2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Learning Group Activities from Skeletons without Individual Action Labels

Fabio Zappardino, Tiberio Uricchio, Lorenzo Seidenari, Alberto del Bimbo
University of Florence, Italy
{name.surname}@unifi.it

***Abstract*—To understand human behavior we must not just recognize individual actions but model possibly complex group activity and interactions. Hierarchical models obtain the best results in group activity recognition but require fine grained individual action annotations at the actor level. In this paper we show that using only skeletal data we can train a state-of-the art end-to-end system using only group activity labels at the sequence level. Our experiments show that models trained without individual action supervision perform poorly. On the other hand we show that pseudo-labels can be computed from any pre-trained feature extractor with comparable final performance. Finally our carefully designed lean pose only architecture shows highly competitive results versus more complex multimodal approaches even in the self-supervised variant.**

I. Introduction

Human behavior understanding can hardly be imagined as a task which requires to explain each individual actions in isolation. Human behavior is for the most part induced by social interactions. For a machine to understand the meaning of multiple humans interacting with each other, so called social behavior or group activity, multiple level of reasoning must be enacted. A naive approach could feed the whole frame to a deep convolutional neural network, but we know, that especially when domain shift is present that a large amount of data is required to learn a good hierarchy of features automatically. Recently hierarchical approaches have emerged. With such methodologies a model of the group behavior is built in a bottom-up fashion starting from the detection and tracking of all actors, following with the understanding of their individual behavior and finally building collective models of the whole action.

As we know a fully supervised system is usually the best bet to obtain high accuracy. Unfortunately such systems must rely on a lot of hand labelling: each person action must be annotated at a not to low frequency, allowing a tracker to propagate such label over time. Considering this issue only few fully annotated datasets are available, limiting the development of group understanding computer vision algorithms. To address this issue in this work we propose a novel approach for group activity recognition based on the concept of pseudo-labels. Loosely inspired by the work of Caron *et al.* [1] we propose to replace costly single action labels with pseudo labels obtained via a simple clustering procedure which can be derived at very little cost. Interestingly, we show that such process is enough to provide such mid-level supervision thus enabling group activity recognition. Ground truth labels can therefore limited to the whole activity sequence with order of magnitude of time spared in the annotation phase.

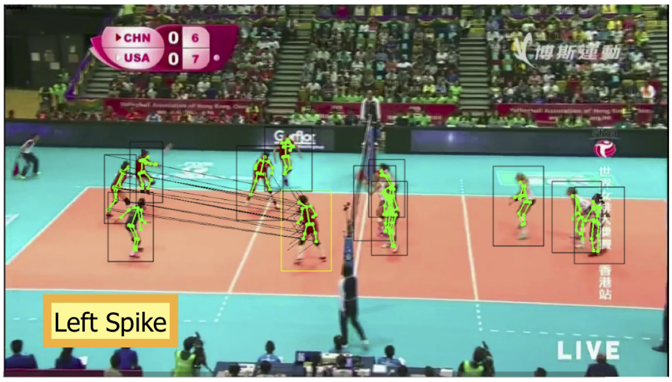

Fig. 1: We classify group activities using skeletons, motion of single actors and their relative positions to a pivot actor.

Recently, the issue of privacy in A.I. appliance has been raised especially in the EU, which enforces extremely strict policies regarding acquisition and protection of user persona information and data. Deploying cameras in public or private places to monitor user behavior has the major drawback of requiring the acquisition and possibly the storage and streaming of people images. While reliance on cryptography may offer a solution using highly anonymized human representation such as the skeleton offers many benefits. Human poses can be acquired in real-time with edge computing devices exploiting cloud computing facilities for the more complex task of action recognition. Moreover, dataset acquisition is made easier, not requiring the abidance to privacy policies if only the substantially anonymous, 3D skeletal data is stored.

In this work we propose to relevant contribution to the field of activity recognition:

- We propose a novel semi-supervised approach allowing to train group activity recognition methods without fine grained ground truth annotation.
- We show how group activity can be efficiently performed using only skeletal representations which have a lower computational burden and have interesting privacy preserving properties.

II. Related Works

The initial methods for group activity recognition used handcrafted features which were extracted for each actor and

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

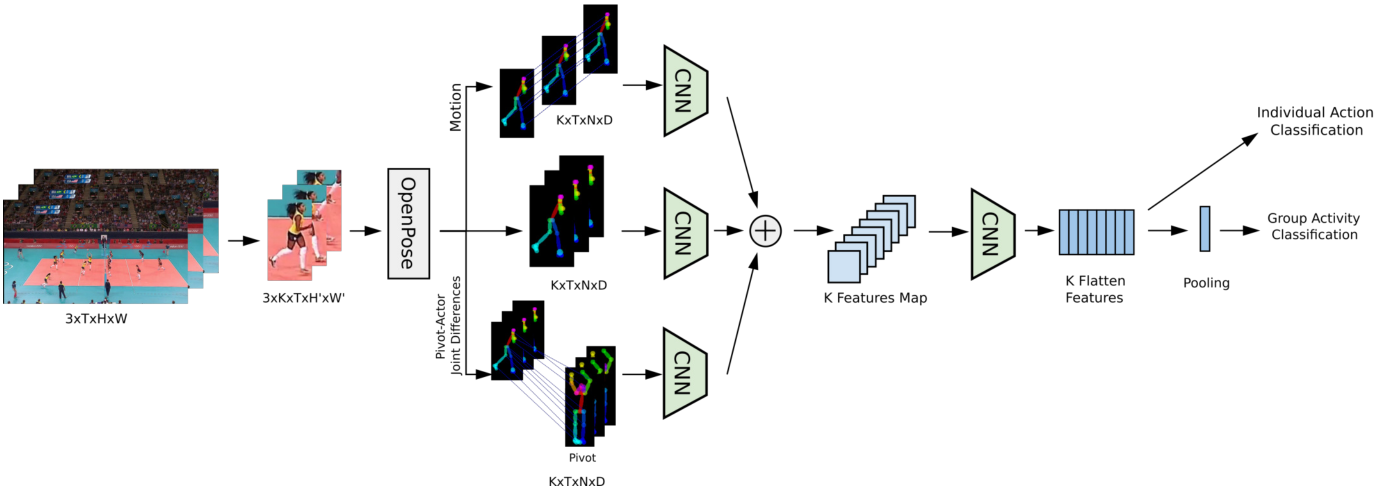

Fig. 2: The proposed architecture employs three structurally identical branches with independent weights. We use openpose estimated skeletons to create three different representations. A concatenation of actor features is fed to a rather shallow network with two convolutional layers. Finally, single actor features are fed to the individual action classification head and single actor features are pooled to compute a group representation which is fed to the group classification head.

combined using probabilistic graphical models [2], [3], [4], [5], [6], [7], [8]. Then, after the emergence of deep learning and the release of the Volleyball [9] dataset, there was a renewed interest which remarkably increased performance. Most approaches use RNN-type networks which were introduced by Ibrahim *et al.* [9] for the task of group activity recognition. They are trained to understand the action dynamics of individual actors and then predict group activity by taking their aggregation. A combination of RNN with graphical models were proposed in [10]. A two-level hierarchy of LSTMs were used in [11] to simultaneously minimize the loss of predictions and maximize the confidence. Bagautdinov *et al.* [12] predicted every action of each actor and the group activity with an RNN which is also used to maintain temporal consistency of detected boxes. Intra-group, inter-group interactions and single person dynamics were considered in [13] and modeled with an LSTM. Each person is also modeled in a relational framework in [14]. Other works considered different approaches, Azar *et al.* [15] exploited activity maps of a CNN and iteratively refined group activity predictions. In [16], they capture the appearance and positions between actors to build a graph of actor relationships from a CNN and graph convolutional networks. Gavrilyuk *et al.* [17] explicitly modeled spatial and temporal relationships of actors with an actor-transformer model that learns and extract relevant information for group activity recognition. Our approach use CNN to perform classification. We share the use of skeletal data with [17], however our approach do not use additional 2D and 3D information to preserve privacy. We also consider the relative position of actors combined with motion and pose to perform individual action classification and group activity recognition.

The use of human poses for recognizing actions of an actor is a popular approach in the literature. The early approaches were using handcrafted pose features [18], [19], then skeletons [20], [21] and attention based pose estimations [22], [23] were all explored for the task of action recognition of single actors. We exploit skeletons to model actions of single actors and then combine them into an holistic representation of the entire scene.

III. Method

In the following we show our approach for partially self-supervised group activity recognition using skeletal data.

A. *Input Skeleton Representation*

A group activity model has to consider individual actors representation its temporal evolution and contextual information. Similarly to [9], the person bounding boxes are firstly obtained through the object tracker in the Dlib library [24]. Then we feed each person track frames as input to openpose [25], obtaining a group of skeleton sequences **GS**. **GS** can be represented with a $K \times T \times N \times D$ tensor, where K is the number of actors in the video clip, T is the number of frames in the sequence, N is the number of joints in the skeleton and D is the coordinate dimension. Given a group of skeletons at time t, $\mathbf{GS}^t = \{\mathbf{S}_1^t, \mathbf{S}_2^t, \ldots, \mathbf{S}_K^t\}$, we represent the skeleton of a person k at time t as

$$\mathbf{S}^{tk} = [J_1^{tk}, J_2^{tk}, \ldots, J_N^{tk}],$$

where $J = [x, y, p]$ is a 2D joint coordinate + precision given by the pose estimator.

Since skeleton coordinates depend on actor camera distance, bounding box dimension and height we normalize each skeleton sequence by subtracting the mid-hip keypoint from each skeleton joint in order to have this last joint as the center of the coordinates system, then we divide each limb by the torso length.

Similarly to [26], we introduce a representation of skeleton motion. The skeleton motion of a person k at time t is defined as the temporal difference of each joint between two consecutive frames:

$$\begin{aligned}&\mathbf{M}^{tk} = \mathbf{S}^{(t+1)k} - \mathbf{S}^{tk} = \\ &[J_1^{(t+1)k} - J_1^{tk}, J_2^{(t+1)k} - J_2^{tk}, \ldots, J_N^{(t+1)k} - J_N^{tk}]\end{aligned} \tag{1}$$

To model the configuration of each actor k we compute a person-to-person interaction $\mathbf{D}^{tk}$ (see Fig.3), defined as the difference between each pair of joints of two persons at time t. In order to obtain a group representation invariant to camera motion and do not rely on global fram coordinates, we select a pivot actor. We used the pivot as a reference to compute the difference between joint pairs.

We then represent each skeleton sequence via pivot-actor Joint differences, computed between an actor k and the actor pivot p at time t is formulated as:

$$\mathbf{D}^{tk} = \mathbf{S}^{tk} - \mathbf{S}^{tp} = [J_1^{tk} - J_1^{tp}, J_2^{tk} - J_2^{tp}, \ldots, J_N^{tk} - J_N^{tp}] \tag{2}$$

Motion $\mathbf{M}^k$ and pivot-actor joint differences $\mathbf{D}^k$ from all actors are stacked obtaining two tensor $\mathbf{GM}$ and $\mathbf{GD}$ having the same shape of the group skeleton sequence $\mathbf{GS}$.

Group sequence of skeletons, motion and pivot-actor joint differences are fed into the network directly as three input streams into three separate network branches sharing the same architecture. However their parameters are not shared and learned separately. Following [26] feature maps from inputs are learned hierarchically with specific convolution layers, then they are flattened and fused by concatenation obtaining a $F \times K$ matrix to represent feature vectors of actors. Feature vectors are used both for action classifications and max-pooled for group activity classification. The whole model can be trained in an end-to-end manner with back-propagation and the extremely small model size allows us to easily train the network from scratch without the need of pretraining. Combining the two standard cross-entropy losses, the final loss function is formed as

$$L_{tot} = L_G(y^G, \hat{y}^G) + \lambda L_I(y^I, \hat{y}^I) \tag{3}$$

where L_I and L_G are the cross-entropy loss, y^G and y^I denote the ground-truth labels of group activity and individual action, $\hat{y}^G$ and $\hat{y}^I$ are the predictions to group activity and individual action. The first term corresponds to group activity classification loss, and the second is the loss of the individual action classification. The weight λ is used to balance these two tasks.

B. Self-Supervised Group Activity Recognition

We use P3D a pretrained 3D-CNN [27] initialized on Kinetics sport dataset [28]. We use the last fully connected layer output as features from video clips of each actor. We cluster features from the training set using k-means and use the cluster assignments as ”pseudo-labels” k^I to compute individual action loss term $L_I(k^I, \hat{y}^I)$. This allows to train our model in a self-supervised way. Before the clustering, features are PCA-reduced from 2048 to 256 dimensions, whitened and l2-normalized.

IV. Experiments

In this section, we evaluate the performance of the proposed method in both supervised and self-supervised variants with several baselines. We also compare our results with the state-of-the-art.

A. Datasets

We evaluate our method on the Volleyball dataset[9], since it is the only public available dataset for group activity recognition that is relatively large-scale and contains labels for people locations, as well as their collective and individual actions. This dataset contains 4830 clips of 55 volleyball games. Each clip central frame is annotated at each player level with the bounding box and one of the 9 individual actions, and the whole scene is labeled with one of the 8 collective activity. Since other frames are not annotated, to get the bounding boxes of people, we used DLIB tracker[24]. We do horizontal flips as data augmentation.

B. Implementation Details

We adopt stochastic gradient descent with ADAM to learn the network parameters with fixed hyper-parameters to $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$. We train the network for 100 epochs using a mini-batch size of 64 and a starting learning rate of 0.001 decreasing it by a factor of 10 every 30 epochs. Individual action loss weight $\lambda = 0.7$ is used. For training all our models (that include the baseline models) we follow the same training protocol using a Tesla K80 GPU and PyTorch Framework.

C. Baseline and Variants for Ablation Studies

Here we evaluate our model by comparing obtained results with several baselines. First, we describe the baseline model then, results on the volleyball dataset are presented. Our model is end-to-end trainable but could also be implemented in a 2-stage style, splitting the model into an action model and a group model. Training then would consists in learning actions from each skeleton sequence and then use this model to extract actor features, which are pooled over all people and fed to the group model to recognize group activity. We test the following approaches:

1 **Two Stage Model without Pivot-Actor Joint Differences**: This baseline is the two stage model with two input streams: skeleton joints and motion.

2 **Two Stage Model**: This baseline is an extension of the previous baseline. Here individual action model receive as input a third stream: the pivot-actor joint differences.

3 **End-to-end Model without Pivot-Actor Joint Differences**: this baseline is the end-to-end version of first baseline.

4 **End-to-end Model**: the end-to-end final version with its three input stream.

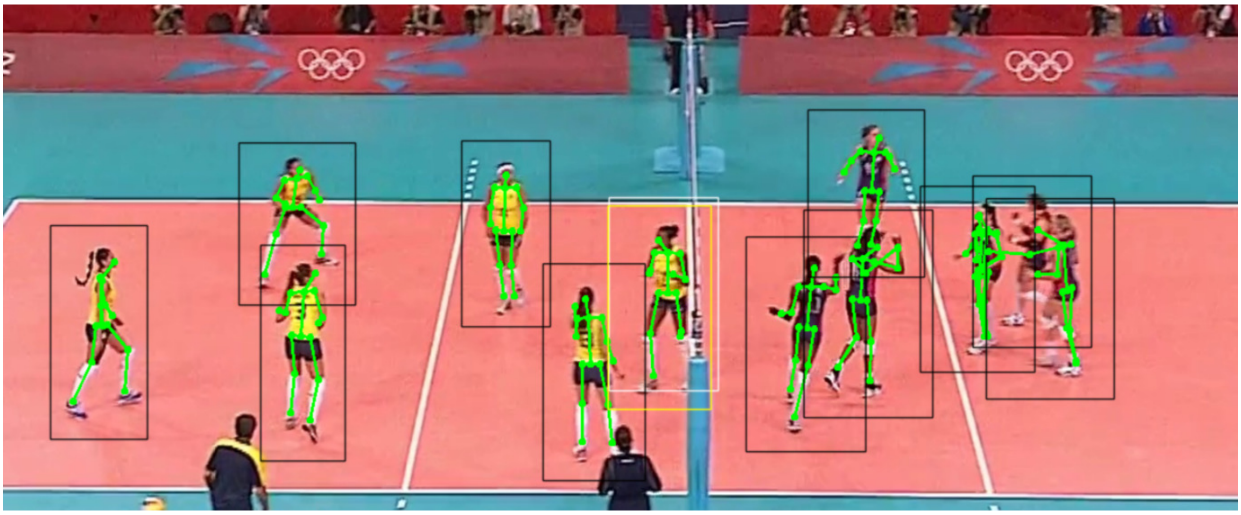

Fig. 3: Pivot selection among players. We pick the players closest to the average joint centroid of all players. Pivot is shown in yellow.

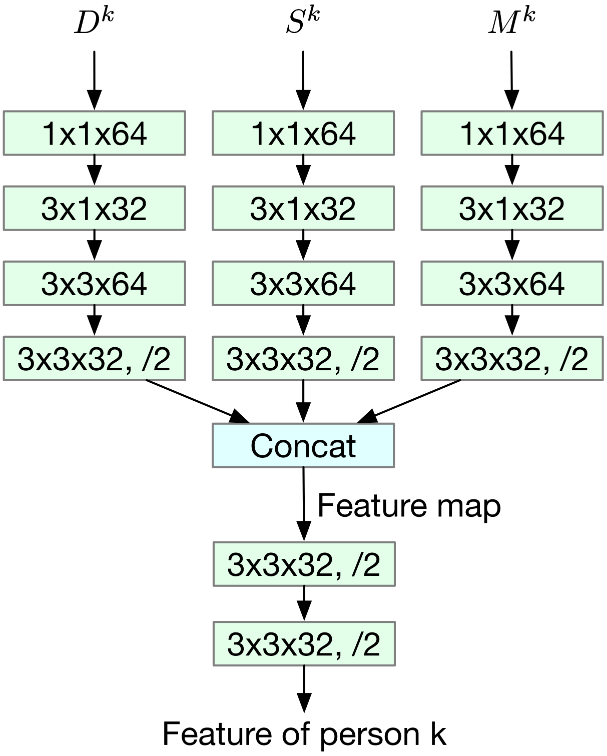

Fig. 4: Architecture of the CNN. Each green block corresponds to a convolutional layer of size WxHxFilters. The layers with $/2$ have a stride of 2. The entire figure corresponds to the CNN blocks of Fig. 2

5 **End-to-end Model with Data Augmentation**: the end-to-end final version with its three input stream and data augmentation.

Method	Accuracy
Two stage model w/o actor-pivot joint pair difference	78.1
Two stage model	80.9
End-to-end model w/o actor-pivot joint pair difference	85.0
End-to-end model	89.2
End-to-end model with data augmentation	91.0

TABLE I: Comparison of our method with baseline methods on the volleyball dataset.

In Table I, the classification results of our method is compared with baselines. As shows in the table, an increase of performance is generated by Pivot-Actor Joint Differences as it is including a person to person context information that helps our model in both the training procedure. End-to-end training helps significantly the model. Also data augmentation is useful for training the model as it provide consistent data for network's learning. Therefore we choose to use both Pivot-Actor Joint Differences and data augmentation with end-to-end training as our final model.

D. *Self-Supervised Ablation Studies*

As we discussed in previous subsection, the use of Pivot-Actor Joint Differences with end-to-end training is able to achieve the highest performance and we choose this combination as our final model, considering the two variants with and without data augmentation. In order to understand how much performance is affected by the use of our pseudo-labels we conduct the following ablation studies.

1 **Group activity labels only**: In this baseline the model has been modified in order to work only with group activity labels. The layers that receive actor feature representation to classify the individual action are removed and the individual loss factor λ is set to zero. This baseline is designed to illustrate the importance of individual activity labels in our model.

2 **Pseudo action labels from 2D-CNN**: In this baseline we adopted pseudo-labels instead of ground truth action

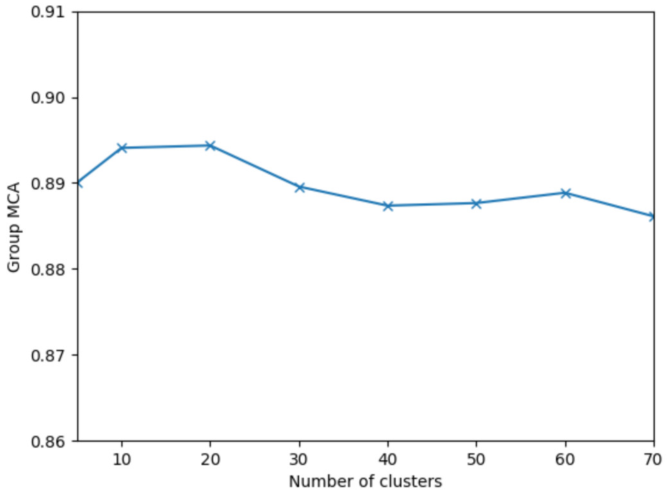

Fig. 5: Mean Classification Accuracy of group activity varying the number of clusters for self-supervised learning.

labels. Visual features are extracted from the central frame of each video clip using a pretrained VGG16 2D-CNN. Number of used cluster k is set to 20. This baseline aims to illustrate the importance of pseudo-action-labels.

3 **Pseudo action labels from 3D-CNN**: Final Self-Supervised model, where we adopt pseudo-labels instead of ground truth action labels. Visual features are extracted from the whole fixed temporal window of each video clip using a pretrained 3D-CNN. Number of used cluster k is set to 20.

4 **End-to-end fully supervised**: this method is the end-to-end final version, previously seen in Table I, trained with the full ground truth.

The results of first and second baselines illustrate the importance respectively of group activity labels and pseudo-action-labels in our model. Comparing with second baseline, our self-supervised method considering the time with a 3D-CNN model obtains better performance. Moreover, our self-supervised method results are very close to supervised variant, especially in case of data augmentation.

Fig.5 shows group accuracy results obtained by varying the number of clusters by a step of 10, note that best results are obtained with k=20. Given that we train our model on the volleyball dataset, one would expect k=9 (actual number of action classes) to yield the best results, but apparently some amount of over-segmentation is beneficial.

E. Comparison with state-of-the-art methods

There are only a few literature [29], [30] referring to use of pose in volleyball dataset, thus we compare our method that exploits only poses with other state of-the-art methods that make use of different input feature like RGB and/or optical flow. As shown in Table 4.5, our method is not so far with the state-of-the-art methods and even outperforms most of the methods that exploit RGB and optical flow input (including PC-TDM [31] and SPA+KD+OF [32]). Although ARG [16], CRM [33], PRL [34] and AT [30] performs somewhat better than our method , note that it is unfair to compare with, because they use image, optical flow input and also a much larger model than ours. In addition, even our self-supervised action method outperformed various results without using ground truth individual action labels, showing that it is possible to obtain good results using action labels generated with visual feature clustering. The confusion matrix of the proposed supervised method is shown in Table 4.6. As we can see, our method achieves promising recognition accuracies (90%) on most of the activities. The main failure cases are from “set” and ”spike” within the left and right subgroups, which is probably due to the very fast actions of the key participants that cause joint extraction very difficult. For the same reason our best cases are in “winpoint” within the left and right sub groups as lackness of rapid movements guarantees a better joint extraction.

Method	No Data Aug.	With Data Aug.
Group activity labels only	84.9	87.1
Pseudo action labels from 2D-Vgg16	87.2	89.3
Pseudo action labels from 3D-Resnet	87.5	89.5
Supervised	89.2	91.0

TABLE II: Comparison of our method with and without action labels. SSAL stands for Self Supervised Action Learning corresponding to centroid indexes assigned by clustering of visual features.

Method	Input	Action Labels	Accuracy
HDTM [9]	RGB	Yes	81.9
SSU [12]	RGB	Yes	89.9
PC-TDM [31]	RGB+OF	Yes	87.7
MS-CNN [29]	RGB+POSE	Yes	90.5
stagNet [35]	RGB	Yes	89.3
RCRG [14]	RGB	Yes	89.5
SPA+KD [32]	RGB	Yes	89.3
SPA+KD+OF [32]	RGB+OF	Yes	90.7
ARG [16]	RGB	Yes	92.6
CRM [33]	RGB+OF	Yes	93.0
PRL [34]	RGB	Yes	91.4
AT [30]	POSE+OF	Yes	94.4
Ours-SSAL	POSE	No	89.4
Ours	POSE	Yes	91.0

TABLE III: Comparison of recognition accuracy (%) on Volleyball dataset. ”OF” denotes optical flow input, while column ”Action Labels” says that a method use ground truth annotations at individual actor level.

V. Conclusion

In this work we have shown a solution to train group activity recognition systems end-to-end without the use of individual action labels. Our method using only skeletal data is able to reach state-of-the art performance without using RGB or Optical flow and requiring only labels at the sequence level. Compared to previous work, the proposed approach needs less supervision to be trained and using only skeleton data, can also be used in contexts where privacy require to avoid saving RGB images.

References

[1] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 132–149.

[2] M. R. Amer, P. Lei, and S. Todorovic, "Hirf: Hierarchical random field for collective activity recognition in videos," in *European Conference on Computer Vision*. Springer, 2014, pp. 572–585.

[3] W. Choi and S. Savarese, "A unified framework for multi-target tracking and collective activity recognition," in *European Conference on Computer Vision*. Springer, 2012, pp. 215–230.

[4] ——, "Understanding collective activities of people from videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 6, pp. 1242–1257, 2013.

[5] W. Choi, K. Shahid, and S. Savarese, "Learning context for collective activity recognition," in *CVPR 2011*. IEEE, 2011, pp. 3273–3280.

[6] H. Hajimirsadeghi, W. Yan, A. Vahdat, and G. Mori, "Visual recognition by counting instances: A multi-instance cardinality potential kernel," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2596–2605.

[7] T. Lan, L. Sigal, and G. Mori, "Social roles in hierarchical models for human activity recognition," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1354–1361.

[8] T. Lan, Y. Wang, W. Yang, S. N. Robinovitch, and G. Mori, "Discriminative latent models for recognizing contextual group activities," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 8, pp. 1549–1562, 2011.

[9] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1971–1980.

[10] Z. Deng, A. Vahdat, H. Hu, and G. Mori, "Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4772–4781.

[11] T. Shu, S. Todorovic, and S.-C. Zhu, "Cern: confidence-energy recurrent network for group activity recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5523–5531.

[12] T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, and S. Savarese, "Social scene understanding: End-to-end multi-person action localization and collective activity recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4315–4324.

[13] M. Wang, B. Ni, and X. Yang, "Recurrent modeling of interaction context for collective activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3048–3056.

[14] M. S. Ibrahim and G. Mori, "Hierarchical relational networks for group activity recognition and retrieval," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 721–736.

[15] S. M. Azar, M. G. Atigh, A. Nickabadi, and A. Alahi, "Convolutional relational machine for group activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7892–7901.

[16] J. Wu, L. Wang, L. Wang, J. Guo, and G. Wu, "Learning actor relation graphs for group activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9964–9974.

[17] K. Gavrilyuk, R. Sanford, M. Javan, and C. G. Snoek, "Actor-transformers for group activity recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 839–848.

[18] B. Xiaohan Nie, C. Xiong, and S.-C. Zhu, "Joint action recognition and pose estimation from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1293–1301.

[19] C. Wang, Y. Wang, and A. L. Yuille, "An approach to pose-based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 915–922.

[20] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[21] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," in *European conference on computer vision*. Springer, 2016, pp. 816–833.

[22] C. Cao, Y. Zhang, C. Zhang, and H. Lu, "Action recognition with joints-pooled 3d deep convolutional descriptors." in *IJCAI*, vol. 1, 2016, p. 3.

[23] G. Chéron, I. Laptev, and C. Schmid, "P-cnn: Pose-based cnn features for action recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3218–3226.

[24] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1755–1758, 2009.

[25] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299.

[26] C. Li, Q. Zhong, D. Xie, and S. Pu, "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," *arXiv preprint arXiv:1804.06055*, 2018.

[27] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5533–5541.

[28] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[29] S. M. Azar, M. G. Atigh, and A. Nickabadi, "A multi-stream convolutional neural network framework for group activity recognition," *arXiv preprint arXiv:1812.10328*, 2018.

[30] K. Gavrilyuk, R. Sanford, M. Javan, and C. G. Snoek, "Actor-transformers for group activity recognition," *arXiv preprint arXiv:2003.12737*, 2020.

[31] R. Yan, J. Tang, X. Shu, Z. Li, and Q. Tian, "Participation-contributed temporal dynamic model for group activity recognition," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1292–1300.

[32] Y. Tang, J. Lu, Z. Wang, M. Yang, and J. Zhou, "Learning semantics-preserving attention and contextual interaction for group activity recognition," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 4997–5012, 2019.

[33] S. Mokhtarzadeh Azar, M. Ghadimi Atigh, A. Nickabadi, and A. Alahi, "Convolutional relational machine for group activity recognition," *arXiv preprint arXiv:1904.03308*, 2019.

[34] G. Hu, B. Cui, Y. He, and S. Yu, "Progressive relation learning for group activity recognition," *arXiv preprint arXiv:1908.02948*, 2019.

[35] M. Qi, J. Qin, A. Li, Y. Wang, J. Luo, and L. Van Gool, "stagnet: An attentive semantic rnn for group activity recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 101–117.