Human activity recognition with capsule networks [⋆]

Laura Llopis-Ibor[1 [0000−0002−6163−9823]],
Alfredo Cuesta-Infante[1 [0000−0002−3328−501X]],
Cesar Beltran-Royo[1 [0000−0002−4628−8340]], and
Juan José Pantrigo[1 [0000−0002−7175−3371]]

Universidad Rey Juan Carlos, Móstoles, Spain.
{laura.llopis, alfredo.cuesta, cesar.beltran, juanjose.pantrigo}@urjc.es

Abstract. Human activity recognition is a challenging problem, where deep learning methods are showing to be very efficient. In this paper we propose the use of capsule networks. This type of networks have proved to generalize better to novel viewpoints than convolutional neural networks. We show that the use of capsule networks into a straightforward architecture, between a convolutional preprocessing stage to extract visual features and a header for carrying out the task, is able to attain competitive results with spatio-temporal data without the use of any kind of recurrent neural network. Moreover, an analysis of the obtained results shows that our architecture is capable of learning the properties that encode the spatio-temporal dynamics of the movements that characterize each activity.

Keywords: Capsule Network · Human Activity Recognition · Skeleton Based Action Recognition · Image Embedding

1 Introduction

Broadly speaking, Human Activity Recognition (HAR) is the task of assigning the correct activity label at the end of a sequence of video frames. Attaining a good HAR performance enables the development of many applications in Human-Computer Interaction (HCI) [7] such as industrial machinery operating without a dashboard, assisting elderly or people with disabilities in everyday tasks [5] as well as at work or developing natural and immersive interfaces.

HAR presents many challenges. If the video sequence has been taken from a single monocular camera it is frequent to have auto-occlusions, there is no direct information of the depth dimension, and obviously it is necessary to rely on computer vision techniques to extract the features out of each frame. On the

[⋆] This work has been funded by the Spanish Government research fundings RTI2018-098743-B-I00 (MICINN/FEDER) and Y2018/EMT-5062 (Comunidad de Madrid). Dr. Beltran-Royo is funded by the Spanish Government research funding RTI2018-094269-B-I00.

other hand, when having multiple views another extra challenge is to match and fuse all the information collected. Some devices provide a sequence of *skeletons*, i.e. a tree representing a series of joints in the human body, in which each node represents 2D or 3D coordinates of the corresponding joint (depending on the device). In this case the challenge is to distinguish tasks that are similar when the information provided by the image is missing, such as combing and waving. Background information is usually irrelevant to the activity being performed. Using 3D skeletons results in more robust models because all the background information or illumination changes present in images are ignored. It is also a less ambiguous representation than its 2D counterpart.

In this paper we present a capsule network (Capsnet) for HAR based on 3D skeletons. As [17] claims, one of the drawbacks of capsule networks is that they try to model every entity found in an image; thus, working on skeletons we get over this problem. Specifically, we use Tree Structure Skeleton Image (TSSI) to convert a sequence of 3D joint arrays into an image. Then, this image is fed into the capsule network. Our hypothesis is that the benefits of capsule networks are able to extract and relate the time dependencies from the TSSI without using any kind of Recurrent Neural Network mechanisms such as Long-Short Term Memories (LSTM) or Gated Recurrent Units (GRU).

2 Related Work

HAR is a field of research that has attracted great interest over the last decade. Early approaches consist of hand-crafted features and classifiers trained to recognize the activities. Among the variety of hand-crafted features proposed are the camera motion corrected descriptors [21], saliency-aware matching kernels [14], simplified Fisher kernel representations [25] or part-based multiple features [11]. However, due to the decreasing price of motion-capture devices in recent years, and the rise of their availability [2], the accessibility to skeletal-based data has increased. Thus, research in the field has recently shifted towards fully automatic methods based on deep learning.

Skeleton-based Deep Learning HAR. Early works focused on the use of recurrent neural networks (RNN) to model the long-term spatial and temporal relationship between joints. Wang *et al.* [22] propose a two-stream RNN architecture that analyzes both the contextual dependence in the time domain and the spatial configuration of the skeletons to then fuse the result for action recognition. Song *et al.* [20] employ a spatio-temporal attention model based on LSTM networks to focus on the most relevant frames and joints for a given action.

Given the success of convolutional neural networks in image-based tasks, they have also been used in the HAR domain. Wang *et al.* [23] propose a convolution-based architecture where they use *Joint Trajectory Maps* as encoding, representing the spatial configuration of the joints and their trajectory in three images through color coding. The resulting maps graphically represent the joint trajectory, motion direction, body parts and motion magnitude. Caetano *et al.* [4]

propose an image encoding of the information regarding joint motion over time. Unlike previous works, Núñez *et al.* [15] do not use a graphical representation of the skeletons. In their work they apply a CNN directly on the 3D skeleton data and use the obtained features to feed a LSTM layer.

The most recent works focus on modeling skeletons as graphs. Following this representation, Yan *et al.* [24] applies a Graph Convolutional Neurnal Network (GCN). They extend this representation to the temporal domain by linking joints between consecutive skeletons. In this line, Huang *et al.* [9] presents a learneable approach to capture body parts information. Their work manages to highlight important body parts in the skeleton and combines this information with joint-level information for activity recognition. Si *et al.* [19] propose a LSTM unit that applies graph convolutions to work with graph-structured skeletal data. They state that their approach is able to learn the co-occurrence relationship between spatio-temporal features in addition to those features.

Capsule Networks for HAR. The idea of capsules was first proposed by Hinton *et al.* [8], however the breakthrough is due to Sabour et al. [17] with their trainable approach to these units. In their work they prove that capsule networks are more robust to affine transformations than convolutional counterparts.

Algamdi *et al.* [1] builds up on the work of Sabour et al. to adapt it to the HAR domain. To do so, they use a deeper CNN to create features from which the capsules are created. This enables the extraction of more complex patterns as the input of the model is a video sequence that includes unnecessary background information for action recognition. Moreover, they implement a weight pooling step on the previous features to reduce the number of created capsules. This results in a reduction of the computational cost of this model.

Jayasundara *et al.* [10] apply capsule networks to estimate the optical flow between pairs of images. They use three consecutive layers of capsules to calculate the motion features that are then fed to an autoecoder network to retrieve the final motion image, which is subsequently used for action recognition. The authors remark that capsules are capable of preserving the structure of entities and therefore they don't need to use a multi-scale approach nor other additional tools to estimate optical flow.

The previously discussed works apply capsule networks on RGB video frames from action sequences. We take advantage of the similarity between the capsule architecture and the CNNs and use an image representation of skeletal data as input to our network. To the best of our knowledge, we are the first approach to use capsule networks on image encoded skeletal data.

3 HAR Capsules

Capsule networks were proposed to solve the shortcomings of CNNs for image-based problem as described in Section 2. In this work, the human activities are defined by a sequence of joint positions. In order to apply a capsule network architecture we encode the data into images following the Tree Structure Skeleton Image (TSSI), introduced in [26].

3.1 Sequence Cutting

The input data of our model is a sequence composed of skeletons, one skeleton per video frame. Each skeleton consists of 25 joints where each joint is a position in a three-dimensional space. Since not all activities last the same time, the sequences will contain a variable number of skeletons. The first step to be carried out in the preprocessing stage is creating sequences of fixed length. To this end, we create sub-sequences form each sequence by selecting quasi-equally spaced skeletons.

We start by defining the number of sub-sequences that can be derived from a sequence as $b = \frac{|S|}{l}$; where $|S|$ is the length of a sequence of skeletons and l is the network input sequence length. Since the number of sub-sequences can be a real value, the last sub-sequence will contain repeated skeletons from the other sub-sequences.

The list of indices for each sub-sequence is the set $I = \{\lfloor s + i \cdot b \rfloor\}$; where $s \in \{1, \ldots, \lfloor b+1 \rfloor\}$, and $i \in \{0, \ldots, l-1\}$ for each s. Applying this list of indices I to a sequence S generates the set of sub-sequences $S' = \{J_i \mid i \in I, J_i \in S\}$.

3.2 Image Embedding

Capsule networks focus on extracting properties that represent entities from images and on putting them together to create the entities that represent the classes sought in a classification problem. The first two layers of our capsule architecture make use of convolutions for the aforementioned extraction. Therefore, to use this type networks it is necessary to convert the skeletal data into images. This problem has already been addressed in the HAR domain by approaches that have had this same issue, such as CNNs.

In order to convert the three-dimensional positions of the joints into an image representation we use the Tree Sctructure Skeleton Image codification [26]. It starts with an arrangement of the joints along the columns of an image and the skeletons along the rows. According to [26], this distribution ensures that a convolution operation will only establish relationships between connected or temporally adjacent joints. Since the position of each joint is three-dimensional, three grayscale images are obtained, which are used as the channels of an image representation of the sequence.

The arrangement that avoids unconnected joints in contiguous columns is obtained by traversing the graph; i.e. the skeleton tree where the root node is the spine joint. This ordering is applied to the joints of each skeleton in a sequence, as shown in Figure 1.

Finally, values are normalized to the unit interval. Aditionally, for this task we propose the use of a bone normalization and scaling process. First, we set the origin point of each joint to their parent joint. Next, we normalize the length of each bone. This process removes inter-subject variability from the data set and makes this representation rotation and translation invariant. In addition, it emphasizes the movement performed by a joint without depending on its predecessors. After this normalization, all the bones of the skeleton share the same motion range, a three-dimensional sphere.

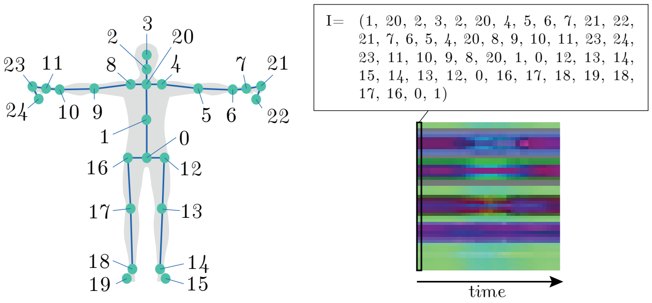
3
2
20
23
11
8
4
7
21
24
10
9
5
6
22
1
0
16
12
17
13
18
14
19
15
I= (1, 20, 2, 3, 2, 20, 4, 5, 6, 7, 21, 22, 21, 7, 6, 5, 4, 20, 8, 9, 10, 11, 23, 24, 23, 11, 10, 9, 8, 20, 1, 0, 12, 13, 14, 15, 14, 13, 12, 0, 16, 17, 18, 19, 18, 17, 16, 0, 1)
time

Fig. 1. TSSI encoding process. On the left, the skeleton graph where joints are the numbered nodes. On the right, the result of the depth-first search on the skeleton employed to encode the first column of the TSSI image. The color of each *pixel* is due to (x, y, z) coordinates used as RBG channels.

3.3 Capsule Network

The architecture used in this work is similar to the one proposed in [17], and depicted in Figure 2. It can be divided in the following components:

Base Feature Extraction. The input image is first processed by a convolutional layer with 236 kernels of size 3 × 3 pixels, stride of 1 and ReLU activation. The output of this layer is a set of 236 feature maps of 21 × 47 pixels.

Low Level Capsules. The next layer applies 18 groups of 11 convolutional kernels of 13 × 13 pixels and stride 2 on the previous feature maps. The results are 18 groups of 11 feature maps of size 5 × 18 pixels. Each group is split along the rows and columns into 90 vectors of length 11, resulting a total of 1620 vectors in $\mathcal{R}^{11}$. Each one of those vectors d_i is then processed by the squash function to rescale its module to the unit interval in a non-linear fashion,

$$squash(d_i) = \frac{\|d_i\|^2}{1 + \|d_i\|^2} \frac{d_i}{\|d_i\|} \tag{1}$$

The result are denoted *low level* capsules $u_i \in [0, 1]^{11}$.

High Level Capsules. The 18 groups of low level capsules represent 18 patterns of motion throughout the image. In this architecture, those patterns are used to establish part-whole relationships with the *high level* capsules. In this work there is a high level capsule for each class in the data set.

First, the properties of each high level capsule are predicted from the low level ones. This is done by a transformation matrix $W_{i,j}$ of size 11 × 21 for each pair of low level capsules and high level capsules. As a result we obtain a *predicted* high-level capsule $\hat{u}_i$. There are $11 \times 21 \times 1620 \times C$ trainable parameters in this layer, where C is the number of classes of the data set.

After each capsule has been transformed we use the dynamic routing algorithm proposed in [17] to cluster together the predictions and to create higher level capsule properties. This clustering is done iteratively based on the similarity between all the predictions for a high level capsule. The obtained values are

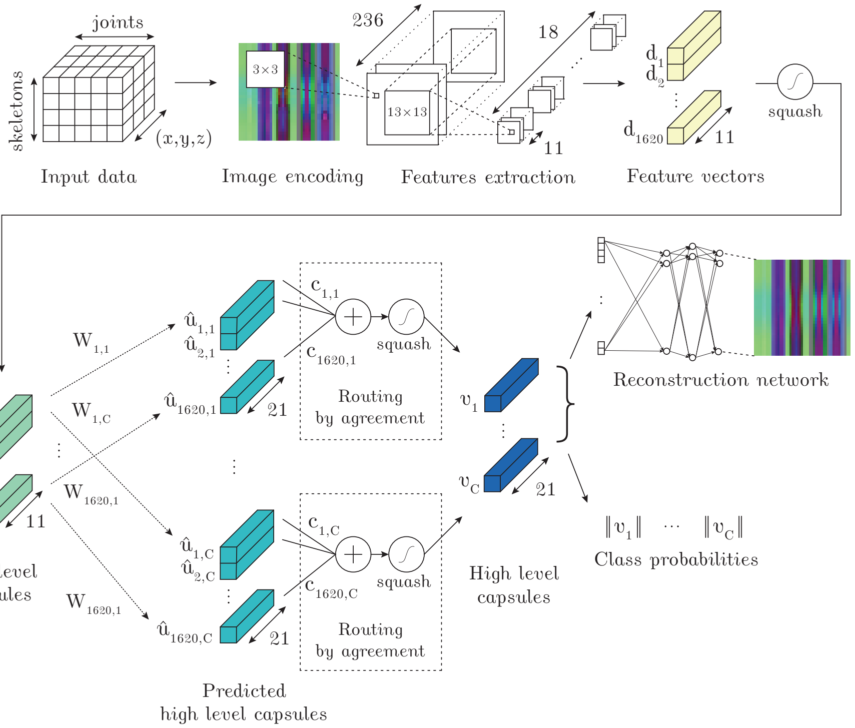

Fig. 2. Diagram illustrating the different phases that compose the capsule network presented in this work. First, a sequence of skeletons is encoded into a TSSI [26] image. Then feature vectors are extracted and squashed in order to obtain low level capsules. These capsules are transformed and clustered together to generate high level capsules. Finally, the encoded image is reconstructed and the probability for each class is obtained.

then processed by the squash function (1) to generate each high level capsule, v_j. The module $\|v_j\|$ of each high level capsule is used as its class probability.

Reconstruction Network. To regularize the training of the capsule network we use a fully connected network to reconstruct the input image from the high level capsules, as suggested in [17]. Only the high level capsule corresponding to the true class is taken into account. This is done by setting all high level capsules to zeros except the one corresponding to the true class. This network has two hidden layers with sizes of 3072 and 6144, respectively.

Loss. The total loss of the architecture is computed from the loss of the capsule network and the reconstruction network, $\mathcal{L}_{cap}$ and $\mathcal{L}_{reco}$ respectively. For the capsule network, the margin loss proposed in [17] is used, with $m^+ = 0.7$, $m^- = 0.3$ and $\lambda = 2$. For the reconstruction network, the loss is the mean squared error between the input image and the reconstructed image. In order to balance these two losses we use two trainable coefficients s_{cap} and s_{reco}. These coefficients are also summed to the total loss to self-balance their influence as detailed in [16]:

$$\mathcal{L} = e^{-s_{cap}}\mathcal{L}_{cap} + s_{cap} + e^{-s_{reco}}\mathcal{L}_{reco} + s_{reco} \tag{2}$$

4 Experiments

In this section we describe the data sets used for the evaluation of our proposal and the setting of our experiments. We analyze the results obtained and perform a comparison against works from the state of the art in skeleton based HAR.

The experiments were conducted on a Intel Xeon E5-2698v4, 2.20 GHz CPU and a NVIDIA Tesla V100 GPU with 32 GB of RAM. We trained our network for 100 epochs with a learning rate of 10^{-5} using the Adam optimization algorithm and batch size of 36. At epoch 50 we decreased the learning rate by a factor of 0.1. All of the hyperparameters have been selected by a constrained random search that ensured the best network accuracy on the validation phase of the training.

4.1 Data Sets

The proposed architecture has been evaluated on two widely used activity recognition data sets. The first one, NTU RGB+D [18], is a data set composed of 56880 sequences representing 60 actions performed by 40 subjects and captured by Kinect V2 cameras from 80 viewpoints. The actions are captured simultaneously by 3 cameras. Each sequence is composed of a variable number of skeletons and each skeleton is composed of 25 joints. For actions involving two subjects we have only used the skeleton of the main actor, as described in [18]. The authors of this data set propose two evaluation protocols. In the first protocol, 20 subjects are used for training and another 20 for evaluation (Cross-Subject). The second protocol uses the sequences captured by camera 2 and 3 for training and those of camera 1 for evaluation (Cross-View). The second data set, NTU RGB+D 120 [12], is an extension of the previous set. The size of the data set increases to 114480 sequences representing 120 actions performed by 106 subjects and captured from 155 viewpoints. As before, the authors propose two evaluation protocols: Cross-Subject and Cross-Setup. For the Cross-Subject protocol, 53 specific subjects should be used for training and 53 for evaluation. On the other hand, for the Cross-Setup protocol, 16 setups are used for training and 16 setups for evaluation.

4.2 Results and Discussion

The results obtained for the above data sets using their evaluation protocols, together with state-of-the-art works are shown in Table 1. Our architecture is not using either recurrent neurons nor graph neural network, and yet our results are competitive with respect to the state of the art. The most similar proposal is

Table 1. Results of the proposed architecture and state-of-the-art works. The Cross-Subject (CS60) and Cross-View (CV60) protocols from the NTU RGB+D were used. For the NTU RGB+D 120 the Cross-Subject (CS120) and Cross-Setup (CST120) protocols were used. These results are reported by their respective authors. The results of the work marked with * are reported in [26].

Model	**CS60(%)**	**CV60(%)**	**CS120(%)**	**CST120(%)**
Deep RNN [18]	56.2	64.0	-	-
HBRNN* [6]	59.1	64.0	-	-
Deep P-LSTM [18]	62.9	70.2	-	-
Trust Gate LSTM [13]	69.2	77.7	-	-
TSSI [26]	73.1	76.5	-	-
TSRJI [3]	73.3	**80.3**	65.5	59.7
SkeleMotion [4]	**76.5**	84.7	**67.7**	**66.9**
Ours	74.2	77.1	63.2	64.6

TSSI [26], which consist of a ResNet-50. Moreover, our results also outperform works that include recurrent networks. We noticed a significant number of false positives between actions that have a similar spatial configuration. In addition, the network is also able to distinguish between actions where the subject performs the same movements in a different order. This leads us to think that our proposal is able to capture spatial-temporal relationships. However these similar activities degrade the overall performance of the classifier.

In the proposed architecture, a fully connected network is used to reconstruct the input image from the high level capsules generated by the capsule network. The output capsule generated by the network for an input sequence contains the properties that define the action performed in it. We can visualize its effect by modifying the values of these properties and depicting the reconstructed skeleton to see which characteristic of the subject's movement each one encodes. We do this modification by adding values from an interval [-0.25, 0.25] with a 0.1 step. Figure 3 shows the result of modifying the values of one property of an output capsule. This modified property characterizes the right and left forearms movement. As the value of this property increases, the forearms movement becomes wider.

5 Conclusions

In the present work, a capsule network architecture for activity recognition based on skeletal data has been presented. The results of our proposal are better than other works employing methods based on CNNs. In the analysis performed using the reconstruction network, it can be observed that the capsule network is able to isolate the properties that define the motion of the human body. This suggests that this type of network has potential in the field of activity recognition and it's able to model spatio-temporal relationships between joints. To further our research we intend to develop a new routing algorithm that models human movement dynamics and to introduce attention mechanisms.

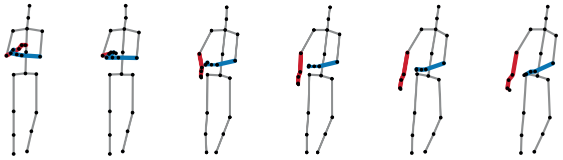

Fig. 3. Reconstructed input skeleton modifying the third high level capsule property of the action "take off jacket". Wider limbs are the most affected by the previous modification.

References

1. Algamdi, A.M., Sanchez, V., Li, C.: Learning temporal information from spatial information using capsnets for human action recognition. In: IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). pp. 3867–3871 (2019). https://doi.org/10.1109/ICASSP.2019.8683720
2. Altun, K., Barshan, B., Tunçel, O.: Comparative study on classifying human activities with miniature inertial and magnetic sensors. Pattern Recognition **43**(10), 3605–3620 (2010). https://doi.org/10.1016/j.patcog.2010.04.019
3. Caetano, C., Brémond, F., Schwartz, W.R.: Skeleton image representation for 3d action recognition based on tree structure and reference joints. In: 32nd SIBGRAPI Conf. on Graphics, Patterns and Images (SIBGRAPI). pp. 16–23 (2019). https://doi.org/10.1109/SIBGRAPI.2019.00011
4. Caetano, C., Sena, J., Brémond, F., Dos Santos, J.A., Schwartz, W.R.: Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. In: 16th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–8 (2019). https://doi.org/10.1109/AVSS.2019.8909840
5. Chamroukhi, F., Mohammed, S., Trabelsi, D., Oukhellou, L., Amirat, Y.: Joint segmentation of multivariate time series with hidden process regression for human activity recognition. Neurocomputing **120**, 633–644 (2013). https://doi.org/10.1016/j.neucom.2013.04.003
6. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2015). https://doi.org/10.1109/CVPR.2015.7298714
7. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. Computer Vision and Image Understanding **108**(1), 52–73 (2007). https://doi.org/10.1016/j.cviu.2006.10.012
8. Hinton, G.E., Krizhevsky, A., Wang, S.D.: Transforming auto-encoders. In: Artificial Neural Networks and Machine Learning – ICANN 2011. pp. 44–51 (2011). https://doi.org/10.1007/978-3-642-21735-7_6
9. Huang, L., Huang, Y., Ouyang, W., Wang, L.: Part-level graph convolutional network for skeleton-based action recognition. In: The 34th AAAI Conf. on Artificial Intelligence. pp. 11045–11052 (2020). https://doi.org/10.1609/aaai.v34i07.6759
10. Jayasundara, V., Roy, D., Fernando, B.: Flowcaps: Optical flow estimation with capsule networks for action recognition. In: Proc. of the IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV). pp. 3409–3418 (2021)
11. Li, M., Leung, H., Shum, H.P.H.: Human action recognition via skeletal and depth based feature fusion. In: Proc. of the 9th Int. Conf. on Motion in Games. p. 123–132 (2016). https://doi.org/10.1145/2994258.2994268

12. Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y., Kot, A.C.: Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. IEEE Trans. on Pattern Analysis and Machine Intelligence **42**(10), 2684–2701 (2020). https://doi.org/10.1109/TPAMI.2019.2916873
13. Liu, J., Shahroudy, A., Xu, D., Wang, G.: Spatio-temporal lstm with trust gates for 3d human action recognition. In: European Conf. on Computer Vision (ECCV). pp. 816–833 (2016). https://doi.org/10.1007/978-3-319-46487-9_50
14. Nguyen, T.V., Song, Z., Yan, S.: Stap: Spatial-temporal attention-aware pooling for action recognition. IEEE Trans. on Circuits and Systems for Video Technology **25**(1), 77–86 (2015). https://doi.org/10.1109/TCSVT.2014.2333151
15. Núñez, J.C., Cabido, R., Pantrigo, J.J., Montemayor, A.S., Vélez, J.F.: Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. Pattern Recognition **76**, 80–94 (2018). https://doi.org/10.1016/j.patcog.2017.10.033
16. Ramírez, I., Cuesta-Infante, A., Schiavi, E., Pantrigo, J.J.: Bayesian capsule networks for 3d human pose estimation from single 2d images. Neurocomputing **379**, 64–73 (2020). https://doi.org/10.1016/j.neucom.2019.09.101
17. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Advances in Neural Information Processing Systems. vol. 30, pp. 3856–3866 (2017)
18. Shahroudy, A., Liu, J., Ng, T., Wang, G.: Ntu rgb+d: A large scale dataset for 3d human activity analysis. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 1010–1019 (2016). https://doi.org/10.1109/CVPR.2016.115
19. Si, C., Chen, W., Wang, W., Wang, L., Tan, T.: An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019). https://doi.org/10.1109/CVPR.2019.00132
20. Song, S., Lan, C., Xing, J., Zeng, W., Liu, J.: An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In: Proc. of the 31st AAAI Conf. on Artificial Intelligence. p. 4263–4270 (2017)
21. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: IEEE Int. Conf. on Computer Vision. pp. 3551–3558 (2013). https://doi.org/10.1109/ICCV.2013.441
22. Wang, H., Wang, L.: Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017). https://doi.org/10.1109/CVPR.2017.387
23. Wang, P., Li, W., Li, C., Hou, Y.: Action recognition based on joint trajectory maps with convolutional neural networks. Knowledge-Based Systems **158**, 43–53 (2018). https://doi.org/10.1016/j.knosys.2018.05.029
24. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence. pp. 7444–7452 (2018)
25. Yang, X., Tian, Y.: Super normal vector for activity recognition using depth sequences. In: IEEE Conf. on Computer Vision and Pattern Recognition. pp. 804–811 (2014). https://doi.org/10.1109/CVPR.2014.108
26. Yang, Z., Li, Y., Yang, J., Luo, J.: Action recognition with spatio–temporal visual attention on skeleton image sequences. IEEE Trans. on Circuits and Systems for Video Technology **29**(8), 2405–2415 (2019). https://doi.org/10.1109/TCSVT.2018.2864148