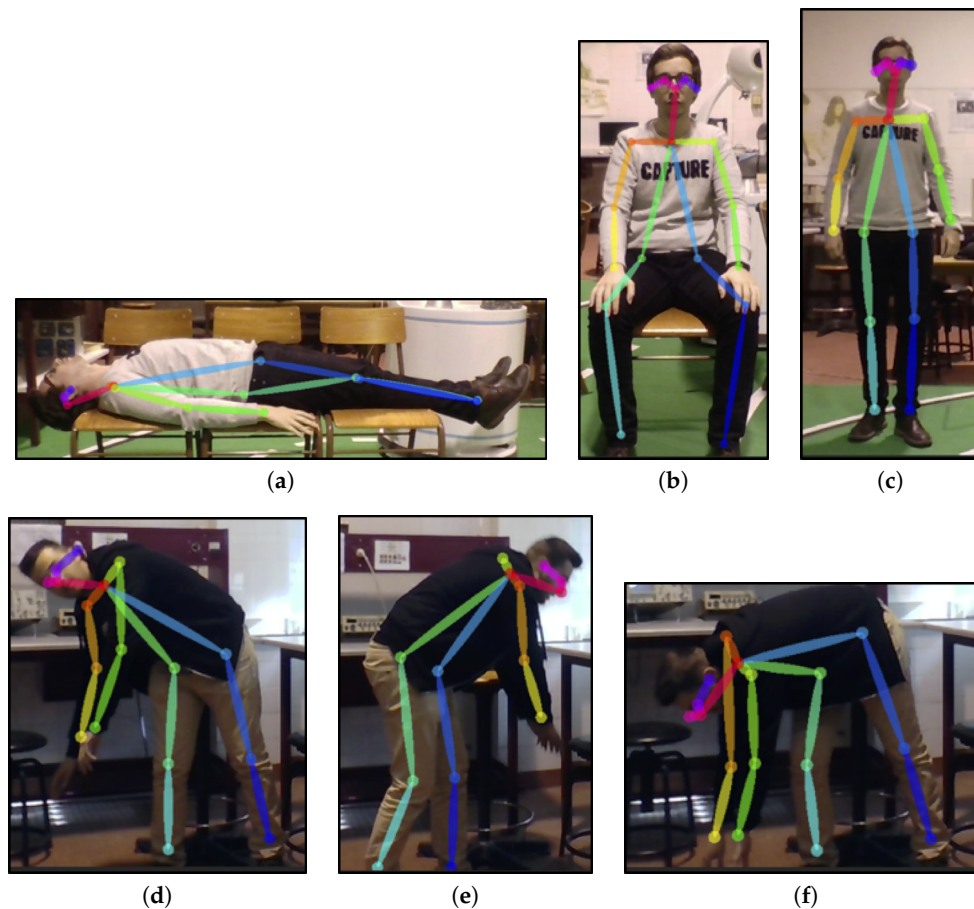


Figure 6. This figure shows examples of all the human poses classes, used in this work: (a) Lying down; (b) Sitting; (c) Standing; (d) Slanted Right; (e) Slanted Left; (f) Crouching.

3.2. Human Activity Detection and Recognition in Homes

From the several activities that a human can perform at home, physical activity was chosen to develop and validate the proposed approach. The physical activity to be performed by the human is a benchmark test, monitor the condition of people. It consists, on walking forward and backwards, 3 m for each leg, starting and ending at the seated position. The time spent at each phase of the exercise is an indicator of the physical condition of the human.

Figure 7 represents two images taken from one image sequence of one exercise, where is depicted the bench of the start and end positions. The procedure of image capturing was the same as the one presented in the previous section, where the 18 node body model was obtained using the OpenPose framework. The sample time for each frame had one second average, where all the pose computations were performed. The sub-images of the right at Figure 7, depicts the computed body models for the images on the left.

The application for human activity detection was developed in Python under the Keras framework [34]. The process was running in an Ubuntu 16.04 desktop machine, with an GPU graphic card NVIDIA Quadro P2000, an i7 processor @ 3.60 GHz and 64 Gb of RAM. The application allows offline and online modes of operation, where the first one can analyse the captured data in batch mode from several activities, or even deploy the processing to another machine, for example in Cloud. The sequence of images to train and test the LSTM model, was captured using the same system of the previous sub-section, that is, an Intel Intellisense camera and the desktop machine that runs the OpenPose algorithm, at two images per second.

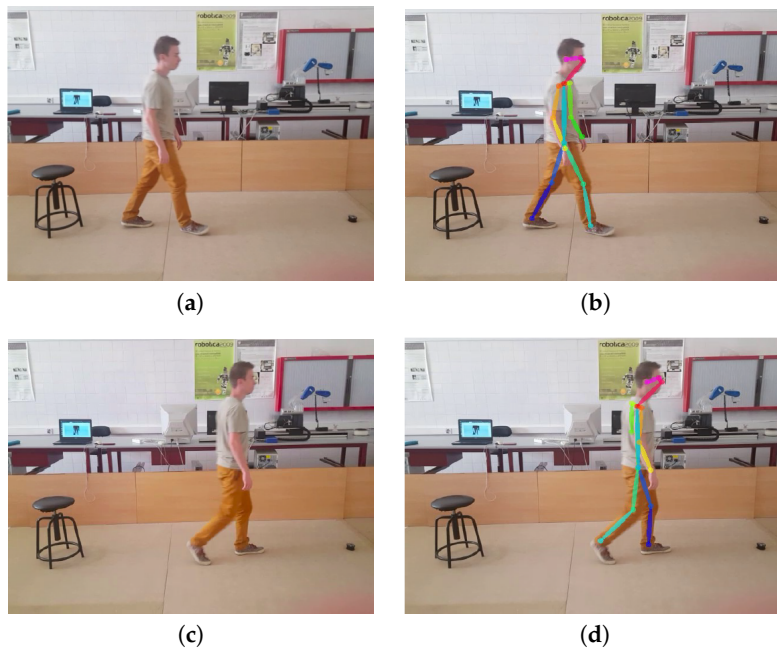


Figure 7. Examples of acquired frames of the exercise performed: (a) frame 39; (b) frame 39 with body model; (c) frame 65; (d) frame 65 with body model.

4. Results and Discussion

In this section, are presented the results obtained from the developed applications in Section 3, that implement the methods presented along with its architectures, in Section 2. This section first presents the devised overall procedure to obtain the Computer Vision Intelligent Models to Extract Human Pose and Activity from Image Sequences. The types of data used and how it was obtained are presented, along with the performance indexes, that will be used to validate the models and the results using the best models. The section ends with the presentation of the results for the two cases: human pose detection and recognition and human activity detection and recognition. For those independent case studies results are presented and discussed, both in training and during the application with novel data.

For training the models, a generic approach was applied for both human pose and activity recognition. In Figure 8 is present that approach. From the acquired images, obtained from an RGB-D camera, presented in Section 3, the human body model is obtained for each image frame. In this work, only one person is present in the image, although OpenPose can recognize several humans. A such, the approach can be extended to extract the poses and activities of humans in the same video frame.

The next step is to normalize the data, within the interval $[-1; 1]$ to avoid large numbers and to scale the data. Moreover, it have an important effect on the results of the training, validation and test procedures, as discussed in Reference [33]. From the acquired data, a separation is needed to distinguish the data from training and validation, as further discussed in the next sub-sections, to increase the robustness of the learned models. The classification models are then obtained with a proper results discussion with the adequate performance metrics.

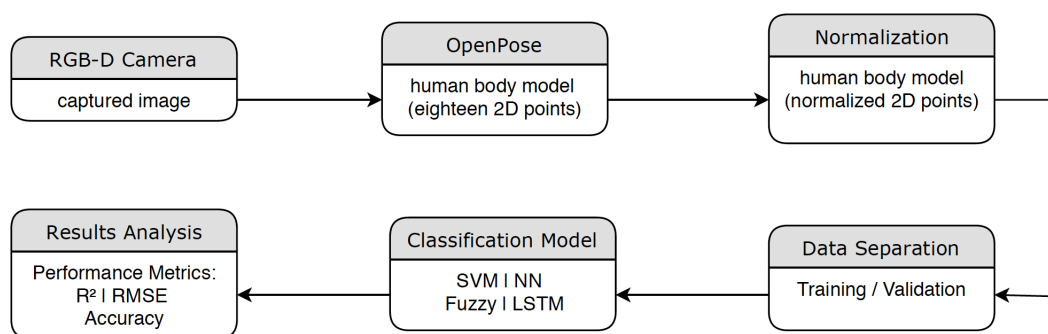


Figure 8. The workflow proposed in the paper to obtain the several models from the acquired data from images, for both human pose and activity detections. In the boxes, in gray are presented the main tasks and in white rectangles the data or sub-tasks are depicted.

4.1. Data Acquisition and Preparation

From the acquired data, that is, the node points of the human body model, as presented in Section 3, several steps needs to be performed to obtain the respective pose and/or activity. The steps are data normalization and separation in training and validation sets. The processes of data separation is important to obtain robust models. If all the available data is used to train the model, those will likely be over fitted to the training data and exceptional results are also likely to be obtained. To obtain a proper model, for example, that can generalize to new situations (data samples) is mandatory to use appropriate training approaches, as presented in Reference [32].

To evaluate the obtained model during training, cross validation [35] is used. In other words, the full dataset is split in training and validation subsets. Before training, some data is removed from the full dataset. After training is complete, the removed data is used to test/validate the performance of the obtained model. Several iterations can be done to check the consistency of the model in several randomized subsets of training and validation. After this process, the classification models are obtained and then applied in new situations.

4.1.1. Pose Detection

Table 1 presents the classes and its size, in number of images. It is also depicted in the table that the data was obtained from eight different persons, students at Instituto Politécnico de Castelo Branco. Its is very important to mention that the data is balanced, that is, the number of images for each class is the same, leading to more accurate models. Since the training algorithms are designed to maximize accuracy, is important to have each class with the same size.

Table 1. Number of images captured to build the data set, by class, for each of the eight persons: $P_i, i = 1, \dots, 8$.

Class Number	Class Description	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	Size
1	Lying down	26	16	21	11	16	11	11	8	120
2	Sitting	22	20	9	10	19	15	12	13	120
3	Standing	18	19	11	21	19	12	10	10	120
4	Slanted Right	12	19	14	14	13	18	18	12	120
5	Slanted Left	18	29	12	14	13	18	12	12	120
6	Crouching	16	21	8	20	14	14	9	11	120

Overall, were used 720 images of humans in the training and testing of the models. After this process, new images were taken from the same persons, to verify the quality of the final application developed.

4.1.2. Activity Detection

Human activity in homes cannot be properly inferred by identifying the pose on static images, because the time is very important in these analysis. That is why the algorithms used for the pose are not the ones to apply. The use of LSTM networks is to be used because it can capture and keeps in memory, for example, in the learned model, the evolution of the human poses during a time interval.

From the physical activity exercise presented in Section 2.2, three atomic activities are present: seating, walking (both forward and backward). These activities are to be learned from videos of persons doing the exercises and become the classes to be detected. In other words, each phase of the exercise, will be an activity to detect and consequently a class to be inferred from the LSTM model.

In this research work, thirty exercises were recorded from three different persons. From the acquired data, the LSTM model for this exercise was learned. After this process, new videos were acquired for offline processing, along with tests with the online version of the application were performed.

4.2. Performance Indexes for Results Analysis

This sub-section presents three types of performance indexes, related to the classification problem at end. These indexes are needed to evaluate the quality of the learned fuzzy model, neural network or support vector machines. The models were developed using Matlab, while using the CLAP platform [32,33] and the toolboxes therein for classification. The toolbox is freely available by request to the authors. The confusion matrix [36] was used to present the results for the assessment of the classification of the human pose, to the dataset not used for training or validation. It was also used the accuracy, which represents the percentage of correctly labelled poses, when compared with the annotated data. During training and for the assessment of the results obtained, two indexes were used that are complementary, that is, the R^2 index and the Root Mean Square (RMSE). R^2 index is related to the variation of the variables, where 1 represents more accurate results, while RMSE represents the mean error when the model fails to classify correctly the pose.

$$R^2 = \left(\frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \cdot \sum(y-\bar{y})^2}} \right)^2 \quad (8)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2}$$

4.3. Human Pose Detection and Recognition

For the training and validation steps, the data was normalized and split with 80% for training and 20% for validation. A 5-fold cross validation approach was used, that is, the data was randomly divided in training and validation and the results are depicted in Table 2.

To estimate the above mentioned models, in Section 2.1, the following parameters were set, after a large set of experiments, for each learned models:

- the fuzzy sugeno inference with 100 clusters using subtractive clustering;
- the support vector machine model using the linear epsilon insensitive cost, while the gaussian kernel's bandwidth has to be set to 0.1, along with the lagrangian multipliers bound set to 20;
- the probabilistic neural network (NN-pnn), with 140 neurons in the radial basis layer

Table 2. Overall results for human pose estimation, obtained from the three methods presented in Section 2.1.

Method	R^2	RMSE	Time Elapsed [msec]
Cubic SVM	0.99523	0.11785	1.3133
Sugeno-SC	0.73216	0.91893	1.7956
PNN	0.85338	0.65279	0.38707

Having the three different learned models, it is clear by observing Table 2 that the one showing more accuracy is the Cubic SVM. The Fuzzy Sugeno model showed inappropriate accuracy levels, which confirms the disadvantage on the large dependency of the training data and difficulty to generalize. The used data was noisy and the validation step was designed with several cross-validations, to enhance the learned models. Despite being more accurate than the fuzzy models, the probabilistic neural network is not more accurate than the SVM, although being able to obtain results more quicker.

In conclusion, the learned cubic SVM model is the one chosen to be implemented and further tested, with new data to verify its robustness. In Figure 9 is presented the confusion matrix of the results for this next step, obtained from the application developed to classify the pose of the human from the images captured. It is clear that crouching estimation obtained the worst results and the lying down the best one. The accuracy of the results is depicted in Table 3, that confirms the excellent performance of the classification application based on SVM, with 29 errors in 720 classified poses, which give an overall accuracy of 95.97%. These results confirms that SVM can generalize and that the model obtained does not show over-fitting to the training data.

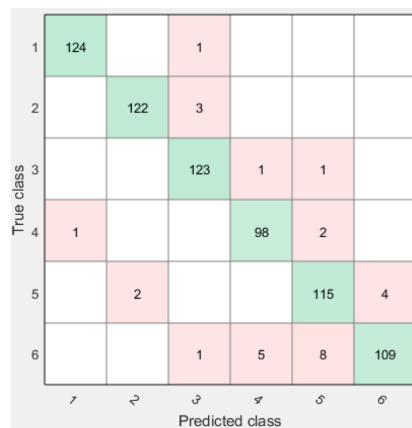


Figure 9. The results obtained from the Cubic SVM model implemented in C++, to operate in the ROS framework.

Table 3. Accuracy of the implemented Cubic SVM learned model, when applied to new data samples.

Class Number	Class Description	Nr. Errors	Accuracy %
1	Lying down	1	99.2
2	Sitting	3	97.6
3	Standing	2	98.4
4	Slanted Right	3	97.0
5	Slanted Left	6	95.0
6	Crouching	14	88.0

After the process of learning the SVM model for the human pose detection and recognition, it was implemented in the application presented in subSection 3.1. The process workflow is depicted in Figure 10, were from the captured RGB-D image, the pose is inferred by the robot manipulator using the ROS node developed for classification. It outputs the number of the class, as presented in Table 1.

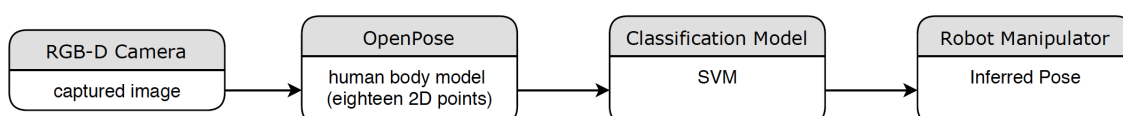


Figure 10. The workflow proposed in the paper to obtain, from the acquired image and using the learned cubic SVM model, the inferred pose to be sent to the robot manipulator.

Figure 11 presents the experimental setup where the human pose estimation is implemented, that is, in the robot equipped with an RGB-D camera facing the human. As seen in the figure, the OpenPose application identified all the nodes of the human pose, that are to be fed to the classification application. The application, when called by the robot, delivers its output, as seen in Figure 12. The learned model output is 2.800800, which clearly is close to the class label number 3, standing, with an error on 0.2. This error is higher than the trained error, which is generally expected when the learned model is applied to new situations, which was the case.

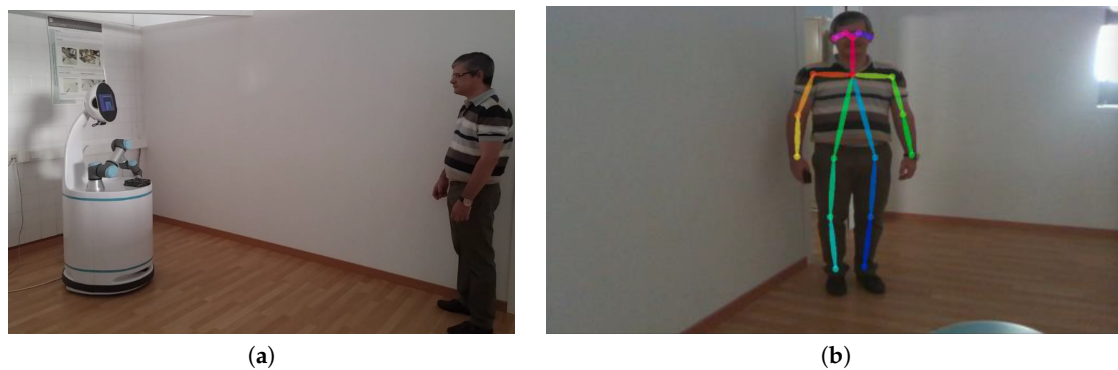


Figure 11. The experimental setup: (a) Robot and the person to estimate the pose; (b) the image captured with the OpenPose eighteen node body model superimposed.

```

672.458000 | 73.040900 | 672.476000 | 153.355000 | 586.320000 | 151.421000 | 552.948000 | 241.
487000 | 553.972000 | 311.591000 | 728.439000 | 157.227000 | 782.183000 | 229.583000 | 881.757
000 | 327.692000 | 631.295000 | 325.466000 | 823.489000 | 462.888000 | 625.222000 | 576.482000
| 721.442000 | 333.512000 | 711.710000 | 462.839000 | 697.958000 | 572.539000 | 656.701000 |
61.266000 | 686.211000 | 61.229000 | 633.249000 | 73.042500 | 711.710000 | 71.063900 |
classificacao: 2.800800 ←
Classe -> En pé
lab301ab3 6L553V0-~/desktop/bernardo/classifyx4

```

Figure 12. The output of the system for the given pose, depicted in Figure 11b .

For the pose estimation, the results are dependent on the video frame, that is 60 Hz at the resolution of 640×480 , and the OpenPose algorithm which outputs an update of the human body model at a rate of two per second. Apart from this bottleneck related to the video systems, the pose classification algorithms are indeed fast. The SVM learned model was the one with best accuracy and implemented in the ROS framework, which have a 1.31 milliseconds time for classification. This factor shows the efficiency of the approach since it is only based on visual features and can refresh its results two times each second. Moreover, it showed very good results with new data, as depicted in Figure 9. This is sufficient for the task tackled, that is, to monitor human body pose.

4.4. Human Activity Detection and Recognition

For human activity detection and recognition, the LSTM model (Section 2.2) was learned from a new gathered dataset, independent from the dataset used in the previous sub-section. During the training and validation steps, the data was normalized in the interval $[-1; 1]$, and split with 80% for training and 20% for validation. A 5-fold cross validation approach was used, that is, the data was randomly divided in training and validation. In the learning process it is adequate to inflate the data to obtain a larger dataset and make the learned model more robust [34], even to missing data. In this sense, the following procedures were performed to obtain new data from the captured ones adding gaussian noise to frames; adding new frames with different focal lengths (by performing homothetys); adding new frames with different rotations; deleting frames. This process was performed which increased the available trained data four times.

In Figure 13a,b are presented examples by adding gaussian noise with zero mean and standard deviation of 20 pixels. In Figure 13c,d are presented examples by performing homothetys that changed the focal length. In the example depicted in the figure the overall size of the body model was diminished

by a factor of two. Following, Figure 14 depicts examples of rotations used, while inflating the data. In the examples shown, the rotations range from zero to ninety degrees. Using this new data in the learning phase, the drawback of the need for large amounts of training data was taking into account.

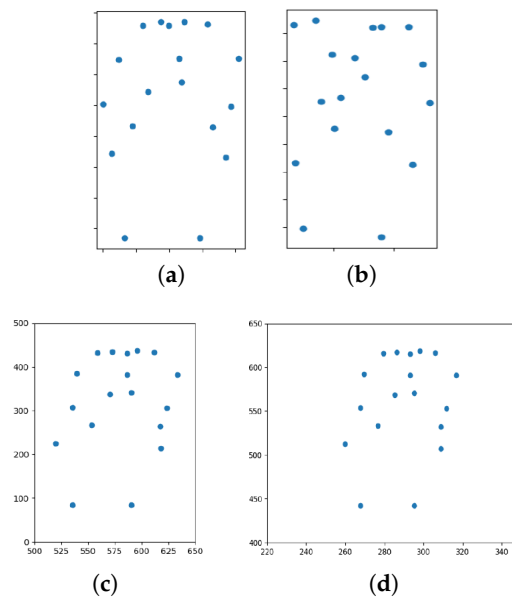


Figure 13. This figure shows examples of data inflation, represented by samples of the 18 nodes body model: (a) acquired body model; (b) body model after adding gaussian noise, to simulate errors in OpenPose; (c) acquired body model; (d) body model after performing an homothety, that is, to simulate different image depths.

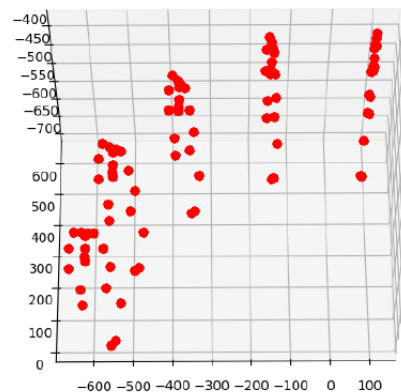


Figure 14. This figure shows examples of data inflation, represented by samples of the 18 nodes body model, due to rotations of the model, from zero to ninety degrees.

After the process of learning the LSTM model for the human activity detection, it was implemented in the application presented in Section 3.2. The process workflow is depicted in Figure 15, where, from the captured RGB-D image frames, the activity is inferred by the robot manipulator using the ROS node developed in python for classification. It outputs the number of the class, as presented in Figure 16.

After several experiments with the LSTM architecture the following parameters were set consider six time steps; the first LSTM with 100 neurons; the second LSTM with 50 neurons. The final feedforward neural network has 30 neurons in the first layer, 10 in the middle layer and 3 neurons in the output that corresponds to the three classes/atomic activities to be detected.

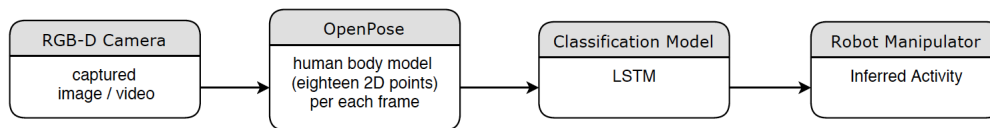


Figure 15. The workflow proposed in the paper to obtain, from the acquired sequence of images/video and using the learned LSTM model, the inferred activity to be sent to the robot manipulator.

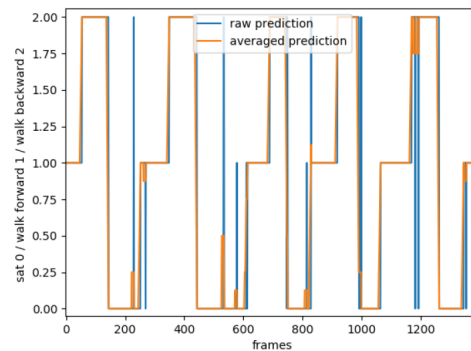


Figure 16. This figure shows the predictions for the activities performed during the exercise, using the LSTM, while walking to the right and left, before ending seated. These steps are the repeated in cycle.

As a final result of the LSTM learned model, an overall accuracy of 88% was obtained. It is worth to mention that the errors appear mainly during the activity changes, that is, from the seating to walking, walking to seating and also in the change of walking forward (from the left) to walking backward (to the right). These errors are depicted in Figure 16 and can be clearly viewed in the *raw prediction*. By applying a simple moving average of three time steps, the results are smoothed and the *averaged prediction* is obtained, where the errors are clearly diminished, as expected. This procedure was the chosen one, instead of trying to obtain a novel class for the transition, which by itself is not a relevant atomic activity. As such, and taking into account the results obtained from the LSTM learned model and the final prediction, the model is considered to be valid for the purposes of this study. Moreover, it was capable of tackling considerable time lags, two events per second in this case, and noisy data, both acquired and inflated.

The activity recognition also has, for the computational power used, an output rate for the OpenPose algorithm of two per second. Despite this fact, the LSTM learned model have an average of 0.13 seconds to refresh the prediction. This fact can be observed from the Figure 17, where is presented its behaviour across some iterations, i.e, sequential time steps, showing a nearly constant variation. This fact ensures that despite the model is more complex than other approaches, for example, neural networks, can be considered stable in computational requirements. The accurate results obtained, combined with the required time to obtained them, show the efficiency of the approach, considering that only visual features were used. This is sufficient for the task tackled, that is, to monitor the human activity in home related activities, such as the one presented in this sub-section.

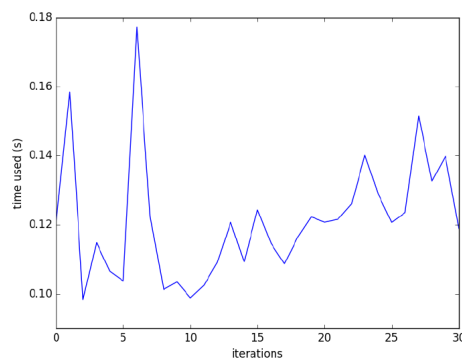


Figure 17. The time used for the LSTM algorithm to output the prediction, at each iteration/time step.

5. Conclusions and Future Work

The paper proposed applications based on computer vision and computational intelligence methods (NN, FM, SVM, LSTM), which allowed to obtain:

- Human Pose semantics from video images, for six different classes. The cubic SVM model obtained 95.97% of correct classification, during tests in situations different than the training phase.
- Human Activity semantics from video image sequences. The implemented LSTM learned model achieved an overall accuracy of 88%, during tests in situations different than the training phase.
- Atomic activities and quantify the time interval that each activity take place.

The computational intelligence approaches used showed excellent results giving the real complexity of the problem related to the several positions, rotations and perspective transformations that the human body model, obtained at each image video frame, can have. Moreover, the gathered data used are only based on visual features and it is noisy.

The developed applications are currently running on a robot in a smart-home environment. This robot is set to monitor a human at home. From the visual features of the human is important to infer its health status and to detect potentially dangerous poses. Moreover, and using the learned LSTM model, the robot can monitor physical activity exercises as that presented in this paper, giving reports of human performance on the benchmark exercise.

Since OpenPose can recognize several humans in the same video frame, the proposed approaches can be extended to extract the poses and activities of several humans in the same image frame. This is one possible direction for future work. Another direction for future work is to gather more data, especially for the activity recognition case, to increase the accuracy of the learned system.

Author Contributions: P.J.S.G., wrote the article, supervised all the work and implemented the applications that validated the methods. B.L., acquired the data for pose detection, and performed the training of the intelligent models. S.S., developed the work on ROS nodes for the robot manipulator. R.B., developed the data acquisition and training of the LSTM models. A.M., developed work on neural networks and with the final tests of the learned models. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: This work was partially supported by FCT, through IDMEC, under LAETA, project UID/EMS/50022/2019. This work was partially supported by project 0043- EUROAGE-4-E (POCTEP Programa Interreg V-A Spain-Portugal). This work was partially supported by FCT, via project SAICT-POL/23811/2016 (GMovE+).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chan, M.; Campo, E.; Estève, D.; Fourniols, J.Y. Smart homes?current features and future perspectives. *Maturitas* **2009**, *64*, 90–97. [[CrossRef](#)] [[PubMed](#)]
2. Bonnefon, J.F.; Shariff, A.; Rahwan, I. The social dilemma of autonomous vehicles. *Science* **2016**, *352*, 1573–1576. [[CrossRef](#)] [[PubMed](#)]

3. Matthias, B.; Kock, S.; Jerregard, H.; Kallman, M.; Lundberg, I.; Mellander, R. Safety of collaborative industrial robots: Certification possibilities for a collaborative assembly robot concept. In Proceedings of the 2011 IEEE International Symposium on Assembly and Manufacturing (ISAM), Tampere, Finland, 25–27 May 2011; pp. 1–6.
4. Veloso, M.; Biswas, J.; Coltin, B.; Rosenthal, S.; Kollar, T.; Mericli, C.; Samadi, M.; Brandao, S.; Ventura, R. Cobots: Collaborative robots servicing multi-floor buildings. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots And Systems, Vilamoura, Portugal, 7–12 October 2012; pp. 5446–5447.
5. Jia, Y.; Zhang, B.; Li, M.; King, B.; Meghdari, A. Human-Robot Interaction. *J. Robot.* **2018**, *2018*, 3879547. [[CrossRef](#)]
6. Zanchettin, A.M.; Ceriani, N.M.; Rocco, P.; Ding, H.; Matthias, B. Safety in human-robot collaborative manufacturing environments: Metrics and control. *IEEE Trans. Autom. Sci. Eng.* **2015**, *13*, 882–893. [[CrossRef](#)]
7. Lasota, P.A.; Fong, T.; Shah, J.A. A survey of methods for safe human-robot interaction. *Found. Trends® Robot.* **2017**, *5*, 261–349. [[CrossRef](#)]
8. Amato, F.; Moscato, V.; Picariello, A.; Sperlii, G. Extreme events management using multimedia social networks. *Future Gener. Comput. Syst.* **2019**, *94*, 444–452. [[CrossRef](#)]
9. Aggarwal, J.; Xia, L. Human activity recognition from 3D data: A review. *Pattern Recognit. Lett.* **2014**, *48*, 70–80. [[CrossRef](#)]
10. Argyriou, V.; Petrou, M.; Barsky, S. Photometric stereo with an arbitrary number of illuminants. *Comput. Vis. Image Underst.* **2010**, *114*, 887–900. [[CrossRef](#)]
11. Gonçalves, P.J.S.; Torres, P.M.; Santos, F.; António, R.; Catarino, N.; Martins, J. A vision system for robotic ultrasound guided orthopaedic surgery. *J. Intell. Robot. Syst.* **2015**, *77*, 327–339. [[CrossRef](#)]
12. Cao, Z.; Simon, T.; Wei, S.E.; Sheikh, Y. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA 21–26 July 2017.
13. Lara, O.D.; Labrador, M.A. A survey on human activity recognition using wearable sensors. *IEEE Commun. Surv. Tutor.* **2012**, *15*, 1192–1209. [[CrossRef](#)]
14. Kim, E.; Helal, S.; Cook, D. Human activity recognition and pattern discovery. *IEEE Perv. Comput.* **2009**, *9*, 48–53. [[CrossRef](#)] [[PubMed](#)]
15. Anguita, D.; Ghio, A.; Oneto, L.; Parra, X.; Reyes-Ortiz, J.L. A public domain dataset for human activity recognition using smartphones. In *Esann*; i6doc.com Publishing: Bruges, Belgium, 2013; ISBN 978-2-87419-081-0.
16. Yuan, G.; Wang, Z.; Meng, F.; Yan, Q.; Xia, S. An overview of human activity recognition based on smartphone. *Sens. Rev.* **2019**, *39*, 288–306. [[CrossRef](#)]
17. Hassan, M.M.; Uddin, M.Z.; Mohamed, A.; Almogren, A. A robust human activity recognition system using smartphone sensors and deep learning. *Future Gener. Comput. Syst.* **2018**, *81*, 307–313. [[CrossRef](#)]
18. Ignatov, A. Real-time human activity recognition from accelerometer data using Convolutional Neural Networks. *Appl. Soft Comput.* **2018**, *62*, 915–922. [[CrossRef](#)]
19. Chen, K.; Yao, L.; Zhang, D.; Wang, X.; Chang, X.; Nie, F. A Semisupervised Recurrent Convolutional Attention Model for Human Activity Recognition. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, 1–10. [[CrossRef](#)]
20. Núñez, J.C.; Cabido, R.; Pantrigo, J.J.; Montemayor, A.S.; Vélez, J.F. Convolutional Neural Networks and Long Short-Term Memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognit.* **2018**, *76*, 80–94. [[CrossRef](#)]
21. Amato, F.; Castiglione, A.; Moscato, V.; Picariello, A.; Sperli, G. Multimedia summarization using social media content. *Multimed. Tools Appl.* **2018**, *77*, 17803–17827. [[CrossRef](#)]
22. Vapnik, V. *Statistical Learning Theory*; Wiley: New York, NY, USA, 1998.
23. Sousa, J.; Kaymak, U. *Fuzzy Decision Making in Modeling and Control*; World Scientific Pub. Co.: Singapore, 2002.
24. Takagi, T.; Sugeno, M. Fuzzy Identification of Systems and its Applications to Modelling and Control. *IEEE Trans. Syst. Man Cybern.* **1985**, *15*, 116–132. [[CrossRef](#)]
25. Chiu, S.L. Fuzzy model identification based on cluster estimation. *J. Intell. Fuzzy Syst.* **1994**, *2*, 267–278. [[CrossRef](#)]

26. Castilho, H.P.; Gonçalves, P.J.S.; Pinto, J.R.C.; Serafim, A.L. Intelligent real-time fabric defect detection. In Proceedings of the International Conference Image Analysis and Recognition, Montreal, QC, Canada, 2–24 August 2007; pp. 1297–1307.
27. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [[CrossRef](#)]
28. Zhang, G.P. Neural networks for classification: a survey. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2000**, *30*, 451–462. [[CrossRef](#)]
29. Specht, D.F. Probabilistic neural networks. *Neural Netw.* **1990**, *3*, 109–118. [[CrossRef](#)]
30. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
31. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [[CrossRef](#)]
32. Gonçalves, P.J.S. The Classification Platform Applied to Mammographic Images. In *Computational Intelligence and Decision Making*; Madureira, A., Reis, C., Marques, V., Eds.; Springer: Dordrecht, The Netherlands, 2013; pp. 239–248.
33. Gonçalves, P.J.; Estevinho, L.M.; Pereira, A.P.; Sousa, J.M.; Anjos, O. Computational intelligence applied to discriminate bee pollen quality and botanical origin. *Food Chem.* **2018**, *267*, 36–42. [[CrossRef](#)]
34. Ketkar, N. Introduction to keras. In *Deep Learning with Python*; Springer: New York, NY, USA, 2017; pp. 97–111.
35. Geisser, S. *Predictive Inference*; Routledge: Abingdon, UK, 2017.
36. Kohavi, R.; John, G.H. Wrappers for feature subset selection. *Artif. Intell.* **1997**, *97*, 273–324. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).