machine learning & knowledge extraction

MDPI

Article

Automated Event Detection and Classification in Soccer: The Potential of Using Multiple Modalities

Olav Andre Nergård Rongved [1,2], Markus Stige [3], Steven Alexander Hicks [1,2], Vajira Lasantha Thambawita [1,2], Cise Midoglu [2,*], Evi Zouganeli [1], Dag Johansen [4], Michael Alexander Riegler [2,4] and Pål Halvorsen [2,5,*]

1 Department of Computer Science, Oslo Metropolitan University, 0167 Oslo, Norway; olav.rongved@gmail.com (O.A.N.R.); steven@simula.no (S.A.H.); vajira@simula.no (V.L.T.); evizou@oslomet.no (E.Z.)
2 SimulaMet, 0167 Oslo, Norway; michael@simula.no
3 Department of Informatics, University of Oslo, 0373 Oslo, Norway; markusstige@gmail.com
4 Department of Computer Science, UIT The Arctic University of Norway, 9037 Tromsø, Norway; dag.johansen@uit.no
5 Forzasys AS, 0167 Oslo, Norway
* Correspondence: cise@simula.no (C.M.); paalh@simula.no (P.H.)

Abstract: Detecting events in videos is a complex task, and many different approaches, aimed at a large variety of use-cases, have been proposed in the literature. Most approaches, however, are unimodal and only consider the visual information in the videos. This paper presents and evaluates different approaches based on neural networks where we combine visual features with audio features to detect (spot) and classify events in soccer videos. We employ model fusion to combine different modalities such as video and audio, and test these combinations against different state-of-the-art models on the SoccerNet dataset. The results show that a multimodal approach is beneficial. We also analyze how the tolerance for delays in classification and spotting time, and the tolerance for prediction accuracy, influence the results. Our experiments show that using multiple modalities improves event detection performance for certain types of events.

Keywords: audio; video; multimodality; event classification; event detection; machine learning; soccer

Citation: Nergård Rongved, O.A.; Stige, M.; Hicks, S.A.; Thambawita, V.L.; Midoglu, C.; Zouganeli, E.; Johansen, D.; Riegler, M.A.; Halvorsen, P. Automated Event Detection and Classification in Soccer: The Potential of Using Multiple Modalities. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 1030–1054. https://doi.org/10.3390/make3040051

Academic Editors: Antonio Fernández-Caballero, Byung-Gyu Kim and Hugo Pedro Proença

Received: 2 November 2021
Accepted: 10 December 2021
Published: 16 December 2021

1. Introduction

The generation of video summaries and highlights from sports games is of tremendous interest for broadcasters as a large percent of audiences prefer to view only the main events in a game. Having worked closely with and observed soccer domain experts for a decade, and lately been part of building a soccer tagging operation center used in the Norwegian and Swedish elite leagues, we see the need for high-performance automatic event detection systems in soccer. The current manual annotation process, where a group of people annotates video segments as they are aired during a live broadcast, is tedious and expensive. The "holy grail" of event annotation has long been anticipated to be a fully automated event extraction system. Such a system needs to be timely and accurate, and must capture events in near real-time.

One of the key components of such a comprehensive pipeline is the detection and classification of significant events in real-time. Existing approaches for this task need significant improvement in terms of detection and classification accuracy. Much work has been done on using visual information to detect events in soccer videos, and promising examples with a good detection performance [1–4] are available. However, in order to be usable in practice, the detection performance must be high, and for events such as goals in soccer, 100% (perfect) detection is required for any method to be considered for deployment as an official tool.

In this paper, we aim to develop an intelligent soccer event detection and classification system using machine learning (ML). In particular, we explore the potential in the data

itself to improve the performance of both *event detection* (also called *spotting*), and *event classification*. Typically, a data stream contains several types of data. For example, a sports broadcast might be composed of several video and audio streams. The final broadcast consumed by the viewers typically consists of a video stream created from various camera views and an audio stream containing, for example, the commentators' voices with the background audio from the audience in the stadium. However, research on detecting particular events has traditionally focused solely on the visual modality of the video. Even in multimedia venues, researchers have for a long time presented work based on video only. Thus, our objective in this study is to evaluate whether the narrow focus of existing multimedia research limits the potential for high-performance event detection, by not utilizing all available modalities.

Using the SoccerNet dataset [1], from which a number of samples are presented in Figure 1, we evaluate two previous unimodal approaches focusing on video [2,4] and experiment with different parameters, showing the trade-offs between accuracy and latency. Subsequently, we evaluate the performance of an audio model, where the audio is sampled and transformed into a Log-Mel spectrogram [5], which is analyzed using a 2D-ResNet model. Following this single-modality event analysis, we fuse the models using softmax predictions. Our results show the various trade-offs between different requirements, but more importantly, that existing state-of-the-art visual models achieve increased performance when a causally related audio analysis is added. Hence, we prove the importance of *multi*media analysis in future event detection systems. The main contributions of this work are as follows:

(**a**) Card.

(**b**) Substitution.

(**c**) Goal.

Figure 1. Sample frames from the SoccerNet dataset [1] for three different event types. The middle frame is at the annotated time for the event.

- We implement, configure, and test two state-of-the-art visual models [2,4] built on the ResNet model and assess their detection performance of various soccer events, such as cards, substitutions, and goals, using the SoccerNet dataset [1]. As expected, adding higher tolerance (enabling the models to use more data) improves the performance, trading off event processing delay for event detection (spotting) accuracy.

- We implement a Log-Mel spectrogram-based audio model, testing different audio sample windows and assessing the model's event detection and classification performance. The results show that using the audio model alone gives poor performance compared to using the visual models.
- We combine the visual and audio models, proving that utilizing all the potential of the data (multiple modalities) improves the performance. We observe a performance increase for various events under different configurations. In particular, for events such as goals, the multimodal approach is superior. For other events such as cards and substitutions, the gain depends more on the tolerances, and in some cases, adding audio information can be detrimental to the performance.

The rest of the paper is structured as follows. In Section 2, we provide an overview of related work. We present our ML models in Section 3 and show how we combine them to achieve a multimodal system for soccer event classification in Section 4. In Section 5, we present our experiments and discuss various aspects in Section 6. We summarize our results and conclude the paper in Section 7.

2. Related Work

This section briefly introduces relevant research on action detection and, more specifically, the automatic detection of events from soccer videos.

2.1. Action Detection and Localization

Computer vision has been an active field of research for many decades. One of the most important areas of computer vision is video understanding. There are several interesting subareas of video understanding, but the two most relevant in our context are *action recognition* and *action detection*, i.e., aiming to find what actions occur and at what time, in videos.

Several different approaches have been proposed for the task of action recognition. In earlier works, features such as histogram of oriented gradients (HOG), histogram of flow (HOF), motion boundary histograms (MBH) [6], and dense trajectories [7,8] have shown promising results. More recently, proposed approaches have started to use some variant of neural networks. For example, Karpathy et al. [9] provided an empirical evaluation of convolutional neural networks (CNNs) on large-scale video classification using a dataset of 1 million YouTube videos belonging to 487 classes. Multiple CNN architectures were studied to find an approach that combines information across the time domain. The CNN architecture was modified to process two streams to improve runtime performance, a context stream and a *fovea* stream. Moreover, the stream was cropped at the center of the image and down-sampled. The runtime performance was increased by up to 4×, and the accuracy of classification was increased by about 60–64% compared to the 2014 state-of-the-art. The learned features generalized to a different smaller dataset, showing benefits of transfer learning. Simonyan and Zisserman [10] also used a two-stream convolutional network, related to the *two-streams hypothesis* [11], and their performance exceeded the previous works using deep neural networks, by a large margin. They used two separate CNNs, with a spatial CNN [12] pre-trained on ImageNet [13] using RGB input by sampling frames from video, and the temporal stream using optical flow fields as input.

Tran et al. [14] proposed 3-dimensional convolutional networks (3D-CNNs) and found out that 3D CNNs are more suitable for spatio-temporal feature learning than 2D CNNs, also having compact features. Later, Tran et al. [15] introduced a new spatiotemporal convolutional block R(2+1)D, which is based on (2+1)D convolutions separating the 3D convolution into two steps. The idea is that it may be easier for the network to learn spatial and temporal features separately. Furthermore, Feichtenhofer et al. [16] have introduced the SlowFast architecture. The model is based on the use of two different frame rates as input, where the idea is to have a high-capacity *slow* pathway that sub-samples the input heavily, and a *fast* pathway that has less capacity but significantly higher frame rate.

Carreira and Zisserman [17] introduced an inflated 3D-CNN (I3D) that is expanded from the 2D CNN inflation, using both 3D convolution and the inception architecture [18]. I3D works much like the two-stream networks, using one network for the RGB stream input and one for the optical flow. To enable 3D convolution, they inflated filters on a pre-trained 2D CNN, i.e., they used transfer learning to initialize the 3D filters. Feichtenhofer et al. [19] explored the fusion process with 3D convolution and 3D pooling. The two-stream architecture is also extended with ST-ResNet [20], which adds residual connections [21] in both streams and from the motion stream to the spatial stream. Research into a combination of hand-crafted features and deep-learned features has also shown promising results [22]. Wang et al. [23,24] proposed temporal segment networks (TSNs), arguing that existing models mostly focused on short-term motion rather than on long-range temporal structures. TSN takes a video input, separates it into multiple snippets, and makes a prediction using two-stream networks for each snippet. C3D [14] explored learning spatio-temporal features with 3D convolutional networks. When compared to existing 2D convolution solutions, 3D convolution indeed proves beneficial by adding temporal information such as motion.

Donahue et al. [25] used long-term recurrent convolutional networks and models combining CNN as a feature extractor, and they fed the features to a long short-term memory (LSTM) model. This approach produced comparable results to Simonyan and Zisserman's two-stream approach on the UCF101 dataset [26]. Qiu et al. [27] presented Local and Global Diffusion (LGD) networks. That is a novel architecture for the learning of spatio-temporal representations capturing long-range dependencies. LGD networks have two paths: a local and a global path for each spatio-temporal location. The local path describes local variation, and the global path describes holistic appearance. The LGD network outperformed several state-of-the-art models at the time on benchmarks, including UCF101 [26], where it reached an accuracy of 98.2%. Kalfaoglu et al. [28] combined 3D convolution with late temporal modeling. With the use of bidirectional encoder representations from transformers (BERT) instead of temporal global average pooling (TGAP), they increased performance for 3D convolution, achieving an accuracy of 85.10% and 98.69% for the HMDB51 [29] and UCF101 datasets, respectively.

Action detection aims to find the time interval in a video where a specific action occurs and to return these time tags along with a classification of the action. The task can be divided into two parts: generating temporal region proposals (also referred to as *temporal action localization*) and classifying the type of action taking place within the proposed time frame. In many studies, these two aspects are considered separately [23,30–32], but they are also adressed jointly with a single model in other works [33,34]. Some success has been observed with sliding window approaches [35]. However, this is computationally expensive and lacks flexibility due to fixed window sizes. Some works have focused on generating the temporal proposals, which can then be used with a classifier [36,37]. Inspired by Faster R-CNN [38], Xu et al. [39] created an end-to-end model for temporal action detection that generates temporal region proposals, followed by classification. There are two approaches to the proposal generation task: *top down* and *bottom up*. The *top down* approach has been used more frequently in previous works [31,40,41] and involves pre-defined intervals and lengths. This has the problems of boundary precision and the lack of flexibility in duration. Some methods have used the *bottom up* approach, like TAG [31] and BSN [37]. The drawback for them is the lack of ability to generate sufficient confidence scores for retrieving proposals. Recently, Lin et al. [37] addressed this challenge by introducing boundary-matching network (BMN), which generates proposals with more precise temporal boundaries and more reliable confidence scores. Combined with an existing action classifier, they reported state-of-the-art performance for temporal action localization.

In this work, we focus on action detection in the context of sports, and, more specifically, soccer videos. We refer to this task as *event detection* (also called spotting), which serves the purpose of identifying actions of interest in soccer games such as goals, player substitutions, and yellow/red cards (bookings).

2.2. Event Detection in Soccer Videos

Researchers have worked on capturing and analyzing soccer events for a long time, for various applications ranging from the verification of goals through goal-line technology [42], to player and ball tracking for segmentation and analysis [43–45], and the automatic control of camera movements for following the gameplay [46]. Events have also been extracted offline using metadata from news and media [47]. In this work, we are interested in the extraction of game happenings such as goals, cards, and substitutions in real-time (from live video streams), and in this context, the detection of events in soccer videos is not a new concept. Earlier approaches for the automated detection of selected events have employed probabilistic models such as Bayesian Networks [48,49], and hidden Markov model (HMM) [50–54]. With the rise of machine learning, there has been growing interest in using support vector machine (SVM) [55–60] and deep learning [45,61–65] due to their relatively higher performance (detection accuracy) and the availability of more advanced computing infrastructures.

After the release of SoccerNet by Giancola et al. [1], works on action detection in soccer videos through neural networks have gained prominence over solutions based on SVMs. SoccerNet was released with a baseline model reaching a mean average precision (mAP) of 49.7% for tolerances ranging from 5 to 60 s for the task of spotting, which has been defined by the authors as finding the temporal anchors of soccer events in a video. Giancola et al. proposed a sliding window approach at 0.5 s stride, using C3D [14], I3D [17], and ResNet [21] as fixed feature extractors. Rongved et al. [2,3] used a ResNet 3D model pre-trained on the Kinetics-400 dataset [66] and reported an average-mAP of 51%, which is an increase from the baseline provided in Giancola et al. [1]. Rongved et al. further showed that the model generalized to datasets containing clips from the Norwegian soccer league Eliteserien and the Swedish soccer league Allsvenskan. The results showed that, in clips containing goals, they could classify 87% of the samples from Allsvenskan and 95% of the samples from Eliteserien, with a threshold of 0.5. Cioppa et al. [4] introduced a novel loss function that considers the temporal context present around the actions. They addressed the spotting task by using the introduced contextual loss function in a temporal segmentation module, and a YOLO [67]-like loss for an action spotting module creating the spotting predictions. This approach increased the average-mAP for the spotting task to 62.5% and is currently considered the state-of-the-art for the SoccerNet spotting task. Another approach to the SoccerNet spotting task was studied in Vats et al. [68], where the authors introduced a multi-tower temporal convolutional network architecture. 1D CNNs of varying kernel sizes and receptive fields were used, and the class-probabilities were obtained by merging the information from the parallel 1D CNNs. They reported an average-mAP of 60.1%, which has a difference of 2.4% from Cioppa et al. [4] with a simpler approach using a cross-entropy loss function. More recently, Zhou et al. [69] presented a solution where they fine-tune multiple action recognition models on soccer data to extract high-level semantic features, and design a transformer-based temporal detection module to locate the target events. They achieve good results in the SoccerNet-v2 spotting challenge with an average-mAP of about 75%.

In this work, we focus on the solutions presented by Cioppa et al. [4] and Rongved et al. [2,3] as representatives of state-of-the-art models and attempt to augment them using a multimodal context that includes both video and audio information.

2.3. Multimodality

The idea of a multimodal analysis of video content goes back at least 20 years. For example, Sadlier et al. [70] performed audio and visual analysis separately, then combined the statistics of the two approaches afterwards, showing the potential of multimodal analysis. Wang et al. [48] extracted audio and visual features from sport videos to generate keyword sequences, where a traditional SVM classifier was used to group the features, and an HMM classifier automatically found the temporal change character of the event instead of rule based heuristic modeling, to map certain keyword sequences into events.

Recently, newer approaches using deep neural networks have also been presented for use cases other than soccer. Ortega et al. [71] combined audio, video, and textual features by first separately using fully connected layers, followed by concatenation. SlowFast [72] used video frames as input at different sample rates and combined these with an audio stream that takes Log-Mel spectrograms as input, with lateral connections and a special training method to avoid overfitting.

In the context of our own research (event detection and classification in soccer videos), AudioVid [73] used a pre-trained audio model to extract features and combine them with the baseline model NetVLAD [1] at different points through concatenation. They found that audio generally increased the performance, and achieved a mAP of 7.43% for an action classification task and 4.19% for an action spotting task using SoccerNet. Gao et al. [74] used a new dataset including video (460 soccer game broadcasts) and audio information (commentator voice for 160 of the games, categorized as "excited" and "not-excited") to benchmark existing methods such as I3D [17], I3D-NL [75], ECO [76], and pySlowFast [16], on detecting events ("celebrate", "goal/shoot", "card", and "pass") and found that I3D-NL achieves the best result. However, despite having a real-world deployment, the dataset from this work is not open.

In this work, we apply the idea of using multiple modalities for event detection in soccer videos, which has more frequently been demonstrated for HMMs and SVM, using deep learning models. More specifically, we focus on the state-of-the-art models from [2,4] (which improve upon the baseline model used by [73]) and augment these with a Log-Mel spectrogram-based audio model. We quantify the performance benefits for different event classes using the same SoccerNet dataset as our benchmark and point out when multimodality might be helpful and when it might actually be detrimental.

3. Tested Models

In this section, we describe the models we have experimented with for both visual and audio information. For the visual analysis, we used the model introduced by Cioppa et al. [4], the model presented by Rongved et al. [2], and a basic 2D CNN model. As mentioned in Section 2.2, the approach proposed in Cioppa et al. is currently considered the state-of-the-art for the SoccerNet spotting task, and Rongved et al. have shown that their model could generalize to multiple datasets. Thus, these can be considered as representative approaches of existing work. The 2D CNN model serves as a baseline. For the audio analysis, we have created another ResNet-based model by transforming audio signals into Log-Mel spectrograms and analyzing these.

3.1. Visual Model: CALF

As mentioned above, Cioppa et al. [4] introduced a loss function that considers the temporal context present around the actions, called context-aware loss function (CALF). Their model consists of a base convolution, a segmentation module that uses their novel loss function, and a spotting module. In the context of this work, we refer to this model as "CALF". Our implementation follows the description provided in the original paper. The inputs to the model are the ResNet [21] features provided with the SoccerNet dataset [1] (see Section 5.1 for details).

3.2. Visual Model: 3D-CNN

Based on the model presented by Rongved et al. [2], we use an 18-layered 3D-ResNet on the video frame inputs. The model is composed of several residual blocks that contain 3D convolutions, with batch-normalization and ReLU. A visual representation of the pipeline is shown in Figure 2. Additionally, the model has been pre-trained on the Kinetics-400 [66] dataset.

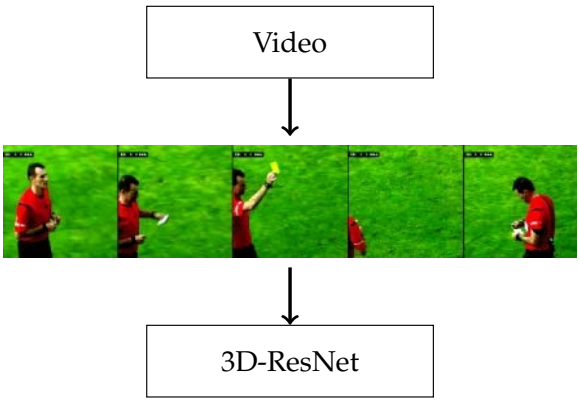


Figure 2. Illustration of the pipeline used by the video-based 3D-ResNet model.

3.3. Visual Model: 2D-CNN

We use a 2D-CNN model that uses the pre-extracted ResNet features provided by SoccerNet. A visual representation of the overall pipeline is shown in Figure 3, and further details of the model are depicted in Figure 4. The model takes in $(2 * W) \times F$ features, where W is the window size used for sampling in seconds and $F = 512$. Inspired by the approach in SoccerNet [1], we first use a 2D convolution with a kernel $1 \times F$. This is followed by batch-normalization and another 2D convolution that has a kernel of $\frac{W}{2} \times 32$, such that it has a temporal receptive field of $\frac{W}{2}$. Finally, we have two fully connected layers and an output layer.

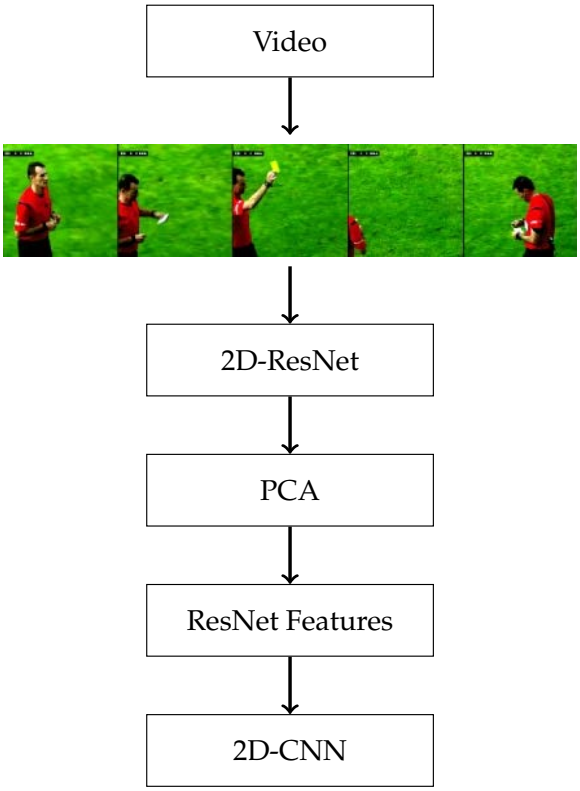


Figure 3. Illustration of the pipeline used by the video-based 2D-CNN model. A pre-trained ResNet is used to extract features from video, followed by PCA [1]. The features can then be used to train a network for action detection tasks.

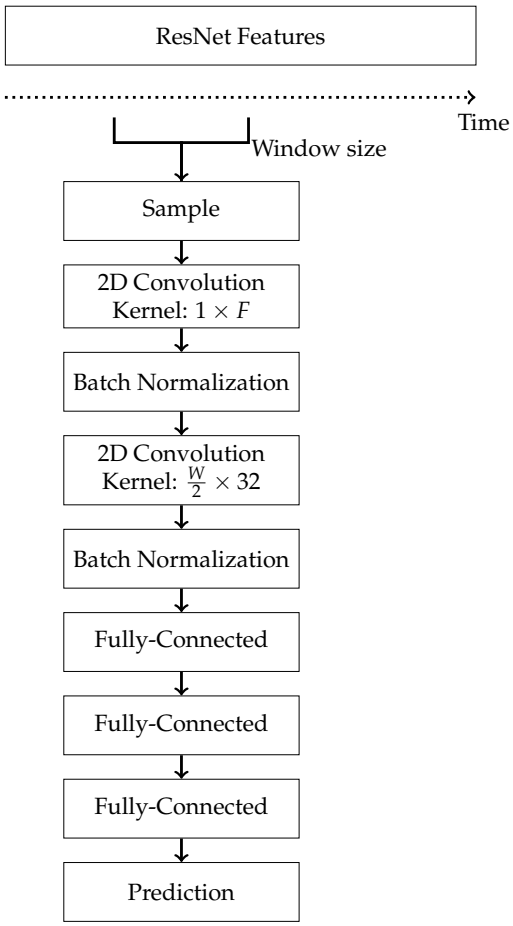


Figure 4. Detailed workflow for the 2D-CNN model. This model uses a pre-computed set of features and is tested with both visual and audio-visual (visual augmented with audio) features in our study.

3.4. Audio Model

Our audio model is based on transforming the audio into Log-Mel spectrograms [5]. Figure 5 shows the pipeline for the audio model. First, audio is extracted from video in wave-form, which again is used to generate Log-Mel spectrograms. As several other approaches using Log-Mel spectrograms, we then use a CNN as a classifier, i.e., an 18-layered 2D-ResNet [21], in the same way as we train our visual models. We also test with different window sizes over which the spectrograms are generated, representing the temporal extent of the input used.

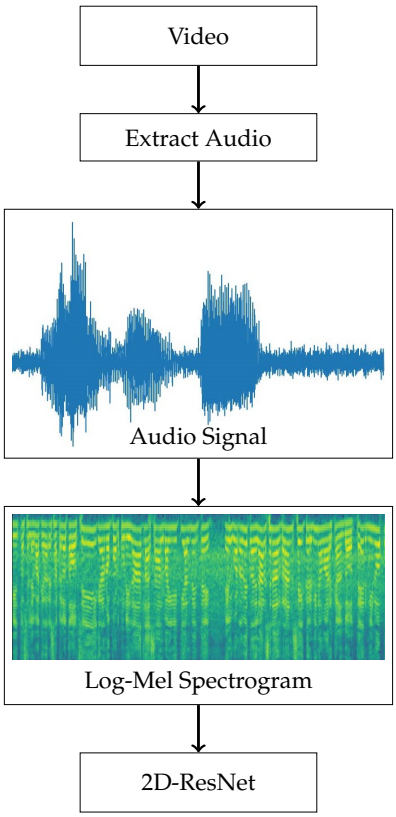


Figure 5. Illustration of the pipeline used by the audio-based model. First, audio is extracted from the video. The audio is used to compute Log-Mel spectrograms, which are then used as inputs to a 2D-ResNet.

4. Model Fusion

In order to use multiple modalities such as video and audio, different models need to be combined. There are multiple ways of fusing models. We focus on two different approaches, which are detailed below.

4.1. Early Fusion through Concatenated Features

Early fusion, also referred to as *data-level fusion* or *input-level fusion*, is a traditional way of fusing data *before* conducting an analysis [77]. Within the context of our work, this approach translates to generating concatenated audio-visual features (i.e., generating audio features and concatenating them with existing visual ResNet features), which can then be used for training. We use the ResNet features supplied in SoccerNet [1] as our visual features. For our audio features, we first train an audio model using Log-Mel spectrograms on $W = 8$. Next, we remove the last layer, resulting in a 512-dimensional feature vector. These features are calculated 2 times per second, and we concatenate them such that we align them over time, resulting in a 1024-dimensional feature vector, 2 times per second. We illustrate this process of concatenated feature generation in Figure 6.

4.2. Late Fusion through Softmax Average and Max

In *late fusion*, also referred to as *decision-level fusion*, data sources are used independently until fusion at a decision-making stage. This method might be simpler than *early fusion*, particularly when the data sources are significantly varied from each other in terms of sampling rate, data dimensionality, and unit of measurement [77].

Within the context of our work, this approach translates to fusing independently built video and audio models by taking the softmax average or max of their predictions at test time. The intuition behind this approach is that, in cases where the video model might make a strong prediction that an event has occurred, the audio model might have weaker

and made more uniform predictions, or vice versa. For *softmax average* fusion, we use the audio and video models, which have been trained on their respective inputs. For each sample, we take the average softmax prediction between the two models. For *softmax max* fusion, we calculate it similarly to softmax average, except that instead of average, we use the maximum softmax prediction between the two models. The process is illustrated in Figure 7.

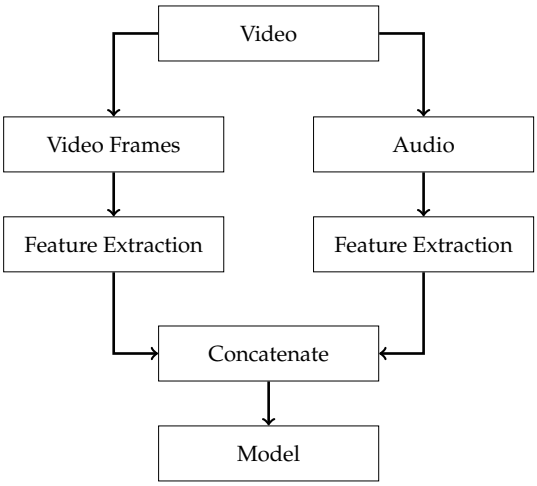


Figure 6. *Early fusion.* Illustration of how audio-visual features can be created. A ResNet is used to compute visual features based on single frames. For the audio, a Log-Mel spectrogram is used to train a 2D-ResNet and further used as a feature extractor by removing the output layer. These features are then concatenated.

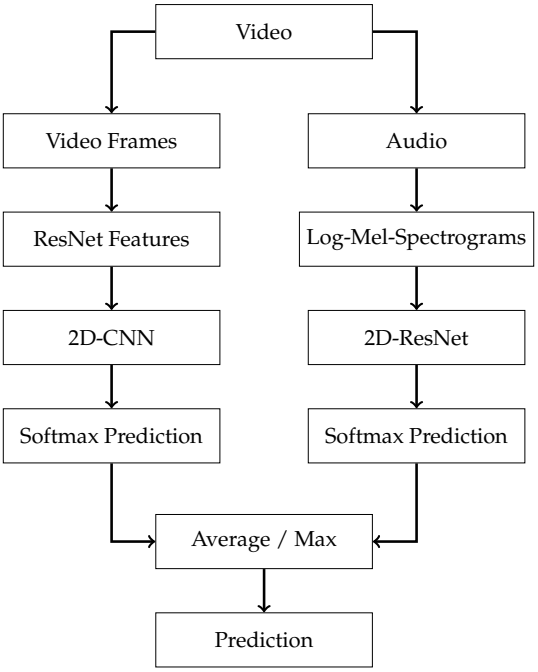


Figure 7. *Late fusion.* Visualization of how two seperate models can be fused through softmax average: a visual pathway that uses a feature extractor with a 2D-CNN, and an audio pathway that uses Log-Mel spectrograms with a 2D-ResNet. These models are trained individually, and the output predictions are fused by softmax (average or max).

5. Experiments and Results

We have performed several experiments to evaluate the performance of different models, using the SoccerNet dataset for both spotting and classification tasks.

5.1. Dataset

We used the soccer-specific SoccerNet [1] dataset released in 2018. This is a dataset containing 500 soccer games from 2014 to 2017 with games from six European elite leagues. It has a total duration of 764 h and includes 6637 annotations of the event types *goal*, (yellow/red) *card*, and *substitution*. This gives a frequency of an event happening every 6.9 min on average. Along with the three classes above, we follow the example in Rongved et al. [2] and add a *background* class by sampling in between events. If the time distance between two consecutive events is larger than 180 s, then a new background sample is added in the center, such that a background sample will never be within 90 s of another event.

In Table 1, we present the distribution of different events and how we have divided the dataset into training, validation, and test splits. The training, validation, and test splits are the same as in Giancola et al. [1], where the authors use 300 games for the training split, 100 games for the validation split, and 100 games for the test split. The augmented dataset containing the added background class is used for the classification task, for which the results are presented in Sections 5.4 and 5.5, while only the original three classes are used by the CALF model for the spotting task, for which the results are presented in Section 5.6.

Table 1. The number of samples per class in our dataset.

Class	Training	Validation	Test
Card	1296	396	453
Substitution	1708	562	579
Goal	961	356	326
Background	1855	636	653
Total	5820	1950	2011

The SoccerNet dataset also comes with extracted visual ResNet features [1]. The features have been extracted from the videos by using a ResNet-152 [21] image classifier pre-trained on ImageNet [13]. Features are extracted from single video frames at a rate of 2 times per second. Subsequently, principal component analysis (PCA) is used to reduce the feature dimension to 512. These features are used in our experiments, as illustrated in Figure 3.

In addition to the visual ResNet features provided by SoccerNet, we also extract information from the video and audio streams in the dataset directly. In Table 2, we give an overview of which type of data is used with which combination of models.

Table 2. Overview of which combination of models use which type of data, indicated by a checkmark. *ResNet features* are the visual features extracted using the ResNet model accompanying the SoccerNet dataset. *Video* is the 3D-ResNet extracted features using video frames from the games in SoccerNet. *Audio* is the Log-Mel spectrograms generated from the audio of the games in SoccerNet.

Task	Model(s)	Data Type		
		ResNet features	Video	Audio
Classification	2D-CNN	✓		
	3D-CNN		✓	
	Audio			✓
	2D-CNN + audio	✓		✓
	3D-CNN + audio		✓	✓
Spotting	CALF	✓		
	Audio			✓
	CALF + audio	✓		✓

5.2. Training and Implementation Details

We implemented the classification models in PyTorch [78], and trained on an Nvidia DGX-2, which consists of 16 Nvidia Tesla V100 graphics processing units (GPUs) and has a total memory capacity of 512 Gigabytes. For the CALF model, we used the implementation from the original paper [4].

Both the 3D-CNN described in Section 3.2 and the 2D-ResNet described in Section 3.4 have similar architectures, with a global average pool (GAP) layer at the end. Due to the GAP layer, varying input dimensions will not have an effect on the number of weights for each model. For both the CALF and 2D-CNN models, described in Sections 3.1 and 3.3, respectively, the models are generated based on input dimensions. This results in a greater number of weights when the input dimensions increase.

5.2.1. Classification Task

For the classification task, we use a minibatch size of 32 for all models, an initial learning rate of 0.001, and momentum of 0.9. We use a scheduler that reduces the learning rate by a multiplicative factor of 0.1 every 10 epochs. During training, we saved the model that had the best accuracy on the validation set and evaluate its performance on the test set.

The 3D-CNN is described in Section 3.2. This approach takes video frames as input. First, due to high memory cost, we use a sample interval of 5, meaning that we reduce each second of video from 25 frames to five frames. Additionally, we downscale the resolution from 224×398 to 112×112. During training, we randomly flip all frames in a sample horizontally, with 0.5 probability. In both training and testing, the samples are normalized. We initialize the model with pre-trained weights, previously trained on the Kinetics-400 dataset [66], and fine-tune the model on the SoccerNet dataset over 12 epochs.

For the audio model, we first generate Log-Mel spectrograms, after which we train on a 2D-ResNet as described in Section 3.4. We use ResNet features to create a 2D-CNN as described in Section 3.3. For both the audio model and for the 2D-CNN on ResNet features, we train for 25 epochs.

5.2.2. Spotting Task

For the spotting task, we use the CALF model and features from the audio model described in Section 3.4 for $W = 8$. We experiment with ResNet features, audio features, and concatenated features that combine the ResNet features and audio features. Our testing is based on the implementation provided by Cioppa et al. [4].

The model takes as input chunks of frames, and the chosen chunk size in the original CALF model covers 120 s of gameplay, with a receptive field of 40. We vary the chunk size and temporal receptive field as they might affect the results for lower tolerances and for when audio features are used. In this paper, we use the models that achieve the best results on the validation, i.e., chunk sizes 120 and 60, with receptive fields of 40, 20, and 5. These three configurations are referred to as "CALF-120-40", "CALF-60-20", and "CALF-60-5".

We use a learning rate of 0.001 for all variations. For our tests with only ResNet features, we train CALF for 300 epochs, validating every 20 epochs. Due to instabilities while training with audio and concatenated features, we train for 30 epochs on audio features and 50 epochs on concatenated features, and validate every 2 epochs.

5.3. Metrics

We use standard metrics like accuracy, precision, recall, and F1-score in our multi-class classification experiments. The task can be interpreted as a one-vs.-all binary classification problem for each class, where a prediction is considered to be a *True Positive (TP)* when the model predicts the correct class, a *False Positive (FP)* when a class is incorrectly predicted, a *True Negative (TN)* when a class is correctly rejected, and a *False Negative (FN)* when a

class is incorrectly rejected. Based on these, accuracy is defined as the number of correct predictions over the total number of predictions:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision, also called Positive Predictive Value (PPV), is the ratio of samples that are correctly identified as positive over all samples that are identified as positive:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall is the ratio of samples that are correctly identified as positive over all positive samples:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Finally, the F1 score is the harmonic mean of precision and recall:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

For spotting, we additionally use average-mAP, which is a metric introduced by Giancola et al. [1], and is expressed as the area under the mAP curve with tolerances ranging from 5 to 60 s. We regard each class separately as a one-vs.-all binary problem and consider a positive prediction as a possible *True Positive (TP)* if it is within a tolerance δ of the ground truth event with a confidence equal to or higher than our threshold. Formally, we use the condition in Equation (5):

$$|gt_{spot} - p_{spot}| < \frac{\delta}{2} \tag{5}$$

where gt_{spot} is a ground truth spot and p_{spot} is a predicted spot in seconds. We take predictions that match the criteria in Equation (5) and create unique pairs of predicted spots and ground truth spots. These are matched in a greedy fashion, where each ground truth spot is matched with the closest prediction. Predicted spots that have no match are considered a *False Positive (FP)*. For a given gt_{spot}, when no predictions are made where this condition holds, we consider it a *False Negative (FN)*. We use the condition in Equation (5) to calculate the average precision (AP) for each class:

$$AP = \sum_{n} (R_n - R_{n-1}) P_n \tag{6}$$

where R_n and P_n are the recall and precision at the n'th threshold, respectively. AP is related to precision-recall and can be calculated as the area under the curve. This is useful as it reduces the PR-curve to a single numerical value. Subsequently, we calculate the mAP:

$$mAP = \frac{\sum_{i=1}^{C} AP_i}{C} \tag{7}$$

where AP_i is AP calculated for the i'th class for C classes and mAP is the mean AP calculated over all classes. This is then calculated for tolerances δ ranging between 5 and 60 s. Finally, we use the mAP scores calculated for different δ to calculate the area under the ROC curve (AUC) and the average-mAP score, which provides some insight into the model's overall performance in the range of 5–60 s.

5.4. Input Window

When training on audio and/or video features, we need to consider what input works best. To that end, we investigate the effect of different temporal windows used for our classification samples, as well as the positioning of our samples relative to the events.

5.4.1. Window Size

The window size determines the temporal size of the input, for example, 1 s of audio features around a given event. Using a large window may include other events or noise, while using a small window might not capture enough relevant information. Another implication of the window size is that it can impact the buffering needed in a real-time scenario, where low latency is required. Regarding the effect the window size has on performance, we observe an improvement for larger temporal inputs across all approaches in Figure 8. Thus, for classification accuracy, a larger window is better. However, as large windows can also have drawbacks, we continue experimenting with different windows sizes in the following experiment.

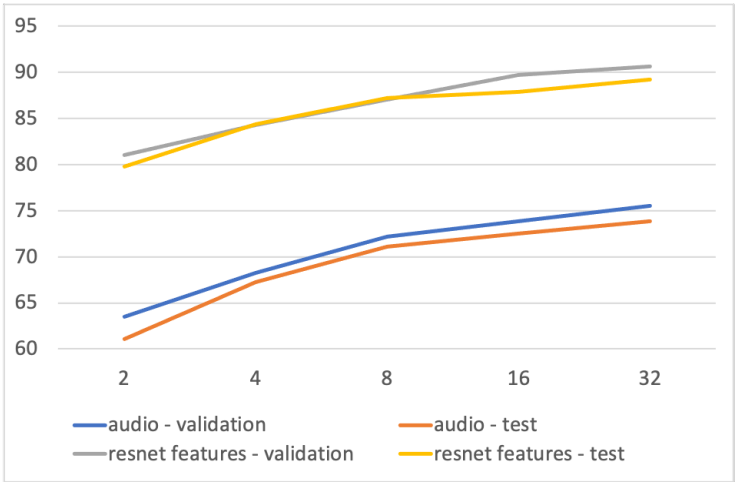


Figure 8. Classification performance with respect to window size (2–32), using the audio model and the ResNet visual features on the validation and test sets. In general, larger windows lead to better results.

5.4.2. Window Position

The window position will impact what information we are using. We illustrate some different positions in Figure 9:

- *Centered window:* A centered window will sample locally around a given event, including both past and future information.
- *Backward (shifted) window:* For backward window, we only use temporal information up to the point of an event. This can be thought of as a slightly different task, where we predict what is about to occur based on information leading up to an event, rather than what has happened.
- *Forward (shifted) window:* Samples from just after the event anchor may contain the most relevant information, such as a soccer ball in a goal, with subsequent celebration without ambiguous prior information.

There are both theoretical and practical consequences associated with the different window positions. In real-time detection, reliance on future information requires buffering before a prediction can be made, thus introducing an undesirable delay.

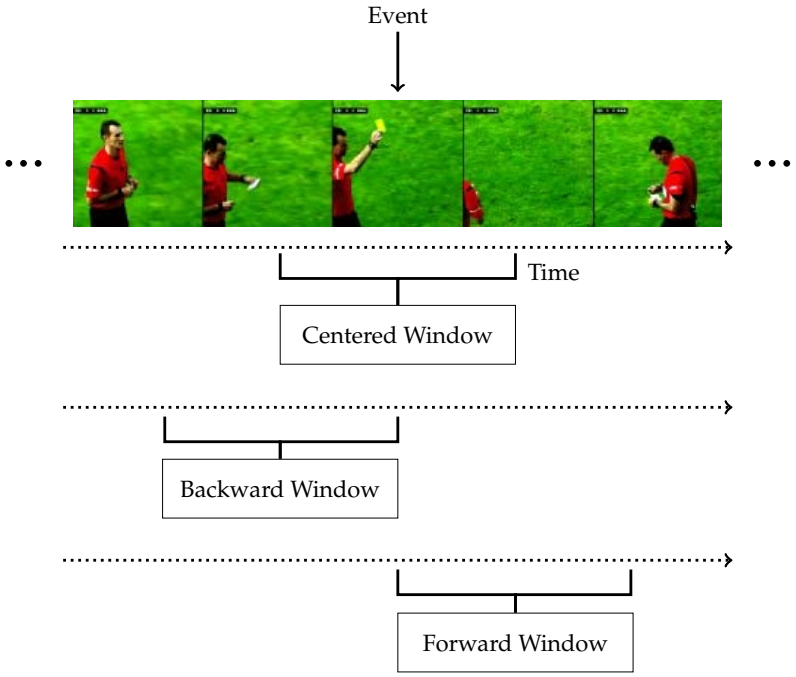


Figure 9. Illustration of window positions relative to the event, for a given sample. A centered window uses both past and future information, while a backward or forward (shifted) window relies only on the past and future, respectively.

From Table 3, we observe that for the audio model, using a centered window performs best at 75% accuracy on both the validation and test set. A forward shift results in a drop in performance, and a backward shift performs the worst. This may be due to the relative change in audio just before and after an event, which is only captured by the center position.

Table 3. Comparison of the accuracy of classification for different models on the validation and test sets, with respect to window position.

		Accuracy (%)		
Model-Dataset	**Window Size**	**Centered Window**	**Backward Window**	**Forward Window**
Audio-validation	16	75.23	66.82	70.62
Audio-test	16	75.09	65.09	69.02
ResNet-validation	16	89.33	76.62	89.08
ResNet-test	16	88.12	75.98	89.01

The results for the visual model show that there is little difference between centered and forward shifted window positions. Backward shifted window performs worse, with close to a 13% absolute drop in accuracy. The reason for this may be that for visual features, the most interesting and discriminating information can be found after the event itself. Thus, in the following experiments, we use a centered window.

5.5. Classification Performance

To evaluate the performance of the classification task, we performed several experiments using the audio 2D-ResNet on Log-Mel spectrograms as described in Section 3.4. This is compared to and combined with the two different visual models based on the 3D-CNN ResNet described in Section 3.2 and the 2D-CNN described in Section 3.3. These models are combined by softmax average and softmax max described in Section 4. Additionally, a separate model trained on concatenated audio features and ResNet features is used.

5.5.1. Overall Performance

We have first assessed the overall results across all events. As observed in Table 4 and Table 5 for the 3D and 2D models, respectively, the audio model generally performs worse than the visual model. Evaluating the results in Table 4 for the 3D visual model, the results indicate that using visual information alone is slightly better. However, using the 2D visual model, the results in Table 5 show improvements in most cases using result concatenation. Thus, it is unclear if the multimodal approach improves performance. When combining the visual and audio models at small window sizes, we observe that the results are worse than the visual model but still competitive. However, the overall best results are present for window sizes W > 16, where the softmax average outperforms the visual-only model. It is therefore interesting to break down the performance into per-event classification to evaluate whether there are differences between event types (which should be expected as, for example, goals often have predictable audio with high sound and fans cheering, whereas events like cards and substitutions have unpredictable audio).

Table 4. Comparison of the accuracy of classification on the validation and test sets, for different models and *late fusion* alternatives. The audio model is described in Section 3.4, and the video-based 3D-CNN model is described in Section 3.2. The fusion of the audio and video features is performed using either softmax average or softmax max, as described in Section 4.

		Accuracy (%)			
Dataset	**Window Size**	**Audio**	**Video**	**Softmax Average**	**Softmax Max**
Validation	8	71.11	88.31	87.71	87.17
	16	72.55	89.90	89.06	88.81
Test	8	72.15	89.69	88.35	87,89
	16	73.90	90.87	89.07	88.92

Table 5. Comparison of the accuracy (%) of classification on the validation and test sets, for different models and fusion alternatives. The audio model is described in Section 3.4, and the video-based 2D-CNN model with pre-extracted ResNet features is described in Section 3.3. The fusion of the audio and video features is performed using *early fusion* (concatenation) or *late fusion* (either softmax average or softmax max), as described in Section 4.

		Accuracy (%)				
Dataset	**Window Size**	**Audio**	**ResNet**	**Concat.**	**Softmax Average**	**Softmax Max**
Validation	2	63.54	81.08	84.46	78.72	78.46
	4	68.26	84.26	87.64	82.31	82.31
	8	72.15	87.08	88.51	86.21	85.79
	16	73.90	89.74	91.23	89.38	89.03
	32	75.49	90.67	92.51	92.05	91.59
Test	2	61.11	79.81	77.27	76.93	76.08
	4	67.28	84.3	81.15	82.40	81.85
	8	71.11	87.22	80.91	85.98	85.83
	16	72.55	87.87	82.15	89.21	88.91
	32	73.89	89.21	87.02	90.85	90.65

5.5.2. Performance per Event Type

For the per-event analysis, we used the same audio and visual models as described above, and softmax average for the fusion of features. Table 6 shows the detailed results using the 3D-CNN model, and Table 7 presents the 2D-CNN model. In Figure 10, we highlighted some of the F1 scores given in the tables. In general, we observe that for goals, combining audio information with visual information almost always improves performance. For other events, it depends a bit on the configuration, but it appears that

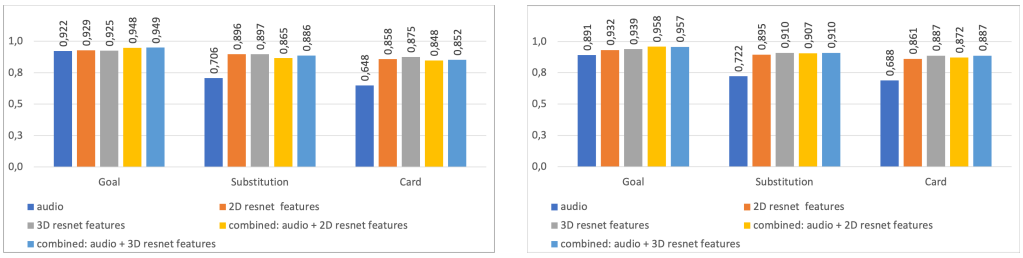
the more audio information is included, i.e., the larger the audio window, the better the results become.

Table 6. Comparison of precision, recall and F1-score per class (event type), for the 3D model (Section 3.2). W is the window size used for the input. The results for the combined input types are obtained using late fusion with softmax average.

		W = 8			W = 16		
Class	**Input Type**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Card	Audio	0.650	0.647	0.648	0.672	0.704	0.688
Card	Video	0.876	0.874	0.875	0.900	0.874	0.887
Card	Video + audio	0.844	0.861	0.852	0.868	0.868	0.868
Substitution	Audio	0.694	0.718	0.706	0.777	0.674	0.722
Substitution	Video	0.925	0.870	0.897	0.949	0.874	0.910
Substitution	Video + audio	0.894	0.877	0.886	0.944	0.877	0.910
Goal	Audio	0.942	0.902	0.922	0.908	0.874	0.891
Goal	Video	0.924	0.926	0.925	0.913	0.966	0.939
Goal	Video + audio	0.954	0.945	0.949	0.954	0.960	0.957
Background	Audio	0.658	0.654	0.656	0.646	0.712	0.677
Background	Video	0.836	0.879	0.857	0.853	0.905	0.878
Background	Video + audio	0.848	0.855	0.851	0.834	0.884	0.858

Table 7. Comparison of precision, recall, and F1-score per class (event type), for the 2D model (Section 3.3). W is the window size used for the input. The input type Combined is the combination of the Resnet and audio models. The results for the Combined input type are obtained using late fusion with softmax average.

		W = 2			W = 4			W = 8			W = 16			W = 32		
Class	***Input type***	***Precision***	***Recall***	***F1***	***Precision***	***Recall***	***F1***	***Precision***	***Recall***	***F1***	***Precision***	***Recall***	***F1***	***Precision***	***Recall***	***F1***
Card	Audio	0.564	0.552	0.558	0.601	0.618	0.609	0.650	0.647	0.648	0.672	0.704	0.688	0.632	0.664	0.648
Card	ResNet	0.811	0.757	0.783	0.836	0.808	0.822	0.869	0.848	0.858	0.873	0.850	0.861	0.873	0.892	0.882
Card	Combined	0.776	0.751	0.763	0.796	0.790	0.793	0.865	0.832	0.848	0.870	0.874	0.872	0.867	0.892	0.879
Substitution	Audio	0.625	0.573	0.598	0.651	0.672	0.661	0.694	0.718	0.706	0.777	0.674	0.722	0.817	0.770	0.793
Substitution	ResNet	0.866	0.794	0.829	0.885	0.874	0.879	0.909	0.883	0.896	0.923	0.869	0.895	0.933	0.870	0.901
Substitution	Combined	0.803	0.741	0.771	0.818	0.841	0.830	0.857	0.872	0.865	0.922	0.893	0.907	0.945	0.914	0.929
Goal	Audio	0.851	0.825	0.838	0.919	0.902	0.910	0.942	0.902	0.922	0.908	0.874	0.891	0.867	0.837	0.852
Goal	ResNet	0.803	0.936	0.864	0.885	0.923	0.904	0.898	0.942	0.919	0.941	0.923	0.932	0.959	0.942	0.950
Goal	Combined	0.878	0.929	0.903	0.937	0.951	0.944	0.940	0.957	0.948	0.969	0.948	0.958	0.960	0.948	0.954
Background	Audio	0.524	0.579	0.550	0.622	0.597	0.609	0.658	0.654	0.656	0.646	0.712	0.677	0.691	0.714	0.702
Background	ResNet	0.734	0.761	0.747	0.793	0.802	0.798	0.830	0.845	0.838	0.820	0.885	0.851	0.842	0.887	0.864
Background	Combined	0.684	0.727	0.705	0.791	0.769	0.780	0.818	0.819	0.819	0.846	0.876	0.861	0.882	0.896	0.889



(**a**) Window size = 8 (**b**) Window size = 16

Figure 10. Classification performance in terms of F1 score, for the 2D (Section 3.3) and the 3D (Section 3.2) models using different input types. We present the results for selected samples from Tables 6 and 7, using window sizes of 8 and 16. The results for the combined input types are obtained using softmax average. We observe that for the *Goal* class, the multimodal approach always performs better.

5.6. Spotting Performance

To test how different combinations of audio and visual features affect spotting performance, i.e., detecting an event in the video stream, once again we experimented with various settings. Here, we used the CALF model described in Section 3.1 with chunk sizes 60 and 120, and receptive fields of 5, 20, and 40. We compared the performance of the models using audio features, visual ResNet features, and combined audio and ResNet features as input. The audio features were extracted from the model described in Section 3.4, and the visual ResNet features were the ResNet features supplied along with SoccerNet [1]. The combination (fusion) of audio and ResNet features was performed using the concatenation process described in Section 4.1.

5.6.1. Overall Performance

From Table 8, we observe that for 2 out of 3 tested configurations ("CALF-60-20" and "CALF-120-40"), using ResNet features alone achieves the highest average-mAP. The only configuration that has an increase in average-mAP for input type of ResNet + audio is the model with the smallest chunk size and receptive field ("CALF-60-5"). This may imply that the benefits of concatenating audio and ResNet features compared to using ResNet features alone are higher for smaller receptive fields.

Table 8. Performance of the CALF [4] model with different configurations, tested on the test set. The mAP values indicate the Average Precision values averaged over all event types. The Combined input type is the integration of the ResNet and the audio models.

		Tolerance = 5			**Tolerance = 20**			**Tolerance = 40**			**Tolerance = 60**			
Model	***Input type***	***Precision***	***Recall***	***mAP***	***Precision***	***Recall***	***mAP***	***Precision***	***Recall***	***mAP***	***Precision***	***Recall***	***mAP***	***Average-mAP***
CALF-60-5	Audio	NaN	0.0735	0.0010	NaN	0.1913	0.0075	NaN	0.3721	0.0187	NaN	0.3501	0.0293	0.0145
CALF-60-5	ResNet	0.1551	0.2947	0.2545	0.4385	0.5507	0.5113	0.5488	0.6110	0.5495	0.5766	0.6147	0.5634	0.5092
CALF-60-5	Combined	0.1425	0.1729	0.2123	0.4330	0.4780	0.5209	0.5453	0.5547	0.5998	0.5889	0.6135	0.6156	0.5408
CALF-60-20	Audio	0.0695	0.0704	0.0771	0.2416	0.1505	0.1940	0.3146	0.2084	0.2321	0.3729	0.2117	0.2475	0.2069
CALF-60-20	ResNet	0.3259	0.3069	0.3655	0.6401	0.5117	0.5493	0.6882	0.5327	0.5871	0.6145	0.5790	0.5971	0.5574
CALF-60-20	Combined	0.2419	0.2303	0.2769	0.5290	0.4659	0.4915	0.5984	0.5016	0.5683	0.6176	0.5398	0.6136	0.5231
CALF-120-40	Audio	NaN	0.0374	0.0007	NaN	0.1519	0.0045	NaN	0.1585	0.0161	NaN	0.2454	0.0264	0.0123
CALF-120-40	ResNet	0.2195	0.2535	0.2869	0.6383	0.5819	0.6067	0.7199	0.6438	0.6425	0.7446	0.6506	0.6530	0.6007
CALF-120-40	Combined	0.1681	0.1855	0.2139	0.5749	0.4674	0.5579	0.6254	0.5836	0.6106	0.6602	0.5931	0.6345	0.5629

The highest average-mAP among all the tested CALF configurations is achieved by the one that uses ResNet features only, with a chunk size of 120 and receptive field 40 ("CALF-120-40"). With respect to average-mAP, we observe that the configuration that performs best with the ResNet + audio input type ("CALF-60-5") does not outperform either of the configurations which perform best with ResNet features alone, overall.

It should also be noted that for some configurations, using audio features alone did not allow one to identify any events correctly. There might be several reasons for this, but still, we see examples where the combination of audio and visual information improves the results. As Table 8 depicts the combined results over all 3 event types (*Card*, *Goal*, and *Substitution* classes), we next look at the spotting performance for each individual class, similar to the classification performance analysis in Section 5.5.

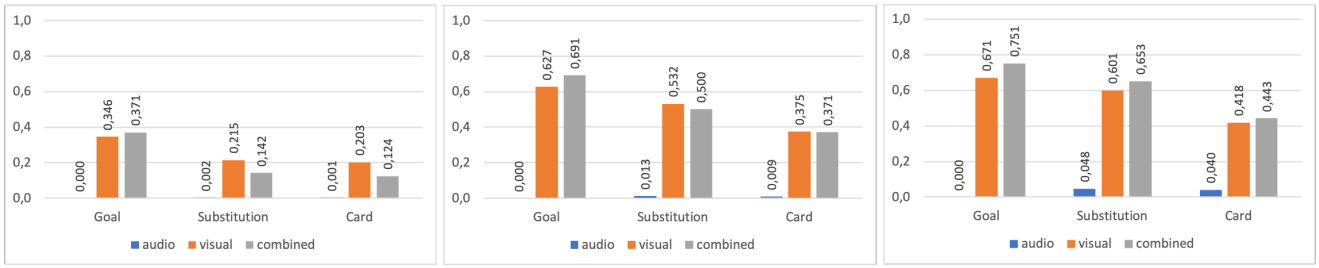
5.6.2. Performance per Event Type

Inspecting the performance indicators in Figure 11 and Table 9, we observe a clear distinction between different classes (event types). While the performance for *Card* and *Substitution* events drops for two of the configurations, there is a significant increase in the Average-AP for all three configurations for *Goal* events. Averaged across all three, the increase is at nearly 7%. This may imply that some events are inherently easier to recognize through audio than others, due to the nature of the event. It is easy to imagine that a goal

might be easier to recognize based on sound than cards and substitutions, due to the loud celebration that often follows a goal.

Table 9. Comparison of precision, recall, and F1-score per class (event type), for the CALF [4] model with different configurations. Highlighted cells indicate the best Average Precision score in each individual experiment. The Combined input type is the integration of the ResNet and the audio models.

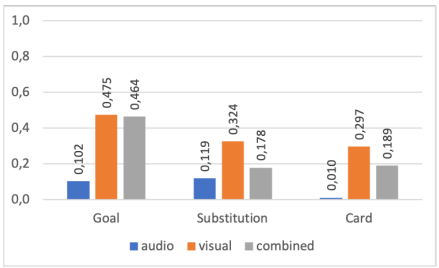
			Tolerance = 5			Tolerance = 20			Tolerance = 40			Tolerance = 60			
Model	*Class*	*Input type*	*Precision*	*Recall*	*AP*	*Precision*	*Recall*	*AP*	*Precision*	*Recall*	*AP*	*Precision*	*Recall*	*AP*	*Average-AP*
CALF-60-5	Card	Audio	0.0030	0.0667	0.0009	0.0096	0.2133	0.0089	0.0197	0.4378	0.0239	0.0254	0.5644	0.0398	0.0183
CALF-60-5	Card	ResNet	0.1271	0.2733	0.2031	0.3489	0.4978	0.3748	0.4348	0.5556	0.4067	0.4486	0.5622	0.4184	0.3701
CALF-60-5	Card	Combined	0.1455	0.0689	0.1242	0.4673	0.2222	0.3716	0.5779	0.2556	0.4218	0.6109	0.3000	0.4427	0.3728
CALF-60-5	Substitution	Audio	0.0022	0.1537	0.0021	0.0102	0.3604	0.0134	0.0192	0.6784	0.0320	0.0275	0.4859	0.0479	0.0254
CALF-60-5	Substitution	ResNet	0.1464	0.2827	0.2148	0.4353	0.5530	0.5323	0.5668	0.6148	0.5868	0.6185	0.6131	0.6007	0.5213
CALF-60-5	Substitution	Combined	0.0941	0.2014	0.1421	0.2912	0.5583	0.5003	0.4183	0.6784	0.6311	0.4732	0.7491	0.6528	0.5299
CALF-60-5	Goal	Audio	NaN	0	0	NaN	0	0	NaN	0	0	NaN	0	0	0
CALF-60-5	Goal	ResNet	0.1918	0.3282	0.3456	0.5312	0.6012	0.6269	0.6448	0.6626	0.6551	0.6626	0.6687	0.6709	0.6111
CALF-60-5	Goal	Combined	0.1879	0.2485	0.3705	0.5406	0.6534	0.6909	0.6398	0.7301	0.7465	0.6825	0.7914	0.7514	0.6879
CALF-60-20	Card	Audio	0.0229	0.0178	0.0101	0.0609	0.0156	0.0580	0.0879	0.0356	0.0761	0.1069	0.0311	0.0885	0.0618
CALF-60-20	Card	ResNet	0.2574	0.2511	0.2973	0.5012	0.4511	0.4144	0.5293	0.4622	0.4495	0.4395	0.5089	0.4670	0.4238
CALF-60-20	Card	Combined	0.2243	0.1889	0.1891	0.4330	0.3444	0.3647	0.5044	0.3800	0.4391	0.5221	0.4200	0.4676	0.3865
CALF-60-20	Substitution	Audio	0.0251	0.1413	0.1193	0.1104	0.3410	0.2367	0.1623	0.4576	0.3111	0.2194	0.4753	0.3439	0.2695
CALF-60-20	Substitution	ResNet	0.2832	0.2862	0.3244	0.6297	0.5318	0.5594	0.7048	0.5654	0.6177	0.6161	0.6237	0.6231	0.5640
CALF-60-20	Substitution	Combined	0.1063	0.2014	0.1779	0.3040	0.5318	0.3896	0.3974	0.5848	0.4708	0.4320	0.6290	0.5550	0.4250
CALF-60-20	Goal	Audio	0.1604	0.0521	0.1019	0.5536	0.0951	0.2872	0.6935	0.1319	0.3090	0.7925	0.1288	0.3101	0.2782
CALF-60-20	Goal	ResNet	0.4371	0.3834	0.4746	0.7895	0.5521	0.6741	0.8304	0.5706	0.6941	0.7880	0.6043	0.7012	0.6653
CALF-60-20	Goal	Combined	0.3952	0.3006	0.4637	0.8500	0.5215	0.7201	0.8934	0.5399	0.7951	0.8986	0.5706	0.8182	0.7383
CALF-120-40	Card	Audio	0.0015	0.0222	0.0004	0.0068	0.1022	0.0042	0.0155	0.2333	0.0184	0.0261	0.3933	0.0343	0.0140
CALF-120-40	Card	ResNet	0.2103	0.2444	0.2394	0.5624	0.5311	0.4802	0.6268	0.5822	0.5239	0.6522	0.6000	0.5399	0.4775
CALF-120-40	Card	Combined	0.1505	0.1378	0.1587	0.5236	0.2956	0.4043	0.5230	0.4044	0.4453	0.5503	0.4133	0.4607	0.4012
CALF-120-40	Substitution	Audio	0.0038	0.0901	0.0017	0.0099	0.3534	0.0091	0.0203	0.2420	0.0299	0.0287	0.3428	0.0449	0.0231
CALF-120-40	Substitution	ResNet	0.1745	0.2155	0.2341	0.5685	0.5795	0.6269	0.6933	0.6590	0.6660	0.7218	0.6555	0.6735	0.6006
CALF-120-40	Substitution	Combined	0.1047	0.1979	0.1562	0.3844	0.5053	0.5102	0.4682	0.6378	0.5629	0.5384	0.6572	0.5988	0.5021
CALF-120-40	Goal	Audio	NaN	0	0	NaN	0	0	NaN	0	0	NaN	0	0	0
CALF-120-40	Goal	ResNet	0.2737	0.3006	0.3872	0.7841	0.6350	0.7131	0.8396	0.6902	0.7375	0.8598	0.6963	0.7455	0.6912
CALF-120-40	Goal	Combined	0.2491	0.2209	0.3269	0.8167	0.6012	0.7592	0.8851	0.7086	0.8237	0.8919	0.7086	0.8441	0.7507



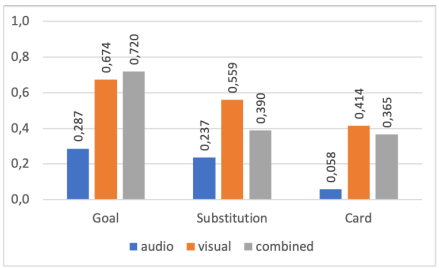
(**a**) CALF-60-5, tolerance = 5

(**b**) CALF-60-5, tolerance = 20

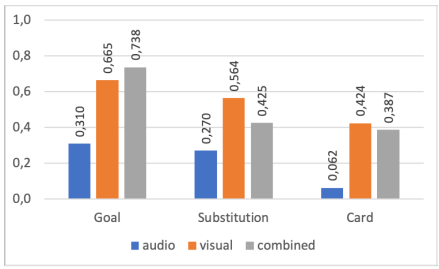
(**c**) CALF-60-5, tolerance = 60



(**d**) CALF-60-40, tolerance = 5



(**e**) CALF-60-40, tolerance = 20



(**f**) CALF-60-40, tolerance = 60

Figure 11. Spotting performance in terms of Average Precision per event type, for the CALF-60-5 and CALF-60-20 models over the tolerances 5, 20, and 60. In general, we can observe that for goals, adding audio information almost always improves performance. For other events, it depends on the configuration, but it seems like the more audio information is included, the better the results get.

Since we observed that the performance has declined for card and substitution events, where it has increased for goal events, it could be interesting to train a model using different

modalities for different events. If one could identify events that benefit from concatenated features and only use these features for the given events, the overall performance of the ensemble model could be improved. As an example, if we combine the results from the concatenated features for goals, and the ResNet features for cards and substitutions, we increase the average-mAP by over 2%.

6. Discussion

In this work, our main aim is not to develop a completely novel model that yields a performance surpassing the state-of-the-art but rather to assess the potential of using multiple modalities, for event detection in soccer videos. We evaluate different model and modality combinations from the state-of-the-art using different configurations, in order to derive insights. In this respect, as also described in Section 2.3, the use of multiple modalities is not a new idea, but the amount of work on deep neural networks is limited. In this paper, we used two existing state-of-the-art visual models based on CNN and assessed the effect of adding audio information on the detection accuracy, for different types of events.

In general, our experimental results show that the benefit of analyzing audio information alone, or in addition to the visual information, is dependent on the context or the type of event. As shown in Tables 7 and 9, there are many events where the combined modalities have a positive effect on detection and classification performance. Still, the improvement is much more profound for events such as goals, which tend to be followed by a more predictable type of sound (e.g., high volume audio with excited commentator or a cheering crowd). This finding is in line with existing works focusing solely on audio information, which typically focus on events where commentator and audience responses to the action are distinguishable. For instance, Xu et al. [55] create seven audio keywords for detecting potential events: long-whistling, double-whistling, multiwhistling, exciting commentator speech, plain commentator speech, exciting audience sound, and plain audience sound, with the justification that these have strong ties to certain types of events. Islam et al. [79] propose an approach to extract audio features using Empirical Mode Decomposition (EMD) that can filter out noises. They conduct a set of experiments on four soccer videos and show that the proposed method can detect goal events with 96% accuracy. Albeit successful, this approach might not be generalizable to different types of events as a standalone solution and is tested on a relatively small scale. However, this is a very interesting area for research. Our results show that, overall, multimodality has a more inconclusive effect on events without any particular audio pattern. Thus, the choice of modalities should be based on the target event. However, this does not discourage the use of finer-tuned and event-specific audio models, which can be used together with visual models conditionally (depending on context).

The visual CNN models we have experimented with are meant as examples of state-of-the-art models. New models are currently being developed and tested. For example, in the SoccerNet-v2 challenge [80], the best dataset benchmark models, CALF [4], AudioVid [73] and NetVLAD [1], have achieved average-mAP values of 72.2%, 69.7%, and 54.9% for goal events, respectively. Moreover, Zhou et al. [69] also achieved good results in the SoccerNet-v2 spotting challenge as the winning team, with an overall average-mAP of about 75% (the authors do not provide event-specific numbers). However, these numbers are lower than our AP of 84% for goal events, achieved by the combined model as shown in Table 9.

Another issue observed both in our results and in various SoccerNet papers (e.g., [1,4,73,80]) is that the input window greatly affects performance. A larger window yields more data to analyze and delays inference prediction in any live event scenario as one must wait for the window to be available and processed. For example, our approach in [2] relies on a small temporal window of 8 s after post-processing, and the values calculated for tolerances $\delta > 8$ can therefore be misleading. The average-mAP metric uses a range of tolerances from 5 to 60 s, and since our model relies on local information rather than

long-range contextual information regarding events, it is expected that higher tolerances will only result in finding spots that are false positives. However, in a real-time setting, it may be important to have as little delay in predictions as possible. For our approach, a live prediction will have a delay of about 4 s (for tolerances $\delta > 8$). This includes buffering future frames and computation. The baseline model would have about 10 s of delay, while the current state-of-the-art [4] would have about 100 s. It seems that long-range contextual features can boost performance, and there is a trade-off between delay and the practical ability to buffer future video frames.

7. Summary and Conclusions

In this paper, we present our research on detecting events in soccer videos, where we evaluate the use of visual, audio, and the combination of visual and audio features. In particular, we focus on three types of events (*cards*, *substitutions*, and *goals*), implement and experiment with existing state-of-the-art visual models based on the ResNet model, using the SoccerNet dataset containing 500 games and a total of 6637 annotated events belonging to the above classes. Additionally, we develop an audio-based model using Log-Mel spectrograms and similarly experiment with this model under different configurations. Finally, we use a multimodal approach where we combine the video and audio models, using both early fusion (feature concatenation) and late fusion (averaging/maxing the softmax prediction values from independent models).

Our experimental results demonstrate the potential of using multiple modalities as the performance of detecting events increases in many of the selected configurations when features are combined. However, we have also seen that there is a difference in the benefits gained from the multimodal approach with respect to different event types. More specifically, the combination of audio and visual features proved more beneficial for the *Goal* events than for *Card* and *Substitution* events. This yields insights into a potential new research direction, where we could decide on the use of multiple modalities based on targeted event categories since different events have different "associated" audio (e.g., for goals, it is usually some predictable high volume cheering, whereas for substitutions and cards, we do not have the same predictable audio). In summary, an ML-based event detection component utilizing several available data modalities can be an important component of future intelligent video processing and analysis systems.

Our ongoing and future work include the following. Firstly, we would like to pursue the path of deciding on the deployment of multimodality based on target event type. Secondly, we would like to investigate more events that are of interest for soccer games, for instance, by using the newer SoccerNet-v2 dataset (https://soccer-net.org, accessed on 3 December 2021), which includes new categories. Thirdly, we would like to focus on the generalizability of our algorithms and findings via cross-dataset analysis, using results from multiple soccer, and, later on, other sports datasets. For instance, in addition to exploring the SoccerNet-v2 dataset, we have also been experimenting with in-house datasets we have been collecting from the Norwegian and Swedish elite soccer leagues, currently containing about 1000 annotated events. Last but not least, we would like to work on optimizing the individual models we have evaluated in this study, such as finding the right trade-offs between detection latency (as determined by input window size and position), processing overhead, and detection accuracy.

Author Contributions: Conceptualization, O.A.N.R., M.S., S.A.H., M.A.R., and P.H.; methodology, O.A.N.R., M.S., S.A.H., M.A.R., and P.H.; software, O.A.N.R. and M.S.; validation, O.A.N.R., M.S., S.A.H., M.A.R., and P.H.; investigation, all authors; resources, O.A.N.R., M.S., S.A.H., D.J., M.A.R., and P.H.; data curation, O.A.N.R., M.S., and P.H.; writing—original draft preparation, all authors; writing—review and editing, all authors; visualization, all authors; supervision, S.A.H., V.L.T., E.Z., D.J., M.A.R., and P.H.; funding acquisition, M.A.R. and P.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Norwegian Research Council, project number 327717 (AI-producer).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The SoccerNet dataset is made available by Giancola et al. [1].

Acknowledgments: The research has benefited from the Experimental Infrastructure for Exploration of Exascale Computing (eX3), which is financially supported by the Research Council of Norway under contract 270053. We also acknowledge the use of video data from Norsk Toppfotball (NTF).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Giancola, S.; Amine, M.; Dghaily, T.; Ghanem, B. SoccerNet: A Scalable Dataset for Action Spotting in Soccer Videos. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1711–1721. [CrossRef]
2. Rongved, O.A.N.; Hicks, S.A.; Thambawita, V.; Stensland, H.K.; Zouganeli, E.; Johansen, D.; Riegler, M.A.; Halvorsen, P. Real-Time Detection of Events in Soccer Videos using 3D Convolutional Neural Networks. In Proceedings of the SMEEE International Symposium on Multimedia (ISM), Naples, Italy, 2–4 December 2020; pp. 135–144. [CrossRef]
3. Rongved, O.A.N.; Hicks, S.A.; Thambawita, V.; Stensland, H.K.; Zouganeli, E.; Johansen, D.; Midoglu, C.; Riegler, M.A.; Halvorsen, P. Using 3D Convolutional Neural Networks for Real-time Detection of Soccer Events. *IEEE J. Sel. Top. Signal Process.* **2021**, *15*, 161–187. [CrossRef]
4. Cioppa, A.; Deliege, A.; Giancola, S.; Ghanem, B.; Droogenbroeck, M.; Gade, R.; Moeslund, T. A Context-Aware Loss Function for Action Spotting in Soccer Videos. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
5. Purwins, H.; Li, B.; Virtanen, T.; Schlüter, J.; Chang, S.; Sainath, T. Deep Learning for Audio Signal Processing. *IEEE J. Sel. Top. Signal Process.* **2019**, *13*, 206–219. [CrossRef]
6. Dalal, N.; Triggs, B.; Schmid, C. Human Detection Using Oriented Histograms of Flow and Appearance. In Proceedings of the ECCV, Graz, Austria, 7–13 May 2006; pp. 428–441. [CrossRef]
7. Wang, H.; Kläser, A.; Schmid, C.; Liu, C.L. Dense Trajectories and Motion Boundary Descriptors for Action Recognition. *Int. J. Comput. Vis.* **2013**, *103*, 60–79. [CrossRef]
8. Wang, H.; Schmid, C. Action Recognition with Improved Trajectories. In Proceedings of the International Conference on Computer Vision (ICCV), Sydney, Australia, 1–8 December 2013; pp. 3551–3558.
9. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale Video Classification with Convolutional Neural Networks. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
10. Simonyan, K.; Zisserman, A. Two-Stream Convolutional Networks for Action Recognition in Videos. *arXiv* **2014**, arXiv:1406.2199.
11. Goodale, M.A.; Milner, A.D. Separate visual pathways for perception and action. *Trends Neurosci.* **1992**, *15*, 20–25. [CrossRef]
12. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
13. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.
14. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning Spatiotemporal Features with 3D Convolutional Networks. In Proceedings of the ICCV, Santiago, Chile, 7–13 December 2015; pp. 4489–4497. [CrossRef]
15. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; Paluri, M. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6450–6459. [CrossRef]
16. Feichtenhofer, C.; Fan, H.; Malik, J.; He, K. SlowFast Networks for Video Recognition. In Proceedings of the International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 6202–6211.
17. Carreira, J.; Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. *arXiv* **2018**, arXiv:1705.07750.
18. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
19. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional Two-Stream Network Fusion for Video Action Recognition. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1933–1941. [CrossRef]
20. Feichtenhofer, C.; Pinz, A.; Wildes, R.P. Spatiotemporal Residual Networks for Video Action Recognition. In Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; pp. 3476–3484.

21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
22. Wang, L.; Qiao, Y.; Tang, X. Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 4305–4314. [CrossRef]
23. Shou, Z.; Wang, D.; Chang, S.F. Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1049–1058. [CrossRef]
24. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In Proceedings of the ECCV, Amsterdam, The Netherlands, 11–14 October 2016; pp. 20–36.
25. Donahue, J.; Hendricks, L.A.; Rohrbach, M.; Venugopalan, S.; Guadarrama, S.; Saenko, K.; Darrell, T. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
26. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv* **2012**, arXiv:1212.0402.
27. Qiu, Z.; Yao, T.; Ngo, C.W.; Tian, X.; Mei, T. Learning Spatio-Temporal Representation with Local and Global Diffusion. *arXiv* **2019**, arXiv:1906.05571.
28. Kalfaoglu, M.E.; Kalkan, S.; Alatan, A.A. Late Temporal Modeling in 3D CNN Architectures with BERT for Action Recognition. *arXiv* **2020**, arXiv:2008.01232.
29. Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; Serre, T. HMDB51: A Large Video Database for Human Motion Recognition. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 13–16 November 2011; pp. 2556–2563. [CrossRef]
30. Singh, G.; Cuzzolin, F. Untrimmed Video Classification for Activity Detection: Submission to ActivityNet Challenge. *arXiv* **2016**, arXiv:1607.01979.
31. Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; Lin, D. Temporal Action Detection with Structured Segment Networks. *arXiv* **2017**, arXiv:1704.06228.
32. Chao, Y.W.; Vijayanarasimhan, S.; Seybold, B.; Ross, D.A.; Deng, J.; Sukthankar, R. Rethinking the Faster R-CNN Architecture for Temporal Action Localization. *arXiv* **2018**, arXiv:1804.07667.
33. Lin, T.; Zhao, X.; Shou, Z. Single Shot Temporal Action Detection. In Proceedings of the ACM MM, Mountain View, CA, USA, 23–27 October 2017. [CrossRef]
34. Buch, S.; Escorcia, V.; Ghanem, B.; Fei-Fei, L.; Niebles, J.C. End-to-End, Single-Stream Temporal Action Detection in Untrimmed Videos. In Proceedings of the BMVC, London, UK, 4–7 September 2017.
35. Idrees, H.; Zamir, A.R.; Jiang, Y.; Gorban, A.; Laptev, I.; Sukthankar, R.; Shah, M. The THUMOS challenge on action recognition for videos "in the wild". *Comput. Vis. Image Underst.* **2017**, *155*, 1–23. [CrossRef]
36. Lin, T.; Liu, X.; Li, X.; Ding, E.; Wen, S. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In Proceedings of the International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
37. Lin, T.; Zhao, X.; Su, H.; Wang, C.; Yang, M. BSN: Boundary Sensitive Network for Temporal Action Proposal Generation. In Proceedings of the ECCV, Munich, Germany, 8–14 September 2018.
38. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
39. Xu, H.; Das, A.; Saenko, K. R-C3D: Region Convolutional 3D Network for Temporal Activity Detection. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
40. Buch, S.; Escorcia, V.; Shen, C.; Ghanem, B.; Niebles, J.C. SST: Single-Stream Temporal Action Proposals. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6373–6382.
41. Heilbron, F.; Niebles, J.C.; Ghanem, B. Fast Temporal Activity Proposals for Efficient Detection of Human Actions in Untrimmed Videos. In Proceedings of the CVPR, Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
42. Spagnolo, P.; Leo, M.; Mazzeo, P.L.; Nitti, M.; Stella, E.; Distante, A. Non-invasive Soccer Goal Line Technology: A Real Case Study. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Portland, OR, USA, 23–28 June 2013; pp. 1011–1018. [CrossRef]
43. Mazzeo, P.L.; Spagnolo, P.; Leo, M.; D'Orazio, T. Visual Players Detection and Tracking in Soccer Matches. In Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Santa Fe, NM, USA, 1–3 September 2008; pp. 326–333.10.1109/AVSS.2008.33. [CrossRef]
44. Stensland, H.K.; Gaddam, V.R.; Tennøe, M.; Helgedagsrud, E.; Næss, M.; Alstad, H.K.; Mortensen, A.; Langseth, R.; Ljødal, S.; Landsverk, O.; et al. Bagadus: An Integrated Real-Time System for Soccer Analytics. *ACM Trans. Multimed. Comput. Commun. Appl.* **2014**, *10*, 1–21. [CrossRef]
45. Thamaraimanalan, T.; Naveena, D.; Ramya, M.; Madhubala, M. Prediction and Classification of Fouls in Soccer Game using Deep Learning. *Ir. Interdiscip. J. Sci. Res.* **2020**, *4*, 66–78.

46. Gaddam, V.R.; Eg, R.; Langseth, R.; Griwodz, C.; Halvorsen, P. The Cameraman Operating My Virtual Camera is Artificial: Can the Machine Be as Good as a Human? *ACM Trans. Multimed. Comput. Commun. Appl.* **2015**, *11*, 1–20. [CrossRef]
47. Johansen, D.; Johansen, H.; Aarflot, T.; Hurley, J.; Kvalnes, R.; Gurrin, C.; Zav, S.; Olstad, B.; Aaberg, E.; Endestad, T.; et al. DAVVI: A Prototype for the next Generation Multimedia Entertainment Platform. In Proceedings of the International Conference on Multimedia (ACM MM), Vancouver, BC, Canada, 19–24 October 2009; pp. 989–990. [CrossRef]
48. Wang, J.; Xu, C.; Chng, E.; Tian, Q. Sports highlight detection from keyword sequences using HMM. In Proceedings of the IEEE International Conference on Multimedia Expo (ICME), Taipei, Taiwan, 27–30 June 2004, Volume 1; pp. 599–602. [CrossRef]
49. Dhanuja, S.P.; Waykar, S.B. A Survey on Event Recognition and Summarization in Football Videos. *Int. J. Sci. Res.* **2014**, *3*, 2365–2367.
50. Xiong, Z.; Radhakrishnan, R.; Divakaran, A.; Huang, T. Audio events detection based highlights extraction from baseball, golf and soccer games in a unified framework. In Proceedings of the International Conference on Multimedia and Expo (ICME), Baltimore, MD, USA, 6–9 July 2003; Volume 3, p. III-401. [CrossRef]
51. Pixi, Z.; Hongyan, L.; Wei, W. Research on Event Detection of Soccer Video Based on Hidden Markov Model. In Proceedings of the 2010 International Conference on Computational and Information Sciences, Chengdu, China, 17–19 December 2010; pp. 865–868. [CrossRef]
52. Qian, X.; Liu, G.; Wang, H.; Li, Z.; Wang, Z. Soccer Video Event Detection by Fusing Middle Level Visual Semantics of an Event Clip. In Proceedings of the Advances in Multimedia Information Processing (PCM), Shanghai, China, September 2010; Springer Berlin Heidelberg: Berlin/Heidelberg, Germany 2010; pp. 439–451.
53. Qian, X.; Wang, H.; Liu, G. HMM based soccer video event detection using enhanced mid-level semantic. *Multimed. Tools Appl.* **2012**, *60*, 233–255. [CrossRef]
54. Itoh, H.; Takiguchi, T.; Ariki, Y. Event Detection and Recognition Using HMM with Whistle Sounds. In Proceedings of the 2013 International Conference on Signal-Image Technology Internet-Based Systems, Kyoto, Japan, 2–5 December 2013; pp. 14–21. [CrossRef]
55. Xu, M.; Maddage, N.; Xu, C.; Kankanhalli, M.; Tian, Q. Creating audio keywords for event detection in soccer video. In Proceedings of the International Conference on Multimedia and Expo (ICME), Baltimore, MD, USA, 6–9 July 2003; Volume 2, p. II-281. [CrossRef]
56. Ye, Q.; Huang, Q.; Gao, W.; Jiang, S. Exciting Event Detection in Broadcast Soccer Video with Mid-Level Description and Incremental Learning. In Proceedings of the ACM International Conference on Multimedia (MM), Singapore, 6–11 November 2005; pp. 455–458. [CrossRef]
57. Sadlier, D.; O'Connor, N. Event detection in field sports video using audio-visual features and a support vector machine. *IEEE Trans. Circuits Syst. Video Technol.* **2005**, *15*, 1225–1233. [CrossRef]
58. Jain, N.; Chaudhury, S.; Roy, S.D.; Mukherjee, P.; Seal, K.; Talluri, K. A Novel Learning-Based Framework for Detecting Interesting Events in Soccer Videos. In Proceedings of the Indian Conference on Computer Vision, Graphics Image Processing, Bhubaneswar, India, 16–19 December 2008; pp. 119–125. [CrossRef]
59. Zawbaa, H.M.; El-Bendary, N.; Hassanien, A.E.; Abraham, A. SVM-based soccer video summarization system. In Proceedings of the the World Congress on Nature and Biologically Inspired Computing, Salamanca, Spain, 19–21 October 2011; pp. 7–11. [CrossRef]
60. Fakhar, B.; Kanan, H.; Behrad, A. Event detection in soccer videos using unsupervised learning of Spatio-temporal features based on pooled spatial pyramid model. *Multimed. Tools Appl.* **2019**, *78*, 16995–17025. [CrossRef]
61. Jiang, H.; Lu, Y.; Xue, J. Automatic Soccer Video Event Detection Based on a Deep Neural Network Combined CNN and RNN. In Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI), San Jose, CA, USA, 6–8 November 2016; pp. 490–494. [CrossRef]
62. Tang, K.; Bao, Y.; Zhao, Z.; Zhu, L.; Lin, Y.; Peng, Y. AutoHighlight: Automatic Highlights Detection and Segmentation in Soccer Matches. In Proceedings of the IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 4619–4624. [CrossRef]
63. Khan, A.; Lazzerini, B.; Calabrese, G.; Serafini, L. Soccer Event Detecion. In Proceedings of the the International Conference on Image Processing and Pattern Recognition (IPPR), Copenhagen, Denmark, 28–29 April 2018. [CrossRef]
64. Hong, Y.; Ling, C.; Ye, Z. End-to-end soccer video scene and event classification with deep transfer learning. In Proceedings of the International Conference on Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 2–4 April 2018; pp. 1–4. [CrossRef]
65. Yu, J.; Lei, A.; Hu, Y. Soccer Video Event Detection Based on Deep Learning. In Proceedings of the MultiMedia Modeling (MMM), Thessaloniki, Greece, 8–11 January 2019, pp. 377–389.
66. Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. The Kinetics Human Action Video Dataset. *arXiv* **2017**, arXiv:1705.06950.
67. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2016**, arXiv:1506.02640.
68. Vats, K.; Fani, M.; Walters, P.; Clausi, D.A.; Zelek, J. Event detection in coarsely annotated sports videos via parallel multi receptive field 1D convolutions. *arXiv* **2020**, arXiv:2004.06172.
69. Zhou, X.; Kang, L.; Cheng, Z.; He, B.; Xin, J. Feature Combination Meets Attention: Baidu Soccer Embeddings and Transformer based Temporal Detection. *arXiv* **2021**, arXiv:2106.14447.

70. Sadlier, D.A.; O'Connor, N.; Marlow, S.; Murphy, N. A combined audio-visual contribution to event detection in field sports broadcast video. Case study: Gaelic football. In Proceedings of the IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Darmstadt, Germany, 17 December 2003; pp. 552–555. [CrossRef]
71. Ortega, J.; Senoussaoui, M.; Granger, E.; Pedersoli, M.; Cardinal, P.; Koerich, A. Multimodal Fusion with Deep Neural Networks for Audio-Video Emotion Recognition. *arXiv* **2019**, arXiv:1907.03196.
72. Xiao, F.; Lee, Y.J.; Grauman, K.; Malik, J.; Feichtenhofer, C. Audiovisual SlowFast Networks for Video Recognition. *arXiv* **2020**, arXiv:2001.08740.
73. Vanderplaetse, B.; Dupont, S. Improved Soccer Action Spotting Using Both Audio and Video Streams. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Seattle, WA, USA, 14–19 June 2020.
74. Gao, X.; Liu, X.; Yang, T.; Deng, G.; Peng, H.; Zhang, Q.; Li, H.; Liu, J. Automatic Key Moment Extraction and Highlights Generation Based on Comprehensive Soccer Video Understanding. In Proceedings of the IEEE International Conference on Multimedia Expo Workshops (ICMEW), London, UK, 6–10 July 2020; pp. 1–6. [CrossRef]
75. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-Local Neural Networks. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 13–18 June 2018.
76. Zolfaghari, M.; Singh, K.; Brox, T. ECO: Efficient Convolutional Network for Online Video Understanding. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
77. Khaleghi, B.; Khamis, A.; Karray, F.O.; Razavi, S.N. Multisensor Data Fusion: A Review of the State-of-the-Art. *Inf. Fusion* **2013**, *14*, 28–44. [CrossRef]
78. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS), Vancouver, QC, Canada, 8–14 December 2019; pp. 8024–8035.
79. Islam, M.R.; Paul, M.; Antolovich, M.; Kabir, A. Sports Highlights Generation using Decomposed Audio Information. In Proceedings of the IEEE International Conference on Multimedia Expo Workshops (ICMEW), Shanghai, China, 8–12 July 2019; pp. 579–584. [CrossRef]
80. Deliège, A.; Cioppa, A.; Giancola, S.; Seikavandi, M.J.; Dueholm, J.V.; Nasrollahi, K.; Ghanem, B.; Moeslund, T.B.; Droogenbroeck, M.V. SoccerNet-v2: A Dataset and Benchmarks for Holistic Understanding of Broadcast Soccer Videos. *arXiv* **2021**, arXiv:2011.13367.