Multi-Camera Temporal Grouping for Play/Break Event Detection in Soccer Games

Chunbo Song and Christopher Rasmussen

Department of Computer & Information Sciences
University of Delaware, Newark, DE 19716, USA
{songcb, ras}@udel.edu

Abstract. Many current deep learning approaches to action recognition focus on recognizing concrete (e.g., single actor) actions in trimmed videos from datasets such as *UCF-101 and HMDB-51*. However, high-level semantic analysis of sports videos often requires recognizing more abstract events or situations involving multiple players with longer time-scale context. This paper builds upon inflated 3D (I3D) ConvNets for video action recognition to detect and differentiate six abstract categories of events in untrimmed videos of soccer games from multiple fixed cameras: normal play, plus breaks in play due to kick-offs, free kicks, throw-ins, and goal and corner kicks. Raw video unit classifications by variants of the basic I3D network are post-processed by two novel and efficient grouping methods for localizing the boundaries of events. Our experiments show that the proposed methods can achieve 84.2% weighted precision for event categories at the level of video units, and boost event temporal localization mean average precision at 0.5 tIoU (mAP@0.5) to 62.0%.

Keywords: Event classification · Event Localization · I3D.

1 Introduction

Computer vision is fast becoming a powerful tool for sports video analysis. All kinds of vision-based tasks traditionally performed by the players themselves, spectators, referees, camera operators, and expert commentators can potentially be automated or enhanced for a myriad of applications. These include training and coaching feedback, enhanced rule enforcement accuracy, replay annotation and explanation for broadcasters, measuring detailed player and team statistics, and even serving as perception modules for robotic sports participants. While the exact purpose of the analysis may vary, as well as the sensors employed, there are certain visual skills such as ball tracking [30, 22], player segmentation [3, 21, 14], recognition [11], and pose estimation [17], and recognition of formations, plays, and situations [1, 31, 32, 12] that many sports vision systems have in common.

One of the most basic forms of sports video understanding, at a high level, is *play/break* categories classification [38, 7, 28]. That is, can one infer whether a particular video sequence depicting part of a game is showing actual game

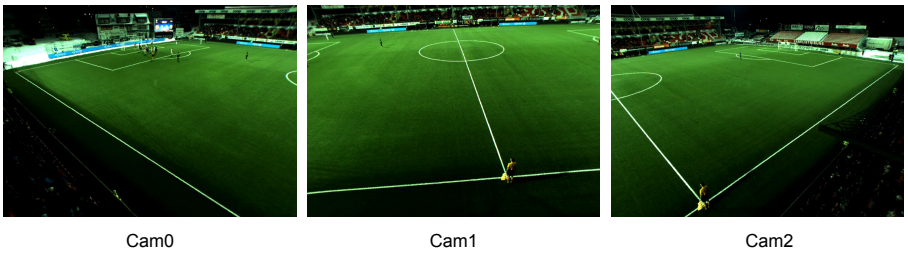

Fig. 1. Example frames of a *corner kick* event defined in the rule of soccer games. In the SVPP dataset, three fixed cameras capture different regions of the field.

play, or is there a *break* in the action? We follow the event definition introduced by Giancola, *et al.* [12] to represent *play/break*s in videos of soccer games, who defined an event as an action that is anchored in a single time instance, defined within a specific context respecting a specific set of rules. Distinguishing between these two game states is not trivial, because during breaks the players (as well as the ball) may still be visible, and still moving. Events like shots, passes, and fouls that occur in the course of play are understandably popular subjects of study for game analysis [28, 3, 31]. However, here we investigate break events, which may be due to a timeout, a foul, halftime, an injury, a ball out of bounds, or any number of sports-specific events[1].

Rather than recognizing events or actions in the long untrimmed video either from one camera or from a broadcast feed (in this case, a video contains camera panning and zooming, shot boundaries and subjects and scenes selected of the action), in this paper we aim to differentiate and localize *play* and *break* events using the Soccer Video and Player Position Dataset (SVPP) [25] which has two complete soccer games from three fixed cameras, like Fig. 1 shows. This dataset doesn't have event categories, we manually annotate them in a frame level. Therefore, the event segment can be extracted. We first consider the Two-Stream Inflated 3D ConvNet (I3D) [4] trained on three cameras be the one worthy for the comparison since it is one of state-of-the-art architectures. The I3D, which takes several seconds of video context or a sequence of frames in a fixed length (which we call video unit for differentiating with the event segment), is able to recognize play and different break categories fairly reliably. Because of multi-camera, an assistant neural network (AN) is then utilized to combine all I3D's predictions on synchronous units from all cameras. We also extend I3D to

[1] In particular, we study soccer *break* event categories as defined in the FIFA rule book [8]: (1) kick-offs (to start each half or after a goal), (2) free kicks (after a foul), (3) penalty kicks, (4) throw-ins (touch line out of bounds), (5) goal kicks (end line out of bounds caused by offensive team), (6) corner kicks (end line out of bounds caused by defensive team), and (7) dropped balls (all other situations), Detecting these *break* event segments in the soccer game video is a difficult task due to the sparsity within a video, but also they have different duration.

our C-I3Ds by integrating observations from multiple cameras, even those not directly viewing the action, are able to boost performance non-trivially. In C-I3Ds, each camera corresponds to one I3D with two-stream (RGBs and Optical Flows). The integration of these I3Ds takes synchronous video units from all cameras as inputs. Outputs are combined to generate predictions.

Here is an assumption: if a classifier performs well with unit inputs, boundaries will be localized easily and efficiently. Unlike recent methods [9, 10, 42] feed by trained deep features for localizing actions or generating action proposals in untrimmed videos, we propose two efficient methods to group adjacent video units for the event localization: probability-based grouping (PBG) and class-based grouping (CBG). Both grouping methods build upon predicted probabilities and classes by our I3D-based model. They and their combination achieve promising performance on our testing.

In summary, our contribution are three-fold: (1) We extend the I3D network to be suitable for the multi-camera case to classify video units. (2) We propose probability-based and class-based grouping methods to facilitate C-I3Ds for event localization. (3) The combination of both grouping methods boosts performance on both classification and localization during testing.

2 Related Work

Deep learning architectures for video classification and action recognition in videos have also shown great promise recently [16, 33, 34], including LSTM networks for human action classification [13] and recognizing pass, shoot, dribble actions from multi-camera video with player and ball trajectories [31]. Strategies for fusing optical flow with spatial information have also achieved considerable success [26, 4, 36, 37], as well as 3D convolutional neural networks which extract features from the spatial and the temporal domains jointly by performing 3D convolutions to capture the motion information encoded in multiple adjacent frames [15]. Based on an Inception module [29], I3D expands 2D filters and pooling kernels to 3D to make it possible to learn seamless spatio-temporal features from video and applied on two-stream (RGB and Optical Flow). The optical flow input may provide some sense of recurrence [4]. The I3D network trained on optical flows carries optimized, smooth flow information. Experimentally it is valuable to classify actions. After pre-training on the Kinetics dataset [6], I3D models have reached 80.9% on HMDB-51 and 98.0% on UCF-101 [4, 27, 18] which is the most state-of-the-art method to our best knowledge.

Despite these advances, localizing action boundaries in a long, untrimmed video is still a difficult problem. Applying temporal sliding window is a typical scheme after classification [24, 35]. The feature extracted from deep neural networks is globally pooled within each window for generating SVM inputs. Yuan *et al.* [39] proposed an approach to address the uncertainty of action occurrence and utilization of information from different scales. Although these works have shown promising performance in their task, the efficiency is still unresolved. Many recent methods have examined this problem as analogous to object de-

tection but in the temporal dimension, they utilize features from deep neural networks to localize action boundaries, including temporal action proposals [10, 9, 5].

In the work which is similar with ours, Giancola, *et al.*[12] try to "spot" three soccer event categories: (*goal, card*, and *substitution*). However, they didn't try to identify the boundaries of an action within a video, but simply the anchor time that identifies an event with one-minute resolution.

Some other soccer datasets include ISSIA [19], which contains player, referee, and ball positions as seen from multiple fixed cameras; and SoccerNet [12]. But, ISSIA is very short – only 2-minute sequences, and while SoccerNet is huge (764 hours of video), it only contains very sparse yellow/red card, goal, and substitution events at essentially 1-minute label resolution. AZADI [2] has play/break labels and Soccer 152-A [20] has a number of actions, including those of referees, coaches, and spectators, but neither of these could be obtained for this work.

3 Dataset and Annotation

The Soccer Video and Player Position Dataset [25] (SVPP) is used in our work. The portion of the dataset that we use consists of two complete soccer game videos captured at 30 fps by three fixed cameras whose overlapping fields of view each roughly cover one-third of the length of the field. These two games are TromsoIL vs. Anzhi (*TvA*) and TromsoIL vs. Stromsgodset (*TvS*). The original resolution of each frame in the video is 1280×960. The video of the games are untrimmed, and no broadcast content. 324,284 frames of each camera are annotated with *play* occupying about 65.9% and *break* 34.1%. There are no instances of penalty kicks or drop balls in the videos, so we remove these two break categories. Of the break frames, 0.4% are kick-off (only at the beginning of the game or after the half, as there are no goals), 32.4% free kick, 24.4% throw-in, 14.7% corner kick, and 28.1% goal kick. And different event categories have various time duration.

4 Methods

4.1 Classification

I3D, Assistant Neural Network (AN) and C-I3Ds Because deep neural networks have displayed good ability of generalization [41, 23], we firstly train one I3D network on units from all cameras. And assign one trained I3D model to a related camera during testing. It implies that different I3D networks share weights with each other. Thus, synchronous units from different cameras are sent to their corresponding I3D networks. Their outputs (confidence scores or logits) are concatenated to feed into AN, which is, in our work, a fully-connected network for outputting event classification results by combining confidence scores from different cameras' related I3D models.

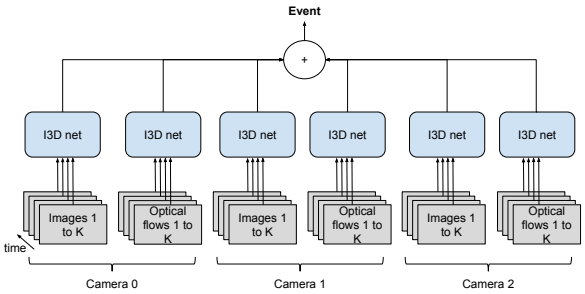

Fig. 2. C-I3Ds: the combination for multiple two-stream I3D framework.

However, in the multi-camera case, both machine and human may be error-prone on pointing out the event when some cameras are unavailable. Training one I3D network on units from all cameras may result in bad recognition. Like the right frame showed in Fig. 1, people cannot tell the exact event. Therefore, deploying several I3D networks for different cameras on training is an alternative way. Unlike the previous way we used, these I3D networks don't share weights with each other. Fig. 2 shows its architecture. Each pair of two-stream I3D networks corresponds to a camera. And the output of these separate I3D networks are combined lately, without applying AN. We call this C-I3Ds. Because synchronous video units have the same categories, we trained these separate I3D networks jointly and averaged their predictions at both training and testing time.

4.2 Event Boundaries Localization

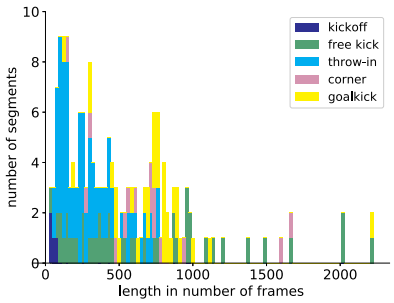

Fig. 3. The distribution of different *break* categories in length on our training set.

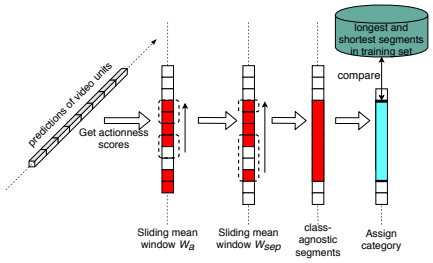

Fig. 4. Probability-based grouping

Probability-based grouping (PBG) A temporal sequence of predicted probabilities may indicate the transition from one state to another. Ideally, such transition would be smooth and precise. But, in videos, classifiers may not always achieve perfect results due to several reasons such as subjective labelling, restrictions of the classifier, limited data and etc. Using good classifiers, false classification on frames/units still commonly exists, and thus makes localization difficult. To address this problem, we applied a sliding window manner on predicted probabilities from deep neural networks to not only filter out some errors, but also can group adjacent segments together. The probability-based grouping has two steps: actionness scores grouping and break categories assigning. Fig. 4 illustrated the full pipeline of PBG.

We extend the definition of actionness scores in [42] and use it to describe the probability of a given video unit is a break event. For an unit k, we get its actionness score by $prob_{k,a} = 1 - prob_{k,p}$, where $prob_{k,a}$ is the actionness score and $prob_{k,p}$ is the probability of 'play' of k. If $\forall k \in [i, j], prob_{k,a} \geq t_a$, we will be able to get a class-agnostic segment $S_{i,j}$, t_a is a threshold of differentiating 'break' with 'play', and i, j are the boundary of a segment. Based on observations, the beginning of any break event is usually very similar with play and the beginning of any play is also very similar with its previous neighbor 'break'. Therefore, for each break segment $S_{i,j}$, we utilize $l_{aw} - 1$ units before i. And apply a mean window W_a with size l_{aw} and stride st from $i - l_{aw} + 1$ to $j - l_{aw} + 1$. It will adjust $S_{i,j}$ to $S_{p,j}$ given t_a. After that, an adjusted segment may overlap or too close with its neighbor segments. We collect these averaged actionness scores from m to n and apply another mean window W_{sep} to determine if separate or group them. We define l_{sep} be the size of W_{sep}. l_{sep} also implies the minimum length (i.e. if the distance between two adjacent segments is less than l_{sep}, we think they are too close). We densely slide W_{sep} across $[m, n]$ with stride 1, and compare every scores with a threshold t_{ma}. If the number of consecutive steps for W_{sep} is more than l_{sep} and the mean scores are less than t_{ma}, then separate them. It is worth noticing that we shrink the size of W_{sep} if $n - t + 1 < l_{sep}$ for outputting $n - m + 1$ mean scores, where t is the index of the current unit.

For assigning break categories, we average probabilities of all categories for all units within the refined segment, denoted by $P_{i',j'}$. Because, for each category c, the shortest and the longest lengths $G_{c,short}$ and $G_{c,long}$ can be obtained from the training set, we iteratively check and assign the most possible category to the segment based on the its length $l_{i',j'}$, if $l_{i',j'} \geq G_{c,short}$ and $l_{i',j'} \leq G_{c,long}$. If no category can satisfy this segment, 'play' will be assigned to it.

Class-based Grouping (CBG) The drawback of PBG is, l_{aw} and l_{sep} might not be very large because large window size will eliminate some short but true predictions. Therefore, many false positives are retained. Based on the rule of professional soccer games, we observed facts that any break category must start at the end of 'play', rather than other break categories, except 'kickoff'. And, any break category will usually not takes too short, similar as what we mentioned in assigning break categories in PBG. So based on these facts, we utilize predicted

classes to further adjust both boundaries and categories. For each input segment $S_{i,j}$ (including 'play') with length is $l_{i,j}$, its two neighbor segments $S_{x,i-1}$ and $S_{j+1,y}$ are extracted if $l_{i,j} < t_{len}$, where t_{len} is a threshold for indicating small segments. Then, group $S_{j+1,y}$ with $S_{i,j}$ and assign its category to $S_{i,j}$ if $l_{x,i-1} < l_{j+1,y}$. Otherwise, combine $S_{x,i-1}$ to it. This step is processed iteratively until all lengths are greater than t_{len}. After that, if any adjacent segment all belong to any 'break' category (except 'kickoff'), we merge the short segment with the adjacent longer one and assign the category to it.

5 Experiment and Analysis

Data Preparation We randomly extract synchronous video units from three halves' videos and all cameras to generate the training set. The three halves are: the 2nd half of *TvA* and the 1st and 2nd half of *TvS*. In our work, each video unit has 64 frames with 1 frame of unit's stride. Fig. 3 displays the distribution of length of event segments in different break categories in our training set. We assign the label of the last frame in an unit to be the category of this video unit. Due to highly imbalanced number of categories in our dataset, we over-sampled video units which are break categories. Thus, for each category (include play), 9,000 synchronous video units from 3 cameras are in the training set. Data augmentation is necessary to improve the ability of generalization of models because of limited instances of some categories. For each frame, we randomly cropped with size 1160×921. Frames in the same video unit are cropped at the same place, as well as the corresponding optical flows images. These frames are re-sized to 224×224 for feeding into I3D and C-I3Ds. We also applied random right-left flipping, frames and corresponding optical flows images in the same video unit do have the same flipping direction. For the test set, we use the 1st half of *TvA* with unit's stride 1 frame as well for both the event classification and boundary detection. There are 81,471 units in our test set.

Implementation Details of Training We train the I3D network in an end-to-end manner, with units of video frames as the input. The optical flows are calculated by Dual TV_L^1 method [40]. The I3D network is trained on randomly selected units from all cameras. For both I3D and C-I3Ds, we use SGD to learn parameters. The learning rates are set to 0.01. And dropout of both I3D and C-I3Ds is 0.5 during training. We make AN have 2 layers of 20 hidden nodes. We deploy the same I3D models to predict confidence scores on different camera units. The input of AN is the confidence score from I3D models on synchronous units. The optimization of AN is launched by Adam optimizer with learning rate 0.0001. The training iterations of both I3D and C-I3Ds are 240K, and they are all trained from scratch. The batch size is 4 because of the memory issue. AN is trained for 20K iterations with batch size 64.

Evaluation Metrics For event classification, we calculate Precision for different event categories (include 'play'). The Weighted Precision is calculated as well

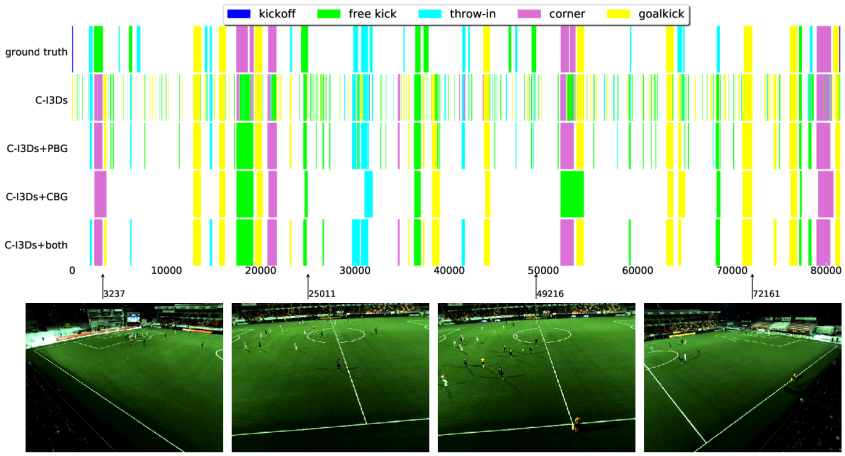

Fig. 5. This is the demonstration of our methods' prediction on the test set. Applying both PBG and CBG after C-I3Ds, precise event boundaries are localized. We sampled four images that are the last frames of their corresponding segments in the ground truth to illustrate some categories are hard to be recognized.

for indicating the overall classification performance. For event localization, we report mean Average Precision (mAP) and Average Recall (AR) using temporal Intersection over Union (tIoU) threshold of 0.5. Because none of the 'kickoff' units is recognized, it is not included in the analysis of the results.

Classification Table. 1 displays the precision of the different model on testing. I3D network trained on units from all cameras doesn't perform really well, even AN is applied. The C-I3Ds perform better than the I3D with AN on almost all categories, except 'free kick'. Without any grouping method, its weighted precision achieves 78.7%. Units can obtain labels after applying grouping methods with the C-I3Ds. If both l_{aw} and l_{sep} are 46, the weighted precision reaches 83.5%. If the CBG is applied with t_{len} is 125, the weighted precision (80.9%) is lower than using PBG, but still higher than C-I3Ds'. We also combine PBG and CBG to adjust predicted categories of units and it achieves relatively good weighted precision performance (84.2%).

The classification result indicates that different levels of difficulty of these event categories. This may be caused by limited number of events in our training set, even though we over-sample frames with this category to make the training set balance. Moreover, owing to diversities, the 'free kick' is also hard to be differentiated from other categories.

Event Localization We use C-I3Ds as the baseline to evaluate performance on the localization by making input video units be in the chronological order and

Table 1. Per-unit (stride 1) classification precision (%)

method	play	free kick	throw-in	corner	goalkick	weighted precision
I3D	88.4	15.8	18.9	51.1	37.6	73.3
I3D+AN	85.4	29.4	31.8	63.4	44.4	74.0
C-I3Ds without grouping	88.7	16.5	50.5	66.8	61.2	78.7
C-I3Ds+PBG($l_{aw}, l_{sep} = 33$)	90.6	17.8	65.5	68.5	72.1	82.2
C-I3Ds+PBG($l_{aw}, l_{sep} = 46$)	91.3	20.3	71.2	72.6	**73.8**	83.5
C-I3Ds+CBG($t_{len} = 65$)	89.5	**29.9**	77.5	**73.3**	**73.8**	83.2
C-I3Ds+CBG($t_{len} = 125$)	88.7	18.3	77.3	58.5	72.6	80.9
C-I3Ds+both($l_{aw}, l_{sep} = 33, t_{len} = 65$)	91.0	19.2	76.2	65.0	72.6	82.9
C-I3Ds+both($l_{aw}, l_{sep} = 33, t_{len} = 125$)	91.2	21.3	76.6	65.0	73.1	83.3
C-I3Ds+both($l_{aw}, l_{sep} = 46, t_{len} = 65$)	91.3	22.0	**79.6**	72.6	72.3	84.0
C-I3Ds+both($l_{aw}, l_{sep} = 46, t_{len} = 125$)	**91.6**	27.7	75.7	72.6	71.6	**84.2**

localizing boundaries because of its decent classification performance. Table. 2 and Table. 3 display the precision and the recall on the event localization.

While the C-I3Ds achieves a decent performance on the classification, the result of event localization is bad. Given tIoU threshold as 0.5, the mAP is less than 1%, and AR is 14.0%. After applying PBG after C-I3Ds with l_{aw} and l_{sep} are 33, the mAP@0.5 and AP@0.5 have reached 33.4% and 41.9%, respectively. If l_{aw} and l_{sep} are all 46, the mAP@0.5 is 39.3% and the AR@0.5 is 46.5%. Because some segments are pretty short in the training set, it appears both window sizes l_{aw} and l_{sep} is small to maintain these correct segments as many as possible.

C-I3Ds with CBG performs well, which achieves 41.3% mAP@0.5 when set t_{len} to be around the unit size (i.e. 65). Assigning it a larger value for t_{len}, some short but true segments will be merged into their neighbors. When t_{len} is much larger (e.g. 125), both mAP and AR will be low (30.2% and 25.6%) due to the incorrect merging. C-I3Ds with PBG achieves higher recalls than CBP (46.5% vs. 34.9%). The PBG will still leave too many short segments because of its short window sizes. In these segments, the number of false positives is far more than true positives'. And, CBG with relatively larger t_{len} can be applied for eliminating them. Thus, we test the combination of these two grouping methods after C-I3Ds. The combination boosts mAP@0.5 up to 62.0% without sacrificing AR much as Table. 2 and Table. 3 showed. Fig. 5 shows qualitative examples on testing. The four frames display some correct and incorrect recognition. Besides 'kickoff', 'free kick' is the most difficult category for recognition, like the first frame with the number 3237. The corresponding segment of the third frame with the number 49216 is eliminated by the grouping since C-I3Ds only predicts a few short segments. 'goalkick' is the easiest category to be detected in the testing, as the rightmost frame shows. From the Fig. 5, although some short segments in the ground truth are hardly detected by C-I3Ds, the predicted boundary can be adjusted accurately by applying our grouping methods. Both PBG and CBG are efficient. Running both after C-I3Ds only spends less than 1 second on testing.

6 Conclusion and Future Work

In this paper, we firstly introduce our construction upon the I3D network to make it be suitable with multi-camera in the soccer game and apply it to classify

Table 2. Results for event localization in precision(%) and mAP(%)@0.5 tIoU

method	free kick	throw-in	corner	goalkick	mAP
C-I3Ds without grouping	0.3	0.0	0.0	1.0	0.3
C-I3Ds+PBG($l_{aw}, l_{sep} = 33$)	5.3	38.5	50.0	44.4	33.4
C-I3Ds+PBG($l_{aw}, l_{sep} = 46$)	9.4	41.7	60.0	56.3	39.3
C-I3Ds+CBG($t_{len} = 65$)	22.2	50.0	50.0	44.4	41.3
C-I3Ds+CBG($t_{len} = 125$)	33.3	0.0	**66.7**	**70.0**	30.2
C-I3Ds+both($l_{aw}, l_{sep} = 33, t_{len} = 65$)	10.0	55.6	50.0	47.1	41.9
C-I3Ds+both($l_{aw}, l_{sep} = 33, t_{len} = 125$)	20.0	**83.3**	50.0	53.3	56.7
C-I3Ds+both($l_{aw}, l_{sep} = 46, t_{len} = 65$)	13.6	62.5	60.0	56.3	48.9
C-I3Ds+both($l_{aw}, l_{sep} = 46, t_{len} = 125$)	**37.5**	**83.3**	60.0	56.3	**62.0**

Table 3. Results for event localization in recall(%) and AR(%)@0.5 tIoU

method	free kick	throw-in	corner	goalkick	AR
C-I3Ds without grouping	12.5	0.0	0.0	55.6	14.0
C-I3Ds+PBG($l_{aw}, l_{sep} = 33$)	25.0	**27.8**	**50.0**	88.9	41.9
C-I3Ds+PBG($l_{aw}, l_{sep} = 46$)	**37.5**	**27.8**	**50.0**	**100.0**	**46.5**
C-I3Ds+CBG($t_{len} = 65$)	25.0	11.1	**50.0**	88.9	34.9
C-I3Ds+CBG($t_{len} = 125$)	25.0	0.0	33.3	77.8	25.6
C-I3Ds+both($l_{aw}, l_{sep} = 33, t_{len} = 65$)	25.0	**27.8**	33.3	88.9	39.5
C-I3Ds+both($l_{aw}, l_{sep} = 33, t_{len} = 125$)	25.0	**27.8**	33.3	88.9	39.5
C-I3Ds+both($l_{aw}, l_{sep} = 46, t_{len} = 65$)	**37.5**	**27.8**	**50.0**	**100.0**	**46.5**
C-I3Ds+both($l_{aw}, l_{sep} = 46, t_{len} = 125$)	**37.5**	**27.8**	**50.0**	**100.0**	**46.5**

soccer game event rather than actions from individuals. We also propose PBG and CBG to localize/adjust event boundaries in the video of the soccer game. The performance demonstrates the combination of these two grouping methods can achieve a promising result. In the future, we will test our methods on the event classification and localization in more general scenarios. And, due to our grouping methods are not in a learning manner, we are still interested in inferring event boundaries by machine learning approaches.

References

1. Assfalg, J., Bertini, M., Colombo, C., Bimbo, A.D., Nunziati, W.: Semantic annotation of soccer videos: automatic highlights detection. Computer Vision and Image Understanding **92**(2), 285–305 (2003)
2. Bozorgpour, A., Fotouhi, M., Kasaei, S.: Robust homography optimization in soccer scenes. In: Iranian Conference on Electrical Engineering (2015)
3. Canales, F.: Automated Semantic Annotation of Football Games from TV Broadcast. Ph.D. thesis, Department of Informatics, TUM Munich (2013)
4. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the Kinetics dataset. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
5. Chao, Y.W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R.: Rethinking the faster r-cnn architecture for temporal action localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1130–1139 (2018)
6. DeepMind: Convolutional neural network model for video classification trained on the Kinetics dataset (2017), https://github.com/deepmind/kinetics-i3d
7. Fani, M., Yazdi, M., Clausi, D., Wong, A.: Soccer video structure analysis by parallel feature fusion network and hidden-to-observable transferring markov model. IEEE Access **5**, 27322–27336 (2017)
8. Fdration Internationale de Football Association (FIFA): Laws of the game (2015), https://img.fifa.com/image/upload/datdz0pms85gbnqy4j3k.pdf

9. Gao, J., Chen, K., Nevatia, R.: Ctap: Complementary temporal action proposal generation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 68–83 (2018)
10. Gao, J., Yang, Z., Chen, K., Sun, C., Nevatia, R.: Turn tap: Temporal unit regression network for temporal action proposals. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3628–3636 (2017)
11. Gerke, S., Muller, K., Schafer, R.: Soccer jersey number recognition using convolutional neural networks. In: IEEE International Conference on Computer Vision Workshop (2015)
12. Giancola, S., Amine, M., Dghaily, T., Ghanem, B.: Soccernet: A scalable dataset for action spotting in soccer videos. In: CVPR Workshop on Computer Vision in Sports (2018)
13. Grushin, A., Monner, D.D., Reggia, J.A., Mishra, A.: Robust human action recognition via long short-term memory. In: Neural Networks (IJCNN), The 2013 International Joint Conference on. pp. 1–8. IEEE (2013)
14. Huda, N., Jensen, K., Gade, R., Moeslund, T.: Estimating the number of soccer players using simulation-based occlusion handling. In: CVPR Workshop on Computer Vision in Sports (2018)
15. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence **35**(1), 221–231 (2013)
16. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1725–1732 (2014)
17. Kazemi, V., Sullivan, J.: Using richer models for articulated pose estimation of footballers. In: British Machine Vision Conference (2012)
18. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: A large video database for human motion recognition. In: IEEE International Conference on Computer Vision (2011)
19. Leo, M., Mosca, N., Spagnolo, P., Mazzeo, P., et al.: A semi-automatic system for ground truth generation of soccer video sequences. In: Advanced Video and Signal Based Surveillance (2009)
20. Liu, T., Lu, Y., Lei, X., Zhang, L., Wang, H., Huang, W., Wang, Z.: Soccer video event detection using 3D convolutional networks and shot boundary detection via deep feature distance. In: International Conference on Neural Information Processing (2017)
21. Lu, K., Chen, J., Little, J.J., He, H.: Light cascaded convolutional neural networks for accurate player detection. In: British Machine Vision Conference (2017)
22. Maksai, A., Wang, X., Fua, P.: What players do with the ball: A physically constrained interaction modeling. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016)
23. Neyshabur, B., Bhojanapalli, S., McAllester, D., Srebro, N.: Exploring generalization in deep learning. In: Advances in Neural Information Processing Systems. pp. 5947–5956 (2017)
24. Ni, B., Yang, X., Gao, S.: Progressively parsing interactional objects for fine grained action detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1020–1028 (2016)
25. Pettersen, S.A., Johansen, D., Johansen, H., Berg-Johansen, V., Gaddam, V.R., Mortensen, A., Langseth, R., Griwodz, C., Stensland, H.K., Halvorsen, P.: Soccer video and player position dataset. In: ACM Multimedia Systems Conference (2014)

26. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)
27. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. Tech. Rep. CRCV-TR-12-01, University of Central Florida (2012)
28. Sozykin, K., Khan, A.M., Protasov, S., Hussain, R.: Multi-label class-imbalanced action recognition in hockey videos via 3D convolutional neural networks. In: IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (2018)
29. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
30. Tong, X., Lu, H., Liu, Q.: An effective and fast soccer ball detection and tracking method. In: International Conference on Pattern Recognition (2004)
31. Tsunoda, T., Komori, Y., Matsugu, M., Harada, T.: Football action recognition using hierarchical lstm. In: CVPR Workshop on Computer Vision in Sports (2017)
32. Wagenaar, M., Okafor, E., Frencken, W., Wiering, M.: Using deep convolutional neural networks to predict goal-scoring opportunities in soccer. In: International Conference on Pattern Recognition Applications and Methods (2017)
33. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision. pp. 3551–3558 (2013)
34. Wang, L., Li, W., Li, W., Van Gool, L.: Appearance-and-relation networks for video classification. arXiv preprint arXiv:1711.09125 (2017)
35. Wang, L., Xiong, Y., Lin, D., Van Gool, L.: Untrimmednets for weakly supervised action recognition and detection. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 4325–4334 (2017)
36. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European Conference on Computer Vision. pp. 20–36. Springer (2016)
37. Wang, Y., Song, J., Wang, L., Van Gool, L., Hilliges, O.: Two-stream sr-cnns for action recognition in videos. In: BMVC (2016)
38. Xie, L., Xu, P., Chang, S.F., Divakaran, A., Sun, H.: Structure analysis of soccer video with domain knowledge and hidden markov models. Pattern Recognition Letters **25**(7), 767–775 (2004)
39. Yuan, J., Ni, B., Yang, X., Kassim, A.A.: Temporal action localization with pyramid of score distribution features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3093–3102 (2016)
40. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-l 1 optical flow. In: Joint pattern recognition symposium. pp. 214–223. Springer (2007)
41. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530 (2016)
42. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2914–2923 (2017)