Action Spotting and Temporal Attention Analysis in Soccer Videos

Hiroaki Minoura, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi
Chubu University
{himi1208, hirakawa@mprg.cs, takayoshi, fujiyoshi@isc}.chubu.ac.jp

Mitsuru Nakazawa, Yeongnam Chae, Björn Stenger
Rakuten Institute of Technology, Rakuten Group, Inc.
{mitsuru.nakazawa, yeongnam.chae, bjorn.stenger}@rakuten.com

Abstract

Action spotting is the task of finding a specific action in a video. In this paper, we consider the task of spotting actions in soccer videos, e.g., goals, player substitutions, and card scenes, which are temporally sparse within a complete game. We spot actions using a Transformer model, which allows capturing important features before and after action scenes. Moreover, we analyze which time instances the model focuses on when predicting an action by observing the internal weights of the transformer. Quantitative results on the public SoccerNet dataset show that the proposed method achieves an mAP of 81.6\%, a significant improvement over previous methods. In addition, by analyzing the attention weights, we discover that the model focuses on different temporal neighborhoods for different actions.

1 Introduction

Action spotting is the task of finding a specific action in a video and is an important part of video understanding. With the recent explosive growth in the number of online videos, efficient methods for search, classification, and further analysis are needed.

A number of video datasets have been introduced, where each video contains multiple different actions [1, 2, 3, 4, 5, 6, 7, 8]. Each video is annotated with action labels and the corresponding time instances or intervals. One challenge is that in many cases actions are temporally sparse in a video, such as goals, substitutions and card scenes in SoccerNet [7]. Much of the video contains game play without any of these actions of interest: The average number of goals in professional soccer games is 2-3 (2.6 at the most recent World Cup [9]). In order to detect such sparse events, it is important to understand the temporal context of the action. For example, a goal can be recognized after the shot itself, when players celebrate and a slow motion replay is shown. In a substitution scene, players leave and enter the pitch shortly after the referee raises the board indicating the player numbers.

Prior work proposed efficient action spotting methods for soccer video [7, 10, 11], which addressed the data imbalance and sparsity of action classes by using appropriate loss functions and temporal pooling of video features extracted by convolutions. For sparse action spotting it is important to capture different related scenes, i.e., the temporal context of each action, which has not fully been captured in prior work.

The goal of this paper is to improve the accuracy of action spotting by allowing the model to automatically use the correct temporal features. Typically, temporal features are extracted using an LSTM auto-encoder or CNN convolutions in the temporal direction. However, LSTM auto-encoders are known to struggle with capturing features over longer time periods [12, 13]. CNN convolutions along the time axis can be considered temporal pooling as in previous work. In this work, we adapt the transformer model [14] to model context over longer time periods in SoccerNet. The transformer captures features by calculating the similarity of feature vectors at different times. Moreover, it is possible to analyze the temporal attention for action spotting by analyzing the attention weight obtained by the transformer. This allows us to clarify the explanation for the action predicted by the model.

In summary, the main contributions of this work are:

- By using temporal attention of each action our transformer model improves the action spotting accuracy using a transformer, achieving 81.6\% mAP, a significant improvement over prior work, shown in Section 4.2.
- To the best of our knowledge, this work analyzes temporal attention for action spotting on soccer video for the first time, see Section 4.2.
- As a result of the internal analysis of the transformer, we discovered that the temporal attention was different for each action label, shown in Section 4.2.

2 Related Work

In this section, we review recent work on video understanding with a focus on action spotting in sports videos.

2.1 Video Understanding

Video understanding includes a number of approaches, such as action recognition [1, 15, 16], spatio-temporal lo-
Figure 1: Proposed Network Architecture. Our model adopts a transformer as a backbone model. The red rectangle frame shows the annotated frame for an action label. The transformer spots appropriate labels in soccer videos using a ResNet-based features extractor (FE), in our case applied to 120 frames extracted every 0.5 seconds. The images are cited from [7].

calization [17, 18, 19, 20], and video classification [6, 17, 21, 22]. Video classification is the task of predicting a label that is relevant to the video. Most recent work on video classification is based on 3D convolutional networks and recurrent neural networks. Donahue et al. [22] proposed a method of propagating CNN feature vectors in the temporal direction using an LSTM. Karpathy et al. [6] used a CNN architecture to learn spatio-temporal features for action classification. Much of the video classification work assigns a single label to the entire video.

Recently, action spotting has been studied as the task of predicting labels of a specific scene in the video [7, 8, 10, 11, 23]. Action spotting is important for long-term summarization, e.g., of cooking or sports videos. Giancola et al. [7] provide a baseline of action spotting tasks on soccer video by introducing pooling and context gating layers [24, 25, 26, 27]. Tomei et al. [11] proposed a method for action spotting and approximate temporal offset regression by taking the maximum features in the time direction after convolution. These methods improve the accuracy by capturing temporal features, however, they do not consider different related scenes for each action in the input video. Here we consider action-related scenes by using a transformer to calculate the similarity between different scenes for each action.

3.2 Datasets


3 Method

In this section, we describe the model for sparsely annotated data and the temporal attention analysis.

3.1 Overview

The architecture of the proposed network is illustrated in Figure 1. Our method consists of a feature extractor and a transformer encoder. The transformer receives as input a sequence of embedding vectors. First, we convert 2D images $x_t \in \mathbb{R}^{H \times W \times C}$, $t = 1, 2, ..., T$ into $d$-dimensional feature vectors $h_t = f(x_t)$. $(H \times W \times C)$ is the resolution of the original image, and $T$ are the input time steps. As proposed in [14] we inject temporal information by adding positional encoding vectors to each feature vector. These vectors are of the same dimension, $d_{\text{model}}$, as the feature vectors, and the components are sinusoids of different wavelength:

$$PE(t, 2i) = \sin\left(\frac{t}{10000^{\frac{2i}{d_{\text{model}}}}}\right)$$

$$PE(t, 2i+1) = \cos\left(\frac{t}{10000^{\frac{2i}{d_{\text{model}}}}}\right),$$

where $t$ is the time index and $i \in \{0, ..., \left\lfloor \frac{d_{\text{model}}}{2} \right\rfloor\}$ is the index of the positional encoding. This results in a sequence, $p_t$, of feature vectors with embedded positional encodings. The transformer learns latent features for the action labels for each time step via a self-attention mechanism with $L$ layers and $N$ heads. The self-attention first learns the query matrix $Q = \phi_q(\{p_t\}_t)$, the key matrix $K = \phi_k(\{p_t\}_t)$ and the value matrix $V = \phi_v(\{p_t\}_t)$.
where $\phi_p$, $\phi_k$, and $\phi_v$ are MLP layers. It computes the attention by

$$\text{MultiHead}(Q, K, V) = \phi_v([\text{head}_i]_{i=1}^N),$$

where

$$\text{head}_i = \text{Attention}_i(Q, K, V),$$

$\phi_v$ is an MLP layer, and the Attention function is the scaled dot-product attention in [14]. The feature vector, obtained by the transformer in each time step, is averaged and passed through an MLP layer $\phi_p(\cdot)$ to obtain the action probability, $y$, as

$$m = \frac{1}{T} \sum_i^T \text{MultiHead}(Q_i, K_i, V_i),$$

$$y = \phi_p(m).$$

### 3.2 Temporal attention analysis

We use the attention weights to indirectly obtain explanations for the action predictions. Inspired by ViT [30], which focuses on spatial image features only, we analyze differences between the temporal attention and the frame annotations using attention rollout [31]. Attention rollout is an intuitive way to approximate features of interest using the attention weight from the first layer to the last layer of self-attention. Given a transformer with $L$ layers and $N$ heads, we compute the attention weight $a_{l,j}$ of all layers $l$, heads $n$, and time steps $i,j$. Note that the attention graph for $n \times T^2$ is constructed in each layer $l$. To compute the attention score $A_i$ for time step $i$, we first aggregate the attention weight of each head $n$ and add all values of the $j$-th frame focused on the $i$-th frame from the attention weight. Finally, the value is calculated as the product over all layers. We select the frame index $\hat{i}$ with the strongest temporal attention by taking the max value over time steps $i$:

$$A_i = \prod_l \left( \sum_j \sum_n a_{l,j}^{i,n} \right),$$

$$\hat{i} = \arg \max_i (A_i).$$

### 3.3 Implementation Details

The image dimensions $(H \times W \times C)$ are set to $224 \times 224 \times 3$ by resizing and cropping the input images. We extract features using a ResNet-152 [32], pre-trained on ImageNet [33]. The dimension of feature vectors is reduced to $d = 512$ using Principal Component Analysis. Features are extracted every 0.5 seconds over $T = 120$ time steps, i.e., the network takes as input features computed over 60-second intervals. Our transformer model has 8 heads and 6 layers. The feature vector dimension for self-attention is set to 512. The dimension of the output at each time step in the transformer is set to 4 via a single MLP layer $\phi_p$ [$512 \times 4$].

We trained using a cross-entropy loss, and an Adam optimizer [34] with an initial learning rate of $10^{-5}$. The model was trained for 400 epochs with a batch size of 16. We handled sparse data by weighting the cross-entropy loss as in [7], which we set to $[\text{background, card, substitution, goal}]$ to $[0.1, 0.5, 1.0, 1.0]$.

### 4 Experiments

#### 4.1 Evaluation Protocols

SoccerNet [7] is a dataset annotated with 4 actions (background, card, substitution, and goal) in 500 soccer videos. The background class accounts for about 70% of all data samples. Following [7], we use a 300/100/100 split for training, validation, and testing, respectively. We compare different feature representations and various pooling methods. Mean and max-pooling are pooling methods to obtain a $d$-dimensional feature vector at each time step. CNN denotes a method convolving $d \times T$ dimension in the time direction. LSTM is a method to spot via 3 layers (input-LSTM-output) to feature vector taken by a feature extractor. SoftDBOW [20], NetFV [25], NetRVLAD [24], and NetVLAD [35] leverage context gating layers, to reweight both the features and the output labels. We use publicly available code to compare these methods with the proposed one. We use classification mAP and AP as evaluation metrics, and analyze the temporal attention for each correctly spotted action by ranking via attention rollout.

#### 4.2 Results

Table 1 shows the mAP and AP scores on SoccerNet. Our method achieves an mAP of 81.6%, an absolute mAP improvement of 26.6% over the next best method in the comparison. The method also achieves the best average precision (AP) for each of the classes. Note that card scenes are detected with lower accuracy than goals or substitutions. We believe this is due to cases when no close-up of the referee is shown and the region where the action takes place is small within the images.

We denote the attention weight obtained by attention rollout the ‘attention score’. Figure 2 shows the visualization of the attention score in a transformer. We show that actions were spotted while focusing on different time steps for different action labels. In Fig. 2(a) and (d), we show that the largest attention score is at the time when the referee raises the yellow card. As a result, the annotated label frame and attention match. Fig. 2(b) and (e) show that for substitutions attention is focused on frames before the annotated frame. The model focuses on the board indicating player numbers prior to the annotated frame. Finally, Fig. 2(c) and (f) show that for goals the attention is focused on the frames after the annotated frame, highlighting the replay and player actions after the goal itself.
Figure 2: Visual examples of attention score. The attention score plotted over time for different scenes and action labels. Red boxes show the annotated frame, green boxes show the time with the highest attention score. Circles (labeled I, ..., V) on the time axis correspond to the five frames below, in left-to-right order. In all cases, the attention score has strong peaks, highlighting discriminative frames for each action. In cases (a) and (d), the two are the same, in the other cases, the highest attention score is at a different time from the annotated frame. The images are cited from [7].

Table 1: Spotting results (mAP and AP) on SoccerNet.

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP</th>
<th>Card</th>
<th>Sub</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Pool.</td>
<td>35.1</td>
<td>25.7</td>
<td>38.6</td>
<td>41.1</td>
</tr>
<tr>
<td>Max Pool.</td>
<td>52.4</td>
<td>52.4</td>
<td>52.9</td>
<td>51.9</td>
</tr>
<tr>
<td>CNN</td>
<td>25.4</td>
<td>21.7</td>
<td>26.6</td>
<td>27.9</td>
</tr>
<tr>
<td>FC</td>
<td>52.4</td>
<td>52.4</td>
<td>52.9</td>
<td>51.9</td>
</tr>
<tr>
<td>LSTM</td>
<td>48.7</td>
<td>49.9</td>
<td>50.5</td>
<td>45.6</td>
</tr>
<tr>
<td>SoftDBOW [26]</td>
<td>48.0</td>
<td>36.0</td>
<td>56.8</td>
<td>51.3</td>
</tr>
<tr>
<td>NetFV [25]</td>
<td>52.7</td>
<td>35.0</td>
<td>64.2</td>
<td>58.9</td>
</tr>
<tr>
<td>NetRVLAD [24]</td>
<td>52.3</td>
<td>40.5</td>
<td>51.4</td>
<td>55.1</td>
</tr>
<tr>
<td>NetVLAD [35]</td>
<td>55.0</td>
<td>44.5</td>
<td>62.6</td>
<td>58.0</td>
</tr>
<tr>
<td>Ours</td>
<td>81.6</td>
<td>63.3</td>
<td>94.3</td>
<td>87.1</td>
</tr>
</tbody>
</table>

Table 2: Attention rollout results. Shown is the average time difference (seconds) between the annotated time and the n-th largest attention weight for each label over all samples.

<table>
<thead>
<tr>
<th>Ranking</th>
<th>Card</th>
<th>Sub</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-1</td>
<td>1.39</td>
<td>-2.69</td>
<td>10.84</td>
</tr>
<tr>
<td>Top-2</td>
<td>3.02</td>
<td>-1.40</td>
<td>11.81</td>
</tr>
<tr>
<td>Top-3</td>
<td>2.43</td>
<td>-5.47</td>
<td>11.60</td>
</tr>
<tr>
<td>Top-4</td>
<td>1.63</td>
<td>-4.52</td>
<td>8.86</td>
</tr>
<tr>
<td>Top-5</td>
<td>-0.02</td>
<td>-2.73</td>
<td>15.35</td>
</tr>
</tbody>
</table>

Table 2 shows the differences between the time of largest attention and the annotated frame of each action. Note that the ranking in this table is the average of the differences between the n-th largest attention score and the annotated frame for each label over all samples. In Table 2 we discover that the interesting time of each action varies. For example, the card scene attention is highest around the same or shortly after the annotated frame, focusing on scenes when the referee holds up the card, see Fig. 2(a) and (d). For player substitution, the highest attention score is before the annotated frame. This is because the model focuses on the board for player change before the annotated label frame, see Fig. 2(b) and (e). For goal scenes, the highest attention is on frames a few seconds after the goal shot itself. This is because our model focused on replay and player actions, shown in Fig. 2(c) and (f).

5 Conclusion

In this paper, we proposed a method for action spotting with a transformer to capture related scenes for actions in soccer videos. We demonstrate the effectiveness of the method on SoccerNet, where it outperforms previous work on action spotting. Furthermore, we show that temporal attention is able to highlight discriminative features in the temporal neighborhood of each action. An avenue for future work is extending the application scope of the model to more action classes and more general types of sports and action videos.
References


