Study of Action Spotting via Transformer with Pre-content and Post-content Encoders

TP19051 Yuzhi Shi Supervisor: Takayoshi Yamashita

1. Introduction

The production of sports broadcasts takes huge manpower and material resources. Automatic production will save huge time for producers. Action spotting is a significant task to understand high-level semantic information, supports automatic production. In this paper, we focus on the action spotting task on videos, which temporally localizes the specific actions.

The existing works generally use same temporal length of video frame features. The chunk size is defined as the temporal length of video clips. We propose a novel model based on transformer encoder for action spotting. In addition, we analyze the influence of chunk size for action spotting per action and use an appropriate chunk size for each action to train our proposed model. The experiment results demonstrate that our proposed method improves the Average-mAP for the action spotting task and achieves state-of-the-art performance.

2. Related work

Action spotting is the localization of an action anchored with a single timestamp. It's a challenging task because of rapid change of scenes in videos and the imbalance number of annotations for each action. Prior works proposed some efficient methods for action spotting. A regression and masking approach with RMS-Net was introduced [5], they drop pre-content during training, expecting the model to focus on the post-content frames. [6] introduces a context-aware loss function, defines a high-level semantic context from different temporal regions far distant, just before and just after an action occurs. NetVALD++ [3] uses a novel pooling module for action spotting that learns a temporally-aware vocabulary for past and future temporal context. However, chunk size is important for localizing action on soccer videos because different actions has different temporal lengths of related video frames. Different from previous works [3, 6] use fixed chunk size for all actions, we resample chunks of an appropriate size to take advantage of different ranges of frames in videos per class.

3. Method

In this section, we describe the process of video encoding at first. Next, we present the structure of our network. At last,the details of inference are introduced.

3.1 Video encoding

Our proposed architecture is shown in Figure 1. We assume that there are specific actions (including background) in soccer video in time $t = 1, 2, \ldots, T$. We use the same feature extractor as SoccerNet-v2 [1]. The features $\mathbf{h}_t \in \mathbb{R}^{1\times d}, t = 1, 2, ..., T$ are extracted from the video at 2fps with a resolution of $\mathbf{x}_t \in \mathbb{R}^{H\times W\times C}, t = 1, 2, ..., T$ by a feature extractor, where $(H \times W \times C)$ is the resolution of the original image and d is the feature vector dimension.

3.2 Network

We propose a method, which split pre-content $T_{pre} = \{1, \ldots, \frac{T}{2}\}$ and post-content $T_{post} = \{\frac{T}{2} + 1, \ldots, T\}$ frames to learn specific actions in the semantics of each subset inspired by NetVLAD++ [3]. Two encoders are designed based on Transformer [4], have different weights. We first input temporal information by adding positional encoding to each feature vector extracted via feature extractor. The value is represented as pre-content $\mathbf{p}_t$ and post-content $\mathbf{p'}_t$, where $t = 1, \ldots, \frac{T}{2}$. The positional encoding is represented the components are sinusoids of different wavelength according to [4].

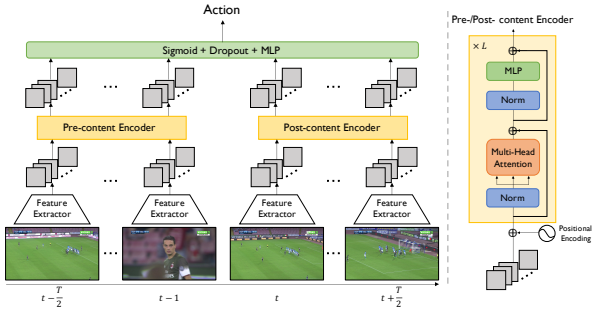

Figure 1: **Proposed Network Architecture**

The two encoders learn latent features for the action labels for each time step via a self-attention mechanism with L layers and N heads, respectively. The self-attention of pre-content encoder learn the query matrix $Q^p = \phi_q^p(\{\mathbf{p}_t\}_{t=1}^{\frac{T}{2}})$, the key matrix $K^p = \phi_k^p(\{\mathbf{p}_t\}_{t=1}^{\frac{T}{2}})$ and the value matrix $V^p = \phi_v^p(\{\mathbf{p}_t\}_{t=1}^{\frac{T}{2}})$, and the self-attention of post-content encoder learn the query matrix $Q^f = \phi_q^f(\{\mathbf{p'}_t\}_{t=1}^{\frac{T}{2}})$, the key matrix $K^f = \phi_k^f(\{\mathbf{p'}_t\}_{t=1}^{\frac{T}{2}})$ and the value matrix $V^f = \phi_v^f(\{\mathbf{p'}_t\}_{t=1}^{\frac{T}{2}})$, where all $\phi_\cdot$ are MLP layers. It computes the attention w.r.t. pre-content by

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

$$P = \phi_o^p([\text{head}_n^p]_{n=1}^N), \quad (2)$$

and post-content by

$$\text{head}_n^f = \text{Attention}_n^f(Q^f, K^f, V^f), \quad (3)$$

$$F = \phi_o^f([\text{head}_n^f]_{n=1}^N), \quad (4)$$

where ϕ_o^p and ϕ_o^f is an MLP layer, and the Attention function the scaled dot-product attention in [4].

We merge the output of the two encoders, and average the output temporally. We then use a sigmoid layer and a dropout layer σ to suppress overfitting. Finally using an MLP ϕ_r layer to accurately classify action as:

$$\mathbf{c} = [P; F], \quad (5)$$

$$\mathbf{m} = \frac{1}{T}\sum_t^T \mathbf{c}_t, \quad (6)$$

$$\mathbf{y} = \phi_r(\sigma(\mathbf{m})), \quad (7)$$

where [;] is concatenation operator.

3.3 Inference

Considering that different actions need different sizes of chunks to localize the specific frame in a video more correctly, we resample features from different sizes of chunks to improve the performance. We set an appropriate chunk size for each class. As an example, we set the chunk size of the goal class as 45 seconds(90 frames). We define the t as the index of chunk's starting frame in soccer videos. First, the features (90×512) of a chunk (from t to $t + 90$ frames) size of 90 frames is used to average resample to a chunk (20×512) size of 40 frames. Secondly, we put the resampled features into the trained network followed by a sigmoid layer. We use the prediction on the goal class as the result of classification

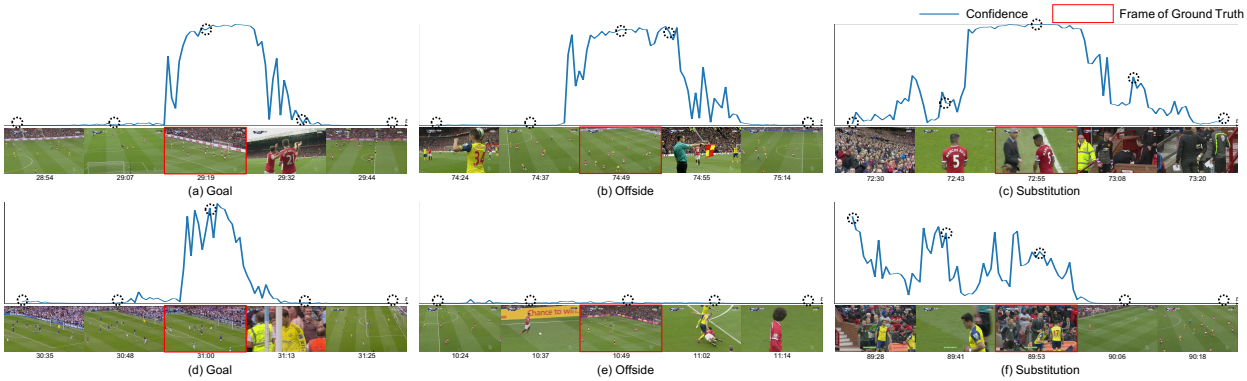

Figure 2: **Visual examples of confidence**.

on the goal class at $t + 45$ frame. Then we use another chunk of different size to predict classification results on other actions. Next, concatenate the prediction results of all actions as the classification result at $t + 45$ frame. We use the same way to predict the classification result at $t + 46$ frames. Finally, we obtain the prediction result of the entire game via sliding chunk frame by frame and predict classification results for every chunk. We use Non-Maximum Suppression (NMS) to improve prediction results. The NMS threshold is 0 and the NMS window is 40 seconds.

4. Experiment

In this section, we introduce the datasets and evaluation metrics used, compare our model with several prior methods and analyze experiment results.

4.1 Dataset

We train and test models on SoccerNet-v2. Both of them have imbalance action labels. SoccerNet-v2 annotated 300k timestamps has 17 classes of actions. SocceNet-v2 is divided into train/val/test (300/100/100 games), and we follow them.

4.2 Results

Table 1: **State-of-the-art comparison**. The visible action is shown in broadcast video and the unshown action must be inferred by the viewer.

Method	Avg-mAP	Visible	Unshown
NetVLAD [7]	31.4	34.4	23.3
CALF [6]	40.7	42.1	29.0
NetVLAD++ [3]	53.4	59.4	**34.8**
Ours	**54.6**	**60.0**	33.8

Table 1 shows that our method achieves an Average-mAP of 54.56% on the test dataset and gets the state-of-the-art performances on the SoccerNet-v2 benchmark. We assume that irrelevant frames provide useless information and relevant frames provide significant information to localize a specific action. The chunk size can control which frames are used to predict. We change chunk size (from 5 seconds to 90 seconds, 5 seconds as step size) on testing to find an appropriate chunk size per action for action spotting. As shown in Table 2, the appropriate chunk size per action is different. For instance, corner action need 10 seconds as the chunk size to get a high Average-AP score. Kick off action's best chunk size is 20 seconds. The best chunk size for each action in vanilla transformer is very similar with our proposed network. There are nine classes of actions have the same best chunk size in vanilla transformer and our proposed network. And the difference of four classes of actions' appropriate chunk size in vanilla transformer and our proposed network only 5 seconds.

Table 2: **Relationship between action and chunk size.**

Chunk Size	10	15	20	25	30	45
Penalty	49.89	69.17	71.67	**73.18**	70.85	67.41
Kick-off	54.19	57.09	**57.97**	57.56	56.70	48.79
Goal	61.03	70.23	69.14	71.02	**71.46**	66.55
Substitution	65.03	**67.66**	67.06	64.93	62.39	50.19
Offside	34.37	40.16	41.42	**41.48**	38.94	33.05
Shots on target	38.89	39.24	**39.51**	39.21	38.59	35.23
Shots off target	39.51	38.21	**39.61**	37.06	34.93	26.08
Clearance	53.59	**54.84**	54.34	50.96	48.08	38.98
Ball out of play	69.43	**69.83**	68.53	66.52	64.10	56.94
Throw-in	65.59	**66.21**	64.52	62.05	59.37	49.69
Foul	62.15	**62.71**	61.86	60.32	58.99	49.64
Indirect free-kick	41.09	42.34	**45.14**	44.82	43.74	37.72
Direct free-kick	**56.54**	56.00	54.09	50.69	47.64	42.47
Corner	**80.33**	78.61	75.88	72.86	69.56	58.50
Yellow card	51.74	**52.35**	51.44	51.38	50.69	39.82
Red card	6.66	14.76	13.45	13.86	17.01	**24.58**
Yellow ->red card	15.73	19.93	**24.39**	21.06	12.08	5.72

5. Conclusion

In this paper, we propose a novel model based on transformer with the pre-content and post-content encoders and obtain 52.94% average-mAP. By setting an appropriate chunk size for each class, we reach 54.6% average-mAP on the test dataset, exceeding the current state-of-the-art by 1.2%. Our future works will focus on improving the performance of the few-shot class for increasing understanding of videos.

References

[1] A. Deliège, *et al.*, "SoccerNet-v2 : A Dataset and Benchmarks for Holistic Understanding of Broadcast Soccer Videos", CVPRW, 2021.

[2] S. Giancola, *et al.*, "SoccerNet: A Scalable Dataset for Action Spotting in Soccer Videos", CVPRW, 2018.

[3] S. Giancola, *et al.*, "Temporally-Aware Feature Pooling for Action Spotting in Soccer Broadcasts", arXiv:2104.06779.

[4] A. Vaswani, *et al.*, "Attention is All you Need", NeurIPS, 2017.

[5] M. Tomei, *et al.*, "RMS-Net: Regression and Masking for Soccer Event Spotting", ICPR, 2021.

[6] A. Cioppa, *et al.*, "A Context-Aware Loss Function for Action Spotting in Soccer Videos", CVPR, 2020.

[7] A. Relja, *et al.*, "NetVLAD: CNN architecture for weakly supervised place recognition", CVPR, 2016.

Research Achievements

[1] Y. Shi, *et al.*, "Action Spotting in Soccer Videos via Transformer with Past and Future Encoders", MIRU, 2021.

(Other: 1 conference presentation)