Action Spotting in Soccer Videos Using Multiple Scene Encoders

Yuzhi Shi[1] Hiroaki Minoura[1] Takayoshi Yamashita[1] Tsubasa Hirakawa[1] Hironobu Fujiyoshi[1]
Mitsuru Nakazawa[2] Yeongnam Chae[2] Björn Stenger[2]

[1]Chubu University, Kasugai, Aichi, Japan
[2]Rakuten Institute of Technology, Rakuten Group, Inc.
Email: {shi, himi1208, hirakawa}@mprg.cs.chubu.ac.jp, {takayoshi, fujiyoshi}@isc.chubu.ac.jp
, {mitsuru.nakazawa, yeongnam.chae, bjorn.stenger}@rakuten.com

***Abstract*—Action spotting, which temporally localizes specific actions in a video, is an important task for understanding high-level semantic information. In this paper, we formulate the action spotting task to one of scene sequence recognition and propose a model with multiple scene encoders to capture scene changes around the timestamp where an action occurs. We divide the input into multiple subsets to reduce the influence of scene context that is temporally distant, and feed every subset into a scene encoder to learn scene context in every subset. Because the optimal temporal length for time windows (chunks) is different for each action, we analyze the influence of chunk sizes for action spotting. The experimental results on the public SoccerNet-v2 dataset demonstrate state-of-the-art accuracy. By using embedding features, our method obtains an Average-mAP of 75.3%. In addition, we confirm that the performance can be improved by using optimal chunk sizes for different actions.**

I. Introduction

The production of sports game summaries requires human and material resources. Automatic or semi-automatic generation of summary videos will reduce the time from event to broadcast. To approach this task, several fundamental technologies of video understanding are required; for example, action recognition [12], action detection [3] and action spotting [2], [8]. Action spotting is the task of localizing an event anchored to a single timestamp. Different from action detection and recognition, it is required to find the most relevant frame where the action occurred.

There are two main challenges in action spotting. The first challenge is how to adequately use temporal information. A video clip generally includes sequential images that are visually similar, but have a different context, such as a football scene where the football field is typically the image background as shown in Fig. 1. Therefore, it is important to consider not only visual information but also underlying temporal information for action spotting in such videos [8], [15]. The second challenge is the different durations of different actions. One may consider an action being composed of different subactions, e.g., a *Goal* action is often composed of *Running*, *Shooting* and *Cheering* subactions. A *Yellow card* action typically includes a scene of a player falling and a scene of the referee raising the card.

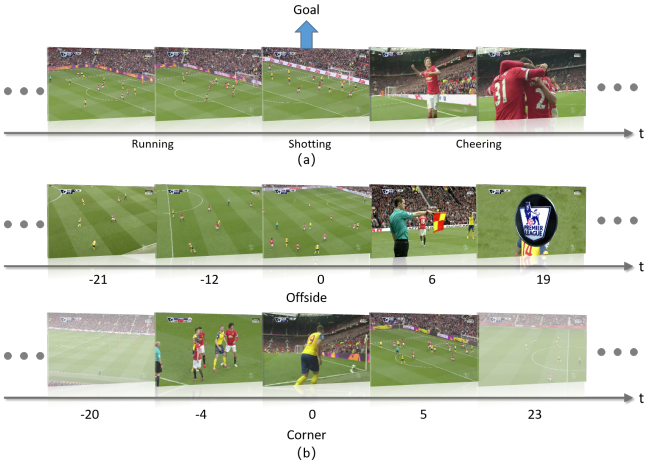


Fig. 1: Action spotting. *A Goal action is shown in (a), which consists of running, shooting and cheering scenes. The top and bottom of (b) show an Offside and a Corner action, respectively. We can see the temporal duration of these actions are different. Note that the action happens in zero second as the center image of each example. Images are cited from [4].*

In this paper, we propose a novel method based on multiple scene encoders for action spotting to tackle these challenges[1]. For better performance in few-shot learning, we use several subsets of sequence images as an important cue for action spotting. Specifically, as shown in Fig. 2, first we extract image features from sequential images of a video as a chunk. A chunk is split into multiple subsets, and fed into multiple transformer encoders, respectively, to recognize scene content and capture the changes of scenes in an action. Finally, we classify actions by recognizing the scene sequence. Multiple encoders help to suppress the influence of different subactions and improve the recognition of subaction sequences. For the second challenge of different action duration, we use different chunk sizes for each action with the aim of using all action-related frames and reducing redundant data in a chunk. In our experiments on SoccerNet-v2 [4], the proposed

[1]We denote input data and its temporal length as chunk and chunk size respectively, following prior work [4], [8].

method reaches 55.2% Average-mAP using ResNet features, and 75.3% using embedding features, representing state-of-the-art performance. In addition, in an ablation study, we confirm that we improve the performance by using optimal chunk sizes for each class.

Our main contributions are summarized as follows: (1) We propose a new action spotting model with multiple scene encoders based on transformer encoders to learn from scene context appearing with actions in every subset, respectively, and recognizing scene sequences. (2) We confirm that the performance can be improved by using an optimal chunk size for different actions. (3) We achieve a state-of-the-art average-mAP of 75.3% for action spotting on the SoccerNet-v2 dataset and confirm the model design choices in ablation studies.

II. Related Work

In this section, we briefly review prior work of action spotting and introduce datasets of sports videos.

A. Action spotting

In video understanding, there are many fundamental tasks, such as video classification [13], action recognition [12] and action detection [3]. Video classification is the task of predicting the label of a video. Action recognition aims to recognize specific actions in videos. Action detection aims to locate temporal regions of particular actions in real-world videos. Action spotting is the task of temporally localizing a specific action anchored with a single timestamp [1], [24]. Action spotting in sports videos is challenging because of rapid scene changes in videos and the various duration of actions. Several approaches have been proposed for action spotting, especially in soccer videos. A regression and masking approach with RMS-Net was introduced in [20]. In this method, frames only after an action occurs (post-action) are considered, without using pre-action frames. CALF [2] introduces a context-aware loss function, which weights frames in different temporal segments according to the distance from the ground truth timestamp, such as 'far before', 'just before' and 'just after' an action occurs. NetVLAD++ [8] uses two NetVLAD [19] modules to handle the context before and after an action, respectively. They disentangle the context from past and future frames and learn specific vocabularies of semantics for each subsets to avoiding blending such vocabulary in time. Inspired by this work, we propose a model based on multiple encoders to handle the temporal relationship in a video. The multiple encoder structure is effective at reducing scene-context interactions and recognizing scene sequences. Moreover, in contrast to prior work, we explore the use of different chunk sizes for different action classes.

B. Datasets of sports videos

Large-scale public datasets are available for different sports, such as GolfDB [16] and MLB-Youtube [18]. A dataset containing 222 sports broadcast videos was released in [26]. SoccerNet [7] is a public dataset of soccer videos including three types of actions (*goal*, *card* and *substitution*). SoccerDB[11] merged a subset of 270 games from SoccerNet with 76 soccer games. SoccerNet-v2[4] extended SoccerNet [7] with 17 actions and over 300,000 annotations. We use the SoccerNet-v2 dataset [4] to evaluate the performance of different action spotting models.

III. Method

We propose a novel method for action spotting using multiple encoders as shown in Fig. 2. We use multiple encoders that learn semantic information in temporal subsets to recognize scene sequences. To consider the different duration of actions, we use optimal chunk sizes during inference.

A. Video encoding

We extract feature vectors from the videos subsampled at 2 fps. Let us denote the frame of an input video at the time index $t = 1, 2, \ldots, T$ by $\mathbf{x}_t \in \mathbb{R}^{H \times W \times C}$, where T is the number of frames in a chunk, H and W are height and width, and C is the number of image channels. We compute a d-dimensional feature vector per frame as $\mathbf{f}_t \in \mathbb{R}^{1 \times d}$. The feature vectors of an input video are fed to our action spotting model, which is explained next.

B. Model

We propose a model with multiple scene encoders to capture scene changes.

1) Multiple encoder structure: We formulate action spotting to the task of recognizing scene sequences. To recognize scene sequences, we propose the multiple encoders structure, as presented on Fig. 2. We partition the set of input feature vectors into multiple subsets and use an encoder for each subset.

2) Scene Encoder: To accommodate different features, which may have different dimensions, we use an MLP layer to map features to a fixed length, denoted E, in every encoder. We inject temporal information by adding a positional encoding to each feature vector to help model the temporal relationship between frames. The positional encoding is represented as a sinusoidal function proposed in [23].

To learn latent features from the temporal relationship among frames, we use a multi-head self-attention mechanism [23]. We denote the number of scene encoders, the number of stacked encoder layers and the number of heads in multi-head attention as N, L and H, respectively. The attention calculated by the query matrix $Q^n = \phi_q^n(\{\mathbf{f}_t\}_{t=1}^{\frac{T}{N}})$, the key matrix $K^n = \phi_k^n(\{\mathbf{f}_t\}_{t=1}^{\frac{T}{N}})$ and the value matrix $V^n = \phi_v^n(\{\mathbf{f}_t\}_{t=1}^{\frac{T}{N}})$, where ϕ_q, ϕ_k and ϕ_v are MLP layers and n is the index of the encoder.

The attention of the h^{th} attention head in the n^{th} encoder is calculated by

$$\text{head}_h^n = \text{Attention}_n^p(Q^n, K^n, V^n), \quad (1)$$

$$O_n = \phi_o^n([\text{head}_h^n]_{h=1}^H), \quad (2)$$

where ϕ_o^n is an MLP layer, and the $Attention$ function is the scaled dot-product attention in [23]. We denote the output

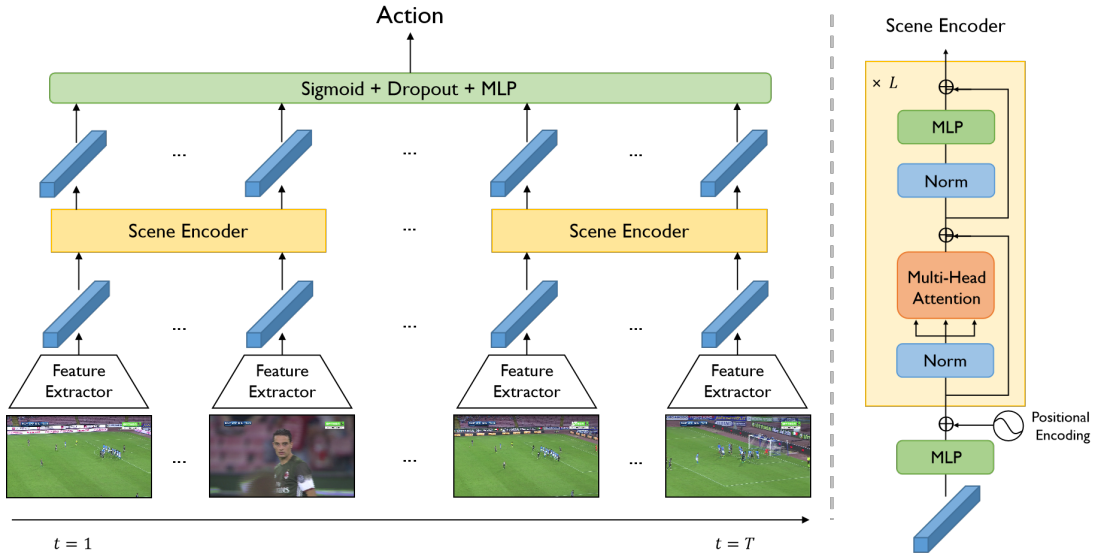


Fig. 2: Proposed Network Architecture. *We split the extracted feature vector into multiple subsets of similar duration, feed them into multiple scene encoders respectively. Each scene encoder is designed based on the transformer's encoder shown on the right. Images of a football game are cited from [4].*

of n^{th} encoder as O_n. We consider that query, key, value matrices are three different representations of the same frames. In Attention, the dot-product of Q and K represents the correlation of every two frames. Two similar frames have a strong correlation, because their features are similar. The result is used as the weight on V to help the model focus on similar frames. Since the frames in a scene are generally similar, using self-attention could help to find scenes in a subset by finding similar images.

We use scene encoders to extract scene features in each subset. We merge the outputs of the multiple encoders, which presents the scene features in the subsets, and average them temporally. We use a sigmoid layer and a dropout layer to avoid overfitting. Finally, we use an MLP ϕ_r layer and a sigmoid layer to classify an action as:

$$\mathbf{C} = [O_1; O_2; ...; O_n], \tag{3}$$

$$\mathbf{m} = \frac{1}{T}\sum_{t}^{T} \mathbf{c}_t, \tag{4}$$

$$\mathbf{y} = \sigma(\phi_r(dropout(\sigma(\mathbf{m})))), \tag{5}$$

where [;] is a concatenation operator, and $\mathbf{c}_t$ is the presentation vector of the t^{th} frame output by encoders.

C. Training

During training, we divide features into multiple non-overlapping chunks. We train the proposed model to predict the actions in chunks. Since multiple actions can occur within a chunk, we formulate it as a multi-label action classification task.

D. Inference

There are two types of inference processes, using a fixed chunk size for all actions and using a optimal chunk sizes for each class. When using a fixed chunk size, we select the chunk label corresponding to the classification result of its center frame. We obtain the prediction result for the entire video via sliding chunks, frame by frame, to predict classification results of all frames. The inference process of using optimal chunk sizes is shown in Figure 3. We input features with different chunk size to the trained model and obtain the classification result. We take the result using a specific chunk size as the classification result of corresponding actions. For example, we

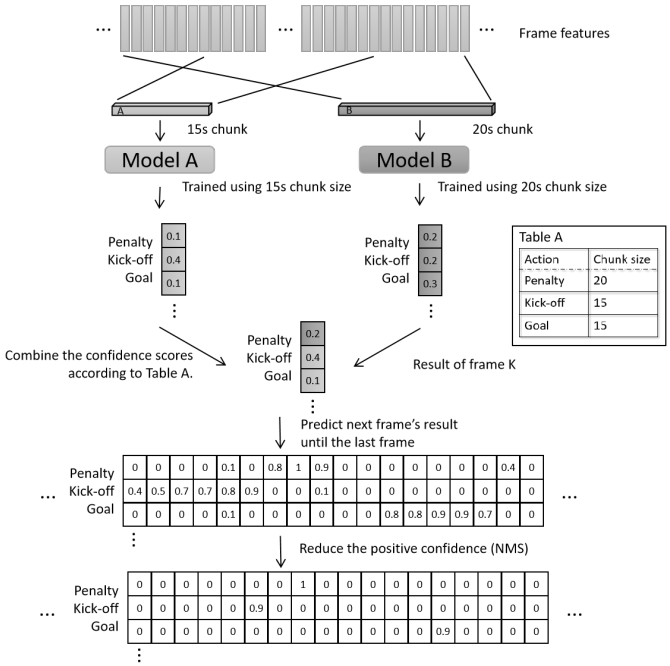


Fig. 3: Inference. *Model A and B are trained with the chunk size of 15 and 20 seconds, respectively.*

use the 20-second chunk result as the classification result of *penalty* action. Similarly, we slide chunks frame by frame to obtain the results of every frame. We select appropriate chunk sizes using the validation dataset.

We use Non-Maximum Suppression(NMS) similar to [7], [2], [4], [8] for reducing positive action spotting results with low confidence. The NMS threshold is 0 and the NMS window size is 60 frames (30 seconds).

IV. Experiments

In this section, we compare our model with several existing methods on the SoccerNet-v2 dataset [4]. We then analyze the influence of chunk size in action spotting to confirm the effectiveness of chunk size optimization. Moreover, we conduct an ablation study to confirm the design choices of our method. Finally, we evaluate our method on the first version of SoccerNet [7] to show the generalization capability of our method.

A. Experimental setting

1) Evaluation protocols: We use SoccerNet-v2 [4] to train and evaluate our method. SoccerNet-v2 contains 765 hours of videos of 500 soccer games, with 300,000 annotated timestamps and 17 action classes, such as *goal*, *ball out of play*, and *yellow card*. SoccerNet-v2 is divided into training, validation and test sets as 300, 100 and 100 games, respectively [4]. We measure performance using the Average-mAP value. If the temporal offset between prediction and its closest ground truth is less than a given tolerance Δ, the prediction is regarded as positive. The average precision (AP) for the prediction per class within the threshold Δ, averaged over action classes to calculate the mAP. The Average-AP is the average of AP values calculated over 12 error tolerance values Δ (from 5s to 60s, the step size is 5s) respectively for each class. The Average-mAP is the average of 17 Average-AP.

2) Implementation details: We use a ResNet-152 [10] pre-trained on ImageNet [5] as the feature extractor. The frame features are extracted at 2 fps videos with a resolution of 224×224. The feature extractor outputs a 2,048-dimensional vector for every frame. The number of scene encoders in the proposed model is set to two. The features are remapped to a 256 dimension vector per frame. Every scene encoder contains 8 heads and 2 encoder layers. We use the Adam optimizer [14], the binary-cross entropy loss function, and a starting learning rate of 0.002. We set the dropout rate to 0.1, the batch size to 128 and the chunk size to 15 seconds on training and validation datasets. We stop the training once the mAP on the validation dataset stops decreasing for 6 continuous epochs. The model achieving the best performance on the validation dataset is used as the model for evaluation on the test set.

B. Results

1) Comparison with state-of-the-art methods: Table I shows the results of our methods and several state-of-the-art methods (NetVLAD [19], AudioVid [22], CALF [2] and NetVLAD++[8]). As seen in Table I, our method achieves an Average-mAP of 55.2% on the test dataset. The Average-mAP is an absolute 1.8% higher compared to NetVLAD++. This improvement is consistently seen for 12 of the 17 action classes, especially for classes with few examples (*Red Card* and *Yellow→Red*). This indicates the advantage of modeling the temporal relationship in every subset, especially for actions with few samples. The proposed method adequately utilizes the temporal relationship thanks to the multiple encoder structure. Performance has low correlation with the number of samples as seen from Table I. In several actions where there are distinctive patterns of frame changes such as *corner* and *goal*, we achieve good performance even with few samples. On the other hand, on actions with less distinctive patterns such as *Indirect free-kick*, the performance is lower. For better recognition of temporal patterns, it is important to consider optimal chunk size because we can include as many as action-related frames in the model and decrease the number of unrelated frames in samples (see next subsection).

2) Visualization of confidence score: For the further analysis of our results, we visualize the confidence scores of several classes, as shown in Figure 4.

Figures 4 (a) and (d) show examples of the confidence score of *Corner* action on a temporal axis. As seen in these figures, the confidence score is high for frames where several people are shown in a close-up view. We assume that these frames are helpful to localize corner actions. Such frames ordinarily last for around 10 seconds; therefore, an appropriate chunk size of *Corner* action would be 10 seconds, as shown in Table II.

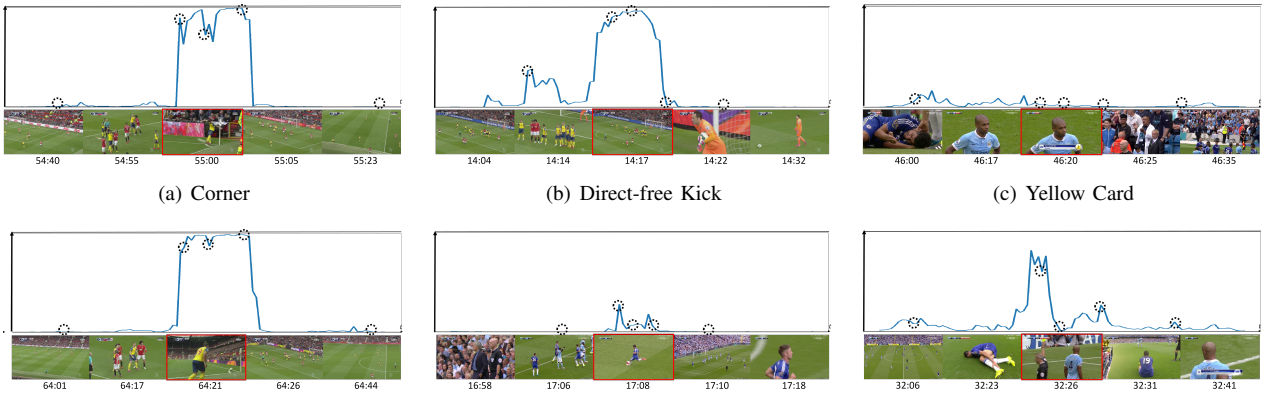
The confidence of the *Direct free-kick* action is high for frames where players gather in front of a goal. As shown in Figure 4 (b), the confidence score increases in frames where players gather in front of the goal. On the other hand, in Figure 4 (e), such frames appear for only a few seconds. Consequently, the confidence becomes low.

In the *Yellow card* action, we often observe scenes of a player falling and a referee appearing. In contrast, no referee appears in Figure 4 (c), however the yellow card information is displayed on the screen. The confidence score is low in this case. In another case, we observe a high confidence score in Figure 4 (f) as the scene of a falling player followed by a referee scene. $10-20$ seconds is an appropriate range of chunk size for yellow card actions. Because the scenes where a player falls often happen $3-10$ seconds before a *Yellow card* action, there is often a replay scene where a player falls, increasing the appropriate chunk size for yellow cards to over 10 seconds.

3) Optimization of chunk size: To find an appropriate chunk size for each action, first we train our model with different chunk sizes, ranging from 10 to 40 seconds (5s as step size). Then, we select the best chunk size that achieves the best performance on the validation dataset. The optimal chunk sizes are shown in Table III. From this table, we confirm that each action requires its own chunk size. Thanks to this optimization, we can see several performance improvements. For example, the average-AP of *Offside* is improved by 3.6% using the chunk size of 20 seconds. The average-AP of *Shot-off target*

TABLE I: State-of-the-art comparison. *The ResNet is a pretrained ResNet-152 and PCA is principal component analysis. Ours(1) uses 15 seconds as chunk size. Ours(2) uses a optimal chunk size for each action. Number of data is the number of samples in every class.*

Method	Feature Extractor	Average-mAP	Penalty	Kick-off	Goal	Substitution	Offside	Shots on target	Shots off target	Clearance	Ball out of play	Throw-in	Foul	Indirect free-kick	Direct free-kick	Corner	Yellow card	Red card	Yel.→Red card
NetVLAD	ResNet+PCA	31.4	47.4	42.4	32.0	16.7	32.7	21.3	19.7	55.1	51.7	45.7	33.2	14.6	33.6	54.9	32.3	0.0	0.0
AudioVid	ResNet+PCA	39.9	54.3	50.0	55.5	22.7	46.7	26.5	21.4	66.0	54.0	52.9	35.2	24.3	46.7	69.7	52.1	0.0	0.0
CALF	ResNet+PCA	40.7	63.9	56.4	53.0	41.5	**51.6**	26.6	27.3	**71.8**	47.3	37.2	41.7	25.7	43.5	72.2	30.6	0.7	0.7
NetVLAD++	ResNet+PCA	50.7	67.7	59.6	70.2	70.3	35.3	37.1	38.3	56.0	68.2	65.3	62.4	43.4	55.2	78.9	50.0	1.5	1.7
NetVLAD++	ResNet	53.4	**79.3**	**62.1**	71.6	68.7	39.3	39.3	41.0	57.0	70.3	69.0	**64.2**	44.4	57.8	79.7	**56.7**	4.0	3.7
Ours(1)	ResNet	54.7	75.8	60.8	72.0	70.6	38.6	41.8	40.2	60.6	71.3	70.3	63.5	49.2	59.9	81.6	53.4	8.0	**11.5**
Ours(2)	ResNet	**55.2**	68.7	60.8	**72.0**	**70.6**	42.2	**41.8**	**42.2**	60.6	**73.1**	**72.3**	63.4	**49.2**	**59.9**	**83.6**	53.9	**15.9**	7.2
Number of data			173	2566	1703	2839	2098	5820	5256	7896	31810	18918	11674	10521	2200	4836	2047	55	46



(a) Corner (b) Direct-free Kick (c) Yellow Card

(d) Corner (e) Direct-free Kick (f) Yellow Card

Fig. 4: Confidence score examples. *Ground truth labeled frames of different actions are shown in frames from [4] within red boxes. The blue line represents the change of the confidence scores in the time series axis. The confidence scores of shown frames are indicated by circles in the graphs above.*

TABLE II: Evaluation results for different chunk sizes. *ResNet-152 is used as the feature extractor. The Average-AP on the validation dataset changes with chunk size on three actions (direct free-kick, corner and yellow card).*

Chunk Size	10	15	20	25	30
Direct free-kick	52.7	**57.4**	52.8	49.7	46.4
Corner	**83.0**	81.1	77.8	74.6	69.6
Yellow card	54.9	54.7	**55.7**	52.3	52.6

is also improved by 2% with the chunk size of 10 seconds. The Average-AP of few-shot actions, such as *Red card* and *Yellow→Red card*, varies considerably with chunk size, and the lack of samples makes it challenging to select an optimal value. On the other hand, actions containing many samples have similar optimal chunk sizes in test and validation datasets. We observe an improvement of 0.5% Average-mAP in total.

C. Ablation study

1) Evaluation of number of encoders: To analyze the influence of the number of encoders, we train our model with different numbers of encoders, from one to five. For each number of encoders, we tune the hyper-parameters, including chunk size, to obtain the best performance for each model. Note that we use a fixed chunk size for all models in this experiment, and the best performance is obtained for 15-second chunks. We observe that models with multiple encoders achieve better results than the single-encoder model, as shown in Table IV. Interestingly, the performance of models with more than two encoders is decreasing as the number of frames in each chunk is reduced when covering the same time window. The model with two encoders achieves the best performance on SoccerNet-v2, modeling pre-action and post-action windows with a duration of 7.5 seconds each.

2) Comparison of feature extractors: We evaluate the proposed method using three feature extractors (ResNet, ResNet+PCA and Embedding feature extractor [27]). The results are shown in Table V. The embedding feature extractor [27] consists of TPN [25], GTA [9], VTN [17], irCSN [21], I3D-Slow [6]. The average mAP increases to 75.3% by using embedding features. To extract scene context in videos, it is required to correctly acquire important information in frames. Therefore, the selection of a feature extractor has a significant effect on action spotting accuracy.

D. Evaluation on SoccerNet-v1

SoccerNet-v1 [7] contains the same soccer videos as SoccerNet-v2, but only includes three action classes and 6,637 annotations. We train and evaluate the proposed model on the SoccerNet-v1 dataset [7] and compare with related work. The

TABLE III: Evaluation of chunk size optimization. *In Fixed size, every class uses a 15 second chunk size. Results with and without chunk size optimization are shown in Fixed and Optimized. The optimized chunk size of action classes is shown in the row Chunk Size. For most action classes, we observe a performance improvement by chunk size optimization.*

	Penalty	Kick-off	Goal	Substitution	Offside	Shots on target	Shots off target	Clearance	Ball out of play	Throw-in	Foul	Indirect free-kick	Direct free-kick	Corner	Yellow card	Red card	Yel.→Red card	Avg-mAP
Fixed	**75.8**	**60.8**	**72.0**	**70.6**	38.6	**41.8**	40.2	**60.6**	71.3	70.3	**63.5**	**49.2**	**59.9**	81.6	53.4	8.0	**11.5**	54.7
Optimized	68.7	**60.8**	**72.0**	**70.6**	**42.2**	**41.8**	**42.2**	**60.6**	**73.1**	**72.3**	63.4	**49.2**	**59.9**	**83.6**	**53.9**	**15.9**	7.2	**55.2**
Chunk Size (sec)	20	15	15	15	20	15	10	15	10	10	10	15	15	10	20	30	20	

TABLE IV: Comparison of different number of encoders. *We evaluate the average-mAP for encoders' number from 1 to 5, using ResNet-152. The highest value is obtained by a model using two encoders. Chunk Size shows the best fixed chunk size for each model.*

Num of Encoder	1	2	3	4	5
Average-mAP(%)	52.1	**54.7**	53.5	52.9	52.2
Chunk Size(s)	15	15	15	15	15

TABLE V: Comparison of different feature extractors. *We evaluate the performance of the proposed method using three feature extractors. Length shows the feature vector length of each feature extractors.*

Feature Extractor	ResNet+PCA	ResNet	Embedding
Average-mAP(%)	52.9	54.7	**75.3**
Length	512	2048	8576

proposed method obtains a 66.8% average-mAP, exceeding CALF [2] by 4.3%, as shown in Table VI. When using ResNet and PCA as the feature extractor, the proposed method exceeds CALF [2] by 2%.

V. Conclusion

In this paper, we propose a new model with multiple encoders for the task of action spotting. We employ multiple scene transformer encoders to learn from the temporal relationship in subsets. In experiments, we show that the proposed model reaches 55.2% average-mAP, increasing the state-of-the-art by 1.8%. When using embedding features, the proposed model obtains an average-mAP. of 75.3%. Furthermore, we confirm that the optimization of the chunk size for each action is effective for action spotting.

TABLE VI: Results on SoccerNet. *The proposed method compares with prior works on SoccerNet and achieves the best performance.*

Model	Feature Extractor	Average-mAP
NetVLAD [19]	ResNet+PCA	49.7
AudioVid [22]	ResNet+PCA	56.0
CALF [2]	ResNet+PCA	62.5
NetVLAD++ [19]	ResNet+PCA	61.1
Ours	ResNet+PCA	64.5
Ours	ResNet	**66.8**

References

[1] 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*, pages 251–266, 2018.

[2] Anthony Cioppa, Adrien Deliège, Silvio Giancola, Bernard Ghanem, Marc Van Droogenbroeck, Rikke Gade, and Thomas B. Moeslund. A context-aware loss function for action spotting in soccer videos. In *Computer Vision and Pattern Recognition*, pages 13126–13136, 2020.

[3] Rui Dai, Srijan Das, Luca Minciullo, Lorenzo Garattoni, Gianpiero Francesca, and Francois Bremond. Pdan: Pyramid dilated attention network for action detection. In *In Winter Conference on Applications of Computer Vision*, pages 2970–2979, 2021.

[4] Adrien Deliège, Giancola Silvio Cioppa, Anthony, Jacob V. Seikavandi, Meisam Dueholm, Kamal Nasrollahi, Bernard Ghanem, Thomas B. Moeslund, and Marc Van Droogenbroeck. Soccernet-v2 : A dataset and benchmarks for holistic understanding of broadcast soccer videos. In *Computer Vision and Pattern Recognition Workshops*, pages 4508–4519, 2021.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *International Conference on Computer Vision*, pages 6202–6211, 2019.

[7] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. In *Computer Vision and Pattern Recognition Workshops*, pages 1824–1834, 2018.

[8] Silvio Giancola and Bernard Ghanem. Temporally-aware feature pooling for action spotting in soccer broadcasts. *arXiv:2104.06779*, 2021.

[9] Bo He, Xitong Yang, Zuxuan Wu, Hao Chen, Ser-Nam Lim, and Abhinav Shrivastava. GTA: Global temporal attention for video action understanding. *arXiv preprint arXiv:2012.08510*, 2020.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[11] Yudong Jiang, Kaixu Cui, Leilei Chen, Canjin Wang, and Changliang Xu. Soccerdb: A large-scale database for comprehensive video understanding. In *Proceedings of the 3rd International Workshop on Multimedia Content Analysis in Sports*, pages 1–8, 2020.

[12] M Esat Kalfaoglu, Sinan Kalkan, and A Aydin Alatan. Late temporal modeling in 3d cnn architectures with bert for action recognition. In *European Conference on Computer Vision*, pages 731–747, 2020.

[13] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[14] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[15] William McNally, Kanav Vats, Tyler Pinto, Chris Dulhanty, John McPhee, and Alexander Wong. Golfdb: A video database for golf swing sequencing. In *Computer Vision and Pattern Recognition Workshops*, 2019.

[16] William McNally, Kanav Vats, Tyler Pinto, Chris Dulhanty, John McPhee, and Alexander Wong. Golfdb: A video database for golf swing sequencing. In *Computer Vision and Pattern Recognition Workshops*, pages 2553–2562, 2019.

[17] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021.

[18] AJ Piergiovanni and Michael S. Ryoo. Fine-grained activity recognition

in baseball videos. In *Computer Vision and Pattern Recognition Workshops*, pages 1740–1748, 2018.

[19] Arandjelovic Relja, Gronat Petr, Torii Akihiko, Pajdla Tomas, Josef, and Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[20] Matteo Tomei, Lorenzo Baraldi, Simone Calderara, Simone Bronzin, and Rita Cucchiara. Rms-net: Regression and masking for soccer event spotting. In *International Conference on Production Research*, pages 7699–7706, 2021.

[21] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *International Conference on Computer Vision*, pages 5552–5561, 2019.

[22] Bastien Vanderplaetse and Stephane Dupont. Improved soccer action spotting using both audio and video streams. In *Computer Vision and Pattern Recognition Workshops*, pages 3921–3931, 2020.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[24] Guillaume Vaudaux-Ruth, Adrien Chan-Hon-Tong, and Catherine Achard. Actionspotter: Deep reinforcement learning framework for temporal action spotting in videos. In *International Conference on Production Research*, pages 631–638, 2021.

[25] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Computer Vision and Pattern Recognition*, pages 591–600, 2020.

[26] Junqing Yu, Aiping Lei, Zikai Song, Tingting Wang, Hengyou Cai, and Na Feng. Comprehensive dataset of broadcast soccer videos. In *Multimedia Information Processing and Retrieval*, pages 418–423, 2018.

[27] Xin Zhou, Le Kang, Zhiyu Cheng, Bo He, and Jingyu Xin. Feature combination meets attention: Baidu soccer embeddings and transformer based temporal detection. *arXiv:2106.14447*, 2021.