Multimedia Tools and Applications
https://doi.org/10.1007/s11042-022-14075-5

Real-time human action recognition using raw depth video-based recurrent neural networks

Adrián Sánchez-Caballero[1] · David Fuentes-Jiménez[1] · Cristina Losada-Gutiérrez[1]

Received: 7 April 2021 / Revised: 4 October 2022 / Accepted: 10 October 2022

Abstract

This work proposes and compare two different approaches for real-time human action recognition (HAR) from raw depth video sequences. Both proposals are based on the convolutional long short-term memory unit, namely ConvLSTM, with differences in the architecture and the long-term learning. The former uses a video-length adaptive input data generator (*stateless*) whereas the latter explores the *stateful* ability of general recurrent neural networks but is applied in the particular case of HAR. This stateful property allows the model to accumulate discriminative patterns from previous frames without compromising computer memory. Furthermore, since the proposal uses only depth information, HAR is carried out preserving the privacy of people in the scene, since their identities can not be recognized. Both neural networks have been trained and tested using the large-scale NTU RGB+D dataset. Experimental results show that the proposed models achieve competitive recognition accuracies with lower computational cost compared with state-of-the-art methods and prove that, in the particular case of videos, the rarely-used stateful mode of recurrent neural networks significantly improves the accuracy obtained with the standard mode. The recognition accuracies obtained are 75.26% (CS) and 75.45% (CV) for the stateless model, with an average time consumption per video of 0.21 s, and 80.43% (CS) and 79.91%(CV) with 0.89 s for the stateful one.

Keywords ConvLSTM · Action recognition · Depth maps · Video-surveillance

This work has been partially supported by the Spanish Ministry of Science and Innovation under projects EYEFUL (PID2020-113118RB-C31/C33), by the Community of Madrid under project CONCORDIA (CM/JIN/2021-015) and by the University of Alcalá under project ARGOS+ (PIUAH21/IA-016)

✉ Cristina Losada-Gutiérrez
cristina.losada@uah.es

Adrián Sánchez-Caballero
adrian.sanchez@uah.es

David Fuentes-Jiménez
d.fuentes@edu.uah.es

[1] Department of Electronics, University of Alcalá, Ctra. Madrid-Barcelona, km. 33600, 28805, Alcalá de Henares, Spain

Published online: 28 October 2022

1 Introduction

Human action recognition (HAR) has received great attention from computer vision researchers in the last decade due to the broad variety of possible applications in different areas, such as automated video surveillance [26], health care services [63], human-computer interaction [48, 59] or autonomous driving.

The firsts works in HAR were based on analyzing RGB image sequences [14, 16, 43] through different approaches, first with handcrafted feature descriptors [4, 5, 30, 46] and, later, as in other areas [9], with deep learning-based techniques [2, 23, 53, 64] or a combination of classical features and deep neural networks (DNN) [3]. In general, handcrafted feature-based methods perform well on small datasets, whereas in the case of large datasets the performance of DNNs is better, but with largest computational costs.

In recent years, due to the rise of affordable RGBD cameras [15], many studies have focused on using this information for HAR. RGBD sensors provide images with rich 3D structural information of the scene with benefits compared to RGB videos such as lighting changes invariance and preservation of personal-privacy, which generates great interest in some domains such as video surveillance [13, 60]. Furthermore, depth maps allow estimating the human joint positions [51] also known as the 3D skeleton, which supposes itself a different data modality for HAR [44].

As in the case of RGB-based HAR works, the first depth-based studies employed methods based on handcrafted descriptors [28, 41, 65, 67], but eventually works using deep learning became the main approach [47]. Besides, DNN-based approaches have been proved [69] to be more robust and suitable for challenging large datasets than handcrafted features-based methods, but with much higher computational costs, which makes it difficult to use them in real-time applications.

The use of DNNs for HAR requires encoding the spatio-temporal information [7]. To do that, there are several approaches that create ad-hoc representations such as depth motion maps [73] or dynamic images [72, 74, 76, 79, 81], whereas other works use specific DNNs, such as the 3D convolutional neural networks (3DCNNs) [36, 47, 54, 80] or the recurrent neural networks (RNN) [11, 31, 49, 50], that processes video-sequences. A particular RNN which solves the exploding or vanishing gradient problem is the long short-term memory (LSTM) [17], which can successfully learn patterns in long sequences like videos by stacking several layers. However, LSTMs cannot directly learn spatio-temporal features from an image sequence. This limitation was overcome by replacing the Hadamard product of the original LSTM with the convolution operation (ConvLSTM) [82].

In this work, we propose and analyze two novel implementations of recurrent neural networks based on ConvLSTMs that receive only depth information as input for real-time HAR for video-surveillance applications. Both models are end-to-end trainable, and their architectures have been optimized for real-time operation. The use of only depth data allows preserving privacy since it does not allow recognizing their identity. Furthermore, input data does not require any prior calculation such as skeleton positions, optical flow or dynamic images, widely used in the literature, that notably increase the computational cost. As it is shown in Section 5, the proposal allows obtaining results comparable to the state-of-the-art in the widely used NTU RGB+D [49] dataset, using only the depth video sequences.

The main contributions of the present work are the following:

1. There have been proposed two different architectures for HAR using only raw depth data based on ConvLSTMs, including a novel implementation of the unusually used *stateful* capability for LSTM layers, in order to fully exploit the long-term memory.

Both proposed architectures are end-to-end, so the raw depth data are fed to the network input without any preprocessing, in contrast to other approaches that require computing 3D-Skeletons or Depth Dynamic Images, which highly increases the computational cost.

2. A careful design of the networks architecture has been done for real-time execution, by using batch normalization [22], LeakyReLU activations [39] and replacing the fully connected layers at the top of the neural network with an average pooling , which drastically reduces the number of parameters and improves model generalization.
3. Training procedures have been configured to exploit the ConvLSTM properties by using different strategies [6, 57]. To reduce the usual subjectivity in the learning rate choice, we use a learning rate range test to estimate optimal values once the batch size is set. Furthermore, a cyclical learning rate [56] allows improving convergence and reducing over-fitting. Moreover, a video-length-adaptive input data generator has been designed to fully exploit the temporal dimension of long videos.
4. This work also includes a comparison with several previous approaches, obtaining comparable accuracy results with much lower computational costs.

It is noteworthy that, to the best of our knowledge, there have not been studies where depth raw videos are directly fed to an RNN or, more specifically, to an LSTM for HAR. It is still more unlikely to find a study that uses an LSTM network in *stateful* operation mode for action recognition, where all the potential of this architecture is exploited.

The rest of this paper is organized as follows. In Section 2, previous works related to HAR are explained, giving special emphasis to depth-based ones. Section 3 includes the description of the analyzed RNNs architecture. Subsequently, Section 4 describes in detail the training stage. Section 5 shows and discusses the main experimental results. Finally, the paper is concluded in Section 6.

2 Related work

As stated in the Introduction, initial works on HAR used visual images recorded with standard RGB cameras and methods with handcrafted features [4, 5, 30, 46]. Motivated by the success in image processing tasks [9, 55, 66], deep learning methods began to be applied also for videos, typically with architectures such as 3D convolutions (3DCNN) and RNNs [2, 23]. In particular, a very common framework found in the literature is the two-stream neural network [12, 53, 70], where one stream operates on RGB frames whereas the other tries to learn motion using optical flow as input. The optical flow is pre-computed with handcrafted methods, which involves a high computational cost. To alleviate this, N. Crasto et al. [8] proposed using a feature-based loss that mimics the motion stream in the two-stream 3D CNN and removes the need for using optical flow. Most of the deep learning-based works on HAR with RGB videos put the effort into solving the problem of how to treat efficiently the temporal dimension of videos. The previous methods use the third dimension in convolutions to deal with the extra dimension.

However, the existence of long videos, which is inherent to certain human actions, may not allow the neural network to process discriminative features due to memory limitations, failing to recognize these actions. This long-term problem in 3DCNNs is especially studied in [64], where they propose to increase the temporal size of the input at the cost of reducing spatial resolution, or in [61] by building motion maps to represent motion from videos of any length. Another alternative is to use RNNs such as LSTM units to learn temporal patterns from features previously extracted with spatial CNNs. This scheme together with a temporal-wise attention model and a joint optimization module to fuse output features is used in [71]. Other researchers have used deep learning to estimate optical flow [21] instead of using traditional and computationally expensive methods. Additionally, novel spatial and temporal pyramid modules for CNN are proposed and aggregated to a Spatial-Temporal Pyramid Network (S-TPNet) in [88] to learn effective spatio-temporal pyramid representations of videos.

Instead of extracting human pose estimations from RGB images, many other works [18, 19, 29, 32, 33, 62, 75] combine directly RGB with depth modality like 3D skeleton or depth maps to leverage the advantages of both types of data.

Regarding depth-based works for HAR, there exist different approaches depending on how 3D information is treated. As explicitly mentioned in [81], depth-based videos are usually divided into three categories depending on the nature of input: human skeleton-based, raw depth-video-based and a combination of both. The evolution and progress of approaches on these three categories have been affected by the growth of deep learning in the last years, especially in computer vision, leading most recent studies to use this technique. In this regard, P. Wang et al. [77] elaborated a very complete survey of recent studies using deep learning in human motion recognition tasks.

In the first category, 3D positions of human body skeletons must be previously extracted from the depth map in each frame or by using MOCAP systems. There are many different approaches to how the 3D skeleton joint positions can be managed as, for instance, computing multiple joint angles [42], extracting discriminative parts for each human action [20] or finding the best viewpoint for recognition as in [35]. Skeleton joint positions, or any other data derived from them, are also fed into neural networks in most recent papers, mainly through RNN-based methods [34, 40, 42, 58, 68, 86], but also with CNNs [10, 89].

Secondly, depth maps are directly used as input to the model. Different descriptors have been proposed as inputs for the classification process, like in [1, 37, 41, 84]. X. Yang et al. [85] proposed Depth Motion Maps (DMM) to represent depth videos through a pseudo coloring image. Later, DNN-based architectures with DMMs as input [73, 74] improved prior results. Alternatively to DMMs, in [72, 78], suggested using three pairs of images for video representation using bidirectional rank pooling. Y. Xiao et al. [81] have recently worked with multi-view dynamic images, reaching state-of-the-art results in the raw depth maps modality.

Finally, some researchers chose to use both 3D skeleton positions and depth maps and reached good results [45, 50], taking the benefits from both modalities at the cost of an increase in model complexity. Indeed, the combination of these two types of data is more often used with traditional hand-crafted feature algorithms.

A recent comparative review of action recognition methods [69] asserts that skeletal data-based models have achieved better accuracy and robustness than depth-based ones. Nevertheless, 3D skeleton joints have some known drawbacks: general information loss, potential failures of 3D position extraction and the impossibility of action detection involving human-object interactions. In addition, 3D skeleton joints can not be directly extracted with a camera, unless a MOCAP system is used, which is not plausible in most applications. On the other hand, depth-based techniques are more similar to how human vision works but with extra 3D information and can be recorded with a camera and immediately used as input to deep learning models without any intermediate calculation. Studies related to this modality are valuable in research fields such as computer vision and scene understanding.

3 Proposed RNN architectures

As explained before, one of the most common approach for HAR is based on the LSTM [17] layer. It is characterized by including a memory cell C_t or cell state, which is modified over time steps through three different gates: input i_t, forget f_t and output o_t, until they build a final state H_t. This algorithm allows LSTM networks to model long-term dependencies.

Additionally, in [82] a modified LSTM layer is proposed, namely the convolutional LSTM or ConvLSTM, and applied directly to videos for weather forecasting. The expressions that model this approach are shown in (1), where, $\circ$ denotes the Hadamard product and $*$ the convolutional operator.

The main difference between ConvLSTM and regular LSTM is that, instead of using 1D arrays, all the input, output, gates and hidden states are 3D tensors where the 2 extra dimensions correspond to the spatial dimensions. Further details and mathematical explanations are presented in [82].

$$\begin{aligned}
i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i)\\
f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f)\\
C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c)\\
o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o)\\
H_t &= o_t \tanh(C_t)
\end{aligned} \tag{1}$$

The showed structure allows ConvLSTM layers to encode both spatial information, as CNNs do, and temporal patterns extracted from previous frames. This makes ConvLSTM a good choice for modelling spatio-temporal sequences like videos, removing any type of previous encoding or preprocessing.

The short-term memory of an LSTM layer is represented by the cell state C_t, which in the case of ConvLSTM is a 3D tensor, whereas the long-term memory is reflected in the trainable weights inside the gates. However, short-term memory is the novel property LSTMs introduce. Ideally, the cell state will not be reset until the entire time sequence is fed to the network. Thus, the cell state can contain full-sequence information, but in the real world, this situation is not always feasible. Training data are provided to neural networks in batches with sizes that are restricted by the CPU/GPU memory capacity of computers. When the used data consist of videos, it is necessary to find a balance between the number of frames in each input sample (the bigger, the more long-term dependencies our model will extract) and samples in each batch (generally the more, the better the model will generalize and avoid over-fitting), both subjected to the hardware memory limitation.

The LSTM *stateless* mode resets its cell state after each batch is processed and the weights are updated. Most studies use LSTMs in stateless mode since these layers usually operate on already extracted features or simplified data, so the memory limit is not significant. However, there exists a solution to this limitation through what is called the *stateful* mode of an LSTM [24]. With this mode, the LSTM layers preserve the cell state from the previous batch, removing any memory restriction. This property allows LSTMs to handle videos of very different lengths as usually happens with HAR samples, where the information from previous frames can be extremely useful.

These properties are also present in the convolutional version of the LSTM layer, ConvLSTM, and therefore, they can be applied to videos. In this paper, we make a performance comparison between stateless and stateful networks on a challenging depth-based HAR dataset [49].

3.1 Stateless ConvLSTM network

LSTM stateless mode is set by default in most machine learning libraries and it is usually omitted in the papers that use this architecture. This operation mode does not require any particular data preparation (as opposed to the stateful mode) and performs well in many cases.

The stateless ConvLSTM network proposed in this work contains two stages: a recurrent block, which extracts features directly from the video frames, and a decision block with convolutional and pooling layers. Furthermore, the network contains two parallel branches (main branch and support branch) that are fused afterwards through addition. Figure 1 shows a general block representation of this architecture where both branches can be seen.

The main branch is composed of 4 stacked ConvLSTM layers with Batch Normalization (BN) after each one (see Fig. 1). BN reduces internal covariance, contributing to speed up the training. The ConvLSTM layers have 32, 32, 128 and 256 3 × 3 filters, respectively. The last recurrent layer removes the temporal dimension and leads to a convolutional layer with 128 3 × 3 filters. At this point, the support branch is added up to the main one.

The support branch (lower branch in Fig. 1) has fewer layers but bigger filters. The input passes through two ConvLSTM layers with 8 7 × 7 and 16 5 × 5 filters, respectively. Next, there is a convolutional layer with 128 3 × 3 filters before the addition to the main branch. A more complete description can be seen in Table 1 including kernel and strides properties of convolutions. The employed activations are LeakyReLU (*Leaky Rectified Linear Unit* [39]) functions. This type of function follows the expression shown in (2), with $\alpha = 0.3$, and it has proven to be more efficient [83] compared with the standard ReLU activation.

$$\begin{aligned} f(x) &= x \quad if \quad x \geq 0 \\ f(x) &= \alpha \cdot x \quad if \quad x < 0 \end{aligned} \tag{2}$$

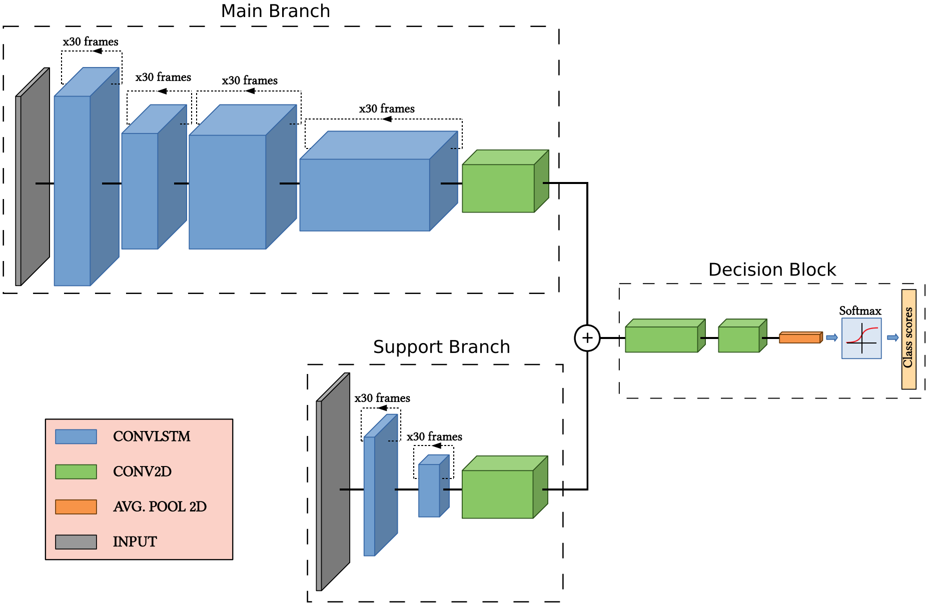


Fig. 1 Schematic of the proposed stateless ConvLSTM network for HAR

Table 1 Summary of stateless ConvLSTM network architecture

	Layer	Parameters	Output size
Support branch	Input	-	(30, 64, 64, 1)
	ConvLSTM	k=(7, 7), s=(2, 2)	(30, 29, 29, 8)
	Batch Normalization	-	
	ConvLSTM	k=(5, 5), s=(2, 2)	(1, 13, 13, 16)
	Batch Normalization	-	
	Conv2D	k=(1, 1), s=(2, 2)	(13, 13, 128)
Main branch	Input	-	(30, 64, 64, 1)
	ConvLSTM 1	k=(3, 3), s=(1, 1)	(30, 64, 64, 32)
	Batch Normalization	-	
	Activation	LeakyReLU	
	ConvLSTM 2	k=(3, 3), s=(2, 2)	(30, 31, 31, 32)
	Batch Normalization	-	
	Activation	LeakyReLU	
	ConvLSTM 3	k=(3, 3), s=(1, 1)	(30, 64, 64, 128)
	Batch Normalization	-	
	Activation	LeakyReLU	
	ConvLSTM 4	k=(3, 3), s=(2, 2)	(1, 15, 15, 256)
	Batch Normalization	-	
	Activation	LeakyReLU	
	Conv2D	k=(3, 3), s=(1, 1)	(13, 13, 128)
Decision block	Add Main Branch + Support Branch	-	(13, 13, 128)
	Activation	LeakyReLU	
	Conv2D	k=(3, 3), s=(2, 2)	(6, 6, 128)
	Batch Normalization	-	
	Activation	LeakyReLU	
	Conv2D	k=(1, 1), s=(1, 1)	(6, 6, 60)
	Global Average Pooling 2D	-	(1, 1, 60)
	Activation	Softmax	

k: kernel size, s: stride

These convolutional layers start the decision block. After the addition, there are other two convolutional layers with 128 3 × 3 filters that precede 2D global average pooling and softmax activation, producing a vector that includes the class likelihoods standardized to the unit.

3.2 Stateful ConvLSTM network

The stateful ConvLSTM architecture is slightly simpler than stateless. It consists of a single branch and the structure is very similar to the main branch of the Stateless ConvLSTM network.

The recurrent block contains four ConvLSTM layers with 32, 64, 128 and 256 3 × 3 filters and BN after each one. After the third ConvLSTM layer, a regular 2D convolution has been placed to reduce the number of features and, consequently, the overall network parameters. The decision block is composed of two convolution layers with 128 3 × 3 filters, followed by BN and Leaky ReLU activations as in the stateless architecture. Next, a final convolution reduces the number of features to match the number of classes, which precedes Global average pooling and softmax activation. A detailed description of every layer of the stateful model is reported in Table 2.

The complexity of the stateful network falls essentially on the particular training and data arrangement, which is explained in the following section.

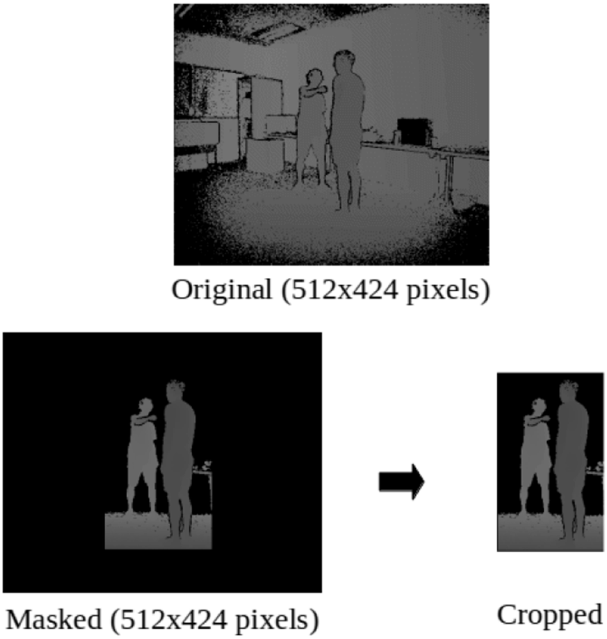
4 Training stage

We use the NTU RGB+D dataset [49], which is one of the largest available human action datasets that include videos in RGB, depth-maps, 3D-skeletons and infrared. It contains 56880 samples with one or more subjects performing a particular action. Videos have been recorded using three simultaneous *Microsoft Kinect II* [87] sensors and, thus, providing multi-view scenes. Resolution of RGB videos is 1920 × 1080 pixels, whereas for depth-map and infrared videos it is 512 × 424 pixels. 3D skeletal data provide three-dimensional locations of 25 main human body joints for every frame. The database contains 60 human actions within three well-defined groups: daily actions, medical conditions, and mutual actions.

This work only uses the depth-map modality and adapts the two data evaluations suggested in [49] by which the training and test are divided: cross-subject (CS) and cross-view (CV). The provided images are foreground masked versions to improve the compression ratio of files and alleviate the processes of downloading and managing such a big amount of data. They are then cropped to the movement area of the action, as shown in Fig. 2. Finally, the model itself takes the cropped images and re-scales them to 64 × 64 pixels to build the network input.

Table 2 Summary of stateful ConvLSTM network architecture, being k: kernel size and s: stride

Layer	Parameters	Output Size
Input	-	(8, 64, 64, 1)
Stateful ConvLSTM 1	k=(3, 3), s=(1, 1)	(8, 64, 64, 32)
Batch Normalization	-	
Stateful ConvLSTM 2	k=(3, 3), s=(2, 2)	(8, 31, 31, 64)
Batch Normalization	-	
Stateful ConvLSTM 3	k=(3, 3), s=(1, 1)	(8, 31, 31, 128)
Batch Normalization	-	
Stateful ConvLSTM 4	k=(3, 3), s=(2, 2)	(1, 15, 15, 256)
Batch Normalization	-	
Conv2D 1	k=(3, 3), s=(2, 2)	(7, 7, 128)
Batch Normalization	-	
Activation	LeakyReLU	
Conv2D 2	k=(3, 3), s=(2, 2)	(3, 3, 128)
Batch Normalization	-	
Activation	LeakyReLU	
Conv2D 3	k=(1, 1), s=(1, 1)	(3, 3, 60)
Global Average Pooling 2D	-	(1, 1, 60)
Activation	Softmax	


Original (512x424 pixels)

Masked (512x424 pixels)

Cropped

Fig. 2 Example of one original video frame (up) from the NTU RGB+D dataset [49] and the cropping process of its masked version (down)

Due to the strongly different strategies followed by the two proposed methods, they need different training procedures. On one hand, the stateless ConvLSTM focuses on how the frames to be processed from the video are selected depending on their length and follows a more conventional training, on the other hand, the training of stateful ConvLSTM needs a special treatment due to the existence of a state that can be updated and reset throughout the processing of each video. Below it is presented in more detail the processes for training both proposed models.

4.1 Training of the stateless ConvLSTM network

Data arrangement concerns how training and test samples are generated from data and fed to the model, and usually has a big influence not only on the ability to train a neural network but also on the final accuracy of the model. It is required a good understanding of the network architecture and taking into account the dataset properties to get an optimum data arrangement.

The temporal sequences that are fed to our network are 30 frames long, which corresponds to 1 second of a video. This value has been experimentally chosen following the previously mentioned balance between the number of frames in the input sample and batch size, which is set to 12, but also subject to the hardware memory limitation. However, in the NTU dataset, video lengths go from 26 to 300 frames. There are only a few videos shorter than 30 frames. In this case, some of the final frames have been smoothly repeated until the desired length is reached. When videos are longer, the starting point of the 30-frames temporal window is randomly selected and, in the case of very long videos, it also skips frames uniformly to cover a wider video range (see Fig. 3 for an explanatory illustration). These

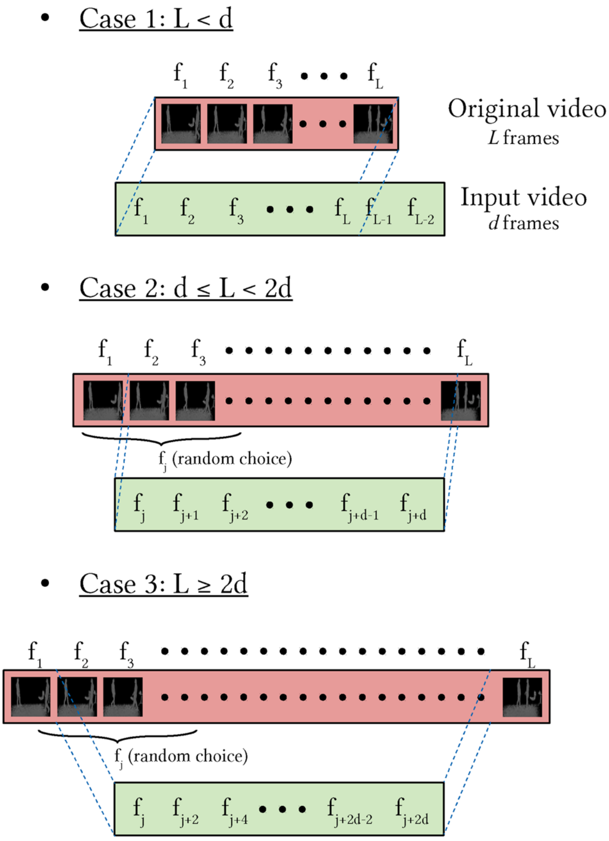

• Case 1: L < d
f1 f2 f3 • • • fL
Original video
L frames
Input video
d frames
f1 f2 f3 • • • fL fL-1 fL-2
• Case 2: d ≤ L < 2d
f1 f2 f3 • • • • • • • • • • • fL
fj (random choice)
fj fj+1 fj+2 • • • fj+d-1 fj+d
• Case 3: L ≥ 2d
f1 f2 f3 • • • • • • • • • • • • • • • • fL
fj (random choice)
fj fj+2 fj+4 • • • fj+2d-2 fj+2d

Fig. 3 Window selection scheme performed by the stateless input data generator

strategies have proven to achieve a better performance of the network for action recognition without increasing the number of input frames.

In Fig. 3, being L the number of frames of a video from the dataset, the input data generator select d frames to build the input video that will be fed to the neural network. When $L < d$, the original video is extended by repeating the last frames until reaching d frames. In the second case, an initial frame f_j is randomly chosen provided that the d subsequent frames fit inside the original video. Finally, when $L \geq d$, the initial randomly chosen frame f_j is followed by $\{f_{j+2}, f_{j+4}, ..., f_{j+2d}\}$ to cover a region of size $2d$ in the original video.

The training method has been as follows. First, a learning rate range test has been performed to find the optimum interval of values, as suggested in [56] when using a cyclical learning rate schedule. As mentioned in [6], there exists a dependency between batch size and learning rate, so we have first set the batch size to 12 and then perform the learning rate range test (see Fig. 4). From the results of this range test, we choose a customized cyclical schedule, which can improve accuracy with faster convergence. In the first 21 epochs, the learning rate moves linearly between a minimum value of 8×10^{-5} and a maximum of 9.8×10^{-4}. After that, boundaries are reduced to 10^{-5} and 10^{-4}, respectively. Finally, after epoch 44 the learning rate is fixed to the minimum 10^{-5} until training completes 48 epochs. The algorithm *Adam* [27] has been used as optimizer. This algorithm performs a

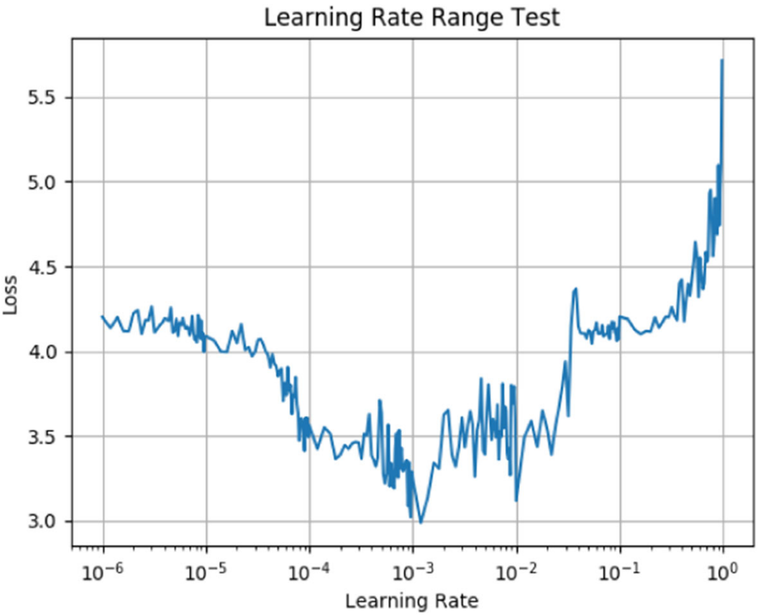


Fig. 4 A learning rate range test performed for the stateless ConvLSTM on NTU RGB+D dataset (CV evaluation) with a batch size of 12. The interval of values where the loss function decreases define the optimal range for the learning rate. In this figure, it would be between 10^{-5} and nearly 10^{-3}

stochastic gradient descent with an adaptive learning rate computed from estimations of first and second moments of the gradients, and it has proven to achieve fast convergence and be computationally efficient with large models and datasets.

Due to the large training times, it has been used a checkpoint technique in training that continuously saves the model weights when validation accuracy improves. Thus, we take the best model between the former 48 epochs and extend training on 27 more epochs using an initial learning rate of 2×10^{-4} that is reduced by half after 4 epochs without accuracy improving.

The followed learning rate schedule together with the training and validation curves are shown in Figs. 5 and 6 for recognition accuracy and loss function, respectively. Here it can be seen how the variation of learning rate affects accuracy and loss function curves. For instance, the big step at epoch 48 shown in both accuracy and loss function appears due to a significant change in learning rate meaning to find a better minimum of the loss function and to reduce over-fitting.

4.2 Training of the stateful ConvLSTM network

Training the ConvLSTM in stateful operation mode requires data preparation as videos have to be sorted by their lengths. This is necessary for the neural network to know where an action ends, and at this point, *reset states* so a new state starts for the next sequences.

Therefore, a video-length analysis of the dataset is required to ensure data balance for training. A distribution of the video-lengths in NTU RGB+D dataset is shown on the left graphic in Fig. 7 for CS evaluation. It shows big differences of video lengths, which range from 26 to 300 frames, but most of them fall into the 44-90 frames region. Thus, we selected

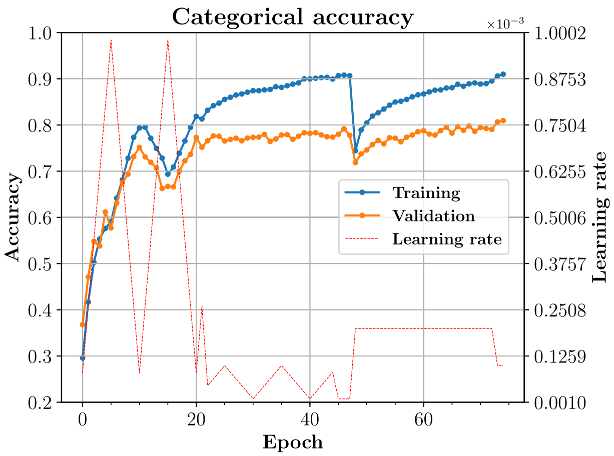


Fig. 5 Curves of training and validation accuracies for the stateless ConvLSTM network. In addition, the learning rate schedule used along the epochs is shown

a customized set of bin edges in order to get a slightly more uniform distribution, which can be seen on the right graphic in Fig. 7.

The left limit of every bin on the right graphic in Fig. 7 is chosen to be the length of the videos inside that bin. For example, a video of 46 frames is reduced to 40 frames and one of 300 to 208. These discrete lengths are chosen to be multiple of 8, which is set as the number of frames in each temporal window or unit clip that is fed to the neural network at each step. Thus, videos inside the bin of 112 frames will have 14 pieces of 8 frames, i.e. the neural network has to look through 14 different windows until the 112 frames are reached.

Every time the network processes one of these windows, it is able to update weights. If we let the network do this with every window of a video, validation metrics will behave abnormally and a strong over-fitting will appear. To solve this, we make the network process the first half of the video without making weight updates, but preserving the cell states, and

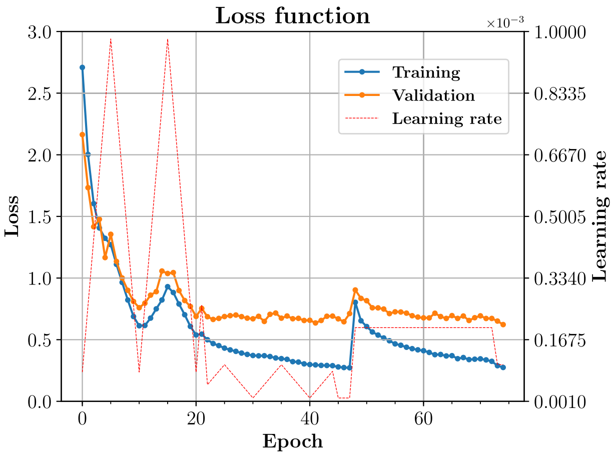


Fig. 6 Curves of training and validation loss function for the stateless ConvLSTM network. In addition, the learning rate schedule used along the epochs is shown

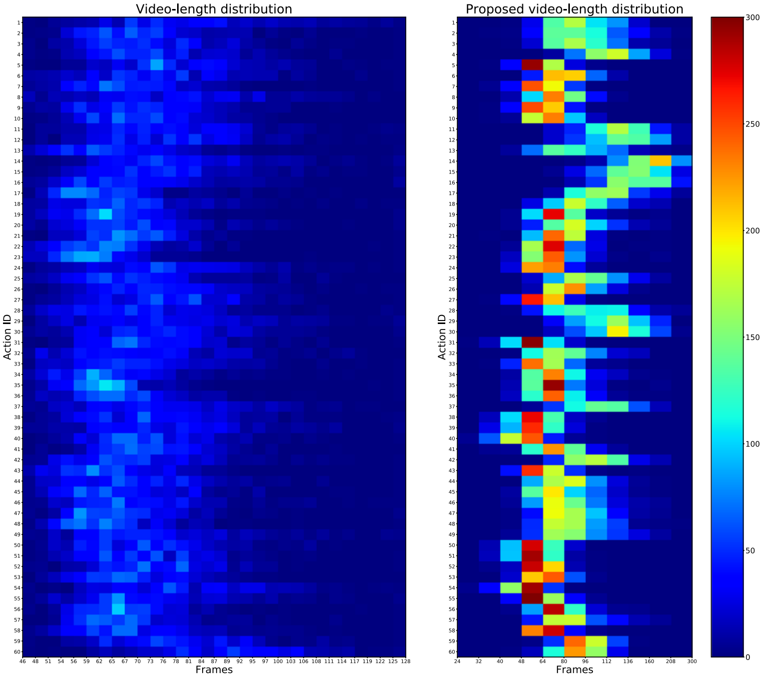


Fig. 7 2D histogram for video-length per class distribution for the NTU depth database training data (CS evaluation), with automatic bin edges on the left (Note the maximum value of 300 frames corresponds to videos in actions 14, 16 and 17) and with customized bin edges on the right

then train on the windows that belong to the second half using here the information gained from the previous frames.

Similarly, we have computed both training and testing metrics considering that late predictions (final windows) are more reliable than the initial ones, where the state does not contain enough information yet. Therefore, for each video, a weighted average is performed using per-window predictions. The distribution of weights $w(t)$ follows the expression shown in (3).

$$w(t) = Nt^a \tag{3}$$

where t is the window number and N, a normalization constant. We chose $a = 3$, whereas the value of N is video-dependent and adopts the expression T^{-a}, being T the total number of windows in a sequence.

We have found that training of the stateful network is more sensitive to learning rate changes than the stateless one. Therefore, to obtain a non-divergent validation loss, we experimentally found some valid learning rate values. Due to the unusual characteristics of this training, the learning rate range test is not used here. The batch size has been set to 6. As in stateless mode, we used Adam as optimizer and a 25%-rate dropout right before the decision block to reduce over-fitting.

We have experimentally observed that small learning rate values are needed to minimize model divergence in training. The applied learning rate schedule for stateful training has been as follows. The initial learning rate is set to 9×10^{-5}, then diminished to 3×10^{-5} in epoch 4, to 8×10^{-6} in epoch 8, to 4×10^{-6} in epoch 15 and, from here, divided by 2 every 4 epochs until complete a total of 25 epochs. The learning rate schedule can be seen in Fig. 8 together with the recognition accuracy curves of training and validation or in Fig. 9, where curves of training and validation loss functions are also shown. In these figures, it can be seen that the stateful ConvLSTM network reaches convergence faster than the stateless mode, but with higher computational time per epoch. It is also noteworthy that a relatively small initial value like 9×10^{-5} for the learning rate still causes a big divergence in the validation curve at first epoch (see Fig. 9). This erratic behavior has been observed in different curves of training at some precise epochs, proving that the stateful operation of the neural network is especially sensitive to the learning rate size.

5 Experimental results

The NTU RGB+D [49] dataset has also been used for the test phase of the proposed methods. The authors of this dataset suggest two different evaluations: cross-subject (CS), where 40 320 samples recorded with 20 subjects are dedicated for training and 16 560 samples with 20 different subjects for test; and cross-view (CV), where 37 920 videos were recorded with 2 cameras from different viewpoints and 18 960 videos from a third different viewpoint for test. Results of both proposed models are shown and analyzed below.

The whole analysis in this work, including training and prediction tests, have been implemented using Tensorflow and the Keras API for Python on a NVIDIA GeForce GTX 1080 with 8 GB and an Intel(R) Core(R) i7-7700 CPU at 3.60 GHz.

5.1 Recognition performance analysis

The confusion matrices for the stateless and stateful model are shown in Fig. 10. As it can be seen, both models exhibit a reasonable performance in the recognition rate over the entire 60 classes of the dataset, and no significant confused actions appear.

In order to further analyze this performance, a more detailed study of the prediction quality of the models has been made, and it is summarized in Table 3 for the cross-subject evaluation. Here they are shown the top 10 recognized actions together with the 10 worst

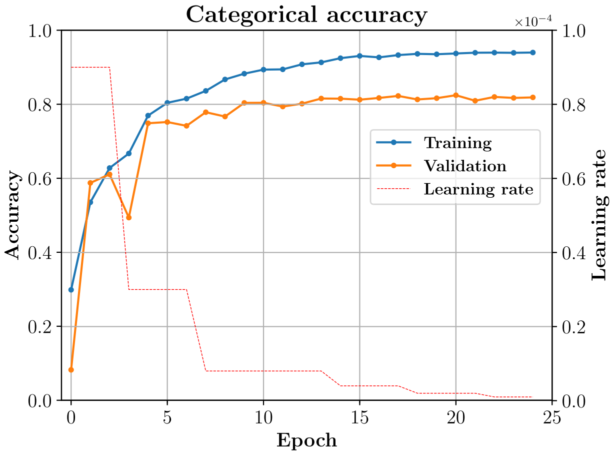


Fig. 8 Curves of training and validation accuracies for the stateful ConvLSTM network. In addition, the learning rate schedule used along the epochs is shown

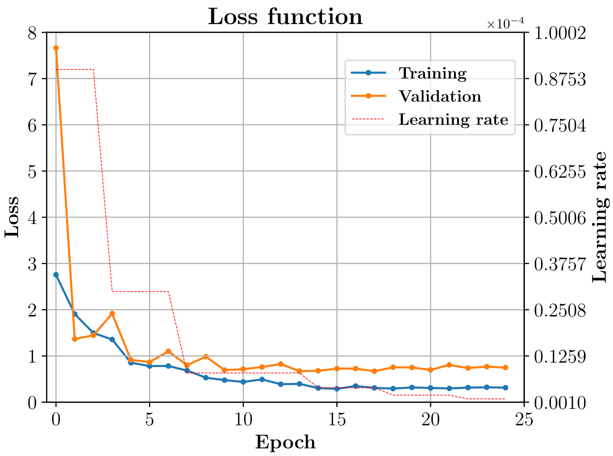


Fig. 9 Curves of training and validation loss functions for the stateful ConvLSTM network. In addition, the learning rate schedule used along the epochs is shown

classified one for both stateless and stateful networks. Also, the average accuracy of these top 10 classes is given for easier comparison.

Regarding the stateless ConvLSTM model, almost all the top 10 recognized actions present an accuracy higher than a 90%. On the other hand, this model finds some difficulties to classify actions like *put on a shoe*, confused with *take off a shoe*, or *reading*, with *writing*, among others. Analyzing the characteristics of these classes it can be seen that correspond to actions that involve similar, short motions and small objects that can not be correctly seen in depth images, thus, this conducts to classification errors and confusion.

In regards to the stateful network, it overcomes the stateless version both within the top 10 recognized and top 10 confused, with some minor exceptions like classes *writing* or *headache*, which slightly decrease their accuracy percentage. On the whole, the top 10 confused actions improve their recognition rate in almost 5% and the top 10 recognized in more than 4% compared with the stateless version. This proves the superiority of using the stateful mode of operation of the LSTM layers over the usual stateless mode. Even with a simpler architecture (less number of layers and branches) and using a challenging dataset, the stateful model achieves higher accuracy rates than the stateless.

Again, the confusion between different classes appears when the actions have similar movements or involve small objects: put on a shoe and take of a shoe, reading and writing, back pain and chest pain, etc.

The total average accuracy on the NTU RGB+D dataset is 75.26% (CS) and 75.45% (CV) for the stateless ConvLSTM network and 80.43% (CS) and 79.91% (CV) for the stateful ConvLSTM network. This proves that, although it is rarely used in the literature, the stateful mode of the conventional LSTM is able to improve dramatically its performance on challenging datasets like NTU RGB+D. Furthermore, it is worth highlighting that the accuracy for both proposed networks is very similar independently of the chosen evaluation set (CS or CV), allowing us to conclude that they are robust against changes in the camera pose and the actors performing different actions.

In the next section, we compare the obtained results and computational costs with state-of-the-art methods.

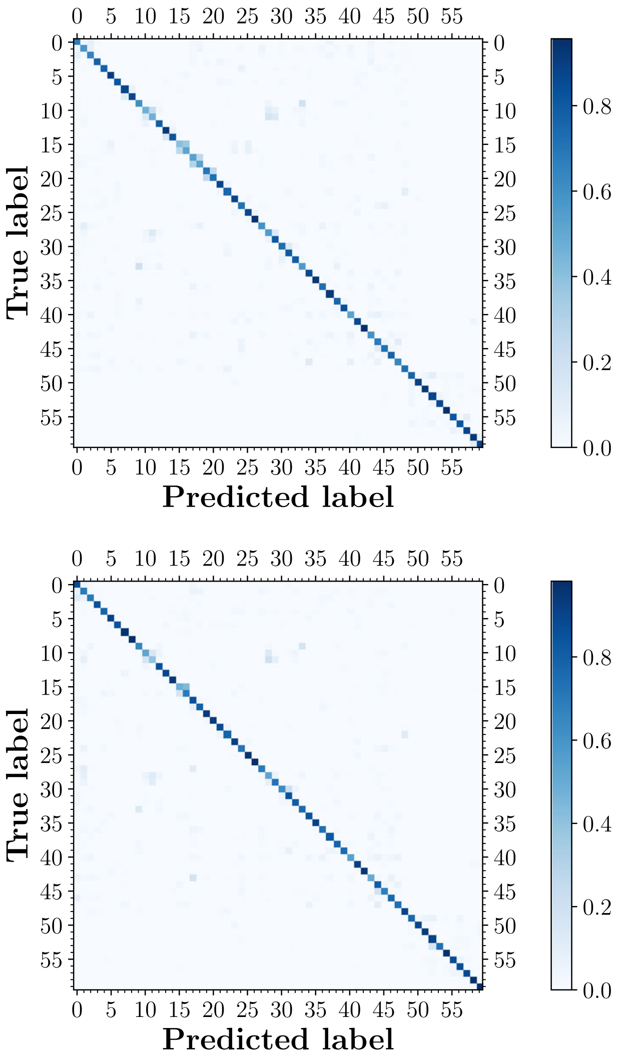


Fig. 10 Confusion matrices for the proposed models stateless (top) and stateful (bottom)

5.2 Comparison with state-of-the-art methods

A performance comparison of the proposed models (stateless and stateful ConvLSTM) with previous state-of-the-art methods is shown in Table 4. Thanks to the innovative deep learning techniques applied, the models proposed in this paper achieve competitive recognition accuracies on the NTU RGB+D dataset, and overcome other ConvLSTM-based methods like in [38]. In addition, although the methods that use dynamic images, as in [72, 79, 81], get the highest accuracies on this dataset within depth modality, they do it at the expense of very high time consumption. The usage of dynamic images prevents these methods from being used in real-time applications like video surveillance, health-care services, video analysis or human-computer interaction, because of the high computational cost

Table 3 Top 10 accurate actions and confused pairs for the proposed model, including accuracy recognition per action (CS evaluation)

Stateless ConvLSTM			
Top 10 recognized actions		Top 10 confused actions*	
1) Falling down	(95.65%)	1) Put on a shoe → Take off a shoe	(40.22%)
2) Hugging	(94.93%)	2) Reading → Writing	(46.37%)
3) Jump up	(93.84%)	3) Writing → Play with phone/tablet	(46.74%)
4) Shake head	(92.75%)	4) Take off a shoe → Put on a shoe	(51.81%)
5) Walking towards	(92.39%)	5) Sneeze/cough → Chest pain	(52.17%)
6) Put on a jacket	(91.67%)	6) Take off glasses → Put on glasses	(52.54%)
7) Salute	(91.67%)	7) Put on glasses → Take off glasses	(52.90%)
8) Pushing	(91.67%)	8) Play with phone/tablet → Writing	(54.71%)
9) Pick up	(90.22%)	9) Rub two hands → Clapping	(55.43%)
10) Kicking	(88.77%)	10) Eat meal → Brush teeth	(57.61%)
Average accuracy	92.36%	Average accuracy	51.05%
Stateful ConvLSTM			
1) Jump up	(98.19%)	1) Writing → Play with phone/tablet	(39.13%)
2) Walking towards	(98.19%)	2) Put on a shoe → Take off a shoe	(48.55%)
3) Stand up	(97.83%)	3) Headache → Put on glasses	(50.00%)
4) Walking apart	(97.83%)	4) Play with phone/tablet → Writing	(52.17%)
5) Hugging	(97.10%)	5) Reading → Writing	(52.54%)
6) Sit down	(96.01%)	6) Sneeze/cough → Chest pain	(54.71%)
7) Hopping	(96.01%)	7) Point to something → Taking a *selfie*	(63.41%)
8) Falling down	(95.29%)	8) Clapping → Rub two hands	(65.58%)
9) Take off jacket	(94.93%)	9) Back pain → Chest pain	(68.48%)
10) Put on a hat/cap	(93.48%)	10) Take off a shoe → Put on a shoe	(69.20%)
Average accuracy	96.49%	Average accuracy	56.38%

*Numbers between parenthesis are the recognition accuracy of true action (before the arrow)

related to dynamic image generation. To illustrate this, the last column of Table 4 includes the reported average processing times per video from the compared methods, accompanied by the results from the two proposed models. The time consumption of the multi-view dynamic images-based method was computed in [81] using an Intel(R) Xeon(R) E5-2630 V3 CPU running at 2.4 GHz and an NVIDIA GeForce GTX 1080 with 8 GB on videos from the NTU RGB+D dataset. Using the same GPU in the present work, the average time consumption was estimated from 10 000 random video samples of the same dataset, giving as a result 0.21 s for the stateless ConvLSTM and 0.89 s for the stateful ConvLSTM. Although the time consumption of the stateful version is small and allows a real-time application, it is still slower than the stateless one since the stateful model analyzes the whole video regardless of its length. Nevertheless, both proposed models are drastically faster than the methods in Table 4 that have reported computational cost information. Although most of the works do not report this information, as they use similar pre-processing strategies (dynamic images or 3D skeleton), it seems reasonable to assume that they would present similar order of magnitude for time consumption as the reported ones. Therefore, although there is an improvement of around 7% in accuracy when using these methods, they are

Table 4 Comparison of total average accuracy (%) on the NTU RGB+D dataset from different modalities

Method	CS	CV	Time/video (s)
Modality: 3D Skeleton			
ST-LSTM + Trust Gate (2016) [49]	69.2	77.7	–
Clips + CNN + MTLN (2017) [25]	79.57	84.83	–
AGC-LSTM (2019) [52]	89.2	95.0	–
Modality: Depth			
Unsupervised ConvLSTM (2017) [38]	66.2	–	–
Dynamic images (HRP) (2018) [72]	87.08	84.22	62.03
HDDPDI (2019) [79]	82.43	87.56	–
Multi-view dynamic images (2019) [81]	84.6	87.3	51.02
Stateless ConvLSTM	75.26	75.45	**0.21**
Stateful ConvLSTM	80.43	79.91	**0.89**

The last column also shows the average time consumption per video of each method as reported by their authors

approximately 100 times slower than the methods proposed in this study, making the latter far more suitable for real-time applications.

It is noteworthy that, as it has been explained before, the proposal outperforms the results provided by the authors [38], that uses a ConvLSTM with raw depth data as input, as well as some of the approaches based on 3D skeletons [49] and [25] on CS evaluation, been able to run in real time. Furthermore, it is worth to highlight that the obtained accuracy for the CS and CV evaluations is very similar for the proposal, whereas there appear larger differences for other state-of-the-art approaches. This allows validating the robustness of the proposed systems for HAR against the change in the point of view or in people performing the actions.

6 Conclusion

In contrast to most previous deep learning-based methods in human action recognition, this paper presents two models based on long short-term memory (LSTM) units for the stage of feature extraction from raw depth videos, followed by an ensemble of convolution and average pooling layers for the classification process. Both proposed models use a variant of LSTM, namely ConvLSTM, that leverages the convolution operation to extract spatial and temporal features from a sequence of images. In addition, to exploit the performance of these models several techniques from deep learning theory have been used, such as learning rate range test, cyclical learning schedule or batch normalization.

The major contribution of this work is the implementation of two novel approaches using ConvLSTMs that aim to boost time efficiency performance while keeping competitive accuracy rates, with two different strategies to directly use the long-term information contained in videos of variable lengths. On the one hand, we proposed an input data generator that takes into account the video lengths and allows the neural network to learn long-term characteristics (stateless ConvLSTM). On the other hand, we leveraged the stateful capability of LSTMs (and ConvLSTMs), by which the states of recurrent layers steadily learn along the video preserving spatio-temporal information of previous frames. That is, we assure that the stateful model processes nearly the whole video length. The main advantage of this approach is that, unlike state-of-the-art methods that generate static video representations such as depth motion maps or dynamic images, the proposed end-to-end trainable stateful model can effectively recognize actions belonging to very long and complex videos. Experiment results on the challenging NTU RGB+D dataset show that both proposed models (stateless and stateful ConvLSTM) reach competitive accuracy rates with very low computational cost compared with state-of-the-art methods because of the absence of any preprocessing. Furthermore, it is observed that the stateful ConvLSTM achieves better accuracy rates than standard or stateless ConvLSTM, proving the effectiveness of this uncommon methodology for videos.

The proven success of the stateful mode operation for HAR may open future research lines that integrate this capability to more complex or robust neural networks that improve accuracy rates in some problematic actions. Additionally, one may leverage its very long-term spatio-temporal pattern learning to design models for real-life continuous/online action recognition, with great interest in the video-surveillance field.

Acknowledgements Portions of the research in this paper used the “NTU RGB+D (or NTU RGB+D 120) Action Recognition Dataset” made available by the ROSE Lab at the Nanyang Technological University, Singapore.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Babu RV, Savitha R, Suresh S, Agarwal B (2013) Subject independent human action recognition using spatio-depth information and meta-cognitive rbf network. Eng Appl Artif Intell 26(9):2010–2021. https://doi.org/10.1016/j.engappai.2013.07.008
2. Baccouche M, Mamalet F, Wolf C, Garcia C, Baskurt A (2011) Sequential deep learning for human action recognition. In: International workshop on human behavior understanding, pp 29–39. Springer
3. Bansal M, Kumar M, Sachdeva M, Mittal A (2021) Transfer learning for image classification using vgg19: Caltech-101 image data set. J Ambient Intell Human Comput:1–12
4. Blank M, Gorelick L, Shechtman E, Irani M, Basri R (2005) Actions as space-time shapes. In: Tenth IEEE international conference on computer vision (ICCV’05) volume 1, vol 2, pp 1395–1402. IEEE
5. Bregonzio M, Gong S, Xiang T et al (2009) Recognising action as clouds of space-time interest points. In: CVPR, vol 9, pp 1948–1955
6. Breuel TM (2015) The effects of hyperparameters on sgd training of neural networks. arXiv:1508.02788
7. Chen J, Wang Z, Zeng K, He Z, Xiong Z (2022) Rethinking lightweight: multiple angle strategy for efficient video action recognition. IEEE Signal Process Lett
8. Crasto N, Weinzaepfel P, Alahari K, Schmid C (2019) Mars: motion-augmented rgb stream for action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7882–7891
9. Dargan S, Kumar M, Ayyagari MR, Kumar G (2020) A survey of deep learning and its applications: a new paradigm to machine learning. Archives Comput Methods Eng 27(4):1071–1092

10. Du Y, Fu Y, Wang L (2015) Skeleton based action recognition with convolutional neural network. In: 2015 3rd IAPR asian conference on pattern recognition (ACPR), pp 579–583. IEEE
11. Du Y, Wang W, Wang L (2015) Hierarchical recurrent neural network for skeleton based action recognition. In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 1110–1118
12. Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1933–1941
13. Günter A, Böker S, König M, Hoffmann M (2020) Privacy-preserving people detection enabled by solid state lidar. In: 2020 16th international conference on intelligent environments (IE), pp 1–4. IEEE
14. Guo G, Lai A (2014) A survey on still image based human action recognition. Pattern Recogn 47(10):3343–3361
15. Han J, Shao L, Xu D, Shotton J (2013) Enhanced computer vision with microsoft kinect sensor: a review. IEEE Trans Cybern 43(5):1318–1334
16. Herath S, Harandi M, Porikli F (2017) Going deeper into action recognition: a survey. Image Vis Comput 60:4–21
17. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
18. Hsu YP, Liu C, Chen TY, Fu LC (2016) Online view-invariant human action recognition using rgb-d spatio-temporal matrix. Pattern Recogn 60:215–226. https://doi.org/10.1016/j.patcog.2016.05.010
19. Hu JF, Zheng WS, Lai J, Zhang J (2015) Jointly learning heterogeneous features for rgb-d activity recognition. In: The IEEE conference on computer vision and pattern recognition (CVPR)
20. Huang M, Cai GR, Zhang HB, Yu S, Gong DY, Cao DL, Li S, Su SZ (2018) Discriminative parts learning for 3d human action recognition. Neurocomputing 291:84–96. https://doi.org/10.1016/j.neucom.2018.02.056
21. Ilg E, Mayer N, Saikia T, Keuper M, Dosovitskiy A, Brox T (2017) Flownet 2.0: evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2462–2470
22. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167
23. Ji S, Xu W, Yang M, Yu K (2012) 3d convolutional neural networks for human action recognition. IEEE Trans Pattern Anal Mach Intell 35(1):221–231
24. Katrompas A, Metsis V (2022) Enhancing lstm models with self-attention and stateful training. In: Arai K (ed) Intelligent systems and applications, pp 217–235. Springer International Publishing, Cham
25. Ke Q, Bennamoun M, An S, Sohel F, Boussaid F (2017) A new representation of skeleton sequences for 3d action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3288–3297
26. Khan MA, Javed K, Khan SA, Saba T, Habib U, Khan JA, Abbasi AA (2020) Human action recognition using fusion of multiview and deep features: an application to video surveillance. Multimed Tools Appl:1–27
27. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. CoRR 1412.6980
28. Klaser A, Marszałek M, Schmid C (2008) A spatio-temporal descriptor based on 3d-gradients. In: Procedings of the british machine vision conference 2008
29. Kong Y, Fu Y (2017) Max-margin heterogeneous information machine for rgb-d action recognition. Int J Comput Vis 123(3):350–371
30. Laptev I, Marszałek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies
31. Li S, Li W, Cook C, Zhu C, Gao Y (2018) Independently recurrent neural network (indrnn): building a longer and deeper rnn. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5457–5466
32. Liu AA, Nie WZ, Su YT, Ma L, Hao T, Yang ZX (2015) Coupled hidden conditional random fields for rgb-d human action recognition. Signal Process 112:74–82. https://doi.org/10.1016/j.sigpro.2014.08.038. Signal Processing and Learning Methods for 3D Semantic Analysis
33. Liu B, Cai H, Ju Z, Liu H (2019) Rgb-d sensing based human action and interaction analysis: a survey. Pattern Recogn 94:1–12
34. Liu J, Shahroudy A, Xu D, Wang G (2016) Spatio-temporal lstm with trust gates for 3d human action recognition. In: European conference on computer vision, pp 816–833. Springer
35. Liu J, Wang Z, Liu H (2019) Hds-sp: A novel descriptor for skeleton-based human action recognition. Neurocomputing. https://doi.org/10.1016/j.neucom.2019.11.048
36. Liu Z, Zhang C, Tian Y (2016) 3d-based deep convolutional neural network for action recognition with depth sequences. Image Vis Comput 55:93–100

37. Lu C, Jia J, Tang CK (2014) Range-sample depth feature for action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 772–779
38. Luo Z, Peng B, Huang DA, Alahi A, Fei-Fei L (2017) Unsupervised learning of long-term motion dynamics for videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2203–2212
39. Maas AL, Hannun AY, Ng AY (2013) Rectifier nonlinearities improve neural network acoustic models. In: Proceedings international conference on machine learning, vol 30, p 3
40. Núñez JC, Cabido R, Pantrigo JJ, Montemayor AS, Vélez JF (2018) Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. Pattern Recogn 76:80–94
41. Oreifej O, Liu Z (2013) Hon4d: histogram of oriented 4d normals for activity recognition from depth sequences. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 716–723
42. Park S, Park J, Al-masni M, Al-antari M, Uddin MZ, Kim TS (2016) A depth camera-based human activity recognition via deep learning recurrent neural network for health and social care services. Procedia Comput Sci 100:78–84
43. Poppe R (2010) A survey on vision-based human action recognition. Image Vis Comput 28(6):976–990
44. Presti LL, La Cascia M (2016) 3d skeleton-based human action classification: a survey. Pattern Recogn 53:130–147
45. Rahmani H, Bennamoun M (2017) Learning action recognition model from depth and skeleton videos. In: Proceedings of the IEEE international conference on computer vision, pp 5832–5841
46. Sadanand S, Corso JJ (2012) Action bank: a high-level representation of activity in video. In: 2012 IEEE conference on computer vision and pattern recognition (CVPR), pp 1234–1241. IEEE
47. Sanchez-Caballero A, de López-Diz S, Fuentes-Jimenez D, Losada-Gutiérrez C, Marrón-Romera M, Casillas-Perez D, Sarker MI (2022) 3dfcnn: Real-time action recognition using 3d deep neural networks with raw depth information. Multimed Tools Appl:1–25
48. Santofimia MJ, Fahlman SE, del Toro X, Moya F, Lopez JC (2011) A semantic model for actions and events in ambient intelligence. Eng Appl Artif Intell 24(8):1432–1445. https://doi.org/10.1016/j.engappai.2011.05.008. Semantic-based Information and Engineering Systems
49. Shahroudy A, Liu J, Ng TT, Wang G (2016) Ntu rgb+ d: a large scale dataset for 3d human activity analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1010–1019
50. Shi Z, Kim TK (2017) Learning and refining of privileged information-based rnns for action recognition from depth sequences. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3461–3470
51. Shotton J, Fitzgibbon A, Cook M, Sharp T, Finocchio M, Moore R, Kipman A, Blake A (2011) Real-time human pose recognition in parts from single depth images. In: CVPR 2011, pp 1297–1304. IEEE
52. Si C, Chen W, Wang W, Wang L, Tan T (2019) An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1227–1236
53. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems, pp 568–576
54. Singh R, Dhillon JK, Kushwaha AKS, Srivastava R (2019) Depth based enlarged temporal dimension of 3d deep convolutional network for activity recognition. Multimed Tools Appl 78(21):30599–30614
55. Singh S, Ahuja U, Kumar M, Kumar K, Sachdeva M (2021) Face mask detection using yolov3 and faster r-cnn models: Covid-19 environment. Multimed Tools Appl 80(13):19753–19768
56. Smith LN (2017) Cyclical learning rates for training neural networks. In: 2017 IEEE winter conference on applications of computer vision (WACV), pp 464–472. IEEE
57. Smith LN (2018) A disciplined approach to neural network hyper-parameters:, Part 1–learning rate, batch size, momentum, and weight decay. arXiv:1803.09820
58. Song S, Lan C, Xing J, Zeng W, Liu J (2017) An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In: Thirty-first AAAI conference on artificial intelligence
59. Song Z, Yin Z, Yuan Z, Zhang C, Chi W, Ling Y, Zhang S (2021) Attention-oriented action recognition for real-time human-robot interaction. In: 2020 25Th international conference on pattern recognition (ICPR), pp 7087–7094. IEEE
60. Sugianto N, Tjondronegoro D, Stockdale R, Yuwono EI (2021) Privacy-preserving ai-enabled video surveillance for social distancing: responsible design and deployment for public spaces. Information Technology & People
61. Sun Y, Wu X, Yu W, Yu F (2018) Action recognition with motion map 3d network. Neurocomputing 297:33–39. https://doi.org/10.1016/j.neucom.2018.02.028

62. 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. IEEE
63. Tan Z, Xu L, Zhong W, Guo X, Wang G (2018) Online activity recognition and daily habit modeling for solitary elderly through indoor position-based stigmergy. Eng Appl Artif Intell 76:214–225. https://doi.org/10.1016/j.engappai.2018.08.009
64. Varol G, Laptev I, Schmid C (2017) Long-term temporal convolutions for action recognition. IEEE Trans Pattern Anal Mach intell 40(6):1510–1517
65. Vieira AW, Nascimento ER, Oliveira GL, Liu Z, Campos MF (2012) Stop: space-time occupancy patterns for 3d action recognition from depth map sequences. In: Iberoamerican congress on pattern recognition, pp 252–259. Springer
66. Wang C, Wang X, Zhang J, Zhang L, Bai X, Ning X, Zhou J, Hancock E (2022) Uncertainty estimation for stereo matching based on evidential deep learning. Pattern Recogn 124:108498
67. Wang J, Liu Z, Chorowski J, Chen Z, Wu Y (2012) Robust 3d action recognition with random occupancy patterns. In: European conference on computer vision, pp 872–885. Springer
68. Wang J, Liu Z, Wu Y, Yuan J (2014) Learning actionlet ensemble for 3d human action recognition. IEEE Trans Pattern Anal Mach Intell 36(5):914–927
69. Wang L, Huynh DQ, Koniusz P (2019) A comparative review of recent kinect-based action recognition algorithms. arXiv:1906.09955
70. Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L (2016) Temporal segment networks: towards good practices for deep action recognition. In: European conference on computer vision, pp 20–36. Springer
71. Wang L, Xu Y, Cheng J, Xia H, Yin J, Wu J (2018) Human action recognition by learning spatio-temporal features with deep neural networks. IEEE Access 6:17913–17922
72. Wang P, Li W, Gao Z, Tang C, Ogunbona PO (2018) Depth pooling based large-scale 3-d action recognition with convolutional neural networks. IEEE Trans Multimedia 20(5):1051–1061
73. Wang P, Li W, Gao Z, Tang C, Zhang J, Ogunbona P (2015) Convnets-based action recognition from depth maps through virtual cameras and pseudocoloring. In: Proceedings of the 23rd ACM international conference on multimedia, pp 1119–1122. ACM
74. Wang P, Li W, Gao Z, Zhang J, Tang C, Ogunbona PO (2015) Action recognition from depth maps using deep convolutional neural networks. IEEE Trans Human-Mach Syst 46(4):498–509
75. Wang P, Li W, Gao Z, Zhang Y, Tang C, Ogunbona P (2017) Scene flow to action map: a new representation for rgb-d based action recognition with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 595–604
76. Wang P, Li W, Liu S, Gao Z, Tang C, Ogunbona P (2016) Large-scale isolated gesture recognition using convolutional neural networks. In: 2016 23rd international conference on pattern recognition (ICPR), pp 7–12. IEEE
77. Wang P, Li W, Ogunbona P, Wan J, Escalera S (2018) Rgb-d-based human motion recognition with deep learning: a survey. Comput Vis Image Underst 171:118–139
78. Wang P, Wang S, Gao Z, Hou Y, Li W (2017) Structured images for rgb-d action recognition. In: Proceedings of the IEEE international conference on computer vision, pp 1005–1014
79. Wu H, Ma X, Li Y (2019) Hierarchical dynamic depth projected difference images–based action recognition in videos with convolutional neural networks. Int J Advan Robot Syst 16(1):1729881418825093
80. Wu H, Ma X, Li Y (2021) Spatiotemporal multimodal learning with 3d cnns for video action recognition. IEEE Trans Circ Syst Video Technol
81. Xiao Y, Chen J, Wang Y, Cao Z, Zhou JT, Bai X (2019) Action recognition for depth video using multi-view dynamic images. Inf Sci 480:287–304
82. Xingjian S, Chen Z, Wang H, Yeung DY, Wong WK, Woo WC (2015) Convolutional lstm network: a machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems, pp 802–810
83. Xu B, Wang N, Chen T, Li M (2015) Empirical evaluation of rectified activations in convolutional network. arXiv:1505.00853
84. Yang X, Tian Y (2014) Super normal vector for activity recognition using depth sequences. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 804–811
85. Yang X, Zhang C, Tian Y (2012) Recognizing actions using depth motion maps-based histograms of oriented gradients. In: Proceedings of the 20th ACM international conference on multimedia, pp 1057–1060. ACM
86. Zhang P, Lan C, Xing J, Zeng W, Xue J, Zheng N (2017) View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In: Proceedings of the IEEE international conference on computer vision, pp 2117–2126

87. Zhang Z (2012) Microsoft kinect sensor and its effect. IEEE multimedia 19(2):4–10
88. Zheng Z, An G, Wu D, Ruan Q (2019) Spatial-temporal pyramid based convolutional neural network for action recognition. Neurocomputing 358:446–455. https://doi.org/10.1016/j.neucom.2019.05.058
89. Zhu J, Zou W, Zhu Z, Hu Y (2019) Convolutional relation network for skeleton-based action recognition. Neurocomputing 370:109–117. https://doi.org/10.1016/j.neucom.2019.08.043

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.