Real-time Recognition of Daily Actions Based on 3D Joint Movements and Fisher Encoding

Panagiotis Giannakeris[1], Georgios Meditskos[1], Konstantinos Avgerinakis[1], Stefanos Vrochidis[1], and Ioannis Kompatsiaris[1]

Centre for Research & Technology Hellas, Information Technologies Institute, Thessaloniki, Greece
{giannakeris,gmeditsk,koafgeri,stefanos,ikom}@iti.gr

Abstract. Recognition of daily actions is an essential part of Ambient Assisted Living (AAL) applications and still not fully solved. In this work, we propose a novel framework for the recognition of actions of daily living from depth-videos. The framework is based on low-level human pose movement descriptors extracted from 3D joint trajectories as well as differential values that encode speed and acceleration information. The joints are detected using a depth sensor. The low-level descriptors are then aggregated into discriminative high-level action representations by modeling prototype pose movements with Gaussian Mixtures and then using a Fisher encoding schema. The resulting Fisher vectors are suitable to train Linear SVM classifiers so as to recognize actions in pre-segmented video clips, alleviating the need for additional parameter search with non-linear kernels or neural network tuning. Experimental evaluation on two well-known RGB-D action datasets reveal that the proposed framework achieves close to state-of-the-art performance whilst maintaining high processing speeds.

Keywords: Action Recognition · 3D Human Joints · Gaussian Mixture Modeling · Fisher Encoding.

1 Introduction

Action recognition in general is an important task in computer vision with a very wide range of possible applications. Recently, many of those are part of ambient assisted living environments where users are monitored by static cameras placed inside their homes. Many of the patients do not feel comfortable wearing any sensory equipment or wearable cameras that may generally feel bulky and invasive.

For those reasons, depth cameras that can capture 3D positions of human skeleton joints are very well suited to this task. Not only the joint locations can be captured at real time speeds, but also the amount of processing time required is small compared to techniques that involve processing RGB video frames. Many of the recent works in action recognition follow closely the deep learning trend [10–12, 19, 20, 27] in order to achieve competitive results, however

they do not give much attention to efficiency. Deep learning techniques usually require lots of training data and careful parameter tuning to avoid overfitting. Additionally, expensive GPUs are needed during training or inference in order to run at acceptable speeds.

In this paper, inspired by [31], we use the Moving Pose descriptor at the basis of our approach for representing actions and extend its applicability by incorporating an aggregation scheme that transforms multiple pose descriptors of an action clip into a compact meaningful encoding using prototypical pose features. The Moving Pose was originally proposed as a descriptor that captured for each frame 3D pose configurations along with differential properties, like the speed and acceleration of human parts. It was paired with a modified kNN classifier which characterized each sequence based on the action label that the majority of the individual frame descriptors had been assigned to.

While the low-level frame descriptors are powerful and are calculated very fast, the application of the modified kNN that is proposed inserts model parameters that need to be optimized (i.e. the number of neighbors and a confidence threshold for classification). Another drawback is that the kNN classifier can be generally slower in some cases of high data dimensionality. Our contribution lies mainly on the creation of compact representations by a sophisticated descriptor aggregation scheme that are suitable for Linear Support Vector Machine classification. The aggregation scheme involves building of a vocabulary of pose prototypes using Gaussian Mixture modeling during training and encoding a sequence of an arbitrary number of frame descriptors into meaningful high-level action representations using Fisher encoding.

Our proposed scheme alleviates the need to search the training data for discriminative neighbors during inference. It also achieves comparable performance even without carefully tuning the weighting parameters that control the relative importance of the pose derivatives like in the original method. Additionally, it results in higher classification accuracy when paired with a Linear Support Vector Machine and improves the overall processing speed which is now faster by 100%. In this work we perform our experiments on public RGB-D datasets that use the Microsoft Kinect V1 sensor to acquire skeletons, but the method can be generally applied to every sensor of this type.

The rest of this paper is structured as follows: In Section 2 we present the state-of-the-art related work, while in Section 3 we describe in detail our proposed methodology. In Section 4 our experimental work and evaluation results are reported and in Section 5 conclusions are drawn and future work is discussed.

2 Related Work

Previous works on third person action recognition can be categorized based on the type of data that they process, whether it is RGB or depth videos. For the first category, most of the pre-CNN era works dealt with motion analysis using optical flow and dense trajectories [1, 8, 9, 25, 26], extracting legacy low-level descriptors like Histograms of Oriented Gradients (HOG), Histograms of

Optical Flow (HOF) and Motion Boundary Histograms (MBH), to represent the visual and motion features around keypoints tracked in time. More advanced techniques quickly began to take advantage of the temporal structure of visual patterns [7, 16]. Shortly after the CNN breakthrough, the direction was steered naturally towards deep learning approaches. Some of them combined various modes of video frames, mainly an RGB channel and an optical flow channel, constructing multi-stream CNNs [6, 23, 29], in order to extract deep CNN visual and motion features and represent actions based on multimodal fusion. In later works, information from human pose was more effectively captured using pose-based CNN features [2]. Other techniques that leverage temporal information rely on RNNs. More specifically, the modern technique of deep visual attention was combined with RNNs in [4]. Very recently, modifications on legacy techniques were presented, like in [22] where optical flow derivation features (OFF) where plugged in a legacy CNN-based action recognition framework, or in [5] where the two-stream CNN approach was modernized by injecting residual connections.

More related to this work are techniques that fall into the second category, i.e. the ones that use information from depth videos and more specifically try to represent actions from 3D human skeleton joint positions. Earlier works focused on extracting features from human joints and characterize their movement [28, 30, 31]. In addition, the features were designed to be invariant to noise, translation and viewpoint temporal misalignment. In [28] LOP features of joints were combined into actionlets that represented particular conjunctions of the joints. A data mining approach was used in order to find the most significant combinations and make an ensemble out of them to represent the actions. In [30] view invariant joint features, named HOJ3D, were extracted and used to discover prototypical poses and Hidden Markov Models were used to describe their temporal evolution in order to represent actions. Temporal joint displacement features were combined with spatial features to form a simple spatiotemporal fusion scheme in [32] by calculating pairwise distances of joint positions. A more elaborate technique was proposed in [24] that represented human actions as curves in a Lie group in order to model 3D geometric relationships between human joints. The processing of those curves involved mapping them from the Lie group to Lie algebra and then using Dynamic Time Warping and Linear SVM for classification.

Later works gravitate towards deep learning methods and specifically Long Short Term Memory networks (LSTM). A a spatio-temporal LSTM was proposed in [14] which aimed to model the temporal dynamics and spatial dependencies. [17] extends upon the Lie group representation of [24] with the inclusion of CNNs and LSTMs in the framework. The transformed vectors sequenced from the Lie group were fed into stacked units of a 1-D CNN across the temporal domain and then a separate LSTM layer was trained to learn temporal dependencies and classify the action segments. A non deep learning method has been recently proposed in [15] which deployed a similar encoding scheme with ours. Its purpose is to encode multiple spatial and temporal features of different joint subgroups. Features were aggregated using VLAD encoding and then optimal feature combinations were found based on metric learning.

3 Methodology

The extraction of human skeleton movement patterns constitute the basis of our approach so as to characterize pre-segmented action sequences. Action videos can be seen as sequential frames of body pose configurations. Therefore, spatial relationships between joints that form body poses in still frames are very informative towards understating the action that is taking place. This holds true for certain actions more than it does for others. For example, "walking" instances can be easily separated from "lying down" instances based on the relative positions of the head or torso joints. However, the movement and transitions of joint configurations is also critical in order to understand actions that cannot be characterized by static pose information only. For example, a still pose taken from a "walking" instance is not enough to uniquely define it and separate it from a "standing up" instance, because movement information is not present. Therefore, we focus on extracting low-level frame descriptors based not only on joint positions but also on joint displacement features inside short temporal windows centered around the current frame. Another important aspect is the expected variance of natural skeletal dimensions between different subjects. Directly computing displacement vectors in 3D space will result in inconsistent results due to lack of invariance to the subjects' natural body shapes. For this reason temporal features that encode velocity and acceleration quantities between adjacent frames are selected instead. The speed and direction of joint movement are informative features that can separate actions that are built by similar static poses but involve different direction of movement and the changes in speed and direction of joint movement can separate actions that involve periodic movements of joints like "walking" or "hand waving". Accordingly, we adopt the Moving Pose descriptor [31] which assumes that the pose, $P(t)$, is a continuous and differentiable function of the joint positions over time; thus, its second-order Taylor approximation in a short temporal window around the current time-step t_0 can be expressed as:

$$P(t) \approx P(t_0) + \delta P(t_0)(t - t_0) + \frac{1}{2}\delta^2 P(t_0)(t - t_0)^2 \tag{1}$$

where $\delta P(t)$ and $\delta^2 P(t)$ are the first and second order derivatives of the pose function and are effectively encoding information about the temporal changes in pose configuration regarding the center frame inside a short temporal window. This approximation can be used in order to describe every action frame with respect to the current static pose and the joint kinematics between adjacent frames, thus, incorporating the temporal dynamics of the movement.

3.1 Low-level Pose Descriptors

We assume that for every video frame the 3D joint positions of the skeleton are available as 3D vectors $j_i = [x_i, y_i, z_i]^T$, where $i = \{1, 2, \dots n\}$ is the total number of joints that the sensor can capture. At any given moment the static pose is given by concatenating all the joint vectors $P = [j_1, j_2, \dots j_n]$. The derivatives

are approximated numerically by using a temporal window of 5 frames centered at the current frame, as such:

$$\delta P(t_0) \approx P(t_1) - P(t_{-1}) \tag{2}$$

$$\delta^2 P(t_0) \approx P(t_2) + P(t_{-2}) - 2P(t_0) \tag{3}$$

The final low-level moving pose descriptor is the concatenation of the static pose vector and the first and second order derivatives. The derivative vectors are rescaled to unit norm. The original descriptor was proposed to insert two weighting parameters for the two derivatives that would effectively control their relative importance. However, we reject the insertion of the parameters in order to avoid optimizing them via cross-validation for specific datasets.

In order to compensate for skeleton variations across different subjects and to deal with noise in the data, the static poses are first normalized before the final low-level descriptor is formed. The average length of each limp of the skeleton is calculated first. Then, all the train and test skeleton limps are normalized to have the same average length for each limp type whilst maintaining the angles between joints (the direction vectors) that form each static pose. Normalizing the skeletons guarantees that the same types of limps will have the same length across not only different subjects but also different instances of the same subject. Each joint's position is also expressed using relative distance from the hip center, by subtracting the coordinates of the root joint. The last step ensures that the descriptor is invariant to camera parameters.

3.2 High-level Action Representations

Each action clip may have arbitrary duration and as such, an arbitrary number of moving pose descriptors are going to be extracted. What follows is our proposed framework for moving pose descriptor aggregation and encoding into high-level meaningful action representations.

We initiate the process by reducing the descriptor dimensionality using Principal Component Analysis. We choose as many principal components in order to guarantee that at least 98% of the original variance of the training set is maintained to the lower-dimensional space. Next, we intent to create a statistical model that can generate a number of prototypical descriptors from the training set. Those will represent the most discriminative static poses and pose transitions from the full training set of low-level descriptors across all subjects and action sequences. We do that using mixtures of Gaussians (GMM) which is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. The EM algorithm is applied in order to fit the mixture of Gaussians to the training set. This way, a moving pose vocabulary based on the most discriminative low-level descriptors is constructed. Any set of low-level action descriptors extracted from a sequence can then be expressed by the gradients of the log-likelihood of each descriptor under the GMM, with respect to the GMM parameters. This

process is known as Fisher Encoding [18] and the final representation is called a Fisher vector (FV).

Let $\{\mu_j, \Sigma_j, \pi_j; j \in R^L\}$ be the set of parameters for L Gaussian models, with μ_j, Σ_j and π_j standing respectively for the mean, the covariance and the prior probability weights of the j^{th} Gaussian. Assuming that the D-dimensional early descriptor is represented as $\overline{M}_i \in R^D; i = \{1, \ldots, N\}$, with N denoting the total number of descriptors, Fisher encoding is then built upon the first and second order statistics:

$$\begin{aligned} f_{1j} &= \frac{1}{N\sqrt{\pi_j}} \sum_{i=1}^{N} q_{ij} \sigma_j^{-1} (\overline{x}_i - \overline{\mu}_j) \\ f_{2j} &= \frac{1}{N\sqrt{2\pi_j}} \sum_{i=1}^{N} q_{ij} [\frac{(\overline{x}_i - \overline{\mu}_j)^2}{\sigma_j^2} - 1] \end{aligned} \tag{4}$$

where q_{ij} is the Gaussian soft assignment of descriptor M_i to the j^{th} Gaussian and is given by:

$$q_{ij} = \frac{exp[-\frac{1}{2}(M_i - \mu_j)^T \Sigma_j^{-1} (M_i - \mu_j)]}{\sum_{t=1}^{L} exp[-\frac{1}{2}(M_i - \mu_t)^T \Sigma_j^{-1} (M_i - \mu_t)]} \tag{5}$$

Distances that are calculated by Eq. 4 are next concatenated to form the final $2LD$-dimensional Fisher vector, $F_X = [f_{11}, f_{21}, \ldots, f_{1L}, f_{2L}]$, that characterizes each action clip. The final Fisher vector is calculated for every action clip sequence in the train and test set. One of the benefits of Fisher encoding is that it leads to representations that can now be classified using cost-less linear SVM classifier with high accuracy.

4 Experimental Work

In order to evaluate our proposed framework we experiment on two well-known RGB-D datasets for action recognition. Namely, they are the UT-Kinect dataset [30] and the MSR-Action3D dataset [13]. The UT-Kinect dataset contains 10 action classes performed twice by 10 subjects. This dataset poses significant intra-class, viewpoint variations and occlusions because the actors are interacting with objects during the action clips. The MSR-Action3D contains 20 classes and 10 subjects, each performing a single action 2 or 3 times. Both datasets provide 20 skeletal joint locations for each action frame. In the case of UT-Kinect not all action frames contain skeletal joint positions which makes the calculation of the pose derivatives tricky since the pose function may not be considered continuous in this case. Still, the method performs well even in this case as we will see.

We followed a common evaluation method among relevant works for the UT-Kinect dataset, which is the leave-one-subject-out-cross-validation (LOOCV). For the MSR-Action3D dataset there are two popular evaluation protocols. The first approach is the 5-fold split where half the subjects are used for training and

the other half is kept for testing. The second approach is the average result among all possible half-splits keeping a different set out for testing in each iteration (cross-validation). It is considered the most stable approach since the results are being averaged across many splits. We followed the second evaluation protocol for the MSR-Action3D.

The modified-kNN classifier that the moving pose authors originally proposed [31] was not evaluated on the UT-Kinect and also on the MSR-Action3D under the second and most reliable protocol, thus, we perform the above experiments ourselves in order to compare it with our approach. The modified-kNN classifier that was proposed works by accumulating votes at every time step using at most k neighbors, but with the modification that a single vote of a training sample is weighted by a "goodness" probability assigning method. A training sample's goodness is derived by treating it as an unknown instance and finding what percentage of its k-neighbors belong to its class. The authors propose this method as a measure to weaken the votes of irrelevant samples or outliers in the training set.

Table 1. Experiments with Moving Pose and modified-kNN.

K neighbors	UT-Kinect accuracy (%)	MSR-Action3D accuracy (%)
5	90.47	92.18
7	**90.47**	**92.51**
10	90.44	91.63
15	87.94	91.47

Table 2. Experiments with Moving Pose and Fisher encoding.

GMM gaussians	UT-Kinect accuracy (%)	MSR-Action3D accuracy (%)
4	**94**	-
8	91.47	-
12	90.47	-
16	-	90.87
24	-	**92.38**
32	-	91.27

For the modified-kNN we experimented with 4 different values for the K number of neighbors, as Table 1 shows. Optimal performance is obtained when setting $K = 7$ for both datasets. This method reaches 90.47% and 92.51% mean classification accuracy on the UT-Kinect and MSR-Action3D respectively. Note that we rejected the derivative weighting parameters as we mentioned earlier. This simply means that we set them both to 1 so that they have no effect on the low-level descriptors, thus, no other parameters were needed to be optimized here. During inference time the modified-kNN approach ran on $50 - 100$ fps depending on the action clip.

For our proposed framework, we experimented with the number of Gaussians for the GMM moving pose vocabulary that is created during training time. it is expected that the optimal vocabulary size is different for each dataset and that it should be generally larger for the MSR-Action 3D since it contains a lot more action classes. Table 2 shows our experiments with different GMM vocabulary sizes for each dataset. We manage to achieve 3.53% increase in classification accuracy with as many as 4 Gaussians on our mixture model on the UT-Kinect dataset compared to the best modified-kNN score. A decrease in performance is observed as the vocabulary size gets larger, however the dataset contains a small set of actions and as such, redundancy in the vocabulary may be prominent in these settings. Similarly, the sweet spot for the MSR-Action3D is around 24 Gaussians. We managed to reach the accuracy of the modified-kNN in this case but not surpass it. Our proposed method ran on average on $100 - 200$ fps during inference time, doubling the efficiency when compared with the modified-kNN. All the experiments were performed using an Intel i7-3770K @ 3.50GHz CPU.

Table 3. Comparison with SoA on UT-Kinect dataset.

Method	Accuracy (%)
Grassmann Manifold [21]	88.5
Histogram of 3D joints [30]	90.9
Riemannian Manifold [3]	91.5
ST-LSTM + Trust Gate [14]	97.0
Lie Group [17]	98.5
Moving Pose + knn	90.47
Moving Pose + Fisher	94.0

Table 4. Comparison with SoA on MSR-Action3D dataset.

Method	Accuracy (%)
Grassmann Manifold [21]	91.21
Feature Learning [15]	90.36
Lie Group [17]	94.27
ST-LSTM + Trust Gate [14]	94.80
Moving Pose + knn	92.51
Moving Pose + Fisher	92.38

Tables 3 and 4 show the comparison of our proposed method with State-of-the-art related works. We can see that our method achieves very good classification accuracy on both datasets when compared to elaborate techniques, like the ones that use manifolds [21, 3], as well as the more simple techniques [30]. Moreover in the case of MSR-Action3D we managed to surpass the approach of [15] which deploys a VLAD encoding scheme. Our method is only topped by a maximum of 4.5% in the UT-Kinect and 2.29% in the MSR-Action3D by the powerful deep learning methods.

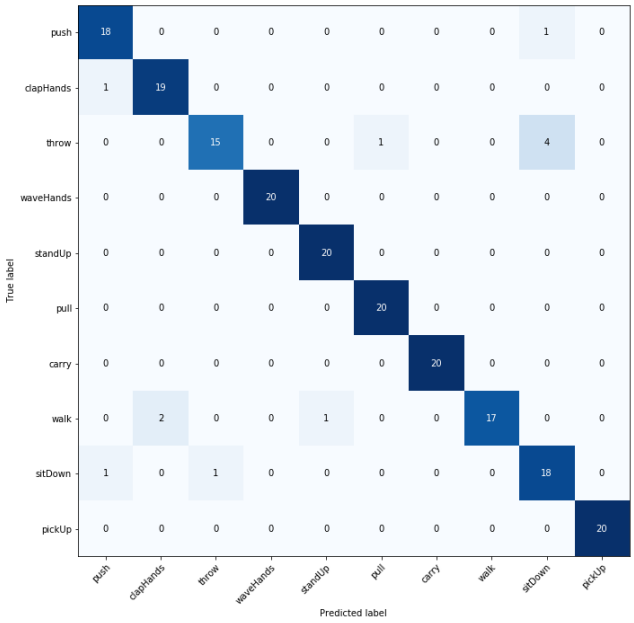

Fig. 1. Confusion matrix for action recognition on the UT-Kinect dataset.

Table 5. Class labels of the MSR-Action3D dataset.

Label	Action name	Label	Action name	Label	Action name
a01	High arm wave	a08	Draw tick	a15	Side-kick
a02	Horizontal arm wave	a09	Draw circle	a16	Jogging
a03	Hammer	a10	Hand clap	a17	Tennis swing
a04	Hand catch	a11	Two-hand wave	a18	Tennis serve
a05	Forward punch	a12	Side-boxing	a19	Golf swing
a06	High throw	a13	Bend	a20	Pick-up and throw
a06	Draw cross	a14	Forward kick		

In Figures 1 and 2 the confusion matrices are shown for action recognition on the UT-Kinect and MSR-Action3D respectively (Table 5 shows the action names for each label). Regarding the UT-Kinect, 5 classes are recognized with a perfect score and the diagonal elements show that True Positive instances never fall bellow 15/20 for any other class. The "throw" action is the most frequently confused class. Due to the object interaction in this "throw" class, occlusion may pose a challenge to the skeleton tracker and cause noise in the data. On the MSR-Action3D dataset the "Hand Catch" (a04) action is the most confused class due to fast arm movement and human-object interaction. The second most confused class is the "High Throw" with 19/30 instances correctly predicted. Nevertheless, for the rest of the classes the True Positive instances do not fall

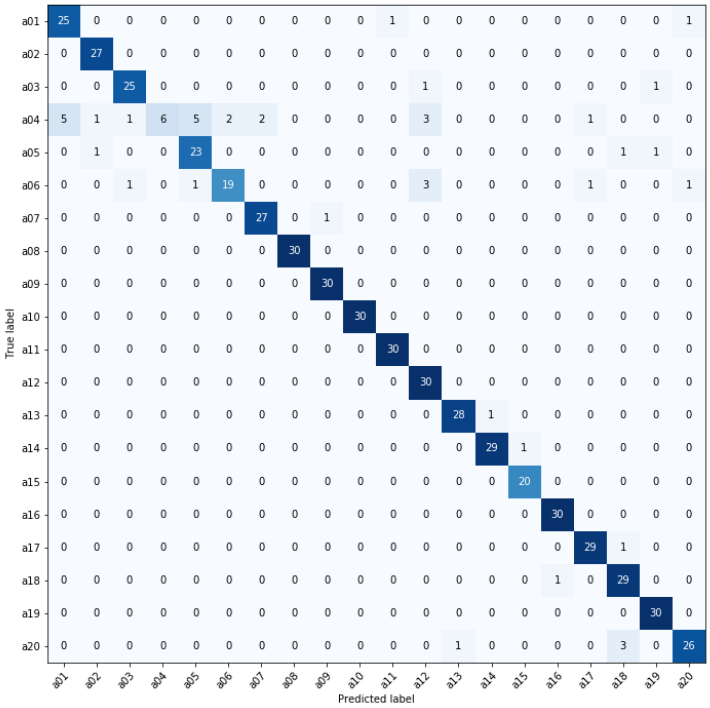

Fig. 2. Confusion matrix for action recognition on the MSR-Action3D dataset.

bellow 25/30 which shows very good performance of the proposed method in a large scale dataset with a higher number of classes and complex actions.

5 Conclusions

We have presented a descriptor aggregation and encoding scheme to pair with the Moving Pose descriptor for action recognition using skeleton data. Evaluation results indicate that our proposed framework can reach close to State-of-the-art performance without the use of deep learning and surpass the modified-kNN method whilst achieving higher processing speed rates. Our future plans are to explore other aggregation techniques and incorporate features regarding human-object interaction into the framework.

Acknowledgments This research has been cofinanced by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE - INNOVATE (T1EDK-00686) and the EC funded project V4Design (H2020-779962).

References

1. Avgerinakis, K., Briassouli, A., Kompatsiaris, Y.: Activity detection using sequential statistical boundary detection (ssbd). Computer Vision and Image Understanding **144**, 46–61 (2016)
2. Chéron, G., Laptev, I., Schmid, C.: P-cnn: Pose-based cnn features for action recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 3218–3226 (2015)
3. Devanne, M., Wannous, H., Berretti, S., Pala, P., Daoudi, M., Del Bimbo, A.: 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. IEEE transactions on cybernetics **45**(7), 1340–1352 (2014)
4. Du, W., Wang, Y., Qiao, Y.: Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3725–3734 (2017)
5. Feichtenhofer, C., Pinz, A., Wildes, R.: Spatiotemporal residual networks for video action recognition. In: Advances in neural information processing systems. pp. 3468–3476 (2016)
6. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1933–1941 (2016)
7. Gaidon, A., Harchaoui, Z., Schmid, C.: Actom sequence models for efficient action detection. In: CVPR 2011. pp. 3201–3208. IEEE (2011)
8. Jain, M., Jegou, H., Bouthemy, P.: Better exploiting motion for better action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2555–2562 (2013)
9. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J.: Towards understanding action recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 3192–3199 (2013)
10. Lee, I., Kim, D., Kang, S., Lee, S.: Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1012–1020 (2017)
11. Li, C., Zhong, Q., Xie, D., Pu, S.: Skeleton-based action recognition with convolutional neural networks. In: 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). pp. 597–600. IEEE (2017)
12. Li, C., Cui, Z., Zheng, W., Xu, C., Yang, J.: Spatio-temporal graph convolution for skeleton based action recognition. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
13. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops. pp. 9–14. IEEE (2010)
14. Liu, J., Shahroudy, A., Xu, D., Kot, A.C., Wang, G.: Skeleton-based action recognition using spatio-temporal lstm network with trust gates. IEEE transactions on pattern analysis and machine intelligence **40**(12), 3007–3021 (2017)
15. Luvizon, D.C., Tabia, H., Picard, D.: Learning features combination for human action recognition from skeleton sequences. Pattern Recognition Letters **99**, 13–20 (2017)
16. Pirsiavash, H., Ramanan, D.: Parsing videos of actions with segmental grammars. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 612–619 (2014)

17. Rhif, M., Wannous, H., Farah, I.R.: Action recognition from 3d skeleton sequences using deep networks on lie group features. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 3427–3432. IEEE (2018)
18. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: Theory and practice. International journal of computer vision **105**(3), 222–245 (2013)
19. Si, C., Chen, W., Wang, W., Wang, L., Tan, T.: An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1227–1236 (2019)
20. Si, C., Jing, Y., Wang, W., Wang, L., Tan, T.: Skeleton-based action recognition with spatial reasoning and temporal stack learning. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 103–118 (2018)
21. Slama, R., Wannous, H., Daoudi, M., Srivastava, A.: Accurate 3d action recognition using learning on the grassmann manifold. Pattern Recognition **48**(2), 556–567 (2015)
22. Sun, S., Kuang, Z., Sheng, L., Ouyang, W., Zhang, W.: Optical flow guided feature: a fast and robust motion representation for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1390–1399 (2018)
23. Tran, A., Cheong, L.F.: Two-stream flow-guided convolutional attention networks for action recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3110–3119 (2017)
24. Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3d skeletons as points in a lie group. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 588–595 (2014)
25. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. International journal of computer vision **103**(1), 60–79 (2013)
26. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision. pp. 3551–3558 (2013)
27. Wang, H., Wang, L.: Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 499–508 (2017)
28. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1290–1297. IEEE (2012)
29. Wang, L., Xiong, Y., Wang, Z., Qiao, Y.: Towards good practices for very deep two-stream convnets. arXiv preprint arXiv:1507.02159 (2015)
30. Xia, L., Chen, C.C., Aggarwal, J.K.: View invariant human action recognition using histograms of 3d joints. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. pp. 20–27. IEEE (2012)
31. Zanfir, M., Leordeanu, M., Sminchisescu, C.: The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2752–2759 (2013)
32. Zhu, Y., Chen, W., Guo, G.: Fusing spatiotemporal features and joints for 3d action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 486–491 (2013)