CNN-LSTM Architecture for Human Action Recognition Using Skeletal Representation

Dimitrios Koutrintzes, dkoutrintzes@gmail.com

Keywords: human activity recognition, Long short-term memory, skeletal representation

Abstract:

1 INTRODUCTION

Human Activity Recognition (HAR) is the problem of identifying actions, activities or events that are performed by humans. Typically, such approaches are based on some sensorial input. The most popular and accessible approach nowadays is to use video input from one or multiple cameras. The problem is usually manifested as a multiclass classification problem, of outputting the predicted class label of the performed activity. When approaching a HAR task, using a computer vision approach, one should select the appropriate way to capture represent, analyze and finally classify visual data to activities. The areas of application of the HAR is broad, including surveillance, assisted living, human-machine interaction, affective computing, etc. According to Wang et al. [Wang et al., 2016], HAR may be divided into a) segmented recognition, segmenting the video into input clips that contain exactly one activity, and b) continuous recognition, wherein the goal is to detect and classify actions within a video, wherein several parts may not contain action while starting and ending points of action should be detected.

In this work, we propose an architecture that combines a Convolutional neural network with a Long short-term memory to utilize better the temporal information of an action.

2 RELATED WORK

In this section, we briefly present related work focused on HAR that is based on deep learning architectures. More specifically, we focus on approaches that are based on intermediate visual representations of 3D motion skeletal joints that are used with a Convolutional Neural Network and Long Short-term Memory.

Skeleton motion image representations are used as input in CNNs. In all approaches, the motivation is to create an artificial image, by mapping features to pixel values. The result is either grayscale or a pseudocolored image, whose color and texture properties somehow reflect the spatial and temporal properties of skeleton motion.

In the work of Hyunh-The et al. [Huynh-The et al., 2020], two geometric features are extracted, namely inter-joint distances and orientations, forming vector representations which are then concatenated to form images.

Pham et al. [Pham et al., 2019] proposed a similar representation, enhanced with an image processing approach for contrast stretching, so as to highlight the textures and edges of the representation.

In the work of Zhu et al. [Zhu et al. 2017] both RGB and depth modalities were exploited for gesture recognition. Short-term Spatio-temporal features were learnt by a 3D CNN and then, long-term Spatio-temporal features were learnt based on the extracted features with the use of convolutional LSTM networks.

Moreover, Haque et al. [Haque et al. 2018] followed an early fusion approach of RGB, Depth and thermal information, to capture complementary facial features related to pain. For feature extraction, a CNN-LSTM model was employed. Imran et al. [Imran et al.] presented a multi-stream network for human action analysis, leveraging CNN and RNN networks, where features from RGB, depth and inertial data were incorporated.

Sun et al. [Sun et al. 2017], fused feature elements of RGB and depth information which were then learnt through an enhanced two-stream LSTM network, called "Lattice-LSTM."

3 PROPOSED METHODOLOGY

3.1 DATASET

For network training and evaluation, our method was tested on the 60 classes from the NTU RGB+D [Shahroudy et al 2016] dataset. NTU RGB+D is a large-scale benchmark dataset for 3D Human Activity Analysis. RGB, depth, infrared, and skeleton videos for each performed action have been also recorded using the Kinect v2 sensor. It contains around 67K videos of actions from multiple angles and multiple people.

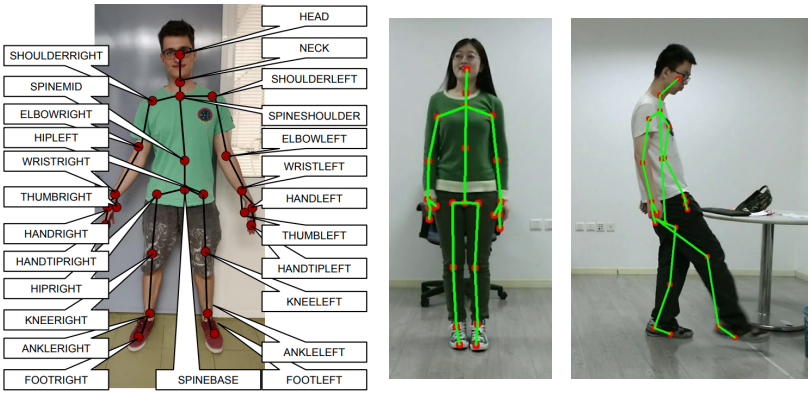

Figure 1. Representation of 25 Skeletal Joints of Kinect V2.

3.2 DATA AND PREPROCESSING

From the NTU dataset, we use the skeletal data. This contains information from 25 skeletal joints for each human body. As illustrated in fig. 1, from the 25 joints, 6 represent each arm,4 for each leg, and the last 5 are for the body, neck, and head. For each joint 3D coordinates are given, and we can have up to two persons in the actions that have interactions between persons.

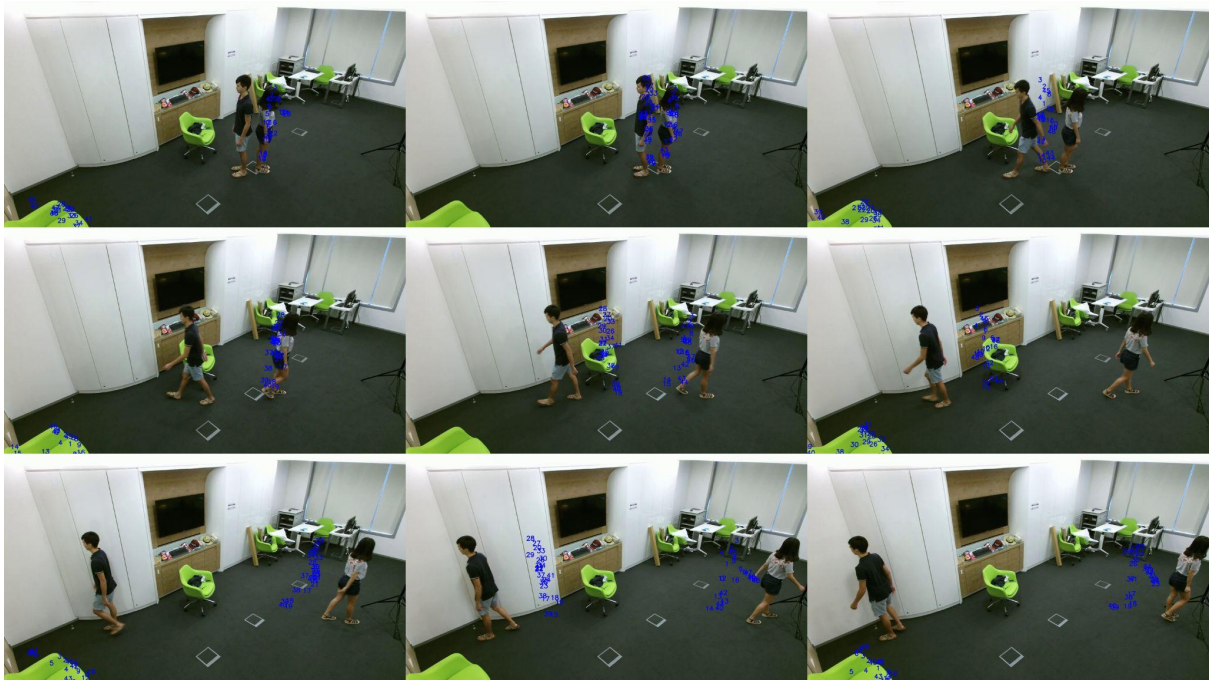

Figure 2. Broken Skeleton File.

For the preprocessing, the first step was to check the files for their information. A step to see if each file contains completed information, by checking each line in a file to look if there are lines with zeros, and if a threshold of 15 zero lines if passed we remove the file. In this check, if a line for the first person has zeros we check if the same line for the second person doesn't, if it does the line doesn't count as a zero line. As a second step, we check the points between two persons if there's any joint that passes between two persons. As we can see in fig. 2 files like that, which have the biggest problem and the zero check doesn't clean, cannot be fixed, as the skeletal information, of both person connected, jumps to a different person every frame, and a couch? so this step didn't provide much help

Another step in preprocessing is to move the start of a movement on the middle of the screen. Technically we move the spine base joint to the middle of the screen and we follow the rest of the joint around that, the rest of the frames don't move to the middle, only are adjusted based on the change we make on the first frame. This step would be useful if we used the coordinates of the joints as features but as we will see after we use the difference of a joint in each frame and this is not affected by the location in the image.

3.2 REPRESENTATION

In order to use the skeletal information as input to the CNN-LSTM, we modified a representation first introduced by Vernikos et al. [Vernikos et al., 2019a]. This representation aims to capture inter-joint distances during the action and use them o create pseudo-colors within an artificial RGB image. The method is using the 3D trajectories of skeletal joint. From the x,y and z coordinates of each ot the M available joints, a set of 3xN signals is collected for a given video sequence depicting an activity. To address the problem o temporal variability between actions and between users, a zero padding step was impose to fill the

image to 200 frames, this covered the ~99% of the dataset.For actions that had more than 200 frames, a linear interpolation step used to manually reduce the frames to 200. For each sequence, coordinate differences between consecutive frames are calculated, while x,y,z coordinates correspond to R,G,B color channels of the pseudo-colored image, respectively.

More specifically, for a $x_i(n)$ that denotes the x-position of the i-th joint in the n-th frame. Let G denote the green Channel of the color image. The value of the $G(i,n)$ is : $G(i,n) = x_i(n+1) - x_i(n)$, where i = 1 … N. Similarly the blue and red channels are calculated. This method creates a Mosaic image that preserving both temporal and the spatial properties of the skeleton trajectories of size 25x199x3. We can see the final images for a correspondent activity in Fig 3. This image contains only the first 25 skeletal joints from the provided file. The NTU dataset contains 11 "Mutual Actions" that include information for both persons. To Evaluate if there are any benefits to including information for both persons. For this, we simply modify the above methodology to include all 50 skeletal joints, 25 per person with a resulting image of size 50x199x3 fig 2. For the classes that don't have two persons, a simple zero-padding was used to fill for the second person, also visible in fig. 2.

Figure 3. Representations of Skeletal information, size of 25x199

Figure 4. Representations of Skeletal information, the first contain 2 person actions, while the other two contain only one, size of 50 x199

3.3 CNN-LSTM ARCITECTURE

The initial CNN architecture that used for the evaluation of the initial representation, and that we used as a baseline for our comparison is illustrated on the fig. 4. It consists of a convolutional layer with 32 kernels of size 3 x 3, followed by a pooling layer. Then, 2 convolutional layers with 64 kernels of size 3 x 3 and a pooling layer follow, succeeded by a convolutional layer with 128 kernels of size 3 x 3 followed by a pooling layer. Each polling layer uses a "max-pooling" to perform 2 x 2 subsampling. Then, a flatten and two dense layers of size 256 and 128 . Finally the output layer of size 60.

The proposed CNN LSTM architecture contains the same convolutional and max pooling layers as before. After that a time-distribution flatten layer is added that creates the

feature vectors for the LSTM layer. We used to LSTM layers with 256 kernels. Follows a flattened layer and after that as before two dense layers of size 256 and 128 are followed by an output layer of size 60. For the images of two person, the model is the same and only change the input.

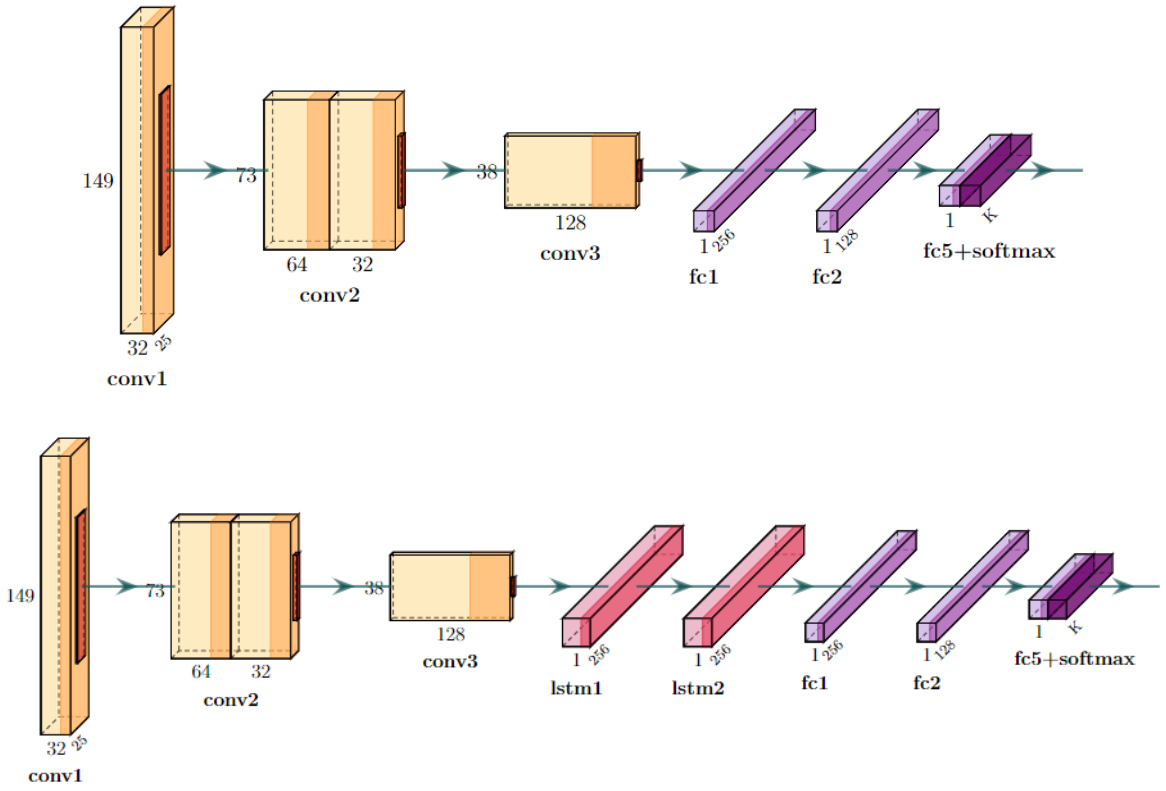

Figure 5. First CNN architecture, Second CNN-LSTM architecture

4 EXPERIMENTAL EVALUATION

4.1 SETUP

Experiments were performed on a personal computer with an Ryzen 5 1600X 6/12 core/threads processor and 16GB RAM, using an NVIDIA Geforce GTX 1060 with 6 GB RAM and Windows 10. The deep architecture has been implemented in python , using Keras and tensorflow 2.6 back end. All data pre-precessing and processing steps have been implemented in python 3.9 using Numpy, SciPy and OpenCV. For all examples the same settings was used. The models was trained for 150 epochs and a checkpointer saved the model after every epoch for evaluation. For the optimizer, Adam was chosen with a starting learning rate of 0.00003 and an ending rate of 0.000001.

4.2 RESULTS

For our results we have two different experiments, one is a cross-view test, were in both train and test, includes all the participants but has different angles for each action and the second is the cross-subject test, which in both train and test, includes data from all the angles but has different subjects. For the evaluation metric we use accuracy, we choose not to include other metrics accuracy is the common evaluation metric for this dataset.

As we can see from the results in Table 1. the inclusion of the LSTM layers provided a big boost to the performance of the model. In the CrossView we have a 12.0 increase in performance while on CrossSubject is 5.7. But the results don't show the complete behavior of the models. The Cnn architecture suffers from overfitting, with the evaluation loss to digressing while the evaluation accuracy is a stack, but the performance gains, on the test dataset, beyond of the 150 epoch are minimum, while in both LSTM the behavior is different, here the evaluation loss is overfitting and starting to grow after a point and the evaluation accuracy and test accuracy of the model continues to grow. This is a very interesting difference on bihevior and its obvious that with more optimization for the CNN - LSTM architecture can produse more performance. As for the number of epoch that was used, the number is pure because time restriction, in experiments with 250 epochs both in CNN and the CNN - LSTM cases we had better results, but couldn't run all the test of Table 1. to have a fair comparison between them, even so, the image there was the same as here with the CNN-LSTM outperforming the CNN comfortably. As for the benefits to having the second person in the image, we can see that there not benefits, this is probably because of the usage of the exact architecture like the single person and internally couldn't utilize correctly the extra information.

Model	CrossView	CrossSubject
CNN	54.2	51.3
CNN - LSTM (Proposed Method)	66.2	57.0
CNN - LSTM 2 Persons(Was broken)	60.6	53.4

Table 1. Experimental Results of the proposed methods (CNN - LSTM / 2 Persons) and the results from the baseline model CNN.

5 CONCLUSION

The proposed method is clearly an evolution of the simple CNN architecture with much more headroom for improvement and future work.

As experiments for feature work, from some experiments we perform, a CNN-BiLSTM architecture gave promising results, we replace the two-layer of LSTM with a single BiLSTM size of 128, and in the 20 epochs produce better results from the proposed method, but as the proposed method was run as referred for 150 epochs, the final result was better and the CNN-BiLSTM was to heavy to let it run to compare.

Also on the preprocessing side, we could test these steps on a dataset like PKU that has more problems like this and could utilize more. Also, a smoothing step could be used to

smooth out the tremble that some joints have between frames. Finally, if we could use more processing power we could test an augmentation step that we know for older works could boost performance, especially on the cross-view test.

REFERENCES

[Wang et al., 2016] Wang, P., Li, Z., Hou, Y., and Li, W. (2016). Action recognition is based on joint trajectory maps using convolutional neural networks.

[Huynh-The et al., 2020] Huynh-The, T., Hua, C.-H., Ngo, T.-T., and Kim, D.-S. (2020). Image representation of pose-transition feature for 3d skeleton-based action recognition.

[Pham et al., 2019] Pham, H. H., Salmane, H., Khoudour, L., Crouzil, A., Zegers, P., and Velastin, S. A. (2019). Spatio–temporal image representation of 3d skeletal movements for view-invariant action recognition with deep convolutional neural networks.

[Zhu et al. 2017] G. Zhu, L. Zhang, P. Shen and J. Song. Multimodal gesture recognition using 3-D convolution and convolutional LSTM.

[Haque et al. 2018] M. A. Haque, R. B. Bautista, F. Noroozi, K. Kulkarni, C. B. Laursen, R. Irani, and T. B. Moeslund, Deep multimodal pain recognition: a database and comparison of spatio-temporal visual modalities

[Imran et al.] J. Imran and B. Raman, Evaluating fusion of RGB-D and inertial sensors for multimodal human action recognition. Journal of Ambient Intelligence and Humanized Computing

[Sun et al. 2017] L. Sun, K. Jia, K. Chen, D. Y. Yeung, B. E. Shi and S. Savarese, Lattice long short-term memory for human action recognition.

[Shahroudy et al 2016] Shahroudy, A., Liu, J., Ng, T. T., & Wang, G. (2016). Ntu rgb+ d: A largecale dataset for 3d human activity analysis.

[Vernikos et al., 2019a] Vernikos, I., Mathe, E., Papadakis, A., Spyrou, E., and Mylonas, P. (2019a). An image representation of skeletal data for action recognition using convolutional neural networks.