Action Spotting using Dense Detection Anchors Revisited: Submission to the SoccerNet Challenge 2022

João V. B. Soares Avijit Shah
Yahoo Research

Abstract

This technical report describes our submission to the Action Spotting SoccerNet Challenge 2022. The challenge is part of the CVPR 2022 ActivityNet Workshop. Our submission is based on a method that we proposed recently, which focuses on increasing temporal precision via a densely sampled set of detection anchors. Due to its emphasis on temporal precision, this approach is able to produce competitive results on the tight average-mAP metric, which uses small temporal evaluation tolerances. This recently proposed metric is the evaluation criterion used for the challenge. In order to further improve results, here we introduce small changes in the pre- and post-processing steps, and also combine different input feature types via late fusion. This report describes the resulting overall approach, focusing on the modifications introduced. We also describe the training procedures used, and present our results.

1. Introduction

This report describes our submission to the action spotting track of the SoccerNet Challenge 2022. Our submission is largely based on our recently proposed method [4]. It uses a dense set of detection anchors, with the goal of increasing the temporal precision of the detections. This method is briefly reviewed here in Section 2. It consists of a two-phase approach, in which first a set of features is computed from the video frames, and then a model is applied on those features to infer which actions are present in the video. Section 3 describes the different features we experimented with, which are derived from previously published ones [2, 6]. Section 4 describes our experimental protocols and training procedures, Section 5 describes the post-processing that we applied (which is based on Soft-NMS [1]), and Section 6 presents our results.

2. Action spotting via dense detection anchors

For our submissions, we used the recent action spotting approach proposed by Soares et al. [4]. The approach uses a dense set of anchors in order to produce temporally precise detections. Each anchor is defined as a pair formed by a time instant and an action class. These are taken at regularly spaced time instants, at the same frequency as the input feature vectors, usually 1 or 2 per second. For each anchor, both a detection confidence and a fine-grained temporal displacement are inferred, with the temporal displacement indicating exactly when an action was predicted to happen, thus further increasing the temporal precision of the results. Both types of outputs are then combined by displacing each confidence by its respective predicted temporal displacement. Finally, a non-maximum suppression (NMS) step is applied to obtain the final detections. For the trunk of the model, we used the 1D version of the u-net, which was shown to be significantly faster than a Transformer Encoder alternative, while producing similar results [4].

3. Features

Our method uses a standard two-phase approach, consisting of feature extraction followed by action spotting. It was implemented in such a way that the set of possible time instants for the detections is the same as that of the extracted feature vectors. Our experiments use the features provided by Zhou et al. [6], which we refer to here as *Combination* features, given that they were produced by concatenating the features from a series of different fine-tuned models. These features were originally computed at 1 feature vector per second, which, for our method, results in output detections of the same frequency. We noticed that this frequency was too low to appropriately compute the tight average-mAP metric, whose tolerance radius (equal to half the tolerance window size δ) increases in half-second increments. We thus resampled the Combination features using linear interpolation, to the desired frequency of 2 feature vectors per second.

We also experimented with combining the resampled Combination features with the ResNet features from Deliege et al. [2], which were already available at the desired frequency of 2 per second. As a straightforward way of combining these features, we adopted a late fusion approach. To obtain the fused detection confidences, we cal-

Features	Experimental protocol	Validation tight a-mAP	Validation a-mAP	Challenge tight a-mAP	Challenge a-mAP
Combination×2	Challenge Validated	65.8	76.5	65.1*	75.9*
Combination×2	Challenge Validated	65.6	76.3	64.6*	74.9*
Combination×2	Challenge	-	-	67.0*	77.3*
Combination×2	Challenge	-	-	66.8*	76.8*
Combination×2 + ResNet	Challenge	-	-	67.6*	77.9*
Combination×2 + ResNet	Challenge	-	-	67.8*	78.0*

Table 1. Results using different protocols for training, validation, and testing, and using different sets of input features. For each configuration determined by a set of features and experimental protocol, we have two runs. Each run uses a different random initialization during training, leading to small variations in the results. Combination×2 refers to the Combination features from Zhou et al. [6] resampled to 2 feature vectors per second, while Combination×2 + ResNet refers to our late fusion approach. * Results provided by the evaluation server.

culated a weighted average of the logits of the confidences from two previously trained models: one trained on the resampled Combination features, and the other trained on the ResNet features. The optimal weights were found through exhaustive search on the validation set. In a similar manner, we experimented with fusing the temporal displacements from models trained on the different feature types. However, this did not provide any improvement, so we opted to use just the single temporal displacement model trained on the resampled Combination features.

4. Experimental protocol and model training

We used two different protocols in our experiments, each combining the SoccerNet data splits in different ways. The dataset provides the following pre-determined set of splits: training, validation, test, and challenge. The challenge split labels are not provided, so metrics on this split can only be obtained by submitting results to the evaluation server. Our first protocol, named *Challenge Validated*, trains on the training and test splits, runs validation on the validation split, and tests on the challenge split. Our second protocol, named *Challenge*, trains on all the available labeled data (training, validation, and test splits), does not include any validation, and tests on the challenge split. The hyperparameters for this last protocol are taken directly from those found using the *Challenge Validated* protocol.

Our confidence and temporal displacement models each have 22.9M parameters when built on the resampled Combination features, and 18.5M when built on the ResNet features. For each model, following [4], we use the validation set to tune each of the hyper-parameters in turn: the learning rate, ρ for Sharpness-Aware Minimization (SAM) [3], weight decay, and α for mixup data augmentation [5].

5. Soft non-maximum suppression

To post-process the detection results, we applied a one-dimensional version of Soft Non-Maximum Suppression (Soft-NMS) [1]. Soft-NMS was originally proposed for post-processing object detection results, as a generalization of the well-known NMS algorithm. Whereas standard NMS will remove any detections that have a high overlap with an accepted high-confidence detection, Soft-NMS opts to instead only *decay* the confidences of those overlapping detections, with the decay being proportional to the overlap. To adapt this to the one-dimensional action spotting setting, we define the decay as being proportional to the temporal distance between detections. Namely, given an accepted high-confidence detection at time t, for any detection of the same class at time s, we define the decay function as $f(s) = \min\{|s - t|/({}^{w}\!/_{2}), 1\}$, where w is a window size that we choose on the validation set. The confidences are then updated by multiplication with the decay function f.

6. Results

All the results presented here were obtained using Soft-NMS. Generally, applying Soft-NMS gives a very small but consistent improvement relative to standard NMS. For example, when using the *Challenge Validated* protocol, on the validation set, one of our models produced 65.78 tight average-mAP using Soft-NMS, versus 65.46 when using standard NMS, where each result corresponds to its respective optimal window size.

Table 1 presents all our results. The table shows an improvement in the challenge tight average-mAP of around 2.1 when switching protocols from *Challenge Validated* to *Challenge* protocol, most likely due to the latter having more training data available. When we use late fusion to add the ResNet features to the resampled Combination features, we see a further average improvement of 0.8 on the challenge tight average-mAP, leading to our best submitted results.

Acknowledgements We are grateful to Topojoy Biswas and Gaurav Srivastava for insightful discussions and feedback.

References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-NMS – improving object detection with one line of code. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5561–5569, 2017. 1, 2

[2] Adrien Deliege, Anthony Cioppa, Silvio Giancola, Meisam Seikavandi, Jacob Dueholm, Kamal Nasrollahi, Bernard Ghanem, Thomas Moeslund, and Marc Van Droogenbroeck. SoccerNet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In *Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4508–4519, 2021. 1

[3] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[4] João V. B. Soares, Avijit Shah, and Topojoy Biswas. Temporally precise action spotting in soccer videos using dense detection anchors. *arXiv preprint arXiv:2205.10450*, 2022. 1, 2

[5] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[6] Xin Zhou, Le Kang, Zhiyu Cheng, Bo He, and Jingyu Xin. Feature combination meets attention: Baidu soccer embeddings and Transformer based temporal detection. *arXiv preprint arXiv:2106.14447*, 2021. 1, 2