Geometric and temporal features for Skeleton-Based Action Recognition using LSTMs

Liliana Antão[1]

DEI/FEUP - Informatics Engineering Department
Faculty of Engineering, University of Porto, Porto, Portugal
lpsantao@fe.up.pt

Abstract. Skeleton datasets are one of the most used inputs when it comes to human action recognition systems. Mainly, remarkable performances have been achieved when using skeleton data with Deep Learning models like Long Short-Term Memory (LSTM) networks. The problem is that most of the solutions with LSTMs either use raw input skeleton data with large and complex models or use simpler models with specifically extracted features according to skeleton particularities and model type, in order to achieve state-of-art performance.
Given this problem, adapting to different use cases, using features that can be applied to different skeleton datasets, or knowing which nature of features helps to improve a lightweight LSTM's performance is skeleton-based action recognition, would bring great insight. With this in mind, geometric, temporal, and combined features, derived from skeleton data, were extracted and tested with a LSTM network. Four different input variations were evaluated: the unfeatured dataset, temporal features, geometric features, and combined features. Results show that the combined features allow attaining the best overall results across different datasets with a simple LSTM, even achieving mean state-of-art results with a much simpler model with low computational power restrains.

Keywords: Action recognition · Deep Learning · LSTMs · Geometric and Temporal features · skeleton datasets.

1 Introduction

Human action recognition systems have a vast set of application areas, from user interfaces to medical assistance, video surveillance, or even robotics, it has been a task with increased interest over the years. Recently, time series classification problems like human action classification, have been solved with Deep Learning models, showing impressive results. In these models, relevant features are learned automatically, where two main Deep Learning approaches are usually used: Convolutional Neural Network and Recurrent Neural Network (RNN), like Long Short-Term Memory (LSTM) networks [15], where these last ones have been showing promising results. To classify human actions, these models usually utilize input data retrieved from different qualities of sensors like vision devices

that collect RGB video frames, or wearables like accelerometers that provide rotation and translation data [4]. Despite this variety in input types, one of the most commonly used input data is articulated human joint coordinates, i.e., skeleton data. These kinds of data have the particularity of allowing an excellent representation for human activities, due to their intrinsic robustness over environment noise when compared to accelerometer data. They also provide direct insight and high-level characteristics of human actions and their movements. Furthermore, skeletons are usually smaller in size than RGB frames, enabling for much more compact models and simplified hardware [10].

In spite of all these advantages, most of the skeleton-based solutions use direct body joint coordinates or specific features extracted from those joints. These features usually depend on the specifications of the data being used, as well as the deep learning model. Generally, these solutions are either not very good in improving the model's performance, or not very adaptable to variances in the model type, input characteristics, or application. This problem is highlighted, for instance, when developers try to apply the same features in skeleton-based models to recognize actions involving the complete body versus just a specific body part (hands, for instance), or with 2D joint coordinates versus 3D [14].

In this paper, we explore the impact of different feature types in skeleton-based action recognition solutions. For this purpose, a lightweight two-layered LSTM model was built. One dataset for body action recognition (JHMDB) and one for hand action classification (DHG-14/28) were tested. Feature extraction was performed in both datasets, where three different types of features were created: geometric, temporal, and combined. Four input variations were evaluated: the unfeatured dataset, temporal features, geometric features, and combined features. With this, we consider that we will be able to get some understanding of which features allow the best performance when applied to the simplest form of LSTM models. Our insight is that by applying combined features state-of-art performance can be achieved with simpler models (with fewer parameters and training samples) for distinct action recognition types (full-body action recognition or specific body parts' action), and also for different data dimensions.

The remainder of this paper is structured as follows: In Section 2, we present the related work on skeleton-based action recognition with LSTM and LSTM-CNN models, as well as geometric and temporal feature extraction; in Section 3 the details on the used datasets and their preprocessing, with visualizations of some of their attribute to get more insight, as well as the used models, are described. In Section 4, the experimental results and discussion are shown. Finally, in Section 5, the final conclusions are drawn.

2 Related Work

Over the past few years, several solutions for human action recognition have been proposed. The majority of these solutions focus in two key aspects: present innovative model architectures [5], [8], [11], [7], [12], and propose novel skeleton-based features [1], [2], [3], [17].

In [5], end-to-end two-stream attention-based LSTM network is proposed to recognize human tasks. In this, the LSTM is used to encode temporal sequence information, and a CLSTM to capture spatial-temporal information, then tuning the learning process, being suited to changing scenarios. In [7], a deep fusion framework is implemented that extracts spatial features from CNNs, combining them with temporal features from LSTM models. The model's fully connected features guide the LSTM to relevant parts of the CNN feature sequence. Hou *et al.* [8] introduce an end-to-end Spatial-Temporal Attention Residual Temporal Convolutional Network (STA-Res-TCN) for hand action recognition. This novel architecture, at each time step, learns distinct degrees of attention and delegates them to each temporal-spatial feature derived from the convolution filters. Also, in [11], an extended version of a LSTM model is presented: Global Context-Aware Attention LSTM (GCA-LSTM). This new network introduces a global context memory cell, which confers selective attention capabilities, that can become more effective gradually. Although in all these solutions, state-of-art performance is achieved in three different action datasets, their models are either very complex to implement and/or have high computational costs.

In [1], a new collection of pose features named Geometric Pose Descriptors (GPD) are proposed. These features exploit geometric properties between body parts, emphasizing relational body part configurations. Several evaluations are performed, showing that GPD outperforms other features. In [3], a fixed size (not dependent on the action's duration) representation of pose motions, called PoTion, is presented. This allows using a conventional CNN for action recognition (no need for recurrent networks or more complex networks). Zhang *et al.* [17] also present novel features for action recognition, specifically for LSTMs. They define eight geometric features that translate relations between joints, in a unique frame or a short sequence of frames. After evaluating four different datasets, joint line distances were the features with better performance. Finally, in [2], a Variational Auto-Encoder (VAE) is used to extract features from the hand skeleton. These features translate finger motions, having as addition global features like rotation and translation of the hand. Experiments with RNN and two datasets demonstrate that these features allow for performance improvements. The problem with all the features mentioned above is that they are very model and data-dependent. Most of them are intended to improve their RNNs' (LSTMs) model performances, not being applicable in other more simple LSTM networks. Also, some of them do not tackle the data to make the features independent from their location or body rotation, or when they do, there is a lack of general movement in the features provided [14]. Besides this, none of these works provide insight on which features one should use according to the model being employed, or the skeleton characteristics.

3 Methodology

As previously stated, for our work, we intend to extract and test different geometric, temporal, and combined features from skeleton datasets to analyze their

performance in action recognition models of different characteristics. Given that action recognition is a machine learning problem with increasing interest and diverse solutions, in our approach, we intend to follow the established CRISP-DM framework for Data Science and Machine learning. Six stages compose CRISP-DM: Business understanding, Data understanding, Data preparation; Modeling; Evaluation, and Deployment. In our approach, given that the background, goals, and requirements are already well defined for the action recognition problem, the first stage was considered done. Throughout this section, details on these stages and implementations are further explained.

3.1 Data Understanding

Considering the vast number of applications for recognizing human actions and activities, many skeleton-based datasets exist. In order to correctly test the generality and applicability of different features in skeleton-based action recognition, we first needed to select datasets that allow supervised-learning (labeled); have sufficient data points for applying deep learning techniques; and differ in types of actions, dimensions, and body parts used. Only by choosing datasets with different characteristics, a correct analysis can be performed. Two different benchmark datasets with distinct properties were chosen: Dynamic Hand Gesture 14-28 (DHG) dataset (used in the Shape Retrieval Challenge in 2017 (SHREC'17)) [6] and Joint-annotated Human Motion Data Base (JHMDB) dataset [9].

Table 1: List of the labels included in the DHG-14/28 dataset

Action	**Type**	**Tag name**
Grab	Fine	*G*
Expand	Fine	*E*
Pinch	Fine	*P*
Rotation CW	Fine	*R-CW*
Rotation CCW	Fine	*R-CCW*
Tap	Coarse	*T*
Swipe Right	Coarse	*S-R*
Swipe Left	Coarse	*S-L*
Swipe Up	Coarse	*S-U*
Swipe Down	Coarse	*S-D*
Swipe X	Coarse	*S-X*
Swipe V	Coarse	*S-V*
Swipe +	Coarse	*S-+*
Shake	Coarse	*Sh*

The DHG dataset provides 3D hand action sequences, with 14 action labels: nine "coarse" which means they are characterized by hand motion (like tap (label 2), shake (label 14) or different swipes (labels 7 to 13)), and five "fine" gestures, i.e., defined by finger movements and shape of the hand through the gesture (grab (label 1), pinch (label 4), Rotations (labels 5 and 6)). The complete list of hand action labels, their type, and tag name are presented in Table 1. These actions can be performed in two manners: using one finger (the original 14 labels)

and the whole hand (plus 14 labels to a total of 28). Each action is performed between 1 and 10 times by 28 participants, resulting in 2800 sequences, of 20 to 50 frames each. Every frame includes 22 joint coordinates of the hand's skeleton (Fig. 1 left): one for the palm center, one for the wrist's position, and four joints for each finger (tip, the two articulations and the base).

The JHMDB dataset derives from the original HMDB dataset (a video dataset for action recognition). It contains 928 clips, incorporating 21 action labels, annotated with a 2D human skeleton puppet model with 13 joints (shoulder, elbow, wrist, hip, knee, ankle, neck) and two landmarks (face and belly)(Fig. 1 right). All actions present in the dataset involve only single-person activities. The resulting 21 actions are the following: *brush hair, catch, clap, climb stairs, golf, jump, kick ball, pick, pour, pull-up, push, run, shoot ball, shoot bow, shoot gun, sit, stand, swing baseball, throw, walk* and *wave*. Each action contains 36 to 55 clips, and each clip 15 to 40 frames, with a total of 31,838 annotated frames.

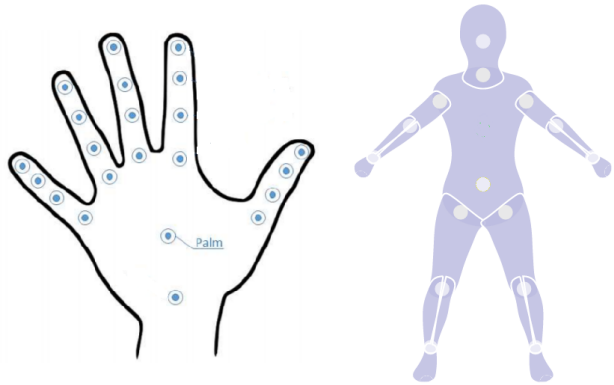

Fig. 1: Hand skeleton with 22 joints of the DHG dataset (left) and body skeleton with 15 joints for the JHMDB dataset (right).

In order to correctly retrieve features, first, we need to better understand the datasets and how the movements are translated in joint positions evolution. For this, some visual representations of datasets were carried out, in order to help simplify the next stage of data preprocessing. Given this, the hand's palm trajectories for six DHG actions, and the joint positions of the complete JHMDB's skeleton for three actions, were plotted and are presented in Fig. 2. As can be seen, the palm's movements in the DHG dataset can clearly indicate which action is being performed most of the time, even by visual inspection. The problem is with similar actions like *swipe +* and *swipe x*, where the palm's motions can lead to errors in classification. In the case of the JHMDB dataset some joints also clearly differ from action to action, as well as their temporal evolution in the action sequence, for instance: the *throw* action expressly manifests coordinates'

variation in all joints throughout the time, whereas the *clap* and *wave* actions only show variations in the arms and hand joints.

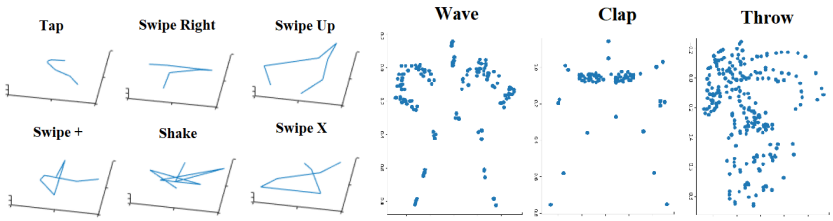

Fig. 2: Hand center movement for six DHG classes (left) and skeleton joints movements for three JHMDB classes (right)

3.2 Dataset Preprocessing

After understanding the dataset, the data preprocessing was performed. Each DHG skeleton file contains a matrix of size *n*x66. Each line contains the 3D hand joints coordinates in the world space. The format is as follows: *x1 y1 z1 ... - x22 y22 z22*. For JHMDB each .mat file contains a 3D matrix with size 2 x 15 x *number of frames*. In the first dimension, the two values correspond to x and y coordinates, and in the second dimension, the values are the 15 joints. Both datasets also provide benchmark splits for training and test sets: DHG with only test and train sets (30% and 70% of the whole dataset, respectively), JHMDB with three different splits for train and test (70% and 30% of each split, respectively). In this phase, given that each dataset was available with single files in separate folders for each action and subject, the files containing the joint coordinates were aggregated in the following shape: $[n^o frames, n^o joints, n^o dimensions]$, for training and test splits/sets.

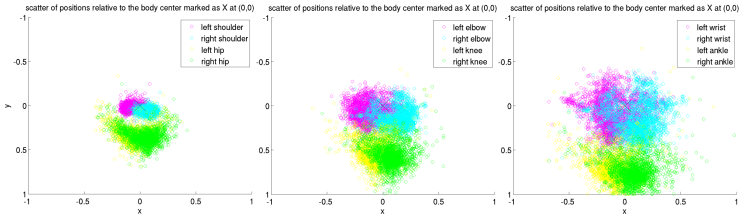

Fig. 3: Scatter plot for JHMDB dataset joint positions relative to the body center marked as X (0,0)

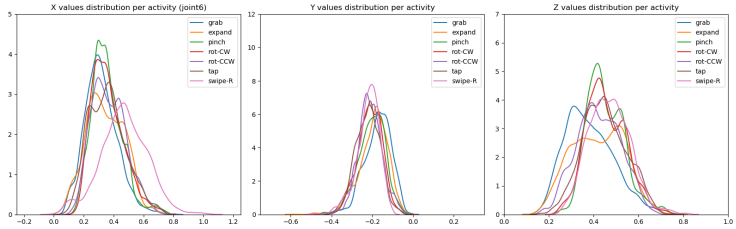

Fig. 4: Distribution of index-tip coordinates (x,y,z) for DHG dataset for different actions

Following this aggregation, joint distribution and scatter plots were made for each action and joint. These plots were done to detect outliers in the dataset, as well as gain some knowledge on coordinates distributions. Some examples of these plots are present in Fig. 3, where the scatter plot of JHMDB joint positions is shown relative to the body center and Fig. 4, where the distribution of x, y, and z are shown for DHG's index tip joint for different actions. As can be seen, not only no clear outliers were detected, as it was possible to conclude that the majority of the joints' coordinates followed a normal distribution.

The next step was normalization, where the joint values were changed to a standard scale, without distorting differences in the ranges of value. All joints suffered a transformation from the camera coordinates frame to the body/hand center coordinate frame. Each point is translated to the "belly" joint in the JHMDB case and the palm's center in DHG. A rotation is also applied in DHG, according to the method used in [13], since it is a three-dimensional dataset, and different rotations of the joint's coordinate frame can affect the calculation of certain features. A uniformed scale of all joint coordinates was also performed in proportion to the body height/hand size. This assists in reducing the effects introduced by distinct skeleton sizes. This normalization step is widespread in input data for neural networks for stable convergence of weight and biases, which is beneficial in our case where deep learning models will be used.

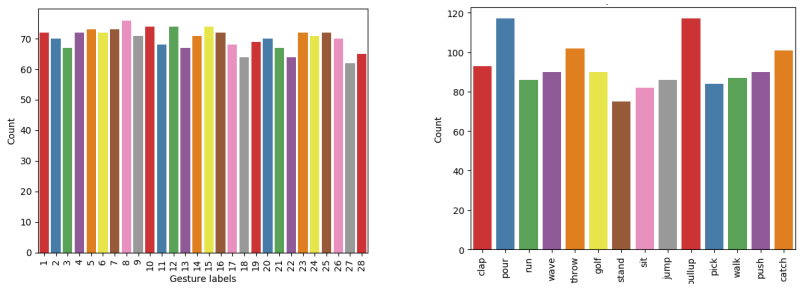

Fig. 5: Number of Skeletons per label in the SHREC train (left) and JHMD train sets (right)

With the datasets aggregated, normalized and divided into train and test sets/splits, the number of skeletons present for each label, in both training and test sets were plotted to check for data imbalances between sets. In Fig. 5, the number of skeletons per action label in the SHREC train (left) and JHMD train sets (right) are shown. As demonstrated, the class distributions in the training sets are very well balanced, with a maximum variance of 20 skeletons between classes. The same happens with the test sets of both datasets. This is not the case for the number of frames in each sample, where both datasets have variable frame numbers between actions, subjects, and samples. Given this, a padding was added to each sample in order to always have a fixed size. This size was established as equal to the maximum number of frames in the whole dataset. After this padding, some memory issues appeared in training, so a rescaling approach was performed, where all samples were resized to 128 frames, using the mean value of every 3 or 2 samples.

The final preprocessing practice, and the most important in this work's context, was feature extraction. As mentioned before, first geometric and temporal features were extracted. These features were based on previously consolidated work with an excellent performance from other authors, for spatial and time domains feature generation. The geometric features extracted were Joint-Joint Distance(JJ_d), i.e., the distance between joint J1 and $J2$ and Joint-Line Distance (JL_d), i.e., the distance from joint J to line L_{J1-J2} using the method described in [17]. These features are directly calculated from skeleton data to reduce the deviation introduced by coordinate transformation, resulting in a total of 105 for JJ_d and 537 for JL_d in the JHMDB dataset, and 231 for JJ_d and 731 for JL_d in DHG. For the time-based features, the Body part Direction (BpD) (adapted from [6]) was extracted, where the direction vector using the center position of a particular the body part is computed (for instance palm if hand, belly if upper body, and elbow if arm), resulting in 1 feature for DHG and 6 for JHMDB. Also, the motion of skeleton joints from two consecutive frames was used. Lastly, the combined features extracted, i.e., features entailing both space and time information, were based in the methods presented in [14]: the Global Motion, i.e., speed (temporal differences) of the joint coordinates, and Joint Collection Distances (JCD) that models euclidean distance throughout the action time with location-viewpoint invariance.

3.3 LSTM

With all the data processed and features extracted, the already mentioned deep learning model typically used for action recognition (LSTM) was implemented. For this, a script for each model was created using Python and the Keras backend in Tensorflow, with NVIDIA CuDNN bindings. A LSTM, a simple two-layer network, was used with the architecture, as shown in Fig. 6. The LSTM network has a total of 224,714 parameters, with two LSTM layers of 100 neurons each, a Dropout layer intended to reduce model overfitting to the training data. A fully-connected layer is used to interpret the features extracted by the LSTM hidden

layer before a final output layer that makes the classification with a softmax activation.

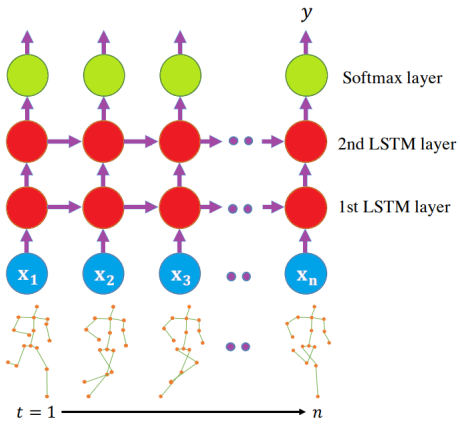

Fig. 6: LSTM network architecture

As input, the LSTM network received a 3D array of shape [*batch_size*, *frames*, *features*], where the *batch_size* is the count of samples sent at once to the network. Categorical cross-entropy was used as loss function, and the class labels were one-hot encoded so that the data was suitable for fitting a multi-class classification model. As an optimizer, the Adam algorithm was used. The LSTM model was fit for a fixed number of 100 epochs and a batch size of 64 samples. The batch size was chosen after evaluating different sizes, where 64 achieved the best trade-off in performance versus training time. Larger windows require large models that are slower to train, whereas smaller windows require smaller models that easier to fit. Finally, to evaluate the model, the test sets/splits are used, and the accuracy of the fitted model on the test set is returned. All the train and evaluation was performed on a NVIDIA GTX 1050-TI.

4 Evaluation: Experiments and Results

To evaluate the LSTM model and test our hypothesis that combined features allow for state-of-art performance even with a simple and lightweight model, four different training and test phases were performed. For the first model, only skeleton data (after preprocessing, but with no further features) was used for training, for the second model the time features extracted were used, for the third the geometric features were applied, and finally for the last LSTM model the combined features were used. All the features, as well as the test and train splits, are explained in Subsection 3.2 and the LSTM network parameters used are detailed in Subsection 3.3. Since the performance of a model should not be

assessed from a single evaluation (since neural networks are stochastic), each evaluation was repeated five times, and the mean performance across those runs computed. The DHG dataset was evaluated in two cases: 14 gestures and 28 gestures, with the recommended train and test sets. The JHMDB dataset was evaluated with manually annotated skeletons, for the three training and test splits. For this last dataset, the results of each split were also averaged. The accuracy results for each dataset and each LSTM model are shown in Table 2.

Table 2: Performance comparison between LSTM with no features (raw inputs only), geometric features, temporal features, and combined feature as input.

Dataset	**Nr. of classes**	**Accuracy**			
		LSTM network			
		Raw Inputs	Temporal	Geometric	Combined
DHG	14	79.1%	81.2%	84.7%	**85.2%**
	28	70.9%	72.7%	79.3%	**81.3%**
JHMDB	21	63.3%	63.9%	68.8%	**71.1%**

When comparing the values of each LSTM model presented in Table 2, we can see that the models with the skeleton data as input show the worst results, as expected, with accuracies of 79% for DHG-14, 70.9% for DHG-28 and 63.3% JHMDB. Even so, decent accuracy levels are reached with a simple LSTM since it is designed to extract temporal information very relevant for skeleton-based action recognition. With just these results, it is clear that the DHG dataset has much stronger joint correlations through frames since it obtained much higher performance levels. For model two (LSTM with temporal features), the increase from the raw input model is not very significant, since as previously mentioned, the LSTM network is already designed to extrapolate time dependencies.

When it comes to model three (geometric features as input), the accuracies increase up to 5% for DHG-14, 9% for DHG-28, and 5% for JHMDB when compared to the initial model. Once again, as LSTMs are suitable for modeling dependences in the time domain, when feeding geometric features, the accuracy increases. Also, by analyzing these values, we can see that the "fine" actions added in the DHG-28 have stronger spatial relations than the "coarse" actions from DHG-14, and the JHMDB dataset has the least spatial correlated actions of all. Finally, as hypothesized, the combined features, with both geometric and time aspects, present the best results for both datasets, showing an accuracy of 71.1% for JHMDB, of 85.2% for DHG with 14 labels and 81.3% for DHG with 28 labels. This represents an increase from the initial model of 6% for DHG-14, 11% for DHG-28, and 8% for JHMDB. From this increase, the statements provided before for space correlated actions are supported. The increase, when compared to model three (with geometric features), is not very significant since the contribution of the extracted time features to the accuracy is small. However,

even with the small increase, these types of features reveal themselves effective, since with the combination less data is necessary for training, tackling memory problems derived from bigger batches being necessary for geometric features or raw joint data.

To better show the evolution in performance during training and test sets for the final model (with combined features), the plots for loss and accuracy during these sets, for each dataset, are presented in Figures 7, 8 and 9. All plots show the evolution of loss and accuracy in 100 epochs. As can be seen, all datasets took only around 40 to 60 epochs to converge in performance, demonstrating the simplicity and low parameter number of our LSTM model, since it is swift to train. On the downside in this approach is the instability throughout training and test phases, since many spikes and slopes can be seen, mostly in JHMDB (Fig. 9). Also, it is clear that the JHMDB dataset should be trained longer than DHG since it is much larger, and it contains data with less correlation (2D).

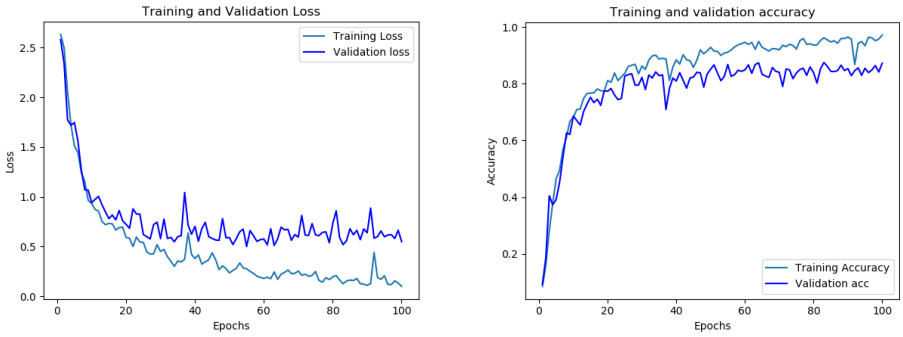

Fig. 7: Training and test loss (left) and accuracy (right) for DHG-14 labels

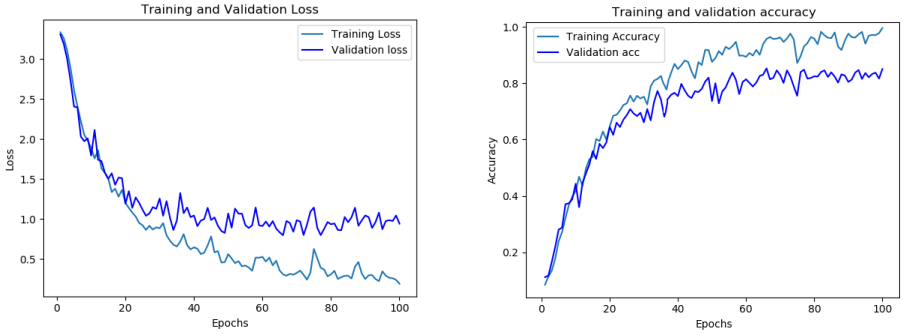

Fig. 8: Training and test loss (left) and accuracy (right) for DHG-28 labels

In order to understand which actions the model recognizes correctly and not, the confusion matrices for each dataset were computed (Figures 10 and 11). We can see that our model performs well on most of the actions, but still, the

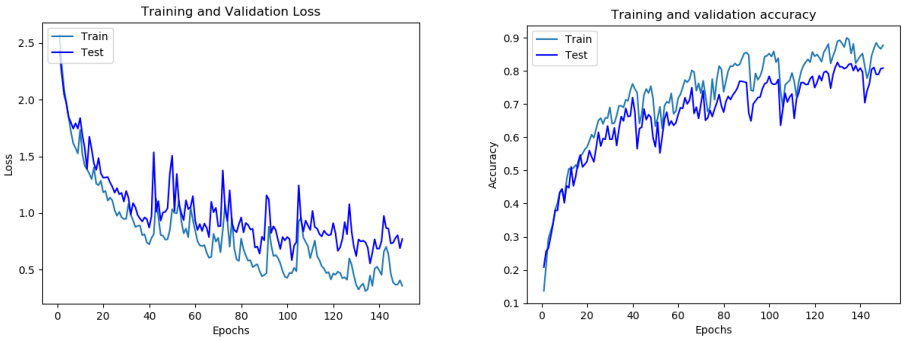

Fig. 9: Training and test Loss (left) and accuracy (right) for JHMD dataset

misclassification is not avoidable, showing some higher false positives in the JHMDB dataset like for the actions *pick* and *stand*(Fig 10 left). For JHMDB, 15 out of 21 actions are more than 80% correctly classified, for DHG-14 12 out of 14 and for DHG-28 20 out of 28. With DHG-28 dataset, more false positives are shown, since the difference between an action performed with one finger or the whole hand is sometimes hard to perceive (case of *tap* and *tap_2*). The confusion matrices show that the LSTM model with combined features is robust to almost every action label, displaying its capability to generalize throughout datasets.

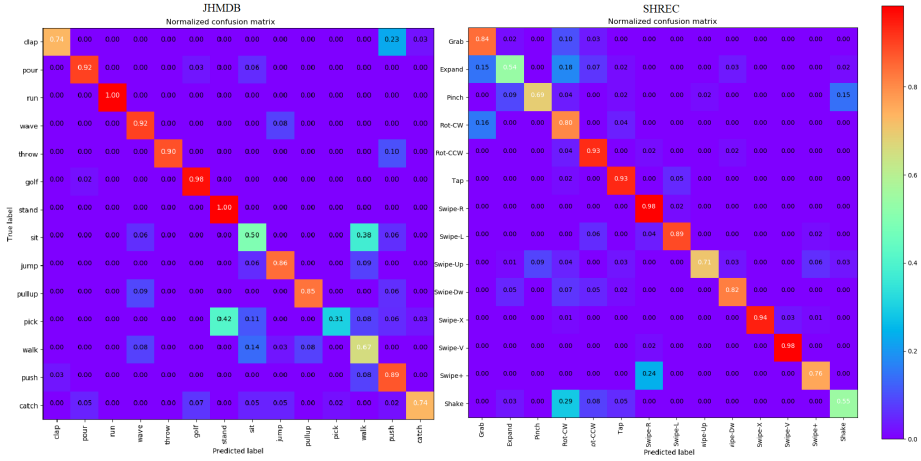

Fig. 10: Normalized Confusion Matrix for JHMDB (left) and DHG-14 labels (right)

To conclude the analysis of the model's performance, the ROC (Receiver Operating Characteristics) curves, AUC (Area Under the Curve) were plotted and calculated for each dataset. The ROC curves for JHMDB and DHG-28 are

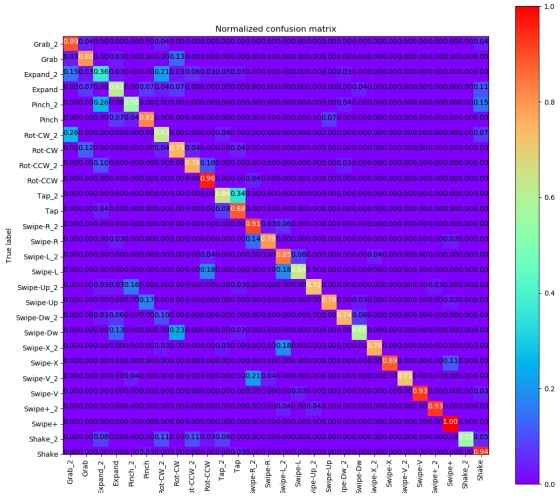

Fig. 11: Normalized Confusion Matrix for DHG with 28 labels

shown in Fig. 12. As can be seen, both datasets show curves well above the random guessing line (blue dashed line). Still, some actions show better results than others, for instance, the *pick* action shows an AUC of only 0.65, which complies with the false positives shown in the confusion matrix (Fig. 10 left), while *run* and *golf* display an AUC of 1, which means there is 100% chance that model will be able to distinguish between a positive class and negative class for these actions. In DHG dataset, the action with worst AUC is *expand_2*, with 0.67, and the best are *swipe +* and *swipe v* with 1 and 0.98 respectively. For the complete model, the AUC mean is 0.944 for the DHG-14 dataset, 0.920 for DHG-28, and 0.895 for JHMDB.

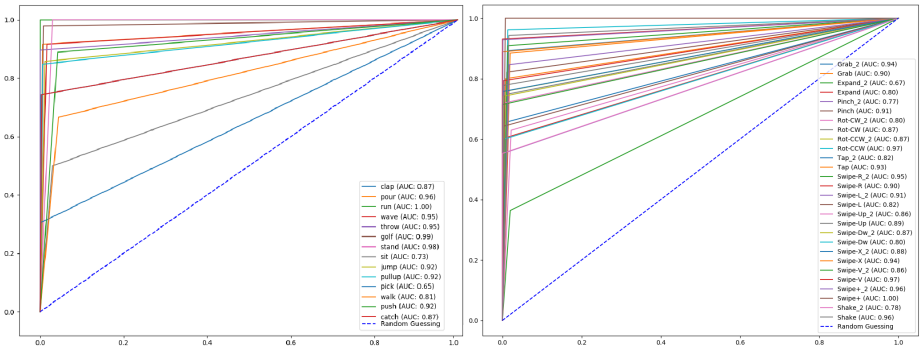

Fig. 12: ROC curve and AUC scores for JHMDB (left) DHG with 28 labels (right)

5 Conclusions

Recent LSTM-based action recognition systems utilize raw data with complex models, or features extracted from the data that need to be adapted according to the dataset being used. In this work, an approach to explore the impact of different feature types in skeleton action recognition solutions with LSTMs was presented. For this, several data preprocessing and feature extraction methods were applied, obtaining multiple features of different natures: time-based, space-based, and combined, adapted from consolidated state-of-art works. We introduced the hypothesis that with combined features from geometric and temporal natures, a lightweight LSTM can be used to achieve state-of-art performance in different types of skeleton datasets. In order to test this hypothesis, a vanilla LSTM model with two layers was built. Two different datasets (JHMDB and DHG) with distinct characteristics were evaluated in four input variations: the unfeatured dataset, temporal features, geometric features, and combined features.

Results show that geometric features cause more significant improvements in the LSTM model (almost 9% in accuracy in some datasets), but the combined features allow to attain best overall results across both models and applications (body actions and hand actions) in all datasets (85.2% for DHG-14, 81.3% for DHG-28 and 71.1% for JHMDB). Since LSTMs are good at exploiting robust temporal information, adding strong spatial information aids the system, but with combined features, simpler models are allowed since the temporal features are also provided as inputs. Not only good accuracies were obtained as mean state-of-the-art results were achieved for our datasets, although with some stability issues.

As future work, we intend to test these features with more benchmark datasets, as well as with different LSTM architectures. An excellent further improvement would also be to extend this study to CNN-based action recognition systems, testing if the same type of features additionally improves the system's performance.

References

1. Chen, Cheng, et al.: Learning a 3D human pose distance metric from geometric pose descriptor. In IEEE Transactions on Visualization and Computer Graphics, 17.11, pp. 1676-1689, 2010.
2. Chen, Xinghao, et al. Mfa-net: Motion feature augmented network for dynamic hand gesture recognition from skeletal data. In Sensors, 19.2: 239, 2019.
3. Choutas, V., Weinzaepfel, P., Revaud, J., and Schmid, C..: Potion: Pose motion representation for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7024-7033, 2018.
4. Cippitelli, Enea, et al.: A human activity recognition system using skeleton data from RGBD sensors. In: Computational intelligence and neuroscience 2016, 2016.
5. Dai, Cheng, Xingang Liu, and Jinfeng Lai.: Human action recognition using two-stream attention based LSTM networks. In: Applied Soft Computing 86 (2020): 105820, 2020.

6. De Smedt, Q., Wannous, H., and Vandeborre, J. P.: Skeleton-based dynamic hand gesture recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1-9, 2016.
7. Gammulle, H., Denman, S., Sridharan, S., and Fookes, C.: Two stream lstm: A deep fusion framework for human action recognition. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 177-186, IEEE, 2017.
8. Hou, Jingxuan, et al.: Spatial-temporal attention res-TCN for skeleton-based dynamic hand gesture recognition. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 0-0.
9. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., and Black, M. J.: Towards understanding action recognition. In Proceedings of the IEEE international conference on computer vision, pp. 3192-3199, 2013.
10. Li, Chao, et al.: Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp. 786-792, 2018.
11. Liu, J., Wang, G., Duan, L. Y., Abdiyeva, K., and Kot, A. C.: Skeleton-based human action recognition with global context-aware attention LSTM networks. In: IEEE Transactions on Image Processing, 27(4), pp. 1586-1599, 2017.
12. Ludl, D., Gulde, T., and Curio, C.: Simple yet efficient real-time pose-based action recognition. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 581-588, IEEE, 2019.
13. Shahroudy, A., Liu, J., Ng, T. T., and Wang, G.: 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, 2016.
14. Yang, Fan, et al.: Make Skeleton-based Action Recognition Model Smaller, Faster and Better. In: Proceedings of the ACM Multimedia Asia on ZZZ, pp. 1-6, 2019.
15. Wang, Jindong, et al.: Deep learning for sensor-based activity recognition: A survey. In: Pattern Recognition Letters 119, pp. 3-11, 2019.
16. Zeng, Ming, et al.: Convolutional neural networks for human activity recognition using mobile sensors. In: 6th International Conference on Mobile Computing, Applications and Services. IEEE, 2014. pp. 197-205.
17. Zhang, S., Yang, Y., Xiao, J., Liu, X., Yang, Y., Xie, D., and Zhuang, Y.: Fusing geometric features for skeleton-based action recognition using multilayer LSTM networks. In IEEE Transactions on Multimedia, 20(9), pp. 2330-2343, 2018.