A Context-Aware Loss Function for Action Spotting in Soccer Videos

Anthony Cioppa*
University of Liège
anthony.cioppa@uliege.be

Adrien Deliège*
University of Liège
adrien.deliege@uliege.be

Silvio Giancola*
KAUST
silvio.giancola@kaust.edu.sa

Bernard Ghanem
KAUST

Marc Van Droogenbroeck
University of Liège

Rikke Gade
Aalborg University

Thomas B. Moeslund
Aalborg University

Abstract

In video understanding, action spotting consists in temporally localizing human-induced events annotated with single timestamps. In this paper, we propose a novel loss function that specifically considers the temporal context naturally present around each action, rather than focusing on the single annotated frame to spot. We benchmark our loss on a large dataset of soccer videos, SoccerNet, and achieve an improvement of 12.8% *over the baseline. We show the generalization capability of our loss for generic activity proposals and detection on ActivityNet, by spotting the beginning and the end of each activity. Furthermore, we provide an extended ablation study and display challenging cases for action spotting in soccer videos. Finally, we qualitatively illustrate how our loss induces a precise temporal understanding of actions and show how such semantic knowledge can be used for automatic highlights generation.*

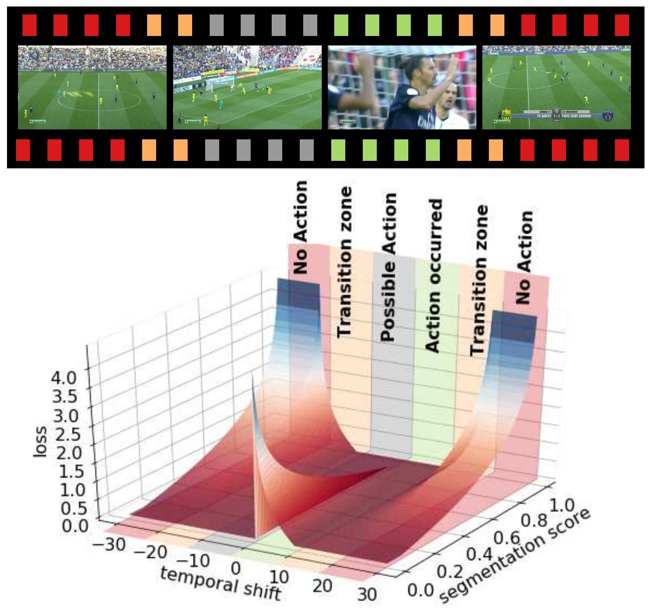

Figure 1. **Context-aware loss function.** We design a novel loss that leverages the temporal context around an action spot (at a temporal shift of 0). We heavily penalize the frames **far-distant** from the action and decrease the penalty for those **gradually closer**. We do not penalize the frames **just before** the action to avoid providing misleading information as its occurrence is uncertain, but we heavily penalize those **just after**, as the action has occurred.

1. Introduction

Aside from automotive, consumer, and robotics applications, sports is considered one of the most valuable applications in computer vision [54], capping $91 billion of annual market revenue [31], with $28.7 billion from the European Soccer market alone [15]. Recent advances helped provide automated tools to understand and analyze broadcast games. For instance, current computer vision methods can localize the field and its lines [17, 24], detect players [12, 63], their motion [18, 40], their pose [7, 67], their team [27], track the ball position [50, 56] and the camera motion [39]. Understanding spatial frame-wise information is useful to enhance the visual experience of sports viewers [47] and to gather players statistics [57], but it misses higher-level game understanding. For broadcast producers, it is of paramount importance to have a deeper understanding of the game actions. For instance, live broadcast production follows specific patterns when particular actions occur; sports live reporters comment on the game actions; and highlights producers generate short summaries by ranking the most representative actions within the game. In order to automate these production tasks, computer vision methods should understand the salient actions of a game and respond accordingly. While spatial information is widely studied and quite mature, localizing actions in time remains a challenging task for current video understanding algorithms.

(*) Denotes equal contributions. Code available at https://github.com/cioppaanthony/context-aware-loss.

In this paper, we target the action spotting challenge, with a primary application on soccer videos. The task of action spotting has been defined as the temporal localization of human-induced events annotated with a single timestamp [21]. Inherent difficulties arise from such annotations: their sparsity, the absence of start and end times of the actions, and their temporal discontinuities, *i.e*. the unsettling fact that adjacent frames may be annotated differently albeit being possibly highly similar. To overcome these issues, we propose a novel loss that leverages the temporal context information naturally present around the actions, as depicted in Figure 1. To highlight its generality and versatility, we showcase how our loss can be used for the task of activity localization in ActivityNet [23], by spotting the beginning and end of each activity. Using the network BMN introduced in [34] and simply substituting their loss with our enhanced context-aware spotting loss function, we show an improvement of 0.15% in activity proposal leading to a direct 0.38% improvement in activity detection on ActivityNet [23]. On the large-scale action spotting soccer-centric dataset, SoccerNet [21], our network substantially increases the Average-mAP spotting metric from 49.7% to 62.5%.

Contributions. We summarize our contributions as follows. **(i)** We present a new loss function for temporal action segmentation further used for the task of action spotting, which is parameterized by the time-shifts of the frames from the ground-truth actions. **(ii)** We improve the performance of the state-of-the-art method on ActivityNet [23] by including our new contextual loss to detect activity boundaries, and improve the action spotting baseline of SoccerNet [21] by 12.8%. **(iii)** We provide detailed insights into our action spotting performance, as well as a qualitative application for automatic highlights generation.

2. Related Work

Broadcast soccer video understanding. Computer vision tools are widely used in sports broadcast videos to provide soccer analytics [42, 57]. Current challenges lie in understanding high-level game information to identify salient game actions [13, 60], perform automatic game summarization [49, 51, 61] and report commentaries of live actions [65]. Early work uses camera shots to segment broadcasts [16], or analyze production patterns to identify salient moments of the game [46]. Further developments have used low-level semantic information in Bayesian frameworks [25, 55] to automatically detect salient game actions.

Machine learning-based methods have been proposed to aggregate temporally hand-crafted features [5] or deep frame features [28] into recurrent networks [44]. SoccerNet [21] provides an in-depth analysis of deep frame feature extraction and aggregation for action spotting in soccer game broadcasts. Multi-stream networks merge additional optical flow [10, 59] or excitement [6, 51] information to improve game highlights identification. Furthermore, attention models are fed with per-frame semantic information such as pixel information [13] or player localization [32] to extract targeted frame features. In our work, we leverage the temporal context information around actions to handle the intrinsic temporal patterns representing these actions.

Deep video understanding models are trained with large-scale datasets. While early works leveraged small custom video sets, a few large-scale datasets are available and worth mentioning, in particular Sports-1M [30] for generic sports video classification, MLB-Youtube [43] for baseball activity recognition, and GolfDB [41] for golf swing sequencing. These datasets all tackle specific tasks in sports. In our work, we use SoccerNet [21] to assess the performance of our context-aware loss for action spotting in soccer videos.

Video understanding. Recent video challenges [23] include activity localization, that find temporal boundaries of activities. Following object localization, two-stage approaches have been proposed including proposal generation [9] and classification [8]. SSN [69] models each action instance with a structured temporal pyramid and TURN TAP [20] predicts action proposals and regresses the temporal boundaries, while GTAN [38] dynamically optimizes the temporal scale of each action proposal with Gaussian kernels. BSN [36], MGG [37] and BMN [34] regress the time of activity boundaries, showing state-of-the-art performances on both ActivityNet 1.3 [23] and Thumos'14 [29]. Alternatively, ActionSearch [4] tackles the spotting task iteratively, learning to predict which frame to visit next in order to spot a given activity. However, this method requires sequences of temporal annotations by human annotators to train the models that are not readily available for datasets outside ActivityNet. Also, Alwassel *et al*. [3] define an action spot as positive as soon as it lands within the boundary of an activity, which is less constraining than the action spotting defined in SoccerNet [21].

Recently, Sigurdsson *et al*. [52] question boundaries sharpness and show that human agreement on temporal boundaries reach an average tIoU of 72.5% for Charades [53] and 58.7% on MultiTHUMOS[64]. Alwassel *et al*. [3] confirm such disparity on ActivityNet [23], but also show that it does not constitute a major roadblock to progress in the field. Different from activity localization, SoccerNet [21] proposes an alternative action spotting task for soccer action understanding, leveraging a well-defined set of soccer rules that define a single temporal anchor per action. In our work, we improve the SoccerNet [21] action spotting baseline by introducing a novel context-aware loss that temporally slices the vicinity of the action spots. Also, we integrate our loss for generic activity localization and detection on a boundary-based method [34, 36].

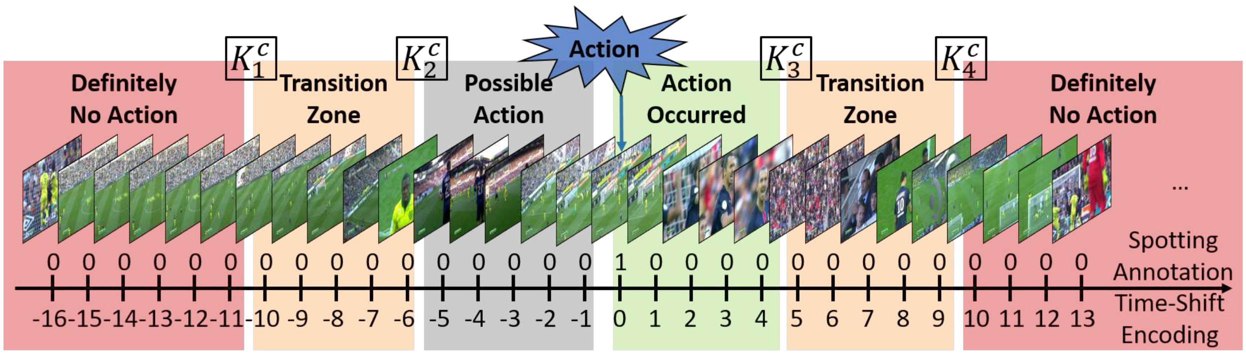

Figure 2. **Action context slicing.** We define six temporal segments around each ground-truth action spot, each of which induces a specific behavior in our context-aware loss function when training the network. **Far before** and **far after** the action, its influence is negligible, thus we train the network not to predict an action. **Just before** the action, we do not influence the network since a particular context may or may not result in an action (*i.e.* an attacking phase *may* lead to a goal). **Just after** the action, its contextual information is rich and unambiguous as the action has just occurred (*i.e.* a goal *leads* to celebrating). Hence, we train the network to predict an action. Finally, we define **transition zones** for our loss function to be smooth, in which we softly train the network not to predict an action. For each class c, the temporal segments are delimited by specific slicing parameters K_i^c and are materialized through our time-shift encoding, which contains richer temporal context information about the action than the initial binary spotting annotation.

3. Methodology

We address the action spotting task by developing a context-aware loss for a temporal segmentation module, and a YOLO-like loss for an action spotting module that outputs the *spotting predictions* of the network. We first present the re-encoding of the annotations needed for the segmentation and spotting tasks, then we explain how the losses of these modules are computed based on the re-encodings.

Problem definition. We denote by C the number of classes of the action spotting problem. Each action is identified by a single *action frame* annotated as such. Each frame of a given video is annotated with either a one-hot encoded vector with C components for the action frames or a vector of C zeros for the background frames. We denote by N_F the number of frames in a video.

3.1. Encoding

To train our network, the initial annotations are re-encoded in two different ways: with a *time-shift encoding* used for the temporal segmentation loss, and with a *YOLO-like encoding* used for the action spotting loss.

Time-shift encoding (TSE) for temporal segmentation. We slice the temporal context around each action into segments related to their distance from the action, as shown in Figure 2. The segments regroup frames that are either *far before*, *just before*, *just after*, *far after* an action, or in *transition zones* between these segments.

We use the segments in our temporal segmentation module so that its *segmentation scores* reflect the following ideas. **(1)** *Far before* an action spot of some class, we cannot foresee its occurrence. Hence, the score for that class should indicate that no action is occurring. **(2)** *Just before* an action, its occurrence is uncertain. Therefore, we do not influence the score towards any particular direction. **(3)** *Just after* an action has happened, plenty of visual cues allow for the detection of the occurrence of the action. The score for its class should reflect the presence of the action. **(4)** *Far after* an action, the score for its class should indicate that it is not occurring anymore. The segments around the actions of class c are delimited by four temporal *context slicing parameters* $K_1^c < K_2^c < 0 < K_3^c < K_4^c$ as shown in Figure 2.

The context slicing is used to perform a *time-shift encoding* (TSE) of each frame x of a video with a vector of length C, containing class-wise information on the relative location of x with respect to its closest past or future actions. The TSE of x for class c, noted $s^c(x)$, is the time-shift (*i.e.* difference in frame indices) of x from either its closest past or future ground-truth action of class c, depending on which has the dominant influence on x. We set $s^c(x)$ as the time-shift from the past action if either **(i)** x is just after the past action; or **(ii)** x is in the transition zone after the past action, but is far before the future action; or **(iii)** x is in the transition zones after the past and before the future actions while being closer to the past action. In all other cases, $s^c(x)$ is the time-shift from the future action.

If x is both located *far after* the past action and *far before* the future action, selecting either of the two time-shifts has the same effect in our loss. Furthermore, for the frames located either before the first or after the last annotated action of class c, only one time-shift can be computed and is thus set as $s^c(x)$. Finally, if no action of class c is present in the video, then we set $s^c(x) = K_1^c$ for all the frames. This induces the same behavior in our loss as if they were all located far before their closest future action.

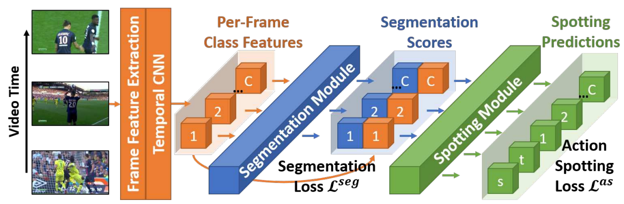

Figure 3. **Pipeline for action spotting.** We propose a network made of a **frame feature extractor** and a **temporal CNN** outputting C class feature vectors per frame, a **segmentation module** outputting per-class segmentation scores, and a **spotting module** extracting $2+C$ values per spotting prediction (*i.e.* the confidence score s for the spotting, its location t and a per-class prediction).

YOLO-like encoding for action spotting. Inspired by YOLO [45], each ground-truth action of the video engenders an *action vector* composed of $2 + C$ values. The first value is a binary indicator of the presence ($= 1$) of the action. The second value is the location of the frame annotated as the action, computed as the index of that frame divided by N_F. The remaining C values represent the one-hot encoding of the action. We encode a whole video containing N_{GT} actions in a matrix $\mathbf{Y}$ of dimension $N_{\text{GT}} \times (2 + C)$, with each line representing an action vector of the video.

3.2. Loss and Network Design

Temporal segmentation loss. The TSE parameterizes the temporal segmentation loss described below. For clarity, we denote by p the segmentation score for a frame x to belong to class c output by the segmentation module, and s as the TSE of x for class c. We detail the loss generated by p in this setting, noted $L(p, s)$. First, in accordance with Figure 2, we compute $L(p, s)$ as follows:

$$L(p,s) = \begin{cases} -\ln(1-p) & s \leq K_1^c \quad (1) \\ -\ln\left(1 - \dfrac{K_2^c - s}{K_2^c - K_1^c}\, p\right) & K_1^c < s \leq K_2^c \quad (2) \\ 0 & K_2^c < s < 0 \quad (3) \\ -\ln\left(\dfrac{s}{K_3^c} + \dfrac{K_3^c - s}{K_3^c}\, p\right) & 0 \leq s < K_3^c \quad (4) \\ -\ln\left(1 - \dfrac{s - K_3^c}{K_4^c - K_3^c}\, p\right) & K_3^c \leq s < K_4^c \quad (5) \\ -\ln(1-p) & s \geq K_4^c. \quad (6) \end{cases}$$

Then, following the practice in [14, 48] to help the network focus on improving its worst segmentation scores, we zero out the loss for scores that are satisfying enough. In the case of Equation (4) when $s = 0$, we say that a score is satisfactory when it exceeds some *maximum margin* τ_{max}. In the cases of Equations (1) and (6), we say that a score is satisfactory when it is lower than some *minimum margin* τ_{min}. The range of values for p that leads to zeroing out the loss varies with s and the slicing parameters in most cases. This is achieved by revising $L(p, s)$ as in Equations (7) and (8). Figure 1 shows a representation of $\tilde{L}(p, s)$.

$$\tilde{L}(p,s) = \begin{cases} \max(0, L(p,s) + \ln(\tau_{\text{max}})) & 0 \leq s < K_3^c \quad (7) \\ \max(0, L(p,s) + \ln(1 - \tau_{\text{min}})) & \text{otherwise.} \quad (8) \end{cases}$$

Finally, the segmentation loss $\mathcal{L}^{\text{seg}}$ for a given video of frames $x_1, \ldots, x_{N_F}$ is given in Equation (9).

$$\mathcal{L}^{\text{seg}} = \frac{1}{C\ N_F} \sum_{i=1}^{N_F} \sum_{c=1}^{C} \tilde{L}(p^c(x_i), s^c(x_i)) \quad (9)$$

Action spotting loss. Let N_{pred} be a fixed number of action spotting predictions generated by our network for each video. Those predictions are encoded in $\hat{\mathbf{Y}}$ of dimension $N_{\text{pred}} \times (2 + C)$, similarly to $\mathbf{Y}$.

We leverage an iterative one-to-one matching algorithm to pair each of the N_{GT} ground-truth actions with a prediction. First, we match each ground-truth location of $\mathbf{Y}_{\cdot,2}$ with its closest predicted location in $\hat{\mathbf{Y}}_{\cdot,2}$, and vice-versa (*i.e.* we match the predicted locations with their closest ground-truth locations). Next, we form pairs of $(\text{ground-truth}, \text{predicted})$ locations that reciprocally match, we remove them from the process, and we iterate until all ground truths are coupled with a prediction. Consequently, we build $\hat{\mathbf{Y}}^M$ as a reorganized version of the actions encoded in $\hat{\mathbf{Y}}$, such that $\mathbf{Y}_{i,2}$ and $\hat{\mathbf{Y}}^M_{i,2}$ reciprocally match for all $i \leq N_{\text{GT}}$.

We define the action spotting loss $\mathcal{L}^{\text{as}}$ in Equation (10). It corresponds to a weighted sum of the squared errors between the matched predictions and a regularization on the confidence score of the unmatched predictions.

$$\mathcal{L}^{\text{as}} = \sum_{i=1}^{N_{\text{GT}}} \sum_{j=1}^{2+C} \alpha_j \left(\mathbf{Y}_{i,j} - \hat{\mathbf{Y}}^M_{i,j}\right)^2 + \beta \sum_{i=N_{\text{GT}}+1}^{N_{\text{pred}}} \left(\hat{\mathbf{Y}}^M_{i,1}\right)^2 \quad (10)$$

Complete loss. The final loss $\mathcal{L}$ is presented in Equation (11) as a weighted sum of $\mathcal{L}^{\text{seg}}$ and $\mathcal{L}^{\text{as}}$.

$$\mathcal{L} = \mathcal{L}^{\text{as}} + \lambda^{\text{seg}}\, \mathcal{L}^{\text{seg}} \quad (11)$$

Network for action spotting. The architecture of the network is illustrated in Figure 3 and further detailed in the **supplementary material**. We leverage frame feature representations for the videos (*e.g.* ResNet) provided with the dataset, embodied as the output of the frame feature extractor of Figure 3. The temporal CNN of Figure 3 is composed of a spatial two-layer MLP, followed by four multi-scale 3D convolutions (*i.e.* across time, features and classes). The temporal CNN outputs a set of $C \times f$ features for each frame organized in C feature vectors (one per class) of size f, as in [48]. These features are input into a segmentation module, in which we use Batch Normalization [26] and sigmoid activations. The closeness of the C vectors obtained in this way to a pre-defined vector gives the C segmentation scores output by the segmentation module, as [14]. The $C \times f$ features obtained previously are concatenated with the C scores and fed to the action spotting module, as shown in Figure 3. It is composed of three successive temporal max-pooling and 3D convolutions, and outputs N_{pred} vectors of dimension $(2 + C)$. The first two elements of these vectors are sigmoid-activated, the C last are softmax-activated. The activated vectors are stacked to produce the prediction $\hat{\mathbf{Y}}$ of dimension $N_{\text{pred}} \times (2 + C)$ for the action spotting task.

4. Experiments

We evaluate our new context-aware loss function in two scenarios: the action spotting task of SoccerNet [21], and activity localization and detection tasks on ActivityNet [23].

4.1. Experiments on SoccerNet

Data. Three classes of action are annotated in SoccerNet by Giancola *et al*. [21]: goals, cards, and substitutions, so $C = 3$ in this case. They identify each action by one annotated frame: the moment the ball crosses the line for *goal*, the moment the referee shows a player a card for *card*, and the moment a new player enters the field for *substitution*. We train our network on the frame features already provided with the dataset. Giancola *et al*. first subsampled the raw videos at 2 fps, then they extracted the features with a backbone network and reduced them by PCA to 512 features for each frame of the subsampled videos. Three sets of features are provided, each extracted with a particular backbone network: I3D [11], C3D [58], and ResNet [22].

Action spotting metric. We measure performances with the action spotting metric introduced in SoccerNet [21]. An action spot is defined as positive if its temporal offset from its closest ground truth is less than a given tolerance δ. The average precision (AP) is estimated based on Precision-Recall curves, then averaged between classes (mAP). An Average-mAP is proposed as the AUC of the mAP over different tolerances δ ranging from 5 to 60 seconds.

Experimental setup. We train our network on batches of *chunks*. We define a chunk as a set of N_F contiguous frame feature vectors. We set $N_F = 240$ to maintain a high training speed while retaining sufficient contextual information. This size corresponds to a clip of 2 minutes of raw video. A batch contains chunks extracted from a single raw video. We extract a chunk around each ground-truth action, such that the action is randomly located within the chunk. Then, to balance the batch, we randomly extract N_{GT}/C chunks composed of background frames only. An epoch ends when the network has been trained on one batch per training video. At each epoch, new batches are re-computed for each video for data augmentation purposes. Each raw video is time-shift encoded before training. Each new training chunk is encoded with the YOLO-like encoding.

Method	Frame features		
	I3D	C3D	ResNet
SoccerNet baseline 5s [21]	-	-	34.5
SoccerNet baseline 60s [21]	-	-	40.6
SoccerNet baseline 20s [21]	-	-	49.7
Vats *et al*. [62]	-	-	57.5
Ours	53.6	57.7	**62.5**

Table 1. **Results on SoccerNet.** Average-mAP (in %) on the test set of SoccerNet for the action spotting task. We establish a new state-of-the-art performance.

The number of action spotting predictions generated by the network is set to $N_{\text{pred}} = 5$, as we observed that no chunks of 2 minutes of raw video contain more than 5 actions. We train the network during 1000 epochs, with an initial learning rate $lr = 10^{-3}$ linearly decreasing to 10^{-6}. We use Adam as the optimizer with default parameters [33].

For the segmentation loss, we set the margins $\tau_{\text{max}} = 0.9$ and $\tau_{\text{min}} = 0.1$ in Equations (7) and (8), following the practice in [48]. For the action spotting loss in Equation (10), we set $\alpha_j = 1$ for $j \neq 2$, while α_2 is optimized (see below) to find an appropriate weighting for the location components of the predictions. Similarly, β is optimized to find the balance between the loss of the action vectors and the regularization of the remaining predictions. For the final loss in Equation (11), we optimize λ^{seg} to find the balance between the two losses.

Hyperparameter optimization. For each set of features (I3D, C3D, ResNet), we perform a joint Bayesian optimization [1] on the number of frame features f extracted per class, on the temporal receptive field r of the network (*i.e.* temporal kernel dimension of the 3D convolutions), and on the parameters $\alpha_2, \beta, \lambda^{\text{seg}}$. Next, we perform a grid search optimization on the slicing parameters K_i^c.

For ResNet, we obtain $f = 16$, $r = 80$, $\alpha_2 = 5$, $\beta = 0.5$, $\lambda^{\text{seg}} = 1.5$. For goals (resp. cards, substitutions) we have $K_1 = -40$ (resp. $-40, -80$), $K_2 = -20$ (resp. $-20, -40$), $K_3 = 120$ (resp. $20, 20$), and $K_4 = 180$ (resp. $40, 40$). Given the framerate of 2 fps, those values can be translated to seconds by scaling them down by a factor of 2. The value $r = 80$ corresponds to a temporal receptive field of 20 seconds on both sides of the central frame in the temporal dimension of the 3D convolutions.

Main results. The performances obtained with the optimized parameters are reported in Table 1. As shown, we establish a new state-of-the-art performance on the action spotting task of SoccerNet, outperforming the previous benchmark by a comfortable margin, for all the frame features. ResNet gives the best performance, as also observed in [21]. A sensitivity analysis of the parameters K_i^c reveals robust performances around the optimal values, indicating that no heavy fine-tuning is required for the context slicing. Also, performances largely decrease as the slicing is strongly reduced, which emphasizes its usefulness.

Ablation study. Since the ResNet features provide the best performance, we use them with their optimized parameters for the following ablation studies. **(i)** We remove the segmentation module, which is equivalent to setting $\lambda^{\text{seg}} = 0$ in Equation (11). This also removes the context slicing and the margins τ_{max} and τ_{min}. **(ii)** We remove the action context slicing such that the ground truth for the segmentation module is the raw binary annotations, *i.e.* all the frames must be classified as background except the action frames. This is equivalent to setting $K_1 = -1 = K_2 = -K_3 = -K_4$. **(iii)** We remove the margins that help the network focus on improving its worst segmentation scores, by setting $\tau_{\text{max}} = 1$, $\tau_{\text{min}} = 0$ in Equations (7) and (8). **(iv)** We remove the iterative one-to-one matching between the ground truth $\mathbf{Y}$ and the predictions $\hat{\mathbf{Y}}$ before the action spotting loss, which is equivalent to using $\hat{\mathbf{Y}}$ instead of $\hat{\mathbf{Y}}^M$ in Equation (10). The results of the ablation studies are shown in Table 2.

From an Average-mAP perspective, the auxiliary task of temporal segmentation improves the performance on the action spotting task (from 58.9% to 62.5%), which is a common observation in multi-task learning [66]. When the segmentation is performed, our temporal context slicing gives a significant boost compared to using the raw binary annotations (from 57.8% to 62.5%). This observation is in accordance with the sensitivity analysis. It also appears that it is preferable to not use the segmentation at all rather than using the segmentation with the raw binary annotations (58.9% vs 57.8%), which further underlines the usefulness of the context slicing. A boost in performance is also observed when we use the margins to help the network focus on improving its worst segmentation scores (from 59.0% to 62.5%). Eventually, Table 2 shows that it is extremely beneficial to match the predictions of the network with the ground truth before the action spotting loss (from 46.8% to 62.5%). This makes sense since there is no point in evaluating the network on its ability to order its predictions, which is a hard and unnecessary constraint. The large impact of the matching is also justified by its direct implication in the action spotting task assessed through the Average-mAP.

Results through game time. In soccer, it makes sense to analyze the performance of our model through game time, since the actions are not uniformly distributed throughout the game. For example, a substitution is more likely to occur during the second half of a game. We consider non-overlapping bins corresponding to 5 minutes of game time and compute the Average-mAP for each bin. Figure 4 shows the evolution of this metric through game time.

	Segm.	Slic.	Marg.	Match.	Result
(i)				✓	58.9
(ii)	✓		✓	✓	57.8
(iii)	✓	✓		✓	59.0
(iv)	✓	✓	✓		46.8
Ours	✓	✓	✓	✓	**62.5**

Table 2. **Ablation study.** We perform ablations by **(i)** removing the segmentation ($\lambda^{\text{seg}} = 0$), hence the slicing and the margins; **(ii)** removing the context slicing ($K_1 = -1 = K_2 = -K_3 = -K_4$); **(iii)** removing the margins that help the network focus on improving its worst segmentation scores ($\tau_{\text{min}} = 0$, $\tau_{\text{max}} = 1$); **(iv)** removing the matching (using $\hat{\mathbf{Y}}$ instead of $\hat{\mathbf{Y}}^M$ in $\mathcal{L}^{\text{as}}$). Each part evidently contributes to the overall performance.

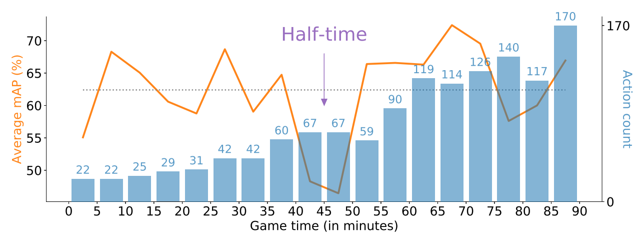

Figure 4. **Performance as function of game time.** **Average-mAP** spotting performance over the game time with all ground-truth actions of the dataset binned in 5 minute intervals. It appears that actions around the half-time break are more challenging to spot. **Number of actions** for each bin. **Our performance** (62.5%).

It appears that actions occurring during the first five minutes of a half-time are substantially more difficult to spot than the others. This may be partially explained by the occurrence of some of these actions at the very beginning of a half-time, for which the temporal receptive field of the network requires the chunk to be temporally padded. Hence, some information may be missing to allow the network to spot those actions. Besides, when substitutions occur during the break, they are annotated as such on the first frame of the second halves of the matches, which makes them practically impossible to spot. In the test set, this happens for 28% of the matches. None of these substitutions are spotted by our model, which thus degrades the performances during the first minutes of play in the second halves of the matches. However, they merely represent 5% of all the substitutions, and removing them from the evaluation only boosts our Average-mAP by 0.7% (from 62.5% to 63.2%).

Results as function of action vicinity. We investigate whether actions are harder to spot when they are close to each other. We bin the ground-truth actions based on the distance that separates them from the previous (or next, depending on which is the closest) ground-truth action, regardless of their classes. Then, we compute the Average-mAP for each bin. The results are represented in Figure 5.

We observe that the actions are more difficult to spot

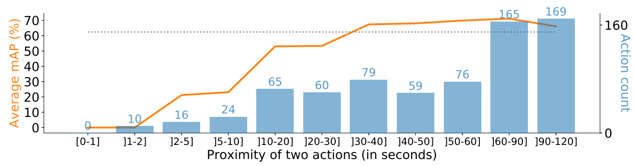

Figure 5. **Performance as function of action vicinity.** **Average-mAP** spotting performance per bin of ground-truth actions grouped by distance (in seconds) from their closest ground-truth action. It appears that nearby actions are more challenging to spot. **Number of actions** for each bin. **Our performance** (62.5%).

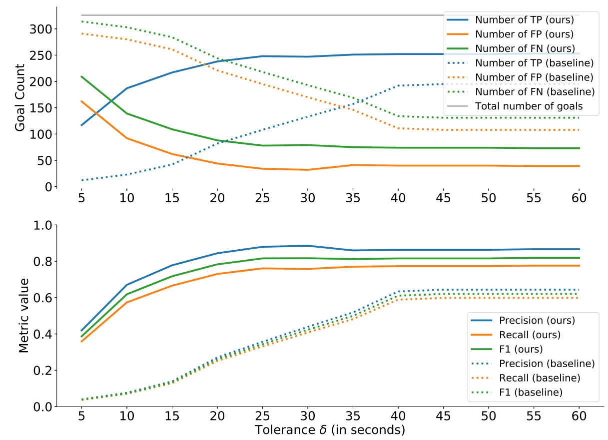

Figure 6. **Per-class results (goals).** A prediction of class *goal* is a **true positive (TP)** with tolerance δ when it is located at most $\delta/2$ seconds from a ground-truth goal. The baseline results are obtained from the best model of [21]. Our model spots most goals within 10 seconds around the ground truth ($\delta = 20$ seconds).

when they are close to each other. This could be due to the reduced number of visual cues, such as replays, when an action occurs rapidly after another and thus must be broadcast. Some confusion may also arise because the replays of the first action can still be shown after the second action, *e.g*. a sanctioned foul followed by a converted penalty. This analysis also shows that the action spotting problem is challenging even when the actions are further apart, as the performances in Figure 5 eventually plateau.

Per-class results. We perform a per-class analysis in a similar spirit as the Average-mAP metric. For a given class, we fix a tolerance δ around each annotated action to determine positive predictions and we aggregate these results in a confusion matrix. An action is considered spotted when its confidence score exceeds some threshold optimized for the F_1 score on the validation set. From the confusion matrix, we compute the precision, recall and F_1 score for that class and for that tolerance δ. Varying δ from 5 to 60 seconds provides the evolution of the three metrics as a function of the tolerance. Figure 6 shows these curves for *goals* for our model and for the baseline [21]. The results for cards and substitutions are provided in **supplementary material**.

Figure 6 shows that most goals can be efficiently spotted by our model within 10 seconds around the ground truth ($\delta = 20$ seconds). We achieve a precision of 80% for that tolerance. The previous baseline plateaus within 20 seconds ($\delta = 40$ seconds) and still has a lower performance. In particular for goals, many visual cues facilitate their spotting, *e.g*. multiple replays, particular camera views, or celebrations from the players and from the public.

4.2. Experiments on ActivityNet

In this section, we evaluate our context-aware loss in a more generic task than action spotting in soccer videos. We tackle the *Activity Proposal* and *Activity Detection* tasks of the challenging ActivityNet dataset, for which we use the ResNet features provided with the dataset at 5 fps.

Setup. We use the current state-of-the-art network, namely BMN [34], with the code provided in [2]. BMN is equipped with a temporal evaluation module (TEM), which plays a similar role as our temporal segmentation module. We replace the loss associated with the TEM by our novel temporal segmentation loss $\mathcal{L}^{\text{seg}}$. The slicing parameters are set identically for all the classes and are optimized with respect to the AUC performance on the validation set by grid search with the constraint $K_1 = 2K_2 = -2K_3 = -K_4$. The optimization yields the best results where $K_1 = -14$.

Results. The average performances on 20 runs of our experiment and of the BMN base code [2] are reported in Table 3. Our novel temporal segmentation loss improves the performance obtained with BMN [2] by 0.15% and 0.12% for the activity proposal task (AR@100 and AUC) and by 0.38% for the activity detection task (Average-mAP). These increases compare with some recent increments, while being obtained just by replacing their TEM loss by our context-aware segmentation loss. The network thus has the same architecture and number of parameters. We conjecture that our loss $\mathcal{L}^{\text{seg}}$, through its particular context slicing, helps train the network by modelling the uncertainty surrounding the annotations. Indeed, it has been shown in [3, 52] that a large variability exists among human annotators on which frames to annotate as the beginning and the end of the activities of the dataset. Let us note that in BMN, the TEM loss is somehow adapted around the action frames in order to mitigate the penalization attributed to their neighboring frames. Our work goes one step further, by directly designing a temporal context-aware segmentation loss.

5. Automatic Highlights Generation for Soccer

Some action spotting and temporal segmentation results are shown in Figure 7. It appears that some sequences of play have a high segmentation score for some classes but do not lead, quite rightly, to an action spotting. It turns

Method	AR@100	AUC	Average-mAP
Lin *et al.* [35]	73.01	64.40	29.17
Gao *et al.* [19]	73.17	65.72	-
BSN [36]	74.16	66.17	30.03
P-GCN [68]	-	-	31.11
BMN [34]	75.01	67.10	**33.85**
BMN code [2]	75.11	67.16	30.67 ± 0.08
Ours: [2] + $\mathcal{L}^{\text{seg}}$	**75.26**	**67.28**	31.05 ± 0.07

Table 3. **Results on ActivityNet** validation set for the proposal task (AR@100, AUC) and for the detection task (Average-mAP). For our experiments, we report the average values on 20 runs.

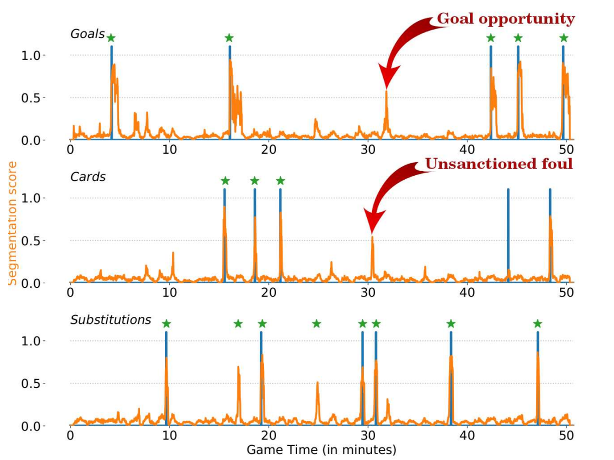

Figure 7. **Action spotting and segmentation** for the 2nd half of the "Remuntada" FCB - PSG. **Ground truth actions**, **temporal segmentation curves**, and **spotting results** are illustrated. We can identify **unannotated interesting actions** using our segmentation.

out that these sequences are often related to unannotated actions of supplementary classes that resemble those considered so far, such as *unconverted goal opportunities* and *unsanctioned fouls*. Video clips of the two actions identified in Figure 7 are provided in the **supplementary material**.

To quantify the spotting results of goal opportunities, we can only compute the precision metric since these actions are not annotated. We manually inspect each video sequence of the test set where the segmentation score for goals exceeds some threshold η but where no ground-truth goal is present. We decide whether the sequence is a goal opportunity or not by asking two frequent observers of soccer games if they would include it in the highlights of the match. The sequence is a true positive when they both agree to include it and a false positive, otherwise. The precision is then computed for that η. By gradually decreasing η from 0.9 to 0.3, we obtain the precision curve shown in Figure 8. It appears that 80% of the sequences with a segmentation score larger than $\eta = 0.5$ are considered goal opportunities.

As a direct by-product, we derive an automatic highlights generator without explicit supervision. We extract

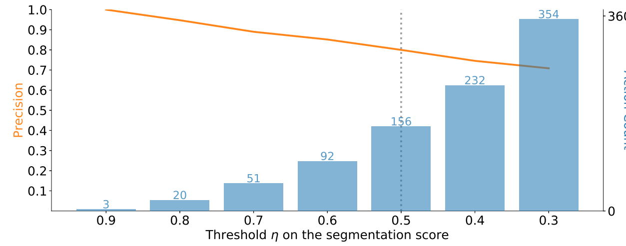

Figure 8. **Precision for goal opportunities**, as a function of the threshold on the segmentation score to exceed for manually inspecting a sequence. For scores larger than $\eta = 0.5$, a **precision** of 0.8 is achieved, *i.e*. 80% of the sequences inspected were goal opportunities. **Number of sequences** inspected per threshold.

a video clip starting 15 seconds before each spotting of a *goal* or a *card* and ending 20 seconds after. We proceed likewise for the sequences with a segmentation score ≥ 0.5 for *goals*. We dismiss substitutions as they rarely appear in highlights. We assemble the clips chronologically to produce the highlights video, provided in **supplementary material**. Evaluating its quality is subjective, but we found its content to be adequate, even if the montage could be improved. Indeed, only sequences where a goal, a goal opportunity, or a foul occurs are selected. This reinforces the usefulness of the segmentation, as it provides a direct overview of the proceedings of the match, including proposals for unannotated actions that are interesting for highlights.

6. Conclusion

We tackle the challenging action spotting task of SoccerNet with a novel context-aware loss for segmentation and a YOLO-like loss for the spotting. The former treats the frames according to their time-shift from their closest ground-truth actions. The latter leverages an iterative matching algorithm that alleviates the need for the network to order its predictions. To show generalization capabilities, we also test our context-aware loss on ActivityNet. We improve the state-of-the-art on ActivityNet by 0.15% in AR@100, 0.12% in AUC, and 0.38% in Average-mAP, by only including our context-aware loss without changing the network architecture. We achieve a new state-of-the art on SoccerNet, surpassing by far the previous baseline (from 49.7% to 62.5% in Average-mAP) and spotting most actions within 10 seconds around their ground truth. Finally, we leverage the resulting segmentation results to identify unannotated actions such as goal opportunities and derive a highlights generator without specific supervision.

Acknowledgments. This work is supported by the DeepSport project of the Walloon region and the FRIA (Belgium), as well as the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-CRG2017-3405.

References

[1] Bayesian Optimization. https://github.com/fmfn/BayesianOptimization. Last accessed: 2019-10-20. 5

[2] Code for BMN. https://github.com/JJBOY/BMN-Boundary-Matching-Network. Last accessed: 2019-10-30. 7, 8

[3] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *European Conference on Computer Vision (ECCV)*, September 2018. 2, 7

[4] Humam Alwassel, Fabian Caba Heilbron, and Bernard Ghanem. Action Search: Spotting Targets in Videos and Its Application to Temporal Action Localization. In *European Conference on Computer Vision (ECCV)*, September 2018. 2

[5] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Action classification in soccer videos with long short-term memory recurrent neural networks. In *International Conference on Artificial Neural Networks (ICANN)*, September 2010. 2

[6] Vinay Bettadapura, Caroline Pantofaru, and Irfan Essa. Leveraging contextual cues for generating basketball highlights. In *ACM international conference on Multimedia (ACM-MM)*, October 2016. 2

[7] Lewis Bridgeman, Marco Volino, Jean-Yves Guillemaut, and Adrian Hilton. Multi-Person 3D Pose Estimation and Tracking in Sports. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1

[8] Shyamal Buch, Victor Escorcia, Bernard Ghanem, Li Fei-Fei, and Juan Carlos Niebles. End-to-End, Single-Stream Temporal Action Detection in Untrimmed Videos. In *British Machine Vision Conference (BMVC)*, September 2017. 2

[9] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. SST: Single-Stream Temporal Action Proposals. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[10] Zixi Cai, Helmut Neher, Kanav Vats, David A. Clausi, and John Zelek. Temporal hockey action recognition via pose and optical flows. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 2

[11] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 5

[12] Anthony Cioppa, Adrien Deliege, Maxime Istasse, Christophe De Vleeschouwer, and Marc Van Droogenbroeck. ARTHuS: Adaptive Real-Time Human Segmentation in Sports Through Online Distillation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1

[13] Anthony Cioppa, Adrien Deliege, and Marc Van Droogenbroeck. A Bottom-Up Approach Based on Semantics for the Interpretation of the Main Camera Stream in Soccer Games. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 2

[14] Adrien Deliège, Anthony Cioppa, and Marc Van Droogenbroeck. HitNet: a neural network with capsules embedded in a Hit-or-Miss layer, extended with hybrid data augmentation and ghost capsules. *CoRR*, abs/1806.06519, 2018. 4, 5

[15] Deloitte. Market size of the European football market from 2006/07 to 2015/16 (in billion euros), 2017. Retrieved October 30, 2019, from https://www.statista.com/statistics/261223/european-soccer-market-total-revenue/. 1

[16] Ahmet Ekin, A Murat Tekalp, and Rajiv Mehrotra. Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing*, 12(7):796–807, 2003. 2

[17] Dirk Farin, Susanne Krabbe, Wolfgang Effelsberg, et al. Robust camera calibration for sport videos using court models. In *Storage and Retrieval Methods and Applications for Multimedia 2004*, volume 5307, pages 80–91. International Society for Optics and Photonics, 2003. 1

[18] Panna Felsen, Pulkit Agrawal, and Jitendra Malik. What will happen next? Forecasting player moves in sports videos. In *IEEE International Conference on Computer Vision (ICCV)*, October 2017. 1

[19] Jiyang Gao, Kan Chen, and Ram Nevatia. CTAP: Complementary Temporal Action Proposal Generation. In *European Conference on Computer Vision (ECCV)*, September 2018. 8

[20] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. TURN TAP: Temporal unit regression network for temporal action proposals. In *IEEE International Conference on Computer Vision (ICCV)*, October 2017. 2

[21] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. SoccerNet: A Scalable Dataset for Action Spotting in Soccer Videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 2, 5, 6, 7

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 5

[23] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2, 5

[24] Namdar Homayounfar, Sanja Fidler, and Raquel Urtasun. Sports field localization via deep structured models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1

[25] Chung-Lin Huang, Huang-Chia Shih, and Chung-Yuan Chao. Semantic analysis of soccer video using dynamic Bayesian network. *IEEE Transactions on Multimedia*, 8(4):749–760, 2006. 2

[26] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning (ICML)*, July 2015. 5

[27] Maxime Istasse, Julien Moreau, and Christophe De Vleeschouwer. Associative Embedding for Team Discrimination. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1

[28] Haohao Jiang, Yao Lu, and Jing Xue. Automatic Soccer Video Event Detection Based on a Deep Neural Network Combined CNN and RNN. In *International Conference on Tools with Artificial Intelligence (ICTAI)*, November 2016. 2

[29] Yu-Gang. Jiang, Jingen Liu, Amir Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. THUMOS Challenge: Action Recognition with a Large Number of Classes. `http://crcv.ucf.edu/THUMOS14/`, 2014. 2

[30] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale Video Classification with Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2

[31] A.T. Kearney. Global sports market - total revenue from 2005 to 2017 (in billion U.S. dollars), 2014. Retrieved 2019-10-30 from https://www.statista.com/statistics/370560/worldwide-sports-market-revenue/. 1

[32] Abdullah Khan, Beatrice Lazzerini, Gaetano Calabrese, and Luciano Serafini. Soccer Event Detection. In *International Conference on Image Processing and Pattern Recognition (IPPR)*, April 2018. 2

[33] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, May 2015. 5

[34] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In *IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2, 7, 8

[35] Tianwei Lin, Xu Zhao, and Zheng Shou. Temporal Convolution Based Action Proposal: Submission to ActivityNet 2017. *CoRR*, abs/1707.06750, 2017. 8

[36] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. BSN: Boundary Sensitive Network for Temporal Action Proposal Generation. In *European Conference on Computer Vision (ECCV)*, September 2018. 2, 8

[37] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity Generator for Temporal Action Proposal. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[38] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian Temporal Awareness Networks for Action Localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[39] Jikai Lu, Jianhui Chen, and James J. Little. Pan-tilt-zoom SLAM for Sports Videos. In *British Machine Vision Conference (BMVC)*, September 2019. 1

[40] Mehrtash Manafifard, Hamid Ebadi, and Hamid Abrishami Moghaddam. A survey on player tracking in soccer videos. *Computer Vision and Image Understanding*, 159:19–46, 2017. 1

[41] William McNally, Kanav Vats, Tyler Pinto, Chris Dulhanty, John McPhee, and Alexander Wong. GolfDB: A Video Database for Golf Swing Sequencing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 2

[42] Thomas B. Moeslund, Graham Thomas, and Adrian Hilton. *Computer Vision in Sports*. Springer, 2014. 2

[43] AJ Piergiovanni and Michael S. Ryoo. Fine-Grained Activity Recognition in Baseball Videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 2

[44] Vignesh Ramanathan, Jonathan Huang, Sami Abu-El-Haija, Alexander Gorban, Kevin Murphy, and Li Fei-Fei. Detecting events and key actors in multi-person videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[45] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 4

[46] Ren Reede and Jose Joemon. Football video segmentation based on video production strategy. In *European Conference on Information Retrieval (ECIR)*, March 2005. 2

[47] Konstantinos Rematas, Ira Kemelmacher-Shlizerman, Brian Curless, and Steve Seitz. Soccer on Your Tabletop. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[48] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic Routing Between Capsules. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, December 2017. 4, 5

[49] Melissa Sanabria, Frédéric Precioso, Thomas Menguy, et al. A Deep Architecture for Multimodal Summarization of Soccer Games. In *ACM International Conference on Multimedia (ACM-MM) Workshops*, October 2019. 2

[50] Saikat Sarkar, Amlan Chakrabarti, and Dipti Prasad Mukherjee. Generation of Ball Possession Statistics in Soccer Using Minimum-Cost Flow Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1

[51] Pushkar Shukla, Hemant Sadana, Apaar Bansal, Deepak Verma, Carlos Elmadjian, Balasubramanian Raman, and Matthew Turk. Automatic Cricket Highlight Generation Using Event-Driven and Excitement-Based Features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 2

[52] Gunnar A. Sigurdsson, Olga Russakovsky, and Abhinav Gupta. What actions are needed for understanding human actions in videos? In *IEEE International Conference on Computer Vision (ICCV)*, October 2017. 2, 7

[53] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision (ECCV)*, October 2016. 2

[54] Statista. Computer vision artificial intelligence (AI) market revenues worldwide, from 2015 to 2019, by application (in million U.S. dollars), 2016. Retrieved October 30, 2019, from https://www.statista.com/statistics/641922/worldwide-artificial-intelligence-computer-vision-market-revenues/. 1

[55] Mostafa Tavassolipour, Mahmood Karimian, and Shohreh Kasaei. Event detection and summarization in soccer videos using Bayesian network and copula. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(2):291–304, 2014. 2

[56] Rajkumar Theagarajan, Federico Pala, Xiu Zhang, and Bir Bhanu. Soccer: Who Has the Ball? Generating Visual Analytics and Player Statistics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 1

[57] Graham Thomas, Rikke Gade, Thomas B. Moeslund, Peter Carr, and Adrian Hilton. Computer vision for sports: Current applications and research topics. *Computer Vision and Image Understanding*, 159:3–18, 2017. 1, 2

[58] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In *IEEE International Conference on Computer Vision (ICCV)*, December 2015. 5

[59] Grigorios Tsagkatakis, Mustafa Jaber, and Panagiotis Tsakalides. Goal!! Event detection in sports video. *Journal of Electronic Imaging*, 2017(16):15–20, 2017. 2

[60] Takamasa Tsunoda, Yasuhiro Komori, Masakazu Matsugu, and Tatsuya Harada. Football Action Recognition Using Hierarchical LSTM. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 2

[61] Francesco Turchini, Lorenzo Seidenari, Leonardo Galteri, Andrea Ferracani, Giuseppe Becchi, and Alberto Del Bimbo. Flexible Automatic Football Filming and Summarization. In *ACM International Conference on Multimedia (ACM-MM) Workshops*, October 2019. 2

[62] Kanav Vats, Mehrnaz Fani, Pascale Walters, David A. Clausi, and John Zelek. Event detection in coarsely annotated sports videos via 1d temporal convolutions, 2019. Preprint at `https://bit.ly/3b4TiTf`. 5

[63] Ying Yang and Danyang Li. Robust player detection and tracking in broadcast soccer video based on enhanced particle filter. *Journal of Visual Communication and Image Representation*, 46:81–94, 2017. 1

[64] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, 126(2-4):375–389, 2018. 2

[65] Huanyu Yu, Shuo Cheng, Bingbing Ni, Minsi Wang, Jian Zhang, and Xiaokang Yang. Fine-Grained Video Captioning for Sports Narrative. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[66] Amir R. Zamir, Alexander Sax, William Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling Task Transfer Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 6

[67] Dan Zecha, Moritz Einfalt, and Rainer Lienhart. Refining Joint Locations for Human Pose Tracking in Sports Videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1

[68] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *International Conference on Computer Vision*, pages 7094–7103, 2019. 8

[69] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *IEEE International Conference on Computer Vision (ICCV)*, October 2017. 2