Analog Integrated Circuits and Signal Processing (2019) 99:23–32
https://doi.org/10.1007/s10470-018-1306-2

An efficient end-to-end deep learning architecture for activity classification

Amel Ben Mahjoub[1] · Mohamed Atri[1]

Received: 14 June 2018 / Accepted: 10 August 2018 / Published online: 22 August 2018

Abstract
Deep learning is widely considered to be the most important method in computer vision fields, which has a lot of applications such as image recognition, robot navigation systems and self-driving cars. Recent developments in neural networks have led to an efficient end-to-end architecture to human activity representation and classification. In the light of these recent events in deep learning, there is now much considerable concern about developing less expensive computation and memory-wise methods. This paper presents an optimized end-to-end approach to describe and classify human action videos. In the beginning, RGB activity videos are sampled to frame sequences. Then convolutional features are extracted from these frames based on the pre-trained Inception-v3 model. Finally, video actions classification is done by training a long short-term with feature vectors. Our proposed architecture aims to perform low computational cost and improved accuracy performances. Our efficient end-to-end approach outperforms previously published results by an accuracy rate of 98.4% and 98.5% on the UTD-MHAD HS and UTD-MHAD SS public dataset experiments, respectively.

Keywords Pre-trained CNN · LSTM · End-to-end model · Feature extraction · Action recognition

1 Introduction

Quite recently, considerable attention has been paid to Deep Neural Networks (DNNs) in challenging computer-vision research area. These DNN models are undergoing a revolution in description and classification tasks due to their capabilities of solving time-series-related issues. It has been utilized in several applications such as speech and image recognition, robot navigation systems, self-driving cars and medical diagnosis. The deep learning architecture, which imitates the human brain working, builds intelligent neural network models to automatically learn complex data sequences. These successful deep learning methods have been extensively used in human action recognition, and various newly researchers have shifted traditional machine learning methods to successful deep learning approaches, as reviewed in [1–6]. Convolutional Neural Networks (CNNs) are neural networks consisting generally of convolutional, pooling and fully connected layers utilized for many image classification tasks [7, 8]. Forward neural networks as CNNs cannot characterize time dependencies of data sequences. Deep CNN forward neural network models have achieved great success for visual-image recognition. However, it has failed to characterize time dependencies of video sequences. Video dynamics detection needs a system that knows the present, previous, and next frames of a given video. Recurrent Neural Networks (RNNs) are then defined to represent times-series information based on recurrent hidden states over time steps. Nevertheless, RNNs remain limited for very long time dependency because of the vanishing gradient problem. Therefore the solution is to use Long Short-term Memory (LSTM) which improve the RNN performances by storing multiple gating neural responses at each time-step to exploit long time memory. Much recent research [9–11] on action recognition based on LSTM has been developed. The focus of recent research [12–15] has been on video-level representation using CNNs to encode convolutional features. In practice, it is hard to directly train a CNN model from scratch with random initial parameters, which needs a high computational architecture and hours or days

✉ Amel Ben Mahjoub
amelbmh@gmail.com

[1] Laboratory of Electronics and Micro-electronics, Faculty of Sciences, Monastir University, 5000 Monastir, Tunisia

of computation time for the training step. Comparatively, training a CNN network with the large ImageNet dataset, which contains 1.2 million images from 1000 different classes, takes 2–3 weeks utilizing multiple Graphic Processing Units (GPUs). Thus, it is common today to apply pre-trained CNN models, which were previously learned on a large challenging dataset, to successfully transfer learning with a reduced runtime for the recognition system based on limited dataset. Transfer learning from pre-trained networks can be executed with two different ways. The first approach consists in training the pre-trained model with a new dataset to update the network parameters and get a well action prediction. In the second method, the pre-trained model is used to extract feature vectors from video sequences by removing the fully connected output layer. Accordingly, the second category has been widely developed for several action recognition architectures followed by a classifier, such as LSTM, to form an end-to-end model. End-to-end deep learning of the CNN-LSTM [15–17] method is currently an efficient technique to recognize actions from the entire long time video sequences. This framework has served to output the complex data for action detection and segmentation and it has achieved superior performances compared to traditional deep learning methods. Few researchers have addressed the problem of sequence-to-sequence framework complexity. These challenges motivate us to call into question how to define an efficient and effective model for learning time-series dependencies of action video sequences. The objective of this paper is to define an efficient end-to-end model for human action recognition with a low computational cost and an effective accuracy improvement by optimizing network parameters. First, convolutional feature vectors are extracted from RGB video sequences based on a pre-trained CNN. The pre-trained Inception-v3 model is utilized for video-level representation. Second, the LSTM recurrent network architecture is defined by optimizing its parameters to get a well classification rate. Finally, the feature outputs of the pre-trained model is applied to the LSTM classifier giving an end-to-end CNN-LSTM architecture for video dynamics detection and recognition.

2 Literature review

Chéron et al. [18] developed a new method of Pose based CNN (P-CNN) features for activity recognition from videos. Color images and optical flow vectors were cropped from video frames and their corresponding positions of body joints. Two CNN layers, with five convolutional and three fully-connected layers, were used to describe informative image regions of body joints. The first CNN network pre-trained on the ImageNet dataset was trained with the RGB frames to represent appearance information. The motion information was defined by applying the optical flow features to the second CNN layer which was pre-trained using the UCF101 dataset. The concatenation of both features presented the final P-CNN descriptor vectors of action videos. Ng et al. suggested in [19] an efficient classification method that combined information over full length videos based on various DNN architectures. Two AlexNet and GoogLeNet pre-trained CNN models were applied to scratch video frames in order to encode convolutional temporal features. The LSTM recurrent network was then connected to the output of the CNN models to perform ordered frame sequences and to predict activity videos. The authors in [20] adapted an effective Region-based CNN (RCNN) model for action classification. The description was done in the region that contained people and secondary regions with additional contextual cues, which helped to improve the activity recognition system. The stochastic gradient descent optimization method was applied in the RCNN training step for prediction. An end-to-end model was defined in [21] to describe action videos. An optical flow descriptor was extracted from consecutive frames to encode motion information. Color data and/or motion descriptor vectors were applied for a pre-trained CNN network in order to generate video representation. After that, the LSTM network was trained by the output of the pre-trained CNN to classify activity videos in a sequence-to-sequence way. Wang et al. defined in [22] an effective Temporal Segment Network (TSN) framework for deep action recognition. TSN aims to encode long-term temporal features by combining sparse temporal and spatial sampling across whole action videos based on CNN models. Two spatial and temporal stream learned CNNs were used so as to represent the dynamic characteristics of complex action sequences. The authors in [23] captured frame order information utilizing a novel temporal convolutional pooling technique inspired from the CNN functionality for action recognition. Motion features were extracted by applying the improved dense trajectory method to video sequences. Frame-level appearance information was encoded by calculating convolution operations at several local image regions using a CNN pre-trained on the ImageNet challenging dataset. An order-aware convolutional pooling approach was applied to the obtained sequence of frame-level characteristics to get dynamic video representation. A multi-region two-stream CNN architecture was proposed in [24] for action detection. Motion and appearance information was represented by applying Region Proposal Networks (RPN) to the optical flow and RGB data, respectively. The improvement in the recognition system was done by embedding a multi-region approach to the CNN model. The end-to-end two-stream CNN performed a well frame-level activity

detection system. Lan et al. [12] developed an action recognition system based on a deep local video descriptor. The TSN method was trained by video sequences to form local spatial and temporal CNN features. The Support Vector Machine (SVM) classifier was matched by the global video representations to get score prediction. Score-level fusion was executed to the spatial and temporal prediction output of SVM to perform the final action label. An efficient and fast hidden two-stream CNN method was defined in [25] to detect and classify human actions. A fully convolutional network called MotionNet was trained by video frames to encode optical flow features. The fine-tuning of a temporal stream CNN was done utilizing this estimated motion information in an end-to-end architecture to predict the action label. A second spatial stream CNN was trained by the input frames to extract appearance features. Hidden two-stream CNNs were performed based on late fusion that combined the obtained spatio-temporal information for a powerful action recognition system. Sargano et al. [16] presented an activity classification approach based on transfer learning of a deep video representation. the extraction of dataset frame features was done using an AlexNet CNN model pre-trained on the ImageNet dataset. A hybrid SVM and K-Nearest neighbor classifier were matched by the information vectors to get human action classes. A real-time and high-precision video dynamics detection technique was introduced in [26] utilizing a deep learning architecture. An RNN was implemented to represent the time-series continuity of video sequences. The dynamic detection of actions with a reduced video size was made by combining the CNN and RNN models together. The authors in [27] suggested two Fully Convolutional Networks (FCNs) models based on Temporal Pyramid Pooling (TPP) to represent video-level features. One FCN layer was focused to encode appearance information from color videos. Optical flow vectors were learned by the other FCN in order to to characterize motion information. A linear weighted method was performed to fuse the output of the two FCNs representing the spatio-temporal features of action videos. The classification step of the activity sequences was executed by the SVM classifier. An extended Dynamic Time Recurrent Attention Model (DT-RAM) was proposed in [28] for video representation. DT-RAM was a deep recurrent network that would extract the informative features from complex video sequences by removing pointless image regions. This kind of networks updated the next attention state and made a decision with an extra binary action whether to stop the computation while giving the classification rate or to continue calculation. DT-RAM was an end-to-end model that would help to improve the action recognition score. An accurate video representation method was studied in [17] for human activity recognition with a Long-Term Temporal CNN (LTC-CNN) model. Five space-time convolutional layers followed by three fully connected layers were used to learn motion estimation. The LTC-CNN architecture was considered as a well spatio-temporal low-level representation approach. Shi et al. defined in [29] a shuttleNet biologically-inspired deep network with feedward and feedback connections for action classification. The ShuttleNet contained different processors of gated recurrent units connected together across several pathways in a shuttle mode. A mechanism of attention was then applied to choose the most efficient pathway with informative features. A novel Human-Related Multi-Stream CNN (HR-MSCNN) framework was proposed in [13] to recognize action sequences. HR-MSCNN aimed to extract the discriminative video information based on different Two-Stream CNN Networks (TS-NETs). Three TS-NETs were trained to encode body motion estimation and three TS-NETs to represent appearance description. The final video-level representation was characterized by the concatenation of the six stream outputs based on the spatio-temporal 3D convolutional fusion method. The authors in [30] presented an end-to-end framework to characterize video features for an action recognition system. TS-NETs were applied utilizing color and optical flow data to encode spatio-temporal feature vectors. Frame-level temporal characteristics of human activities were captured by a deep TPP layer. Yan et al. [31] solved the problem of sequence-to-sequence RNN complexity by suggesting a Hierarchical Multi-scale Attention Network (HM-AN). The HM-AN algorithm was performed by concatenating hierarchical multi-scale RNNs with hard and soft attention mechanisms in order to learn the relevant video information. Reinforcement learning with the Gumbel-Softmax method was implemented to generate a stochastic hard attention. Hierarchical temporal features were detected by HM-AN to learn the long-term dependencies of the video-action recognition system. An important video-level representation approach was introduced in [14] using an Attention-based Temporal Weighted (ATW) CNN technique. Video data were grouped into different snippets, where each snippet contained three ATWs of the spatial RGB ResNet, temporal flow ResNet and Warped flow ResNet image features to get action probabilities. The obtained sequences of temporal weights were learned utilizing an attention mechanism, and the action-video prediction was done with a weighted sum method to fuse the three snippet modalities. The authors developed in [15] a Recurrent Spatial-Temporal Attention Network (RSTAN) model for person identification in complex video sequences. An improvement of the classical LSTM network was performed by adding a spatial-temporal attention technique that helped end-to-end trained model to learn relevant space-time video dependencies. Actor-attention regularization was

introduced in RSTAN for the reinforcement learning of additional information around actors, which was helpful to improve the action recognition rate.

3 Proposed architecture

We present an end-to-end CNN-LSTM architecture, as shown in Fig. 1, for human action representation and classification which will be detailed in this section.

3.1 CNN

CNN is a feed-forward artificial neural network type which has been successfully applied in several image classification and recognition systems [13, 14, 32]. CNN like all networks, includes input, hidden and output layers. Generally, CNN hidden layers are structured as a series of convolutional, pooling and fully connected layers. The first convolutional layer is intended to extract the local spatial features of the input images based on the kernel filter involving multiple channels. The filter is slid over the entire input image to calculate a dot product between the filter weights and the pixel values, as illustrated in Fig. 2. Given an input image I with C = 3 channels (RGB) and a Kernel filter K as a weight matrix of a k_1*k_2 shape, the convolutional equation can be expressed as follows:

$$\sigma(I * K)_{ij} = \sigma\left(\sum_{m=o}^{k_1-1}\sum_{n=o}^{k_2-1}\sum_{c=o}^{C} K_{m,n,c} * I_{i+m,j+n,c}\right) \quad (1)$$

where σ is the activation function, typically a Rectified Linear Unit (ReLU). The ReLU layer is an element-wise approach that consists in replacing the negative values in the feature map by zero, based on the following equation:

$$\sigma(x) = max(x, 0) \quad (2)$$

Pooling, or also called downsampling layers are applied after the ReLU operation to reduce the spatial dimension of the input volume, thus diminishing the computation cost, controlling the overfitting network problem and keeping the most relevant features invariant to scale and orientation changes. Several pooling categories have been utilized in deep learning literature include max, average and stochastic pooling methods. The max-pooling layer is considered as the most widely used method in the CNN architecture by sliding a window across the input to find the highest values. The final layer represents the fully-connected layer in which all the input neurons are completely connected to the previous layer nodes. These dense layers perform a classification step by transforming the features extracted by the convolutional layers and downsampled by the pooling layers to a class scores. The fully-connected layer with a K-dimensional vector is followed by a softmax activation function to generate a value in the range of (0, 1), which is given by:

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \quad (3)$$

3.2 LSTM

The LSTM network [9–11] is an RNN-developed architecture with memory blocks in recurrent hidden layers to remember cell information over a long time. This kind of recurrent networks solves the vanishing gradient problems of classical RNNs by inserting input, output, and forget gates that maintain long-term memory. LSTM cell states are modified by gate units in order to control information using the previous cell hidden state h_{t-1} and the current cell input X_t. The forget gate f_t decides the information parts that will be forgotten from the previous state (c_{t-1}) and the relevant parts that will be stored in the cell state

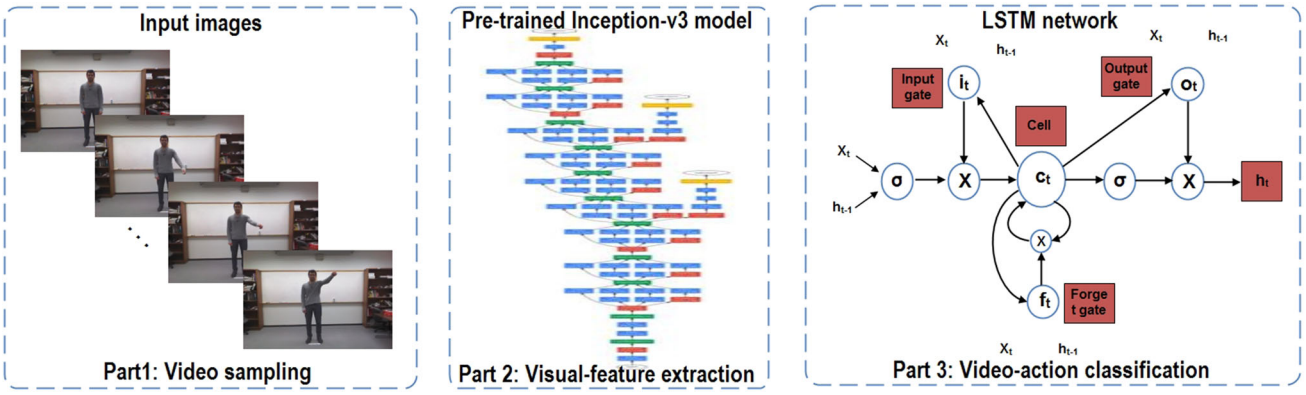

Fig. 1 Proposed architecture

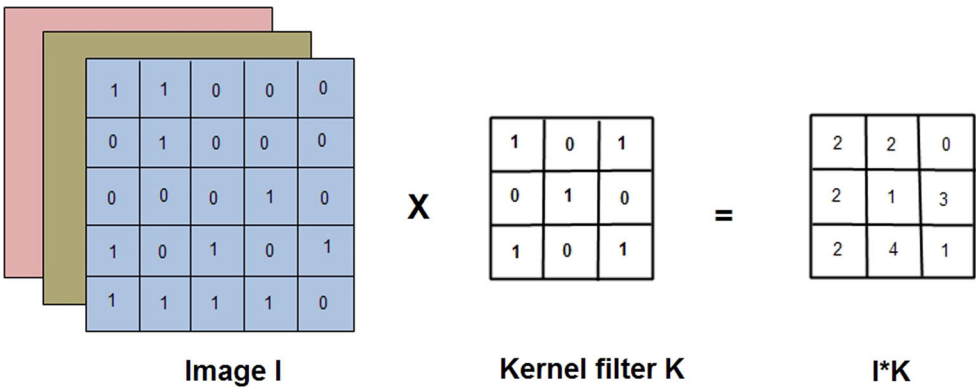

Fig. 2 Convolutional layer

multiplying matrix by zero or one, respectively, according to information significances. New information was added to the cell state based on the input gate i_t determinated by w_i and b_i parameters. A new internal memory state c_t is defined combining c_{t-1} multiplied by the f_t values with the latterly computed hidden state. The final hidden state cell output h_t is calculated utilizing the output gate, which highlights the c_t parts that will be stored to the next hidden state. The equations of the LSTM gate units are given as follows:

$$f_t = \sigma(w_f[h_{t-1}, X_t] + b_f) \tag{4}$$

$$i_t = \sigma(w_i[h_{t-1}, X_t] + b_i) \tag{5}$$

$$c_t = f_t \odot c_{t-1} + \tanh(w_c[h_{t-1}, X_t] + b_c) \tag{6}$$

$$o_t = \sigma(w_o[h_{t-1}, X_t] + b_o) \tag{7}$$

$$h_t = \tanh(c_t) \odot o_t \tag{8}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is a sigmoid activation function, $w_{f,i,o}$ are the weight matrices, $b_{f,i,o}$ are the bias, and $\odot$ is an element-wise product operation.

3.3 End-to-end model

The CNN architecture has two functions: (i) a visual feature representation approach using convolutional, ReLU and pooling layers and (ii) classification technique with fully-connected and softmax layers. In practice, very little research work has learned a CNN network directly from image inputs, which is a very hard architecture that takes days of training on large datasets such as ImageNet. Instead, it is common to fine-tune pre-trained CNN models using a new dataset by updating network parameters. Pre-trained CNN networks are trained on a challenging dataset to get network parameters. Transfer learning is done by applying these shared pre-trained parameters as an initialization or a fixed feature extractor for the representation and classification task in order to reduce the computation time. There are many challenging datasets utilized to define pre-trained CNNs such as ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [8] with 1.2 million images from 1000 classes, CIFAR10 including 60,000 images from ten categories and UCF101 with 13,320 videos divided into 101 actions. In our work, we apply mxnet models pre-trained with ILSVRC, which include CaffeNet [8], NiN [33], SqueezeNet [34], VGG [35], resnet [36], resnext [37], inception-BN and Inception-v3 [38] models, as detailed in Table 1.

4 Experimental results

We analyze in this part the obtained results of our end-to-end architecture for human action recognition.

4.1 Dataset

To illustrate the validity of our proposed method, several experiments are carried out based on the University of Texas at the Dallas Multimodal Human Action Dataset (UTD-MHAD). UTD-MHAD is a multimodal human action dataset in an indoor environment defined by Chen et al. in [39]. Four data modalities including color, depth, skeleton joint positions and inertial sensor signals were collected from a kinect camera and a wearable inertial sensor. Twenty-seven human actions were performed in this dataset, as detailed in Table 2. Each action was repeated four times by eight subjects to get 861 total data sequences after eliminating three invalid videos. Two different experiments based on the UTD-MHAD dataset are defined in order to compare our method with the state of the art. In the first Half-Subject (UTD-MHAD HS) experiment, the data provided by the subject 1, 3, 5, and 7 are chosen for the training step and the remaining subject sequences for the testing step. The second Subject-Specific (UTD-MHAD SS) experiment consists in using the two first repetitions in training and the last three repetitions in testing.

Table 1 Mxnet pre-trained models

Pre-trained model	Year	Layer numbers	Parameter size (MB)	Top-1 accuracy (%)	Top-5 accuracy (%)
CaffeNet	2012	5 conv + 3 FC	233	54.5	78.3
Network in network (NiN)	2014	12 conv + 1 softmax	29	58.8	81.3
SqueezeNet v1.1	2016	26 conv + 1 softmax	4.7	55.4	78.8
VGG16	2015	13 conv + 3 FC	528	71.0	89.8
VGG19	2015	16 conv, 3 FC	548	71.0	89.8
Inception-BN	2012	69 conv + 1 FC	43	72.5	90.8
Inception-v3	2015	94 CON + 1 FC	91	76.88	93.3
ResNet-50	2015	53 conv + 1 FC	98	75.4	92.6
Resnext-50	2015	49 res-conv + 1 FC	96	75.4	92.6

Table 2 UTD-MHAD dataset actions

Swipe left	Swipe right	Wave
Clap	Throw	Arm cross
Basketball shoot	Draw X	Draw circle (clockwise)
Draw circle (counter clockwise)	Draw triangle	Bowling
Boxing	Baseball swing	Tennis swing
Arm curl	Tennis serve	Push
Knock	Catch	Pickup and throw
Jog	Walk	Sit to stand
Stand to sit	Lunge	Squat

4.2 Implementation details

The implementation of our end-to-end architecture is done using the python algorithm with the flexible and efficient mxnet deep learning library.

RGB video actions are downsampled in frame sequences of 229*229, which are used to finetune the Inception-v3 [38] mxnet model pre-trained on the ImageNet dataset. As it can be seen from Table 1, Inception-v3 is the best pre-trained model to encode convolutional video features. The visual features of the UTD-MHAD dataset are extracted by removing the last fully-connected layer of Inception-v3 CNN network. For the video action classification part, we utilize a four-layer LSTM recurrent network with 1000 hidden layer units for UTD-MHAD HS and 1100 ones for UTD-MHAD SS experiments, which give good results as depicted in Figs. 3 and 4, respectively. The signum approach [40] is implemented for model-parameter optimization. we adapt our algorithm with a learning rate of $5 * 10^{-4}$, a batch size of 32, a number of epochs of 800, and a dropout with a ratio of 0.5 in order to capture the complexity of data. With GPU memory and these optimized hyperparameters, we design our end-to-end deep learning architecture that shows performing results.

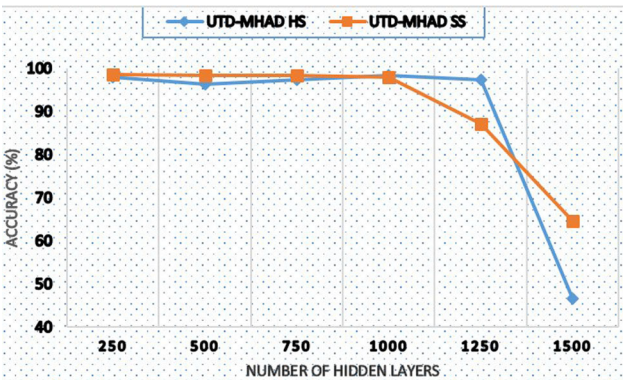

Fig. 3 Accuracy variation according to number of hidden layers

4.3 Comparison with state-of-the-art

To verify the efficiency of our end-to-end human action recognition architecture, we carry out the experimental simulations based on the UTD-MHAD public dataset. Our approach was compared with previous activity recognition methods, as presented in Tables 3 and 4. It is clear from these tables that our optimized architecture shows an important advantage in recognition accuracy over current methods. Table 3 represents the first simulation with the UTD-MHAD HS experiment, It is found that our method advances the approach [41] which encodes features based

on extracting the Histogram of Oriented Gradients (HOG) from Depth Motion Maps (DMM) by a recognition rate of around 17%. There is also an accuracy improvement of 19.3% comparing to the kinect and inertial feature fusion approach presented in [39]. Our solution outperforms the score fusion [32] method using the naive Bayesian approach by a rate of 7.9%. The second UTD-MHAD SS experiment results are given in Table 4. These results show as well that our proposed architecture enhances the methods defined in [48] and in [32] by a recognition rate of 1.3% and 6.9% respectively. This finding further strengthens our conviction that the end-to-end deep learning approach is an effective way to improve human action system performances.

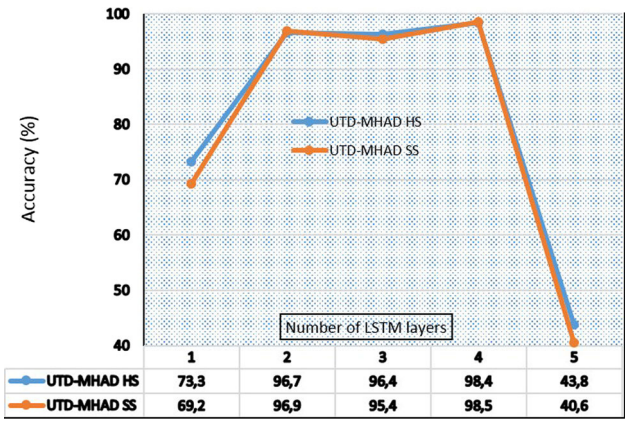

Fig. 4 Accuracy variation according to number of LSTM layers

Table 3 Comparison of state-of-art for UTD-MHAD HS experiment

References	Year	Architecture	Score (%)
[41]	2012	DMM–HOG	81.5
[39]	2015	Kinect	66.1
[39]	2015	Inertial	67.2
[39]	2015	Kinect and inertial	79.1
[42]	2015	DMM	73.4
[43]	2015	LOGP for decision fusion	88.4
[44]	2016	GF + LF	84.8
[45]	2016	Deep CNN	87.9
[46]	2017	3D HOT-MBC	84.4
[47]	2017	VDDM + CRC	85.1
[32]	2017	Score fusion with naive Bayesian	90.5
Our method	2018	End-to-end CNN-LSTM	98.4

4.4 Runtime and memory consumption

These experiments are carried out in computer with a GPU and a Central Processing Unit (CPU). The CPU is an Intel XEean E5-2620v4 with memory of 32 GB DDR4-2400 and 8 cores. The GPU is a Quadro M4000 with memory of 8 GB GDDR5, 1664 core numbers and 256 bits of memory interface, and PCI Express 3.0 × 16 of system interface. The GPU implementation is provided by the python algorithm based on the mxnet deep learning library to extract and classify RGB human action videos. Table 5 provides the runtime and memory consumption of feature extraction and classification parts provided in the UTD-MHAD HS dataset experiment.

Convolutional feature extraction utilizing the pre-trained Inception-v3 model takes 6 h of computing time and 566 MiB of memory cost with Quadro GPU. Human action classification based on the LSTM network is executed during 2.2 h with 1123 MiB of memory consumption.

Figure 5 compares the execution time between CPU and GPU for the classification part given with UTD-MHAD HS dataset experiment. The network classification step lasts 2.2 h and 50.4 h on GPU and CPU, respectively. Quadro GPU speeds up the used solution compared with CPU with a factor of around 23.

4.5 Discussions

The main concern of the paper is to define an efficient end-to-end architecture for human activity representation and classification. Particular attention is paid to optimize network hyperparameters to get a low computational cost and an accurate solution. As depicted in Tables 3 and 4, our

Table 5 Runtime and memory consumption for UTD-MHAD HS

	Time (h)	Memory consumption (MiB)
Feature extraction	6	566
Classification	2.2	1123

Table 4 Comparison with the state of art for UTD-MHAD SS experiment

References	Year	Architecture	Score (%)
[48]	2016	Kinect	85.1
[48]	2016	Inertial	88.3
[48]	2016	Kinect and inertial	97.2
[32]	2017	Score fusion of CRC, SRC, and KELM	91.6
Our method	2018	End-to-end CNN-LSTM	98.5

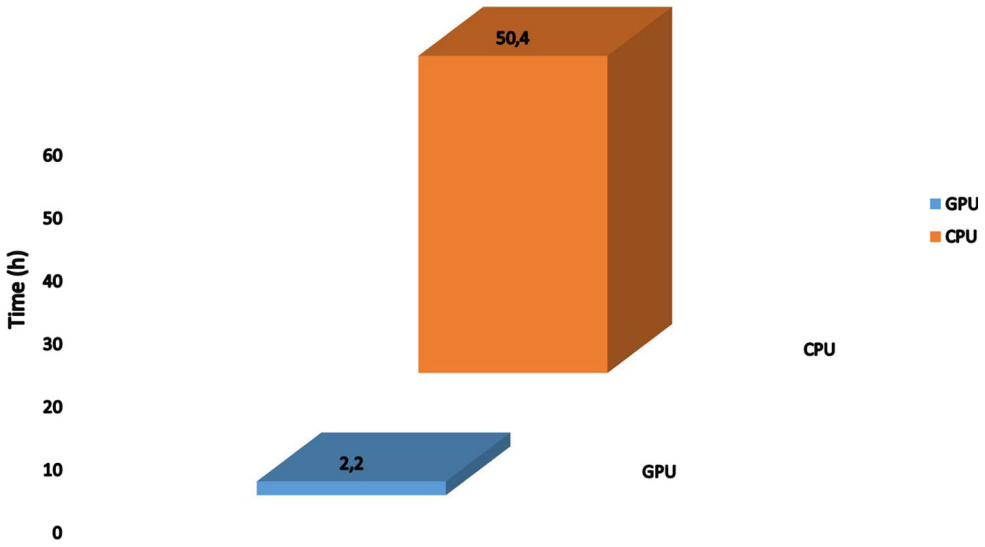

Fig. 5 CPU and GPU runtime comparison

end-to-end architecture achieves a recognition rate of 98.4% for UTD-MHAD HS and 98.5% for UTD-MHAD SS dataset experiments. These findings point to the usefulness of our method as an efficient action information representation and classification solution. This used technique improves the best published results by a rate of 7.9% and 1.3% for the UTD-MHAD HS and UTD-MHAD SS dataset experiments respectively. The obtained experiments are in good agreement with other studies which demonstrate the effectiveness of the deep learning model for activity recognition. It is noticeable that end-to-end neural network techniques are not new, but we optimize in this paper the network parameters in order to make the model more powerful for action recognition. The results have a number of possible limitations, namely the execution of the used algorithm in real time. A hardware implementation can be applied utilizing field-programmable gate arrays and compared with GPU results, which may speed up our proposed architecture.

5 Conclusion

In this paper, we have described an optimized end-to-end deep learning architecture for human action recognition. First, color action videos have been sampled into sequence frames. Second, convolutional feature vectors have been encoded from the frames based on an Inception-v3 model pre-trained in the ImageNet dataset. Finally, video action classification has been done using deep LSTM network. The evidence from this study points towards the idea that the end-to-end deep learning model is a powerful method to represent and classify complex informations. The findings insure that we have succeeded in describing an accurate model by optimizing network parameters. Further work needs to carry out the hardware implementation of our algorithm in order to get a faster and accurate method.

References

1. Asadi-Aghbolaghi, M., Clapes, A., Bellantonio, M., Escalante, H. J., Ponce-López, V., Baró, X., Guyon, I., Kasaei, S., & Escalera, S. (2017). A survey on deep learning based approaches for action and gesture recognition in image sequences. In *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)* (pp. 476–483). IEEE.
2. Koohzadi, M., & Charkari, N. M. (2017). Survey on deep learning methods in human action recognition. *IET Computer Vision*, *11*(8), 623–632.
3. Herath, S., Harandi, M., & Porikli, F. (2017). Going deeper into action recognition: A survey. *Image and Vision Computing*, *60*, 4–21.
4. Dhillon, J. K., & Kushwaha, A. K. S. (2017). A recent survey for human activity recoginition based on deep learning approach. In *2017 fourth international conference on image information processing (ICIIP)* (pp. 1–6). IEEE.
5. Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, *42*, 146–157.
6. Zhang, Q. S., & Zhu, S. C. (2018). Visual interpretability for deep learning: A survey. *Frontiers of Information Technology & Electronic Engineering*, *19*(1), 27–39.
7. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
8. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).
9. Liu, J., Wang, G., Hu, P., Duan, L. Y., & Kot, A. C. (2017). Global context-aware attention LSTM networks for 3D action recognition. In *CVPR*.
10. 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). Cham: Springer.

11. Lee, I., Kim, D., Kang, S., & Lee, S. (2017). Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 1012–1020). IEEE.
12. Lan, Z., Zhu, Y., Hauptmann, A. G., & Newsam, S. (2017). Deep local video feature for action recognition. In *Computer vision and pattern recognition workshops (CVPRW)* (pp. 1219–1225). IEEE.
13. Tu, Z., Xie, W., Qin, Q., Poppe, R., Veltkamp, R. C., Li, B., et al. (2018). Multi-stream CNN: Learning representations based on human-related regions for action recognition. *Pattern Recognition*, *79*, 32–43.
14. Zang, J., Wang, L., Liu, Z., Zhang, Q., Niu, Z., Hua, G., & Zheng, N. (2018). Attention-based temporal weighted convolutional neural network for action recognition. arXiv preprint arXiv:1803.07179.
15. Du, W., Wang, Y., & Qiao, Y. (2018). Recurrent spatial–temporal attention network for action recognition in videos. *IEEE Transactions on Image Processing*, *27*(3), 1347–1360.
16. Sargano, A. B., Wang, X., Angelov, P., & Habib, Z. (2017). Human action recognition using transfer learning with deep representations. In *2017 international joint conference on 2017 neural networks (IJCNN)* (pp. 463–469). IEEE.
17. Varol, G., Laptev, I., & Schmid, C. (2017). Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*, 1510–1517.
18. Chéron, G., Laptev, I., & Schmid, C. (2015). P-CNN: Pose-based CNN features for action recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 3218–3226).
19. Ng, J. Y. H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015) Beyond short snippets: Deep networks for video classification. In *Computer vision and pattern recognition (CVPR)* (pp. 4694–4702). IEEE.
20. Gkioxari, G., Girshick, R., & Malik, J. (2015). Contextual action recognition with r* CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 1080–1088).
21. Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., & Saenko K. (2015). Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision* (pp. 4534–4542).
22. 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). Cham: Springer.
23. Wang, P., Liu, L., Shen, C., & Shen, H. T. (2016). Order-aware convolutional pooling for video based action recognition. arXiv preprint arXiv:1602.00224.
24. Peng, X., & Schmid, C. (2016). Multi-region two-stream R-CNN for action detection. In *European conference on computer vision* (pp. 744–759). Cham: Springer.
25. Zhu, Y., Lan, Z., Newsam, S., & Hauptmann, A. G. (2017). Hidden two-stream convolutional networks for action recognition. arXiv preprint arXiv:1704.00389.
26. Zheng, K., Yan, W. Q., & Nand, P. (2017). Video dynamics detection using deep neural networks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *2*, 224–234.
27. Yu, S., Cheng, Y., Xie, L., & Li, S. Z. (2017). Fully convolutional networks for action recognition. *IET Computer Vision*, *11*(8), 744–749.
28. Li, Z., Yang, Y., Liu, X., Wen, S., & Xu, W. (2017). Dynamic computational time for visual attention. arXiv preprint arXiv:1703.10332.
29. Shi, Y., Tian, Y., Wang, Y., Zeng, W., & Huang, T. (2017). Learning long-term dependencies for action recognition with a biologically-inspired deep network. In *Proceedings of the IEEE conference on computer vision and pattern recognition 2017* (pp. 716-725).
30. Zhu, J., Zou, W., & Zhu, Z. (2017). End-to-end video-level representation learning for action recognition. arXiv preprint arXiv:1711.04161.
31. Yan, S., Smith, J. S., Lu, W., & Zhang, B. (2018). Hierarchical multi-scale attention networks for action recognition. *Signal Processing: Image Communication*, *61*, 73–84.
32. Mahjoub, A. B., Khedher, M. I., Atri, M., & Yacoubi, M. A. E. (2017). Naive Bayesian fusion for action recognition from Kinect. In *Computer science & information technology (CS & IT)* (Vol. 7, pp. 53–69).
33. Lin, M., Chen, Q., & Yan, S. (2013). Network in network. arXiv preprint arXiv:1312.4400.
34. Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360.
35. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
36. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
37. Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5987–5995). IEEE.
38. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).
39. Chen, C., Jafari, R., & Kehtarnavaz, N. (2015). UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *2015 IEEE international conference on image processing (ICIP)* (pp. 168–172). IEEE.
40. Bernstein, J., Wang, Y. X., Azizzadenesheli, K., & Anandkumar, A. (2018). signSGD: Compressed optimisation for non-convex problems. arXiv preprint arXiv:1802.04434.
41. 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.
42. Elmadany, N. E. D., He, Y., & Guan, L. (2015). Human action recognition using hybrid centroid canonical correlation analysis. In *2015 IEEE international symposium on multimedia (ISM)* (pp. 205–210). IEEE.
43. Bulbul, M. F., Jiang, Y., & Ma, J. (2015). DMMs-based multiple features fusion for human action recognition. *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, *6*(4), 23–39.
44. Escobedo, E., & Camara, G. (2016). A new approach for dynamic gesture recognition using skeleton trajectory representation and histograms of cumulative magnitudes. In *2016 29th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)* (pp. 209–216). IEEE.
45. Imran, J., & Kumar, P. (2016). Human action recognition using RGB-D sensor and deep convolutional neural networks. In *2016 international conference on advances in computing, communications and informatics (ICACCI)* (pp. 144–148). IEEE.
46. Zhang, B., Yang, Y., Chen, C., Yang, L., Han, J., & Shao, L. (2017). Action recognition using 3D histograms of texture and a multi-class boosting classifier. *IEEE Transactions on Image Processing*, *26*(10), 4648–4660.

47. Jin, K., Min, J., Kong, J., Huo, H., & Wang, X. (2017). Action recognition using vague division depth motion maps. *The Journal of Engineering*, *1*(1), 77–84.
48. Chen, C., Jafari, R., & Kehtarnavaz, N. (2016). A real-time human action recognition system using depth and inertial sensor fusion. *IEEE Sensors Journal*, *16*(3), 773–781.

Amel Ben Mahjoub got her license and MS degree in Microelectronics from the Faculty of Sciences of Monastir, Tunisia, in 2011 and 2013, respectively. Currently, she is preparing her Ph.D. degree in Electronics and Microelectronics laboratory of the Faculty of Sciences of Monastir. Her main research includes computer vision, deep learning and image processing.

Mohamed Atri received his Ph.D. Degree in Micro-electronics from the Science Faculty of Monastir, Tunisia, in 2001 and his Habilitation in 2011. He is currently a member of the Laboratory of Electronics and Micro-electronics. His research includes Circuit and System Design, Pattern Recognition, Image and Video Processing.