Group Activity Recognition via Computing Human Pose Motion History and Collective Map from Video

Hsing-Yu Chen and Shang-Hong Lai(✉)

Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
andy19933@gapp.nthu.edu.tw, lai@cs.nthu.edu.tw

Abstract. In this paper, we propose a deep learning based approach that exploits multi-person pose estimation from an image sequence to predict individual actions as well as the collective activity for a group scene. We first apply multi-person pose estimation to extract pose information from the image sequence. Then we propose a novel representation called pose motion history (PMH), that aggregates spatio-temporal dynamics of multi-person human joints in the whole scene into a single stack of feature maps. Then, individual pose motion history stacks (Indi-PMH) are cropped from the whole scene stack and sent into a CNN model to obtain individual action predictions. Based on these individual predictions, we construct a collective map that encodes both the positions and actions of all individuals in the group scene into a feature map stack. The final group activity prediction is determined by fusing results of two classification CNNs. One takes the whole scene pose motion history stack as input, and the other takes the collective map stack as input. We evaluate the proposed approach on a challenging Volleyball dataset, and it provides very competitive performance compared to the state-of-the-art methods.

Keywords: Activity recognition · Action recognition · Human pose estimation · Deep learning

1 Introduction

In a scene consisting of a group of people, the collective activity can be seen as integration of actions for all individuals. To recognize individual human action, human pose, which is the configuration of all the main joints, is an important cue [19]. In fact, human actions, especially sport actions, are directly related to the spatio-temporal dynamics of human body parts or joints. For instance, the process of a volleyball player performing "setting" comprises representative evolution of his or her joints, which is different from the one of another player simply standing in a same place.

Previous works on this problem are basically appearance based [1,2,10,16,28]. Given a sequence of images, these appearance based approaches first used ground-truth tracking information or human detection plus human identity association

S. Palaiahnakote et al. (Eds.): ACPR 2019, LNCS 12047, pp. 705–718, 2020.
https://doi.org/10.1007/978-3-030-41299-9_55

to localize the bounding box of each individual in the group, then used CNNs to extract visual features from the corresponding region of each individual in each frame of the sequence, and constructed the rest of their RNN-based models upon these visual features. With recent impressive achievement of bottom-up multi-person pose estimation [5,25], we believe that it is sufficient now to use 2D human pose from pose estimation as the input for DNNs to learn the dynamics of both individual actions and collective activities in a group scene.

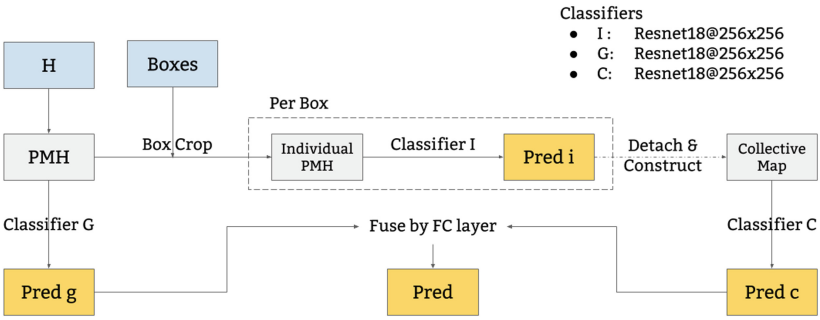

Fig. 1. Overview of the proposed system. **H** denotes joint heat maps. **Boxes** denotes bounding boxes given by annotation, or converted from pose estimation results. **Pred i** denotes the individual action prediction scores classified from Indi-PMH. **Pred g** and **Pred c** denote the collective activity prediction scores classified from PMH and collective map respectively.

In recent years, 2D pose information has been exploited in video-based action or activity recognition tasks to focus on human body parts or joints in the input sequence images. Some previous works proposed to use human joints as guidance to aggregate or attend to partial appearance or motion features from the whole RGB images or optical flow [4,6,12]. Lately, some works started to utilize joint confidence maps as input. For example, [24] proposed to apply spatial rank pooling on joint confidence maps and use body guided sampling on estimated human pose to obtain two kinds of complementary description images. The recognition task is then performed on these two descriptions. [9] proposed to use color coding to aggregate joint confidence maps from different time steps into a single stack of feature maps which is called PoTion. The activity of the whole scene is classified from PoTion.

In this paper we propose a novel approach utilizing multi-person pose information and fusion of individual actions through two novel representations, pose motion history and collective map, to recognize the action of each individual and the group activity. We use intensity retaining mechanism instead of color coding to perform temporal aggregation. The aggregated pose motion history feature map takes less memory than PoTion [9] and thus the recognition task can be done more efficiently. Also, the collective map that encodes both positions and actions of all individuals enhances the group activity recognition.

In this work, we adopt OpenPose [5] to retrieve the positions where people are in the sequence frames, and extract multi-person pose features without the need of cumbersome combination of human detection plus single-person pose estimation. Moreover, with the aid of the two novel representations that represents the spatio-temporal dynamics of multi-person human joints, and the integration of individual actions as well as individual positions in sequences, it is unnecessary to use RNN or its variants to learn the mapping from the input to individual action or group activity prediction. Instead, we use a simple CNN model, such as Resnet-18, for the classification tasks, while still achieving very competitive performance.

In summary, the contributions of this work are three-fold. Firstly, we propose two novel representations, pose motion history and collective map, for representing individual actions and the group activity of a multi-person scene. To the best of our knowledge, we are the first one to utilize multi-person pose estimation to classify both individual actions and collective activities at the same time. Secondly, we design a simple CNN model without the help of human tracking on these two novel representations for the classification task. Finally, we evaluate the proposed system on the Volleyball dataset, and achieve competitive performance even we just use simple CNNs without human identity association compared to the previous RNN-based works.

2 Related Works

Fig. 2. Some examples of applying OpenPose [5] on sequences of Volleyball dataset.

Action Recognition in Videos. Action recognition plays an essential role in various domain such as surveillance, robotics, health care, video searching, and human-computer interaction [36]. With recent revival of deep learning, given video sequences, many works successfully exploit the power of DNNs to learn spatio-temporal evolution of human actions, and report impressive results on several popular benchmarks [13,29,35]. Since deep learning based approaches outperform previous hand-crafted feature based methods, we only review deep learning based ones here. [29] used two-stream CNNs which consist of one spatial stream learning appearance features and one temporal stream learning motion features to recognize actions in video sequences. Several works proposed further enhancement based on this kind of multi-stream architectures [13,35,37]. [18] introduced an 3D CNN which extended traditional 2D CNNs to a 3D one to convolve spatio-temporal information. From then on, 3D CNNs have been used

and improved in many works [14,32,33]. RNNs are also popular for action recognition in videos since they naturally extract temporal features from sequence input [11,26,34].

2D Pose-Based Action Recognition from Video. Since the target of action recognition is mainly human, pose is a natural input cue for classifying human actions in videos. Many works have proposed to use pose information in videos to learn spatial and temporal evolution of human actions [4,6,9,12,17,24]. Some used joint positions to further aggregate or pool appearance or motion features [4,6,12]. Some directly used estimated pose or joint confidence maps as the input for their models [9,17,24]. Our method is most similar to PoTion [9] in which color coding is applied on joint confidence maps to aggregate human joints information from different sequence images into single compact stack of feature maps. We use intensity retaining mechanism instead of color coding to construct our proposed pose motion history feature map stacks which consume less memory and thus results in higher learning efficiency. Our method is different from all the works above, since all these methods only deal with single person, double people settings, or predict only the activity of the whole sequence, while we not only predict the group activity, but also the action of each individual in the input sequence.

Group Activity Recognition from Video. Group activity recognition has attracted a lot of work in past years. Many former methods used hand-crafted features as input to structured models [7,8,20–23]. While with recent revival of deep learning, more and more papers started to take advantage of the superior classification performance of Deep Neural Networks [1,2,15,16,27,28,30,31,38]. In [1,16], hierarchical models consisting of two LSTMs are used, one for representing individual action dynamics of each person in a video sequence, and the other for aggregating these individual action dynamics. [1] combined human detection module into their hierarchical model through reusing appearance features for detection and recognition. [2,15,27] combined graph structures with DNNs to model the actions of individuals, their interactions, and the group activities. [27,28,30,38] utilized attention mechanisms to focus on more relevant individuals or temporal segments in video sequences.

3 Proposed System

Our goal is to recognize the individual action of each individual from a video by utilizing their pose information, and also recognize the collective activity of the whole group based on both collective pose information and also individual predictions. To achieve this goal, we propose a two-stream framework for group activity classification based on the pose motion history and collective map.

The overview of our framework is given in Fig. 1. For a given input sequence, first we apply a multi-person pose estimation algorithm, OpenPose [5], to estimate the joint positions of each individual and also the confidence maps (heatmaps) of these joints for each frame. OpenPose [5] would produce a joint

heatmap stack of 18 channels where each channel is the heatmap corresponding to a certain joint. The value of each pixel in a heatmap indicating the probability a joint locating there. We depict some pose estimation results on the Volleyball dataset in Fig. 2. Second, to construct the whole scene pose motion history stack of each frame, we first multiply a intensity retaining weight w to the joint heatmap stack of the first sequence frame, and sum it to the second sequence frame. Then the pose motion history stack of the second sequence frame is multiplied by the same retaining weight w and summed to the joint heatmap stack of the third sequence frame. We repeat this process until we derive the pose motion history stack of the last sequence frame. This very pose motion history stack $\mathbf{P}$ is the input for the following two streams: individual stream and collective stream.

For the individual stream, we first obtain the bounding box of each individual by simply finding the minimal rectangle containing all joints of the individual based on the joint positions output by OpenPose [5]. Second, these individual bounding boxes are used to crop their corresponding individual pose motion history stacks $p_b, b = 1 \ldots B$, which are then input into our individual PMH CNN to classify individual actions. We then construct a collective map stack $\mathbf{M}$ of I channels where each channel representing one individual action, and for each individual, we fill its individual softmax scores for all actions to its bounding box area in the corresponding action channel. This collective map stack thus encodes individual positions and their actions in a simple feature map stack. See the subsequent sections for more details.

For the collective stream, we first simply input the pose motion history stack $\mathbf{P}$ into our collective PMH CNN to obtain initial collective activity predictions. Second we input the collective map stack $\mathbf{M}$ into another collective map CNN to obtain auxiliary collective activity predictions. Next these two parts of collective activity predictions are fused by a fusion FC layer, where the fused result is the final collective activity predictions.

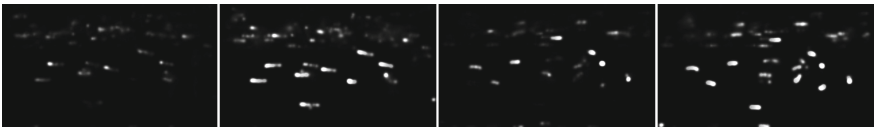

Fig. 3. Some examples of PMH maps computed from Volleyball dataset using retaining weight value $w = 0.95$.

3.1 Pose Motion History

Pose motion history(PMH) can be seen as an idea extended from MHI (Motion History Image) [3]. [3] proposed to form the temporal history of pixel points into a motion history image. In this image, more recently moving pixels are brighter. We observe that human actions are naturally highly related to the

spatial-temporal dynamics of human joints. To form pose motion history, given an input multi-person sequence of T frames, we consider human joints in the sequence as the interest points. To represent their motion history from the past to the current frame, we apply the recursive overlaying mechanism on frame 2 to frame T as given by the following equation:

$$P_t = P_{t-1} * w + H_t, \tag{1}$$

where H_t is the whole scene 18-channel joint heatmap stack of frame t generated by OpenPose [5], w is the intensity retaining weight, and P_t denotes the 18-channel pose motion history stack of frame t. In this paper, the intensity retaining weight w is a fixed chosen value in $[0, 1]$ so that joint positions in latter frames would be more obvious than those in the earlier frames; however it could also be learned through training. Effect of different w values would be discussed in Sect. ?. We clip pixel values in P_t larger than 255 to 255. See Fig. 3 for pose motion history examples. We use OpenPose [5] to retrieve whole scene joint heatmap stacks and joint positions of the input sequence. For a frame image, OpenPose [5] would generate an 18-channel stack of whole scene joint heatmaps. The value of each pixel of a joint heatmap indicating the probability or the confidence that a joint locates at that position. The joint positions output by OpenPose [5] are grouped by individuals. We use the grouped joint positions in the individual stream to crop individual pose motion history stacks (Indi-PMH).

Given the grouped joint positions output by OpenPose [5], we calculate the bounding box of each individual through finding the minimal rectangle able to contain all joints of a person. We enlarge bounding boxes of the last frame with a scale s to crop the corresponding individual pose motion history stack for each individual from P_T. We crop from P_T since it contains all the joint motion history of the whole sequence from the first to the last frame. We enlarge the bounding boxes before cropping so that the cropped individual pose motion history stacks could contain more complete joint motion dynamics. We input these cropped individual pose motion history stacks (Indi-PMH) into a Resnet-18 to classify their individual actions.

3.2 Collective Map

The collective map stack of an input sequence could be computed through summing all the collective map stacks of each of its frames, but in this paper we simply use the one of the last sequence frame instead. We illustrate the construction of a collective map of a single frame in Fig. 4. We denote the number of individuals in an input sequence frame by N. To construct the collective map stack M_{H,W,A_I} of the sequence frame we first fill the stack with all zeros, where A_I denotes the number of individual action classes, H, W the height and the width of the frame. For each individual prediction $R_n, n \in [1, N]$ of this frame, which is a vector with A_I elements, we first apply softmax on it to restrict all the values of its elements in the range $[0, 1]$, so that each element represents the probability of a individual action class. Next we fill each softmax score to

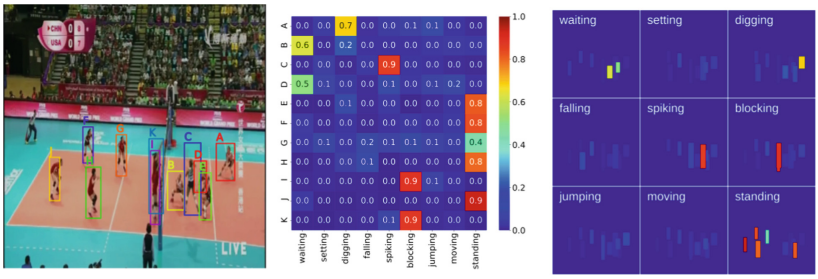

Fig. 4. An illustration for constructing the collective map of a sequence frame. In the leftmost picture, there are 11 players in the scene, denoted as **A** to **K**. First we construct a zero-filled stack. The matrix in the middle picture shows the softmax scores of individual action classes of each player. We fill each softmax score of each player to the area of his/her bounding box of the corresponding individual action channel in the collective map stack. The rightmost picture shows the result collective map stack.

the bounding box area of each individual to the corresponding individual action channel of M_{H,W,A_I}. The constructed collective map thus encodes both the positions and the actions of all the individuals.

We use two CNNs to obtain the collective activity prediction of the input sequence. The first CNN takes PMH of the last frame P_T as input, and the second CNN takes previously constructed collective map stack M_{H,W,A_I} as input. For both collective CNNs, we use Resnet-18 for the classification network. The final collective activity prediction is fused from the output of these two CNNs by an FC layer. We do not need hierarchical RNNs or similar recurrent networks here modeling the integration among individuals for collective activity recognition task like those in [16].

3.3 Training and Loss Function

We use Resnet-18 for each CNN for its learning efficiency. Separate softmax layers are applied on the outputs of individual PMH CNN, collective map CNN, collective PMH CNN, and collective fusion FC layer to obtain the predictions $p_{I,n}$, p_{C_1}, p_{C_2}, and p_{C_3}, respectively, where n is in $[1, N]$. For each CNN, we compute the loss between predictions and targets using the cross entropy, optimized by an Adam optimizer during training. The loss for a training sample of the individual part (individual PMH CNN) and the group part (collective map CNN, collective PMH CNN, and collective fusion FC layer) is defined, respectively, as follows:

$$L_I = -\sum_{i=1}^{A_I} \hat{\mathbf{p}}_i^I log \mathbf{p}_i^I,\ L_C = -\sum_{c=1}^{A_C} \hat{\mathbf{p}}_c^C log \mathbf{p}_c^C, \tag{2}$$

where $\hat{\mathbf{p}}_*^I$ and $\hat{\mathbf{p}}_*^C$ denote the one-hot-encoded ground truth probabilities for individual action classes and group activity classes, respectively, and $\mathbf{p}_*^I$ and $\mathbf{p}_*^C$

denote the softmax scores of the corresponding classes. Here, A_I and A_C are the total numbers of individual action classes and group activity classes, respectively.

4 Experimental Evaluation

We evaluate our approach on the challenging Volleyball dataset collected in [16], as it is the only relatively large-scale dataset with individual action, group activity labels, and individual locations of multi-person scenes. This dataset contains totally 4830 sequences trimmed from volleyball match videos, where 3493 for training plus validation, and 1337 for testing. Each sequence consists of 41 frames, and only the center frame is annotated with the ground truth bounding box of each player in the scene, the individual action of each player, and the group activity of the scene. We follow [16] to obtain the ground truth bounding boxes of people for those unannotated frames. There are totally 9 classes of individual actions, and 8 classes of group activities in this dataset.

4.1 Implementation Details

In previous works [1,16] on the same Volleyball dataset, each sequence is trimmed to a temporal window of length $T = 10$, corresponding to 4 and 5 frames before the annotated frame, and 5 and 4 after the annotated frames respectively. We find out that there are quite a large amount of sequences contain obviously different camera views when trimmed by these configurations. We manually find all sequences with different camera views among the temporal window corresponding to 9 frames before the annotated frames, and 5 frames after the annotated frames and remove these sequences from training. During training, if temporal sampling is applied, we trim a temporal window of length $T = 10$ from the range of 9 frames before the annotated frame, and 5 frames after it for each sequence. Otherwise we use the same temporal window as in the testing stage by fixing the temporal window corresponding to 9 frames before the annotated frame plus the annotated frame itself for each sequence. We use Resnet-18 for all CNNs in our proposed approach for its efficiency, and Adam optimizer for optimizing the model parameters.

We try two training strategies: sequence-by-sequence like in typical RNN training procedure, and batch-by-batch like in typical CNN training procedure. With sequence-by-sequence training strategy, choosing different data preprocessing related hyper parameters is more convenient, such as trying different values of enlarging scale s for constructing Indi-PMH, and retaining weight w for constructing PMH. With batch-by-batch training strategy, we can accelerate the training process of our all-CNN based approach. We first store PMH and Indi-PMH to disk, and then train the individual and collective CNNs separately by random sampling large batches of the corresponding PMH/Indi-PMH data. For the sequence-by-sequence training strategy, due to GPU memory constraint, we random retrieve one sequence at once and accumulate parameter gradients of several forwards before per backward. We first use batch-by-batch strategy to

train our individual PMH CNN and collective PMH CNN. Next, our collective map CNN is trained by loading and freezing the pretrained weights of the individual PMH CNN. Finally, the collective fusion FC layer is trained by loading and freezing the pretrained weights of these three CNNs. Sequence-by-sequence strategy is used to train both the collective map CNN and the collective fusion FC layer. With pretraining and freezing, the time spent for each epoch when training collective map CNN and the final collective fusion FC layer could thus be greatly reduced.

4.2 Pose Estimation Quality

Since we do not have ground truth pose annotation of Volleyball dataset, we evaluate the quality of pose estimation generated by OpenPose [5] by calculating the recall rate of ground truth individual bounding boxes given by the annotation of Volleyball dataset. The estimated bounding boxes are converted from the grouped joint positions output by OpenPose [5]. As joints are center points of human body parts, we increase each side of a converted bounding box by 5 pixels. In Table 1, we report the recall rates on the Volleyball dataset with different IoU threshold values: $0.5, 0.4, 0.3$. We find the gap between different IoU threshold values is resulted from cases where OpenPose [5] cannot generate very complete pose estimation for some individuals with occlusion, or sometimes generating mixed pose for occluded people. Some examples are shown in Fig. 5. For the rest of our experiments, we set the IoU threshold value to 0.3 when finding the matched bounding box for each of the ground truth ones, to make use of those imperfect but still partially informative pose.

Table 1. Recall rates of ground truth bounding boxes given by the annotation of Volleyball dataset with different IoU threshold values.

Threshold	Recall (train)	Recall (test)
0.5	84.7	86.4
0.4	91.4	92.7
0.3	94.8	95.8

Fig. 5. Examples for bounding boxes converted from incomplete or mixed pose of individuals with occlusion generated by OpenPose [5]. The red boxes are ground truth bounding boxes, and the yellow ones are converted from the pose estimation results. (Color figure online)

LSTM vs. CNN for Individual Part. We try an LSTM architecture taking pose features extracted by a CNN (Resnet-18) from individual joint heatmaps cropped from the whole scene version. Since we do not know human identities association across sequence frames, we try two matching mechanisms: bounding box IoU based and long term pose feature similarity based. In the first mechanism, human identities across sequence frames are associated through matching pairs with highest bounding box IoUs between adjacent frames. In the second mechanism, identities are associated through matching instances in different frames with highest long-term pose feature similarities. We compare the LSTM based architecture with these two matching mechanisms to a CNN based architecture taking Indi-PMH as input, which does not need to know human identities association. For an input sequence, we can simply use the bounding boxes converted from pose estimation result of the last frame, enlarged with certain scale (to contain more history information), to crop Indi-PMH from the whole scene version. In this way, the matching stage can be totally removed and thus reduce the execution time. We compare the testing individual action accuracy and recall rate of these methods in Table 2. We use Resnet-18 as the CNN model. We can see that using CNN for the individual part not only results in better accuracy, but also better recall rate since it can avoid potential human identity miss associated by the matching mechanisms. Using a CNN architecture with our proposed PMH representation on the individual action recognition task saves us from the need of any matching mechanisms, which would help in a real-time sequence-to-sequence scenario as it takes less time for the inference.

Table 2. The performance of using LSTM based and CNN based architectures. We report the testing individual action accuracy, and testing recall rate in the second and third columns, respectively. "ID Assoc." at the fourth column stands for human identity association needed or not when forming representation for each individual across frames.

Method	Individual Acc.	Recall	ID Assoc.
LSTM-matching-1	67.9	85.9	Yes
LSTM-matching-2	72.3	**95.4**	Yes
Resnet-18 w/Indi-PMH	**74.3**	**95.4**	**No**

4.3 PMH vs. PoTion

The biggest difference between our PMH representation and PoTion [9] is the temporal aggregation mechanisms. In PoTion [9], temporal relationship between joint heat maps of different sequence frames is represented through color coding with at least 2 color channels, while in our proposed PMH it is represented through intensity retaining with only 1 channel needed. Since both PoTion [9] and our proposed PMH can be simply classified by CNNs, the number of parameters of the classification networks would be nearly the same. However our only-1-channel needed PMH would be naturally more efficient than PoTion [9], which

takes at least 2 channels. Because the authors [9] did not release their code, we evaluate with our PoTion implementation with 3 color channels here. In Table 3, we report testing individual action and collective activity accuracy resulted from using PoTion and our proposed PMH on Volleyball dataset. We can see that although PMH only uses one color channel, our intensity retaining mechanism is still as effective as the color coding mechanism in PoTion [9].

Table 3. Accuracies of using PoTion and our PMH for individual action and group activity classification on Volleyball dataset. C denotes the number of color channels.

Representation	C	Individual Acc.	Collective Acc.
PoTion [9]	3	75.8	80.5
(Indi-)PMH	1	75.3	81.0

4.4 Collective Stream

The performance of our proposed collective PMH CNN, collective map CNN, and their fusion is reported at the bottom of Table 4. We use $w = 0.95$ and $s = 1.75$ in this experiment. We first pretrain the individual PMH CNN with learning rate $1e-4$. Then we train the collective PMH CNN from the scratch, and the collective map CNN using pretrained individual PMH CNN for generating individual action predictions (only the last frame of each sequence), with learning rates $1e-3$ and $1e-4$, respectively. The collective fusion FC layer is finally trained using these pretrained CNNs with their parameters frozen with learning rate set to $1e-2$. The individual and collective PMH CNNs are trained with batch-by-batch strategy, while the collective map CNN and collective fusion FC layer trained with sequence-by-sequence strategy. When training all these modules, random horizontal flipping is applied. Since in Volleyball dataset ground truth player positions and their individual action labels are provided, we conduct feasibility assessment of collective map representation first to see whether it is really discriminative for group activity recognition (using learning rate $1e-2$). The results show that collective maps built from ground truth information generate high testing accuracy, thus proving its effectiveness. Also, fusing both collective CNNs results in about 3% higher accuracy than the best of the two, suggesting that PMH and collective map representations are both effective and complementary.

We compare the performance of our system with the state-of-the-art methods on Volleyball dataset in Table 4. Our approaches denoted with "EST" use representations constructed from pose estimation only; ours with "GT" use representations constructed from ground truth human positions and individual action labels, noting that they directly access the ground truth individual action labels, so they should not be compared to other previous methods but just for reference. All our models are trained with training settings described in Sect. 4.4. From Table 4, it is evident that our fusion approach provides the best result among all the state-of-the-art methods without using the ground truth information [1,27].

Table 4. Comparison of group activity accuracy by using the proposed system with the state-of-the-art methods on the Volleyball dataset. Accuracy results generated by [1,27] without using ground truth information are denoted by MRF and PRO, respectively.

Method	Collective Acc.
HDTM [16] (GT)	81.9
SSU-temporal [1] (MRF/GT)	87.1/89.9
SRNN [2] (GT)	83.5
RCRG [15] (GT)	89.5
stagNet [27] (PRO/GT)	85.7/87.9
stagNet-attention [27] (PRO/GT)	87.6/89.3
PC-TDM [38] (GT)	87.7
SPA [30] (GT)	90.7
ours (collective PMH, Resnet-18) (EST)	84.6
ours (collective map, Resnet-18) (EST/GT)	77.3/92.7
ours (fusion) (EST/GT)	**87.7**/95.4

5 Conclusion

In this paper, we proposed two novel representations: pose motion history and collective map to represent the spatio-temporal dynamics of multi-person joints and the integration among individuals in a group scene. Based on these two representations, we developed a CNN based architecture without the need of any human identity association mechanisms, achieving superior performance on the challenging Volleyball dataset for the group activity recognition task. Future work would be to fuse pose estimation networks into an end-to-end model, and also to compensate the camera motion when constructing the pose motion history.

References

1. Bagautdinov, T., Alahi, A., Fleuret, F., Fua, P., Savarese, S.: Social scene understanding: end-to-end multi-person action localization and collective activity recognition. In: CVPR (2017)
2. Biswas, S., Gall, J.: Structural recurrent neural network (SRNN) for group activity analysis. In: WACV (2018)
3. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. TPAMI **23**(3), 257–267 (2001)
4. Cao, C., Zhang, Y., Zhang, C., Lu, H.: Action recognition with joints-pooled 3D deep convolutional descriptors. In: IJCAI (2016)
5. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2D pose estimation using part affinity fields. In: CVPR (2017)
6. Chéron, G., Laptev, I.: P-CNN: pose-based CNN features for action recognition. In: ICCV (2015)

7. Choi, W., Savarese, S.: A unified framework for multi-target tracking and collective activity recognition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7575, pp. 215–230. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33765-9_16
8. Choi, W., Shahid, K., Savarese, S.: Learning context for collective activity recognition. In: CVPR (2011)
9. Choutas, V., Weinzaepfel, P., Revaud, J., Schmid, C.: PoTion: pose MoTion representation for action recognition. In: CVPR (2018)
10. Deng, Z., Vahdat, A., Hu, H., Mori, G.: Structure inference machines: recurrent neural networks for analyzing relations in group activity recognition. In: CVPR (2016)
11. Donahue, J., et al.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR (2015)
12. Du, W., Wang, Y., Qiao, Y.: RPAN: an end-to-end recurrent pose-attention network for action recognition in videos. In: ICCV (2017)
13. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: CVPR (2016)
14. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In: CVPR (2018)
15. Ibrahim, M.S., Mori, G.: Hierarchical relational networks for group activity recognition and retrieval. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11207, pp. 742–758. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01219-9_44
16. Ibrahim, M.S., Muralidharan, S., Deng, Z., Vahdat, A., Mori, G.: Hierarchical deep temporal models for group activity recognition. TPAMI (2016)
17. Iqbal, U., Garbade, M., Gall, J.: Pose for action – action for pose. In: FG (2017)
18. Ji, S., Xu, W., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. TPAMI **35**(1), 221–231 (2013)
19. Johansson, G.: Visual perception of biological motion and a model for its analysis. Percept. Psychophys. **14**(2), 201–211 (1973)
20. Khamis, S., Morariu, V.I., Davis, L.S.: A flow model for joint action recognition and identity maintenance. In: CVPR (2012)
21. Khamis, S., Morariu, V.I., Davis, L.S.: Combining per-frame and per-track cues for multi-person action recognition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7572, pp. 116–129. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33718-5_9
22. Lan, T., Sigal, L., Mori, G.: Social roles in hierarchical models for human activity recognition. In: CVPR (2012)
23. Lan, T., Wang, Y., Yang, W., Robinovitch, S.N., Mori, G.: Discriminative latent models for recognizing contextual group activities. TPAMI **34**(8), 1549–1562 (2012)
24. Liu, M., Yuan, J.: Recognizing human actions as the evolution of pose estimation maps. In: CVPR (2018)
25. Newell, A., Huang, Z., Deng, J.: Associative embedding: end-to-end learning for joint detection and grouping. In: NIPS (2017)
26. Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: deep networks for video classification. In: CVPR (2015)

27. Qi, M., Qin, J., Li, A., Wang, Y., Luo, J., Van Gool, L.: stagNet: an attentive semantic RNN for group activity recognition. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11214, pp. 104–120. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01249-6_7
28. Ramanathan, V., Huang, J., Abu-El-Haija, S., Gorban, A., Murphy, K., Fei-Fei, L.: Detecting events and key actors in multi-person videos. In: CVPR (2016)
29. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS (2014)
30. Tang, Y., Wang, Z., Li, P., Lu, J., Yang, M., Zhou, J.: Mining semantics-preserving attention for group activity recognition. In: ACM MM (2018)
31. Tora, M.R., Chen, J., Little, J.J.: Classification of puck possession events in ice hockey. In: CVPR Workshop (2017)
32. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: ICCV (2015)
33. Tran, D., Wang, H., Torresani, L., Ray, J., Lecun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR (2018)
34. Veeriah, V., Zhuang, N., Qi, G.J.: Differential recurrent neural networks for action recognition. In: ICCV (2015)
35. Wang, L., et al.: Temporal segment networks: towards good practices for deep action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_2
36. Wu, D., Sharma, N., Blumenstein, M.: Recent advances in video-based human action recognition using deep learning: a review. In: IJCNN (2017)
37. Wu, Z., Jiang, Y.G., Wang, X., Ye, H., Xue, X.: Multi-stream multi-class fusion of deep networks for video classification. In: ACM MM (2016)
38. Yan, R., Tang, J., Shu, X., Li, Z., Tian, Q.: Participation-contributed temporal dynamic model for group activity recognition. In: ACM MM (2018)