Intelligent Service Robotics (2021) 14:175–185
https://doi.org/10.1007/s11370-021-00358-7

ORIGINAL RESEARCH PAPER

Combining CNN and LSTM for activity of daily living recognition with a 3D matrix skeleton representation

Giovanni Ercolano[1] · Silvia Rossi[1]

Received: 3 April 2020 / Accepted: 28 January 2021 / Published online: 10 March 2021

Abstract
In socially assistive robotics, human activity recognition plays a central role when the adaptation of the robot behavior to the human one is required. In this paper, we present an activity recognition approach for activities of daily living based on deep learning and skeleton data. In the literature, ad hoc features extraction/selection algorithms with supervised classification methods have been deployed, reaching an excellent classification performance. Here, we propose a deep learning approach, combining CNN and LSTM, that exploits both the learning of spatial dependencies correlating the limbs in a skeleton 3D grid representation and the learning of temporal dependencies from instances with a periodic pattern that works on raw data and so without requiring an explicit feature extraction process. These models are proposed for real-time activity recognition, and they are tested on the CAD-60 dataset. Results show that the proposed model behaves better than an LSTM model thanks to the automatic features extraction of the limbs' correlation. "New Person" results show that the CNN-LSTM model achieves 95.4% of precision and 94.4% of recall, while the "Have Seen" results are 96.1% of precision and 94.7% of recall.

Keywords Social robots · Deep learning · Activity recognition

1 Introduction

Personal service robotics applications are already available on the market to be used in human-populated environments such as working, public and domestic ones. Such applications are designed to accomplish tasks on behalf of the user, with different degrees of autonomy and interaction capabilities. However, they are typically designed to show the same behavior regardless of humans' possible activities and reactions [11]. The foreseen market success of such a technology is still far to be realized due to several research challenges that are mainly related to the need for a high degree of personalization of the robot behavior to the specific user's needs and preferences and acceptance. Indeed, to be acceptable the robot behaviors have to take into account the user activities to not interfere with them [24] and to adapt to them [22]. The robot is required to sense its environment but also to understand what happens within it in terms of human activities.

The work presented in this paper is part of a project aiming at using mobile robots for monitoring the activities of daily living, or ADLs. In this direction, we aimed to develop an activity of daily living recognition algorithm in a home environment using the skeleton data of an RGB-D camera [23]. In the literature, the combination of ad hoc features selection and extraction algorithms with supervised classification techniques has reached excellent classification performance [4,10,25]. In contrast, we propose the use of a simple deep neural networks structure to automatically extract features and to find long-term temporal dependencies. The proposed model relies on the use of a convolutional neural network (CNN) that is able to extract the spatial dependencies from a grid and it works well on image recognition. To do so, we define a simple but informative 3D matrix skeleton representation to be the input of the CNN that can learn limb correlations. Moreover, according to the literature, CNNs are effectively deployed in action recognition due to their ability of representation learning exploiting the spatial relationships

This work has been supported by MIUR within the PRIN2015 research project "User-centered Profiling and Adaptation for Socially Assistive Robotics - UPA4SAR" (Grant No. 2015KB-L78T).

✉ Silvia Rossi
silvia.rossi@unina.it

Giovanni Ercolano
giovanni.ercolano@unina.it

[1] Department of Electrical Engineering and Information Technologies, University of Naples Federico II, Strada Vicinale Cupa Cintia, 21, 80126 Napoli, NA, Italy

of the extracted features. However, to take into account the temporal dimension of an action, temporal data have to be opportunely fused in one complex image, as in [7], or 3D matrix representations have to be deployed [1]. Here, taking inspiration from the work of [6,20], where the authors propose a spatiotemporal classification, respectively, for video description from images and activity recognition from wearable devices data, here we aim at achieving the same results by combining the use of CNNs with LSTM gaining benefits of both spatial and temporal learning.

Following this idea, we investigate the possibility of training the recognition module considering both spatial dependencies due to the relationships among the RGB-D skeleton joints by the use of a CNN and the temporal patterns of the activities by the use of an LSTM. In a previous work [9], we briefly introduced the framework and provided the first experimentation with respect to another approach dealing only with the use of a multi-scale LSTM. Here, we detailed the proposed approach and provided more in-depth experimentation with respect to the state-of-the-art approaches on the cornell CAD-60 activity dataset [28] to highlight the contribution of the feature extraction level of the CNN.

Results on short video sequences (i.e., 140 frames) show that the approach is able to evaluate in real time the activity performed by the human user with performance beyond the state-of-the-art approaches. Indeed, this result allows for a robot to react instantly adapting its behaviors with respect to human behaviors. Moreover, when comparing the performance on the whole duration of a video the approach performs as the other state-of-the-art approaches.

2 Related works

In this section, we accounted for different approaches to activity recognition working on the same dataset. We also discuss some deep learning approaches applied to other datasets that inspired our work.

In [8], the authors introduced an approach where the human skeleton data are analyzed by considering five parts (i.e., arms, legs, and torso), and a hierarchical bidirectional RNN network with a final LSTM layer is deployed to extract features for building a higher-level representation. Subsequently, a fully connected deep LSTM network is proposed in [32] to recognize action with a framework composed of three LSTM and two feed-forward layers, incorporating the co-occurrence regularization into the loss function, so exploring the conjunctions of discriminative joints and different co-occurrences for several actions. In [16], a deep LSTM framework, based on RNN, is proposed to better localize the start and end of action with a regression module, to automatically extract the features. This joint classification–regression RNN considers the sequence frame by frame and does not require a sliding window approach. A hierarchical approach is also presented in [31], where they propose three exploration fusion methods based on multilayer LSTM. The first LSTM layer takes geometric features computed on the 3D coordinates of the human joints; then, the upper LSTM layers investigate into more detail the input features, abstracting them into a high level of knowledge. All these approaches are characterized by a deep/hierarchical structure aiming at recognizing high-level features for temporal data. Indeed, in the presented work, a single LSTM layer is used but in combination with CNN, so relying on the possibility to extract spatial dependencies on the skeleton's joints patterns. In [9], we showed that a multi-scale LSTM approach resulted in slightly lower performances with respect to the proposed one.

Other approaches dealt with the use of CNN for activity recognition. For example, in [7], the skeleton sequence is represented as a matrix concatenating all frames together in chronological order. This allows us to treat the time sequence of joints in a single image that is fed into a CNN model for feature extraction and activity recognition. In [3], the whole images, and not the skeletons, are used to extract joint heatmaps (using CNNs) for each video frame and colorize them using a specific color depending on the relative time. To obtain a fixed-size representation independent of the duration of the video, they aggregate the colorized heatmaps with different methods to obtain the clip-level representation with a fixed dimension. The necessity of compressing temporal data into single images is overcome by the use of 3D CNN that recognizes spatial–temporal features applying convolutions on a time series of frames [1,12]. Also in [19], the authors consider as input for CNN all the skeleton joints of all frames, arranged in a 3D matrix. This 3D matrix has in the first dimension the number of joints, in the second dimension the number of consecutive frames and in the third dimension the 3D coordinates of the joints. Results of the CNN are then combined with an LSTM using a two-stage training strategy that focuses first on CNN training and then on the entire CNN+LSTM method. In our approach, the CNN takes as input each frame of the sequence independently since temporal relationships are deployed at the LSTM layer. In [30], the authors propose a novel model of dynamics skeletons called spatial–temporal graph convolutional networks (ST-GCN) tested on Kinetics and NTU-RGBD datasets. The ST-GCN implementations are different from 2D or 3D CNN since the temporal properties of the skeleton are kept together as in a graph. It follows the similar implementation of graph convolution [14]. In [18], the authors consider the action recognition and the human pose estimation as one problem that they solve with a multi-task CNN. The human pose estimation is composed of a CNN with one entry flow and K prediction blocks to estimate both the 2D and the 3D pose by volumetric heatmaps. Appearance-based recognition relies on local visual features considering also the objects used during the performed action. The results

are combined to estimate the action. Finally, in [15], the authors use a combination of CNN and LSTM to extract spatiotemporal information, but differently from our approach by merging the individual scores obtained from the CNN and the LSTM. Also in this case, contrary to the method proposed by us, they consider all the joints of the skeleton, extrapolating also other information of distance and trajectory between the joints and the poses. The 3 LSTM models take in input the real positions, distances between joints, distances between joints and lines, while the 7 CNN models take in input the joint distances maps and the joint trajectories maps in time to generate color image to be fed into a CNN module. The innovation in our proposed approach compared to similar works presented so far is in proposing a new spatial representation of the features of human pose.

Different approaches are presented in the literature that are evaluated by the use of the CAD-60 dataset. These approaches are mainly characterized by different features extraction initial processes. In [4], for example, a k-means clustering algorithm computes the "key poses" to describe the activity for each sequence with K centroids that composed the features vector. In [25], the key poses are identified by recognizing poses with the kinetic energy close to zero to perform a sequence segmentation. This approach is shown to be robust with respect to the temporal stretching of an action. In [13], the fusion of 5-CNN is proposed for activity recognition, using motion history image (MHI), depth motion maps (DMMs) (front, side, top) and skeleton images (an image representation of the skeleton joints) as input. Each different type of data is trained on a different CNN and the softmax scores are fused to classify the activity. In [10], the distances and motion features (evaluated as the distances between the initial position of a joint and the position in the following frames) form a total of 14 features that characterize the 12 activities of the dataset. A dynamic Bayesian mixture model (DBMM) is proposed to classify the activity considering the temporal information. Depth-based action recognition is evaluated in [33] using the spatial–temporal interest point (STIP) with the combination of different interest point detectors and descriptors. The SVM classifiers are used to detect the activity. A neurobiologically motivated approach is presented in [21] to recognize action in real time with the growing when required (GWR) networks. The GWR network is a set of neurons that dynamically change their topological structure according to the input creating new neurons with different weights. The architecture proposed is a two-stream hierarchy of GWR networks that can learn spatiotemporal dependencies processing in parallel the pose and motion features extracted from video sequences.

In a recent work [17], Liu et al. proposed a classical machine learning technique, selecting the features from the skeleton data. First, they preprocessed the skeleton data denoising, transforming and normalizing the pose. Then, they considered the position, the velocity and the acceleration of the poses. The recognition method is a three-step weighted voting process based on k-nearest neighbors (kNN). They evaluated their method on MSR-Action3D and CAD-60 datasets, obtaining good results. Currently, this approach is the one obtaining the best performance on precision and recall for CAD-60 considering a whole video sequence. The main difference between our work and [17] is that we use a sliding window to solve a different problem. The obtained model trained on 140 frames instances can classify activities in real time on videos of a few seconds. We also tested our model on the whole videos to compare our approach with the others. Unlike the approaches applied on the CAD-60 dataset that select and extract features manually, we propose a deep learning model for automatic feature extraction that uses CNNs to extract spatial dependencies from human poses and LSTMs to extract temporal dependencies between poses.

3 The proposed approach

The proposed model aims to explore the combination of CNN for representation learning and of LSTM for temporal dependencies learning, which is proposed in applications that concern spatiotemporal classification, like in [6] for video description and in [20] for activity recognition from wearable devices data.

A CNN can be thought of as a hierarchy of convolutional modules that progressively learn higher-level features. Each convolutional module can be composed of: convolution layers that are banks of affine transformations of input (also called kernels) applied on the grid input; detector layers that apply a nonlinear activation function; and pooling layers for reducing the input size and improving the statistical efficiency. The CNNs are deep neural networks for processing grid-like topology data (i.e., image data). Indeed, also the skeleton data can be mapped into an image, but the proper representation has to be investigated. Our initial aim was to automate also the extraction of the spatial features considering all possible connections between the skeleton joints. However, we found a reduced and concise representation that could well describe the human pose.

To be efficiently applied for action recognition, the first step is the transformation of the input data, the coordinates (x_i, y_i, z_i) of each of the ith joint of the human body at time t extracted by an RGB-D camera. Here, a novel representation of the joints values is proposed. Given the vector $f = [x_1, y_1, z_1, ..., x_N, y_N, z_N]$ of N skeleton joints, we combine these features in a three-dimensional matrix considering the spatial dependencies between the limbs. We have built a three-dimensional matrix to be invariant to translation, rotation, and scale. This matrix is the representation of the

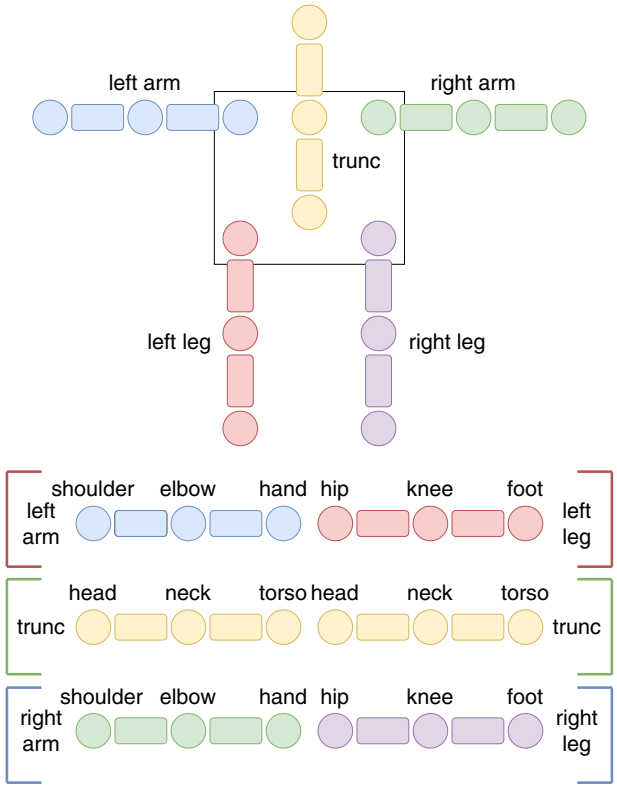

Fig. 1 An abstract diagram of the proposed three-dimensional matrix for human pose representation

posture and it is the input of the CNN that can automatically extract the spatial features.

The input is composed of three matrices referring to data related to the left arm (a_l)/leg (l_l), the trunk (t) and the right arm (a_r)/leg (l_r) of the human skeleton joints for each frame. Every considered limb is constituted by three joints each. For example, in the case of the left arm, the three joints are the left shoulder ($a_l[0]$), the left elbow ($a_l[1]$) and the left hand ($a_l[2]$). In the case of the left leg, the three joints are the left hip ($l_l[0]$), the left knee ($l_l[1]$) and the left foot ($l_l[2]$). The same is for the right arm and leg. In the case of the trunk, we have the head ($t[0]$), the neck ($t[1]$) and the torso ($t[2]$). The aim is to recognize the spatial dependencies between the limbs. Therefore, we model the following matrix representation:

$$\begin{bmatrix} a_l[x_0] & a_l[x_1] & a_l[x_2] & l_l[x_0] & l_l[x_1] & l_l[x_2] \\ a_l[y_0] & a_l[y_1] & a_l[y_2] & l_l[y_0] & l_l[y_1] & l_l[y_2] \\ a_l[z_0] & a_l[z_1] & a_l[z_2] & l_l[z_0] & l_l[z_1] & l_l[z_2] \end{bmatrix}$$

$$\begin{bmatrix} t[x_0] & t[x_1] & t[x_2] & t[x_0] & t[x_1] & t[x_2] \\ t[y_0] & t[y_1] & t[y_2] & t[y_0] & t[y_1] & t[y_2] \\ t[z_0] & t[z_1] & t[z_2] & t[z_0] & t[z_1] & t[z_2] \end{bmatrix}$$

$$\begin{bmatrix} a_r[x_0] & a_r[x_1] & a_r[x_2] & l_r[x_0] & l_r[x_1] & l_r[x_2] \\ a_r[y_0] & a_r[y_1] & a_r[y_2] & l_r[y_0] & l_r[y_1] & l_r[y_2] \\ a_r[z_0] & a_r[z_1] & a_r[z_2] & l_r[z_0] & l_r[z_1] & l_r[z_2] \end{bmatrix}$$

Figure 1 shows an abstract diagram to explain the disposition of the limbs in our proposed three-dimensional matrix for human pose representation and feature extraction with the CNN. Each limb is composed of three joints and represents the rows of the three coordinates x, y and z. From this matrix representation, CNN learns the spatial features that involve the spatial limb correlations.

The proposed CNN is a three-layer deep network (see Fig. 2). The three-matrix representation of the posture is given as input to the first convolutional layer. It sizes $3 \times 6 \times 3$ and has a set of kernels of size 1×1 and stride 1 to consider the spatial limb dependencies. Since the kernels size 1×1, the first convolutional layer linear recombines the weights based on the input feature maps as a parametric pooling layer. Therefore, its output sizes $3 \times 6 \times k_1$ and it is the input of the second convolutional layer. The second layer has a set of kernels of size 3×3 with stride 1. Its output sizes $3 \times 6 \times k_2$ where k_2 is the number of kernels. A max-pooling layer of size 2×2 with stride 2 halves the resolution of the third layer output and its output sizes $1 \times 3 \times k_2$. The size of 2×2 instead of the size of 3×2 is due to consider more information of the coordinates x and y than the information of coordinate z. A final layer flattens the output of the third layer concatenating the values in a vector with a length of $1 \cdot 3 \cdot k_2$.

The features extracted by the CNN are the input of the LSTM to identify the temporal dependencies of the change of the postures during the instance sequence. Hence, the LSTM layer takes as input a sequence of CNN output accumulating the temporal dependencies between each frame of the video. The LSTM input is a feature vector that contains the con-

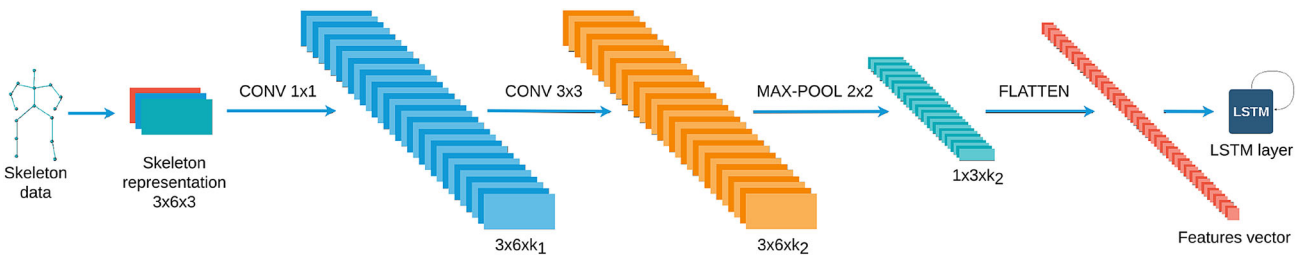

Fig. 2 Combination of a CNN for automatic features extraction from the skeleton representation and an LSTM

catenation of the weight matrix. The LSTM is composed of a single layer and a number of neurons equal to the number of the feature vector extracted from the CNN. A full-connected layer with a softmax activation function classifies the activities performed in the video from the extracted features of the LSTM.

4 Experimental evaluation

In this paragraph, we first introduce the dataset used for the experimental evaluation. Then, we describe the configuration of the proposed models and the results. Specifically, we compared the CNN-LSTM model based on our 3D skeleton representation with an architecture composed only by an LSTM layer to highlight the possible contribution of using CNN and the proposed joint matrix representation in accounting for spatial dependencies. Moreover, we will discuss our results in comparison to other state-of-the-art approaches tested on the same dataset.

4.1 Dataset

Our project aims at recognizing the ADL to monitor the daily activities of elderly people. In this direction, we use the cornell activity dataset (CAD-60) for training and testing the deep networks. The CAD-60 [28] is composed of 60 RGB-D videos captured by a Microsoft Kinect, with twelve activities performed in five environments. These videos are accomplished by four subjects, two males and two females, with one left-handed. The 12 labeled activities are: rinsing mouth, brushing teeth, wearing contact lenses, talking on the phone, drinking water, opening pill container, cooking (chopping), cooking (stirring), talking on couch, relaxing on couch, writing on whiteboard and working on computer. The CAD-60 dataset has two more activities (random and still) which are used together for classification assessment on testing sets. The 5 environments are office, kitchen, bedroom, bathroom and living room. The dataset is made up of RGB and depth images, and the tracked skeleton. Fifteen of the skeleton information are extracted for each frame. The total number of videos is 68: 17 videos for each user.

We decided to use a temporal sliding window for considering all the contiguous frames, unlike [29] where they used a deep learning approach by selecting one frame every six frames of the videos to reduce redundancy and complexity. The smallest video of the CAD-60 is of 147 frames; therefore, we have set the instances of 140 frames (e.g., we obtain 8 instances with a video of 147 frames). Thus, the input sequence to the CNN-LSTM and LSTM models sizes 140 frames. Further considerations on the choice of the 140 frames window size can be found in the results of the CNN-LSTM model. In all model configurations, the validation set is 33% of the training set.

Table 1 shows the frequency distribution of the instances extracted from the CAD-60 dataset with 140 frames for each instance. In Table 1, we considered the environment and the activity class performed by each user. Note that the numbers of the instance are not balanced between the 13 activities. In particular, the "random + still" activity has a number proportional to the sum of the other activities for classification assessment.

4.2 Data preprocessing

The number of skeleton joints tracked in CAD-60 is 15. Eleven joints have both joint orientation and joint position while 4 joints have only the joint position. We considered only the joint positions of the 15 joints. To train our model on 140 frame instances, a temporal sliding window was applied. For each 140 frame instance, we have performed three preprocess steps for the coordinates of the skeleton joints as follows:

1. *Symmetrization.* Since in the dataset, there is one left-handed person, for each subject, we also considered mirrored skeleton data. To mirror the skeleton sequences, we took the opposite values of the x coordinate that are on the horizontal axis. In other words, the point coordinates $J = (x, y, z)$ become $J_{new} = (-x, y, z)$. This step doubles the number of dataset instances.
2. *Translation.* We set the midpoint between the points of the torso, left and right shoulder, left and right hand as the origin of the coordinates system. Once the midpoint was calculated, it was subtracted from the coordinates of the joints to have the midpoint as the center of the skeleton pose. For example, if we have a joint $J = (x, y, z)$ and the midpoint is $J_{mid} = (x_{mid}, y_{mid}, z_{mid})$, the new joint will be

$$J_{new} = (x - x_{mid}, y - y_{mid}, z - z_{mid}).$$

3. *Normalization.* We compute the mean and the standard deviation for each instance to normalize the translated data on a new origin using the standard score: $J_{new} = (J - \mu)/\sigma$. The new coordinates are calculated following the previous formula applied to each coordinate (x, y, z). For each coordinate c (x, y, z), the following equation applies to all the elements i of each sequence:

$$J_{new_{ci}} = \begin{cases} (J_{c_i} - \mu_c)/\sigma_c, & \text{if } \sigma_c \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

Table 1 Number of 140 frame instances for each environment and each activity class

Environment	Class	User 1	User 2	User 3	User 4	Total
Bathroom	Brushing teeth	1212	1536	1644	1441	5833
	Random + still	2684	2074	2729	2785	10272
	Rinsing mouth with water	1607	1307	1364	1726	6004
	Wearing contact lenses	557	1137	544	822	3060
Bedroom	Drinking water	1448	639	1171	1390	4648
	Opening pill container	332	546	204	595	1677
	Random + still	2684	2074	2729	2785	10272
	Talking on the phone	1386	691	1149	1169	4395
Kitchen	Cooking (chopping)	1426	1525	1615	1771	6337
	Cooking (stirring)	1207	1210	1328	1696	5441
	Drinking water	1448	639	1171	1390	4648
	Opening pill container	332	546	204	595	1677
	Random + still	2684	2074	2729	2785	10272
Living room	Drinking water	1448	639	1171	1390	4648
	Random + still	2684	2074	2729	2785	10272
	Relaxing on couch	1308	1358	1240	1714	5620
	Talking on couch	1542	1400	1573	1673	6188
	Talking on the phone	1386	691	1149	1169	4395
Office	Drinking water	1448	639	1171	1390	4648
	Random + still	2684	2074	2729	2785	10272
	Talking on the phone	1386	691	1149	1169	4395
	Working on computer	1126	1391	1083	1523	5123
	Writing on whiteboard	1653	1498	1458	1653	6262

where μ_c is the mean of the whole 140 frame sequence on the c coordinate while σ_c is the standard deviation of the whole 140 frame sequence on the c coordinate.

4.3 Model settings

The settings of the deep models have an important role in the gradient convergence, preventing overfitting on this tiny dataset. We used the Glorot normal initializer, also called Xavier normal initializer for the initialization of the LSTM weights for each deep model. The experiments showed that the deep models performed well with a dropout set at 0.25 after the max-pooling layer of the CNN and at 0.5 on the LSTM layer. CNN has 32 kernels in the two convolutional layers for a reduction of the number of parameters.

The CNN-LSTM model is compared with an LSTM model. The latter model is the same as the CNN-LSTM model without the CNN level. To make the comparison, we left the same LSTM layer configuration for both models. Both models receive an input sequence of human poses. Thus, in the LSTM model, we have consecutively a single LSTM layer, that extracts the temporal dependencies from the features vector $f = [x_1, y_1, z_1, ..., x_N, y_N, z_N]$ of N skeleton joints representing the human pose (without considering the spatial dependencies with a CNN), and a full-connected layer with a softmax activation function, that classifies the activities.

4.4 Implementation details

We used the API of Keras library that is designed to simplify the development of the neural network. Originally developed on top of Tensorflow, now it is part of the Tensorflow library with the Tensorflow version 2.0. During the experiments, we ran the training and the testing on Keras version 1.2.2 with Tensorflow version 0.12.0[1].

5 Classification results

Two different settings are considered in the original work on CAD-60 [27]: "New Person" and "Have Seen" settings. The most considered experimental setting in all the research works on CAD-60 is the "New Person" to guarantee the generalization of the classifier. The "New Person" setting is defined as a "Leave One Out (LOO)" cross-validation that is, the training set consists of three of the four people and the test set consists of the fourth one. In the "Have Seen" setting,

[1] The code is available upon request

Table 2 Precision (P) and recall (R) of the LSTM and CNN-LSTM models on sequences of 140 frames for "New Person" and "Have Seen" setting on 140 frames window

Location	Activity	New Person				Have Seen			
		LSTM		CNN-LSTM		LSTM		CNN-LSTM	
		P (%)	R (%)	P (%)	R (%)	P (%)	R (%)	P (%)	R (%)
Bathroom	Brushing teeth	96.8	100.0	100.0	98.7	96.6	100.0	100.0	99.8
	Random + still	71.4	90.5	94.0	93.4	91.8	91.2	93.3	93.5
	Rinsing mouth	93.9	92.3	94.6	87.7	92.8	84.4	95.8	88.0
	Wearing lens	89.7	98.9	89.9	98.0	95.5	97.3	88.8	94.0
	Average	94.9	94.1	94.9	93.9	93.6	92.4	94.9	93.9
Bedroom	Drinking water	94.7	97.5	94.9	90.7	96.7	89.9	95.8	93.0
	Opening pill container	82.3	97.3	94.0	96.9	93.1	99.2	89.4	100.0
	Random + still	99.5	91.1	99.5	96.9	99.5	90.8	99.8	95.2
	Talking on phone	91.2	95.6	89.3	94.4	88.7	99.5	90.3	95.9
	Average	95.2	93.9	95.9	94.7	95.2	92.8	96.1	95.11
Kitchen	Cooking (chopping)	95.4	74.3	88.8	94.5	80.0	100.0	87.5	100.0
	Cooking (stirring)	98.8	94.4	91.1	74.8	96.8	66.4	99.7	77.1
	Drinking water	94.9	100.0	99.1	99.7	92.5	100.0	95.3	99.4
	Opening container	80.7	92.4	85.5	95.5	83.3	99.0	87.3	99.2
	Random + still	98.2	91.5	95.3	94.7	100.0	90.1	98.9	94.5
	Average	96.4	95.3	93.1	91.6	92.9	89.9	95.3	93.7
Living room	Drinking water	91.7	95.2	99.8	93.1	90.4	99.1	98.9	97.8
	Random + still	100.0	93.1	99.0	98.5	99.7	91.9	100.0	97.7
	Relaxing on couch	100.0	100.0	100.0	100.0	99.8	100.0	100.0	84.8
	Talking on couch	100.0	100.0	100.0	100.0	100.0	99.5	100.0	100.0
	Talking on phone	91.0	98.1	92.5	99.2	92.4	97.9	88.4	99.7
	Average	97.4	96.8	98.7	98.5	97.4	96.7	98.4	96.4
Office	Drinking water	91.7	93.5	95.0	90.5	92.4	90.8	97.8	83.6
	Random + still	97.2	87.3	97.8	93.9	100.0	84.3	99.1	91.2
	Talking on phone	78.2	96.3	80.5	95.5	73.7	99.2	89.1	98.7
	Working on computer	100.0	100.0	100.0	100.0	98.8	100.0	100.0	100.0
	Writing on whiteboard	89.4	75.7	94.3	85.3	65.2	76.7	90.5	100.0
	Average	91.6	89.5	94.3	93.0	87.9	88.3	95.7	94.27
Overall average		95.1	93.9	95.4	94.4	93.3	92.0	96.1	94.7

the model is trained with half of the testing subject's data and the other half is included in the tests. In the literature, the CAD-60 is split according to the considered environment. The final results are the average precision and recall among all the environments.

Table 2 shows the classification precision and recall of the proposed CNN-LSTM model in comparison with the LSTM model for each environment and for both the "New Person" and the "Have Seen" settings. First, we notice that the test set results of the CNN-LSTM model are better than the LSTM model and they are similar in both the settings, but with slightly better performance in the Have Seen setting.

Both models are expected to suffer from overfitting with a small training set. Especially in cases where there are a small number of training examples, the model may adapt to features that are specific only to the training set; therefore, in the presence of overfitting, the performance of the prediction on the training data will increase, while the performance in the test set will be worse. Hence, overfitting on data could have an impact more on the "Have Seen" setting, leading to better results, since training and testing are both obtained from the same subjects. Indeed, since performance in the "New Person" setting is very similar to the Have Seen case, we can consider overfitting as marginal.

From now on, we will make considerations only on the "New Person" setting.

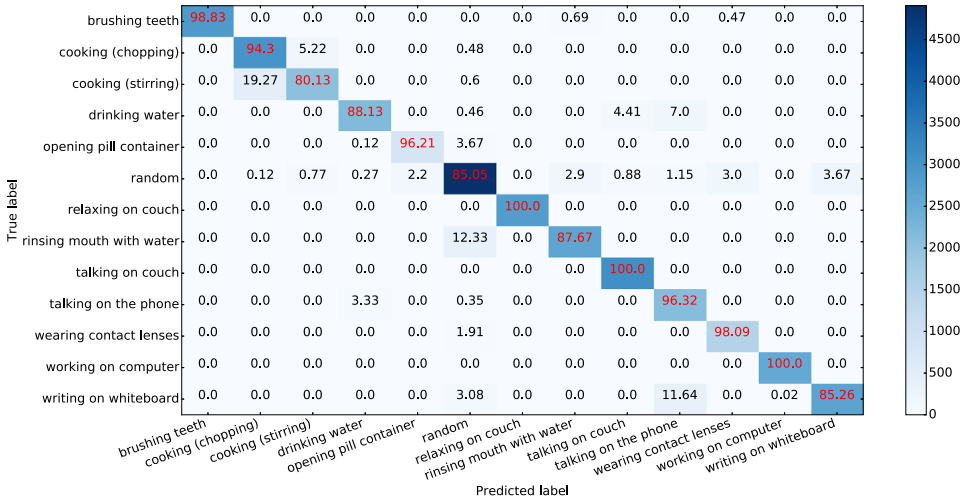

Fig. 3 Overall activity confusion matrix in "New Person" setting with the CNN-LSTM model on 140 frames window

5.1 LSTM results

In the Living Room (97.4% and 96.8%) and the Kitchen (96.4% and 95.3%) environments, using the LSTM model, we achieved the best results in the "New Person" setting (see Table 2) thanks to the recognition of the activity temporal patterns. The worst results are achieved in the Office environment (91.63% and 89.49%). "Relaxing on couch" and "talking on couch" are discriminated at 100%, perhaps for the stationary character of the activities, while the LSTM model has difficulty in the disambiguation of "talking on phone" and "drinking water" in the Living Room and Office environment probably due to their similarities. For "writing on whiteboard," the LSTM model predicts "talking on phone" in the 20.6% of cases or "random + still" in the 3.6% of cases, thus its accuracy is lower than "talking on phone" accuracy.

The overall activity confusion matrix, presented in Fig. 3, shows the results in the "New Person" setting with the CNN-LSTM model. We can see that "cooking (stirring)," "drinking water," "random + still," "rising mouth with water" and "writing on whiteboard" have lower accuracy than the other activities considering only the 140 frames as an instance.

Thanks to the representation of the skeleton with a 3D matrix, the results obtained with the CNN-LSTM model improve in comparison with LSTM. To evaluate the impact of the proposed approach, different combinations of input matrix have been texted leading to lower performance. For example, by inverting the left leg with the right arm, so to have in the first matrix the two arms, and in the last one the two legs, we got 92.74% of precision and 92.30% of recall against 95.40% and 94.38% of the proposed 3D matrix representation.

5.2 CNN-LSTM results

Considering the CNN-LSTM model, we have an improvement in the results compared to the LSTM model results. This is particularly evident in the Office environment. The lowest results are obtained in the Kitchen that, as previously discussed, has activities with periodic patterns as chopping. The CNN-LSTM model behaves better where the LSTM gets worse. We can see in Table 2 that the CNN-LSTM model has better results in precision in the Bathroom environment with the "random + still" (71.4% vs 94.0%), and in the Bedroom and the Kitchen environments with the "opening container" (82.3% vs 94.0% in the Bedroom and 80.7% vs 85.5% in the Kitchen). There are also different results in the Office environment in precision and recall respectively for the "talking on the phone" (78.2% and 96.3% vs 80.5% and 95.5%) and "writing on whiteboard" (89.4% and 75.7% vs 94.3% and 85.3%).

5.3 Statistical hypothesis test

In general, the model that best predicts unseen data might be the model with the maximum accuracy or minimum error for classification or regression problems. We can trust the model selected with the maximum accuracy or minimum error by applying a statistical hypothesis test. We applied the McNemar's test to check whether the slightest differences we have between the CNN-LSTM model (97.00% of precision and 98.00% of recall) and the LSTM model (95.07% of precision and 96.46% of recall) are significant. The McNemar's test strongly confirmed that the CNN-LSTM model was significantly better than the LSTM model ($\chi^2 = 136026$, $p - value < 0.0001$) at a 95% confidence interval. In short, the results of the CNN-LSTM models were statistically significant at a significance level of 0.05.

5.4 Window size results

Let us now consider the possible impact on the performance of the instances' window size. In order to do so, we made additional experimentation considering other frame windows: 50 and 100 frames. The results are shown in Table 4. With respect to 140 frames, as expected, considering fewer frames yields a decrease in performance (precision and recall). However, in view of the application of the proposed approach in real settings, fewer frames can still be considered since achieving good performance.

5.5 Comparison with the SoA

The CNN-LSTM model achieves, in average, 95.4% and 94.4% on precision and recall. In Table 3, we reported our average results with respect to other approaches in the literature. We must emphasize the fact that we get such results considering instances of 140 frames, while all the other works, reported in Table 3, considered the activity recognition on the entire videos. The shortest video is of 147 frames, while the longest video is of 1961 frames. The average number of frames is about 1181 frames with 595 for standard deviation. Hence, our approach achieves a better performance with respect to all the other cases only considering small video sequences and skeleton data only. The only exception is the work of [17].

Applying the proposed model on the entire videos with the "New Person" setting, we obtained 96.46% of recall and 95.07% of precision with the LSTM model and we obtained 98.00% of recall and 97.00% of precision with the CNN-LSTM model reaching such state-of-the-art results in activity recognition on the CAD-60 dataset. Such results are obtained with a sliding window of 140 frames applied to each video, and by considering, for each classification result, only the output with an accuracy greater than 80%. The result of a classification process is then the most recognized activity. For example, on a video of the activity "drinking water" formed by 1448 instances of 140 frames we considered only the results of classifications with a probability greater than 80%. We obtained 1291 instances that are classified as "drinking water," 12 as "random + still" and 41 as "talking on the phone." The predicted activity is, therefore, "drinking water" as it has been predicted more times over the entire video.

The comparison is made on the state of the art applied to the CAD-60 dataset. The classification in these SoA works is performed on the entire frame sequence of each video using manual features extraction and a classic machine learning algorithm. The latter essentially involves the extraction of the characteristic poses of an activity using mainly clustering to select the significant poses that best describe the activity performed. We want to emphasize instead that the results we have obtained on the single instances are not comparable with the other works. On the contrary, the results obtained by applying the sliding window on the entire video are comparable. Moreover, as a difference of the SoA, we have carried out an automatic extraction of the features that is the basis of the potential of deep learning models. However, a preprocessing phase, which does not include feature selection, is necessary to train and run neural network models.

Table 3 State-of-the-art results on CAD-60 dataset

Algorithm	"New Person"	
	Precision	Recall
Zhu W. et al. [33]	93.2%	84.6%
Faria D.R. et al. [10]	91.1%	91.9%
Shan J. et al. [25]	93.8%	94.5%
Parisi G.I. et al. [21]	91.9%	90.2%
Cipitelli E. et al. [4]	93.9%	93.5%
Khaire P. et al. [13]	93.1%	90.0%
Liu T. et al. [17]	97.97%	95.75%
Our LSTM	95.07%	96.46%
Our CNN-LSTM	97.00%	98.00%

Table 4 Results of our approach using different frame window on CAD-60 dataset with "New Person" setting

Model	"New Person"	
	Precision	Recall
LSTM on 50 frames	**91.21%**	**89.13%**
LSTM on 100 frames	**93.08%**	**91.55%**
LSTM on 140 frames	**95.10%**	**93.88%**
CNN-LSTM on 50 frames	**90.02%**	**88.89%**
CNN-LSTM on 100 frames	**92.22%**	**90.54%**
CNN-LSTM on 140 frames	**95.40%**	**94.38%**

On average, only 4% of the frames for each video were discarded due to lower accuracy. Only two videos, regarding the third user, were not correctly recognized. Respectively, in the Kitchen environment, the "cooking (stirring)" activity was classified as "cooking (chopping)" and, in the Office environment, the "writing on whiteboard" activity was classified as "talking on the phone." We must emphasize that the third user is left-handed and the "cooking (stirring)" and the "cooking (chopping)" as the "writing on whiteboard" and the "talking on the phone" are very similar if we consider the movement of the human skeleton.

Considering the confusion matrix reported in Fig. 3, we can observe that, although on average some activities have a lower recognition rate, we reached 98% of recall and 97% of precision on the entire videos with 140 frames sliding window approach. In this case, we supposed that some instances, i.e., subsequences of the videos, are the most likely to provide relevant information to correctly identify an activity

while others are not. Indeed, this issue has to be taken into account when performing online recognition on sequences with a small number of frames.

5.6 Real setting configuration

The UPA4SAR project aimed at assisting and monitoring elderly people in their homes. Hence, we conducted the experimentation in real houses of the participants. Seven patients participated in the trials interacting with the robot for 2 weeks each. The experiments were performed by the robot in full autonomy, without the presence of an operator. For privacy and security reasons, it was not possible to save any video or audio and the robot had no internet access during the experimentation.

For training the network, we collected data from real patients during preliminary experiments in a laboratory resembling a house environment. The considered activities were, "talking and relaxing on the couch," "watching TV," "working on PC," "ironing," "making coffee" and "talking on the phone."

The robotic system used for experimentation consisted of a Sanbot robot and an Intel NUC (Intel NUC 8i7BEH2, Intel Core i7-8559U 4,5 GHz, 16 GB RAM, 250 GB SSD) for the execution of artificial intelligence algorithms that required computing power. During the daily experiments, a Workflow Manager [5], running on the Intel NUC, planned and scheduled the activities to be performed by the robot.

Among the activities, delivered as services [2] at particular times during the day, the robot was requested to monitor the user activity in order to check whether a specific activity was being performed by the user or not. This request was followed by the user search [26]. The robot searching for the user positioned itself in front of the user and, once identified the user through facial recognition, the robot recorded 10 seconds of video, sending the frames to the Intel NUC to extract the skeleton poses. From the extracted skeleton poses, we applied a sliding window of 140 frames and we classified the activity performed on each instance. The recognized activity is the one with the highest number of recognitions from the ones with the confidence greater than 80%. In case the recognized activity was not the one "expected" the robot performed the recognition process three times leading eventually to a dialogue with the user in the case of mismatch.

The running time for a single 140 frames classification was about 0.015 seconds on the Intel Core i7-8559U 4.5GHz, while it was 2.42 seconds for processing the whole 10 seconds of data. Classification data cannot be reported because for privacy reasons videos were not saved and so it was not possible to get a ground truth.

6 Conclusions

In this work, we presented a CNN-LSTM model for activity recognition working on a matrix representation of the skeleton joints. To handle the spatial dependencies the CNN-LSTM model uses a CNN, while an LSTM is used to deal with the temporal dependencies. The LSTMs are used as memory cells for learning periodic pattern from the sequence. The issues faced during this work are due to the tiny datasets, the RGB-D camera errors and the different activities' speed of motion. The CNN-LSTM model exceeds the speed (short and long term) dependencies and it is made up of two small convolutional layers, a pooling layer and an LSTM to automatically extract spatial patterns from the skeleton data and temporal patterns from the sequences of frames. Regarding preprocessing and feature extraction, our model differs from the others proposed in the state of the art since it automatically extracts the features from the raw data.

The model is applied to short subsequences of the videos to be used for real-time activity recognition. We decided to classify the activity on a sequence of 140 frames that correspond to 4.7s with 30fps. The running time for the classification of a sequence of 140 frames is about 0.015 seconds on Intel Core i7-8559U 4.5GHz. The obtained results were compared with the results of a simple LSTM.

Starting from solving a different problem with respect to the literature, the results of CNN-LSTM approach on the entire videos of CAD-60 (each video is about 45s) with the setting "New Person" show such performance in line with the state of the art (98.00% of recall and 97.00% of precision). The main difference of our model with a classical machine learning approach is that, if we train our model with enough data, we can run the model in a real environment without having to train or tune the parameters and without the need for preprocessing and feature extraction. As future work, we will conduct additional experiments to test the performance of our approach on real HRI experiments and larger datasets.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11370-021-00358-7.

Funding Open access funding provided by Universitá degli Studi di Napoli Federico II within the CRUI-CARE Agreement.

References

1. Baccouche M, Mamalet F, Wolf C (2011) Sequential deep learning for human action recognition. In: International workshop on human behavior understanding, pp 29–39
2. Busetta P, Kuflik T, Merzi M, Rossi S (2004) Service delivery in smart environments by implicit organizations. In: The first annual international conference on mobile and ubiquitous systems: networking and services, MOBIQUITOUS, pp 356–363
3. Choutas V, Weinzaepfel P, Revaud J, Schmid C (2018) Potion: Pose motion representation for action recognition. In: CVPR 2018
4. Cippitelli E, Gasparrini S, Gambi E, Spinsante S (2016) A human activity recognition system using skeleton data from RGBD sensors. Comput Intell Neurosci 2016:4351435
5. Di Napoli C, Rossi S (2019) A layered architecture for socially assistive robotics as a service. In: 2019 IEEE international conference on systems, man and cybernetics (SMC), pp 352–357
6. Donahue J, Anne Hendricks L, Guadarrama S (2015) Long-term recurrent convolutional networks for visual recognition and description. In: IEEE conference on computer vision and pattern recognition, pp 2625–2634
7. Du Y, Fu Y, Wang L (2015) Skeleton based action recognition with convolutional neural network. In: 3rd IAPR Asian conference on pattern recognition (ACPR), pp 579–583
8. Du Y, Wang W, Wang L (2015) Hierarchical recurrent neural network for skeleton based action recognition. In: IEEE conference on computer vision and pattern recognition, pp 1110–1118
9. Ercolano G, Riccio D, Rossi S (2017) Two deep approaches for ADL recognition: a multi-scale LSTM and a CNN-LSTM with a 3d matrix skeleton representation. In: 2017 26th IEEE international symposium on robot and human interactive communication (RO-MAN). IEEE, pp 877–882
10. Faria DR, Premebida C, Nunes U (2014) A probabilistic approach for human everyday activities recognition using body motion from rgb-d images. In: The 23rd IEEE intern. symp. on robot and human interactive communication, RO-MAN. IEEE, pp 732–737
11. Hersh M (2015) Overcoming barriers and increasing independence service robots for elderly and disabled people. Int J Adv Robot Syst 12(8):114. https://doi.org/10.5772/59230
12. Ji S, Xu W, Yang M, Yu K (2013) 3d convolutional neural networks for human action recognition. IEEE Trans Pattern Anal Mach Intell 35(1):221–231
13. Khaire P, Kumar P, Imran J (2018) Combining CNN streams of RGB-D and skeletal data for human activity recognition. Pattern Recognit Lett 115:107–116
14. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907
15. Li C, Wang P, Wang S, Hou Y, Li W (2017) Skeleton-based action recognition using LSTM and CNN. In: 2017 IEEE international conference on multimedia & expo workshops (ICMEW). IEEE, pp 585–590
16. Li Y, Lan C, Xing J, Zeng W, Yuan C, Liu J (2016) Online human action detection using joint classification-regression recurrent neural networks. In: 14th European conference on computer vision – ECCV, Part VII. Springer, pp 203–220
17. Liu T, Wang J, Hutchinson S, Meng MQH (2019) Skeleton-based human action recognition by pose specificity and weighted voting. Int J Soc Robot 11(2):219–234
18. Luvizon DC, Picard D, Tabia H (2018) 2d/3d pose estimation and action recognition using multitask deep learning. arXiv preprint arXiv:1802.09232
19. Nunez JC, Cabido R, Pantrigo JJ, Montemayor AS, Velez JF (2018) Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. Pattern Recognit 76:80–94
20. Ordóñez FJ, Roggen D (2016) Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors 16(1):115
21. Parisi GI, Weber C, Wermter S (2015) Self-organizing neural integration of pose-motion features for human action recognition. Front Neurorobotics 9:3
22. Rossi S, Rossi A, Dautenhahn K (2020) The secret life of robots: perspectives and challenges for robot's behaviours during non-interactive tasks. Int J Soc Robot 12:1265–1278
23. Rossi S, Staffa M, Bove L, Capasso R, Ercolano G (2017) User's personality and activity influence on hri comfortable distances. Social Robotics: 9th international conference, ICSR 2017, Tsukuba, Japan, November 22–24, 2017, proceedings. Springer International Publishing, Cham, pp 167–177
24. Sasabuchi K, Ikeuchi K, Inaba M (2018) Agreeing to interact: understanding interaction as human-robot goal conflicts. Companion of the 2018 ACM/IEEE international conference on human-robot interaction, HRI '18. Association for computing machinery, New York, NY, USA, pp 21–28
25. Shan J, Akella S (2014) 3d human action segmentation and recognition using pose kinetic energy. In: IEEE international workshop on advanced robotics and its social impacts. IEEE, pp 69–75
26. Staffa M, De Gregorio M, Giordano M, Rossi S (2014) Can you follow that guy? In: 22th European symposium on artificial neural networks, ESANN 2014, Bruges, Belgium, April 23-25, 2014, pp 511–516
27. Sung J, Ponce C, Selman B, Saxena A (2012) Unstructured human activity detection from rgbd images. In: 2012 IEEE international conference on robotics and automation, pp 842–849
28. Sung J, Ponce C, Selman Bea. CAD-60 and CAD-120. http://pr.cs.cornell.edu/humanactivities/data.php
29. Ullah A, Ahmad J, Muhammad K, Sajjad M, Baik SW (2017) Action recognition in video sequences using deep bi-directional lSTM with CNN features. IEEE Access 6:1155–1166
30. Yan S, Xiong Y, Lin D (2018) Spatial temporal graph convolutional networks for skeleton-based action recognition. arXiv preprint arXiv:1801.07455
31. Zhang S, Yang Y, Xiao J, Liu X, Yang Y, Xie D, Zhuang Y (2018) Fusing geometric features for skeleton-based action recognition using multilayer LSTM networks. IEEE Trans Multimedia
32. Zhu W, Lan C, Xing J, Zeng W, Li Y, Shen L, Xie X (2016) Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In: Proceedings of the AAAI conference on artificial intelligence, pp 3697–3703
33. Zhu Y, Chen W, Guo G (2014) Evaluating spatiotemporal interest point features for depth-based action recognition. Image Vision Comput 32(8):453–464