Temporally Accurate Events Detection Through Ball Possessor Recognition in Soccer

Marc Peral[1][a], Guillem Capellera[2][b], Antonio Rubio[2][c], Luis Ferraz[2][d],
Francesc Moreno-Noguer[1][e] and Antonio Agudo[1][f]

[1]*Institut de Robòtica i Informàtica Industrial CSIC-UPC, Barcelona, Spain*

[2]*Kognia Sports Intelligence, Barcelona, Spain*

{mperal, fmoreno, aagudo}@iri.upc.edu, {guillem.capellera, antonio.rubio, luis.ferraz}@kogniasports.com

Abstract: Recognizing specific actions in soccer games has become an increasingly important research topic. One key area focuses on accurately identifying when passes and receptions occur, as these are frequent actions in games and critical for analysts reviewing match strategies. However, most current methods do not pinpoint when these actions happen precisely enough and often fail to show which player is making the move. Our new method uses video footage to detect passes and receptions and identifies which player is involved in each action by following possession of the ball at each moment. We create video clips, or tubes, for every player on the field, determine who has the ball, and use this information to recognize when these key actions take place. Our results show that our system is better than the latest models in spotting passes and can identify most events with an accuracy down to 0.6 seconds.

1 INTRODUCTION

Soccer has grown in popularity in recent years, as the increase in the revenue of top clubs reflects (Deloitte, 2023). This growth comes hand-in-hand with the multiplication of data acquisition in terms of players and ball positional information (Capellera et al., 2024a; Capellera et al., 2024b), video footage of games and gathering of events statistics. The enormous amount of collected data calls for ways to exploit its potential (Goes et al., 2021). Soccer clubs have a team of analysts that study the behavior of their team and the habits of their next opponent to design a strategy for the upcoming games. To do so, they devote a vast number of hours rewatching recorded games. Understanding soccer semantics lets them plan future tactics, but they have to stare at game footage and there is not much time left for pondering as schedules are tight. The industry has identified this pain point in the sector and some companies emerged

[a] https://orcid.org/0000-0001-6521-6476

[b] https://orcid.org/0009-0006-7266-078X

[c] https://orcid.org/0000-0002-6771-8645

[d] https://orcid.org/0000-0001-7851-9193

[e] https://orcid.org/0000-0002-8640-684X

[f] https://orcid.org/0000-0001-6845-4998

to automate the process of spotting actions.

The latest technological improvements are achieving a pruning of the limitations of soccer analysis in terms of time and subjectivity (Cossich et al., 2023). A strong area of this automation is the detection of events. Recent research (Giancola and Ghanem, 2021; Zhou et al., 2021; Denize et al., 2024) tries to spot high-level actions providing a big picture of the match to follow the flow of the game, although that is not enough for analysts who need to conduct a deep dive examination. To perform a proper match review, they focus their attention on the key events that give more details about team tactics, such as passes and receptions. These are undoubtedly the most common events in a soccer game and we call them touch events because they happen every time a player touches the ball. That is why this work focuses on finding which is the player in the field that has the ball at every frame, if any, to afterwards retrieve who is executing those touch events and when those occur. Taking into account these touch events, we find approaches that use information from previous events to define the next up (Yeung et al., 2023; Simpson et al., 2022), methods employing trajectory data (Vidal-Codina et al., 2022; Kim et al., 2023; Sanford et al., 2020), and others that work with video recordings of matches (Sorano et al., 2021;

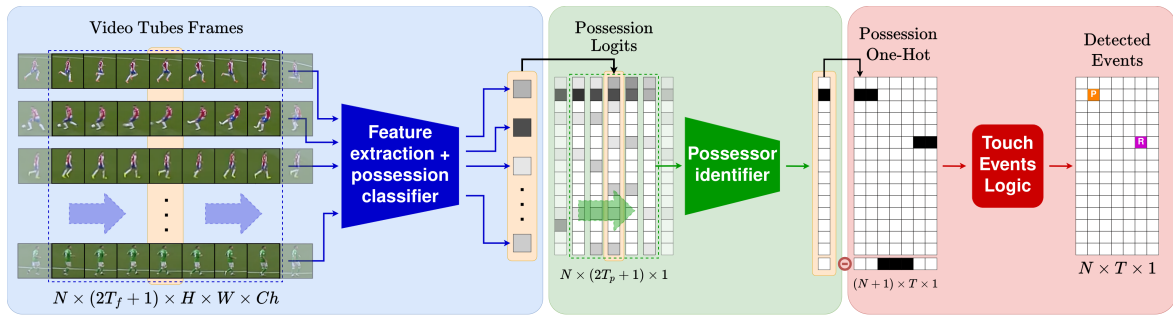Figure 1: **Events detection pipeline.** Our approach is composed of three blocks: 1) video tube computation where feature extraction is also performed to get logits per player and per frame. 2) Possession is chosen between the N players, if any, resulting into a one-hot possession vector per frame. 3) The detection of events shows how a pass P and a reception R are detected in this chunk of T frames.

Sanford et al., 2020; Baikulov, 2023; Philipp Singer, 2022). Some models correct manually annotated data and achieve better synchronization (Biermann et al., 2023). As a consequence, we observe that most of the proposed approaches do not fulfill the temporal accuracy requirements, and the ones that get closer to it are missing some crucial details like identifying which player from the ones in the pitch is the passer or the receiver.

The approach we propose is the first to achieve the detection of touch events and the identification of which of the players in the field is conducting those in a reasonable temporal precision for soccer analysts. As illustrated in Figure 1, our method crops a video tube (Yu and Yuan, 2015) for each player in the frame and finds which one has ball possession. A tube corresponds to the visual information embedded inside consecutive bounding boxes for the same player in the video space. From per player possession information, we spot when those key events take place, additionally yielding which player in the field performed them. We test our model in a large dataset of matches from top European leagues. Despite no other methods providing all the indispensable information, we still compare to state-of-the-art methods that partially meet our requirements and prove that ours outperforms the latest. We bring a valuable contribution to the state of the art as we enable the detection of passes and receptions in a way that is useful for soccer analysts.

2 RELATED WORK

Previous work approaches events detection trying to assign a start and end time, but the state of the art evolved to consider events occurring in a specific frame avoiding ambiguity and subjectivity. As stated in (Giancola et al., 2018), events are defined as in-

stantaneous in the soccer rulesbook. For example, a goal happens at the moment the ball crosses the goal line between the goalposts and the crossbar. For this reason, we consider events as occurrences in a certain frame.

Table 1 depicts how recent methods range in a wide variety of inputs utilized to detect various types of event. In this section we go over those approaches elucidating their strengths and weaknesses. Despite all this, one can notice that only our model is capable of spotting events precisely in time, even when missing information, as well as identifying the player performing those.

Data Observability: A common obstacle in current soccer applications is the lack of data. Algorithms that use previous events (Yeung et al., 2023; Simpson et al., 2022) need every previous action that happened and their position in the field to work. But the frequent missing data problem appears when the information of a player is absent. It may be caused by a failure in the tracking systems or because the player falls out of the camera view (Gutiérrez-Pérez and Agudo, 2024a) for some seconds. Most vision-based algorithms (Giancola and Ghanem, 2021; Zhou et al., 2021; Denize et al., 2024; Sorano et al., 2021; Sanford et al., 2020; Baikulov, 2023; Philipp Singer, 2022) found a way to disregard this lack of information, even methods that use video tubes apply some type of padding or ignoring strategy. But the ones based on the players' trajectories (Vidal-Codina et al., 2022; Kim et al., 2023) are not robust to this lack of data as they need full visibility of the arrangement of the players on the pitch.

Sparse vs. Touch Events: Sparse events refer to soccer actions scattered in time, like foul, corner, penalty, goal, yellow or red card. These can help to give a brief understanding of the course of the game but are not enough for a soccer analyst to examine the match in depth, check if formation decisions outperformed,

Table 1: **Modalities of events detection methods.** Inputs are depicted as: ● - previous events class and their position in field, ■ - players and ball trajectories, ▲ - broadcast video, ▼ - tactical camera video, ◆ - method uses video tubes. Missing data column illustrates: ✓ - can work when missing players information either visual or positional, ✗ - needs dense information of all players or events. Player identification is represented as: ✓ - outputs which player is performing the event, □ - outputs in which region of the field the event happens but not the player, ✗ - does not output player information.

Paper	Event type	Input	Missing data	Player identification	↓ Window [s]
NetVLAD++ (Giancola and Ghanem, 2021)	Sparse	▲	✓	✗	5-60
Zhou et al. (Zhou et al., 2021)	Sparse	▲	✓	✗	5-60
Comedian (Denize et al., 2024)	Sparse	▲	✓	✗	1-5
NMSTPP (Yeung et al., 2023)	Touch	●	✗	□	-
Seq2Event (Simpson et al., 2022)	Touch	●	✗	□	-
Vidal-Codina et al. (Vidal-Codina et al., 2022)	Touch	●■	✗	✓	20
BallRadar (Kim et al., 2023)	Touch	■	✗	✓	2
PassNet (Sorano et al., 2021)	Touch	▲◆	✓	✗	1-4
Sanford et al. (Sanford et al., 2020)	Touch	■ / ▼◆	✓	✗	1.7
Baikulov (Baikulov, 2023)	Touch	▼	✓	✗	1
Singer et al. (Philipp Singer, 2022)	Touch	▼	✓	✗	0.15-0.7
Ours	Touch	▼◆	✓	✓	0.6

or recognize which players stood out. Sparse events provide the context to interpret the wider information that events like passes and receptions unravel. These other events that provide sufficient details to go over soccer matches are touch events.

Touch events occur when a player touches the ball. These are passes and receptions that may later be subdivided into types of passes such as crosses or throw-ins. To glimpse the difference between sparse and touch events notice that the average of events per match in Premier League season 2020-2021 is 2.7 for goals and 2.9 for yellow cards, while receptions and passes appear 696 and 940 times, respectively, per match.

SoccerNet-v2 dataset (Deliege et al., 2021) is one of the largest available soccer events datasets, with more than 500 match recordings from TV broadcast and 17 types of events annotated. Although many strategies (Giancola and Ghanem, 2021; Zhou et al., 2021; Denize et al., 2024) had been designed to detect actions, their task focuses only on sparse events that fall short of what soccer analysts need. Soccer-Net (Deliege et al., 2021) creators identified the need to spot more fine-grained actions and released a new challenge called Ball Action Spotting. As these new actions are closer in time, one cannot use broadcast video with changes in scenes, zooms, and replays. For this reason, a new dataset was published (Deliège et al., 2023) with continuous footage that keeps most of the players on shot, which is called a tactical camera. Our approach works on the mentioned tactical camera videos to detect those touch events that provide the crucial data that soccer experts need.

Prediction with Past Events Sequences: Previous research (Yeung et al., 2023; Simpson et al., 2022) uses information from previous events to try to predict the next up, which could be a good initial approxi-

mation of the problem. However, their results may be distorted by the use of limited information. Their methods are biased toward predicting shots because of the high influence of the event positional information they use on the type of events in the processed sequence.

Detection with the Location of Players and the Ball: Using trajectories data has a clear drawback, the low availability of datasets. This lack of public data relies on the high cost of its gathering, as it requires expensive hardware and meticulous computation of homographies.

The majority of methods that employ trajectories use a set of rules that rely on meticulous spatial information, and some of them need 3D coordinates. If one wanted to obtain the requested data from a video recording, one would need to perform a camera calibration (Gutiérrez-Pérez and Agudo, 2024b) delimiting the field using the visible lines and then compute a set of homographies per frame that would probably produce not accurate enough location information. Today, the required level of accuracy is collected by placing GPS sensors on players and the ball or multiple cameras around the pitch, between 16 and 20 in every stadium (Liga, 2020), and doing so reaches exorbitant prices.

We were not able to compare to methods employing coordinates from players, ball and/or events because their datasets not only are inaccessible but do not have match recordings either. Apart from the high cost for accessing the data, the inconvenience of this practice is that they do not have enough time precision. State-of-the-art methods (Kim et al., 2023; Vidal-Codina et al., 2022) spot actions using positional information in the field by first determining the ball possessor, but their temporal accuracy is of 2 and 20 seconds, respectively. We know from soccer ex-

perts that less than a second of precision is needed for touch events. Our model does not need real coordinates from players, ball, or previous events and still achieves finer temporal precision.

Spotting with Full Frame from Video Recording: Recent works have switched to exploiting video footage because images constitute a richer source of information. (Sanford et al., 2020) conducted an ablation for both trajectories and image-based solutions. Although their acceptance window takes 1.7 seconds, they show how most of their successful predictions fall inside a closer range of 0.5 seconds. Despite these models being able to detect in which frame events happen with a reasonable temporal precision, the obstacle for soccer analysts to use the information provided is that they only provide temporal information about detected events, but give no clue of which is the player performing those.

The winner of the SoccerNet Ball Action Spotting challenge (Baikulov, 2023) used a transfer learning approach fine-tuned with a sampling strategy to combat class imbalance. Both this and the winner of the Bundesliga Data Shootout (Philipp Singer, 2022), were using grayscale neighbor frames stacked in triplets as the color channels of an image before extracting the features with a 2DCNN. There is no possibility of testing our method on the previously mentioned challenges data because they do not contain bounding boxes for the players. Nevertheless, we still compare with the state of the art as we tested (Baikulov, 2023) in our dataset matches, proving our superiority, as will be shown later.

Locating with Video Tubes: As (Yu and Yuan, 2015) state, video tube proposals work well for dynamic action recognition with moving cameras. (Honda et al., 2022) use both trajectories information and video tubes to predict who the receiver of a pass is. However, their dataset only considers successful passes in situations with all players visible.

A Bundesliga Data Shootout contender (Yamamoto, 2022) proved the adequacy of focusing on the region where the ball is to spot events. They used a transfer learning approach pretraining the feature extraction with a ball detection task that upgraded their final results. Nevertheless, as the other participants in the challenge, they provide only temporal information of predicted events. The same obstacle arises with (Sanford et al., 2020). They prove employing players bounding boxes improves over their baseline, but they aggregate features from the tubes and do not retrieve which player is preforming each action.

In (Sorano et al., 2021) an object detection module that finds the bounding boxes for the players that are closer to the ball is used, affirming that it makes a significant contribution to their detections. They used the information obtained from this detector and combined it with features extracted from the whole video frame. Hence, their visual information is deficient as they shrink the image resolution and downsample the framerate to 5 Hz. Although they are in a similar task of detecting passes, we cannot access their private dataset of four matches. Nonetheless, we test their method in our dataset and prove their method cannot find passes as ours does.

Overall, none of the aforementioned approaches is capable of detecting touch events and the associated player with accurate temporal precision. Our method accomplishes this by (1) operating with visual information from video footage, (2) focusing on ball possession making use of video tubes and (3) applying a set of rules to spot passes and receptions with a temporal accuracy.

3 OUR APPROACH

In this section, we explain how our method finds the events and their performers. Figure 1 shows the three-stage pipeline we propose to sort out the problem, with video data and players bounding boxes as input. In the first block (in blue), a video tube for each visible player is cropped, extracting features and providing a score of the player being in possession of the ball. In the second one (in green), all scores in a particular frame are combined to find the player in possession of the ball, if any. Finally, in the last block (in red), a set of rules is exploited in order to determine precisely when touch events happen and which player is performing them. Next, we introduce in depth each of the blocks in our proposal.

3.1 Video Tubes Extraction

This first stage makes use of spatio-temporal video tubes, which pay off in moving camera situations for dynamic action identification (Yu and Yuan, 2015). To generate the video tubes, we consider a collar of T_f frames, i.e., we collect one of the input bounding boxes, cropping a region $H \times W \times Ch$ in a specific frame together with the corresponding one in the previous and next T_f frames, obtaining a video tube of $2T_f + 1$ frames, where H, W and Ch indicate the color image resolution. To normalize the data, for each box we enlarge the smallest rectangle side to make it a square, then add 20% extra margin to include more visual context, and resize them to 128×128 pixels (see Figure 2), obtaining the same size for all the players.
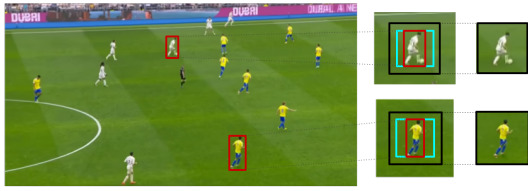
A clear way to identify a possessor is by finding

Figure 2: **Video tubes extraction. Left:** In every input image, a bounding box extraction (in red) per player is applied. Two instances are displayed in the figure. **Middle:** Zooming the selected player by means of a bounding box in red. To normalize data, the square that results from enlarging smallest rectangle side and the square after adding extra margin are displayed in cyan and black, respectively. **Right:** Final region to be cropped.

the ball in their bounding box region. We know that our algorithm does that because some of the false positives we detect include players without the ball but moving their legs forward when they have completely white boots. After visually analyzing that effect, we found that three frames is the time the ball takes to get from the edge of the video tube to the player or to leave the cropped region when a pass is performed. Without loss of generality, the parameter T_f is then set to 3 frames because there should be enough frames to appreciate the action that is being performed. This observation is consistent with (Honda et al., 2022), where they demonstrate that longer sequences lead to a decrease in accuracy due to excess context information.

3.2 Possession Likelihood from Video Tubes

In this stage, we propose to obtain the possession likelihood from the video tubes extracted above. To this end, we extract a feature vector independently for every image of a player video tube using ResNet50 (He et al., 2016), fine-tuned for this task and with a Temporal Shift Module (TSM) (Sudhakaran et al., 2020). We included this module because, as the authors manifested, it shifts channels in time to highlight the narrow differences between consecutive frames. As we use dense visual information without downsampling the frame rate, the dissimilitude from a frame to the next one is subtle. We take the penultimate layer output of the ResNet (He et al., 2016) and concatenate the features from all tube frames. With a fully connected layer, we reduce the time dimension, resulting in a 256-feature vector.

After finding the embedding for a video tube, another dense layer breaks it down to two values, being the possessor or not. From this classification problem we derive the likelihood of a video tube following a player in possession of the ball, which is necessary

to determine the actual possessor from all the players in the field and subsequently find the touch events they may be performing. In training, we use Additive Angular Margin (AAM) loss over common Softmax. This was implemented first in deep face recognition tasks (Deng et al., 2019) to enforce inter-class diversity and to solve intra-class appearance variations, which also apply to our problem, as in the same class we may find players with different positions or clothes. The AAM loss function for the *i*-th sample belonging to the *y*-th class can be written as:

$$\mathcal{L}_f = -\log \frac{e^{s \cdot cos(\theta_{y_i} + m)}}{e^{s \cdot cos(\theta_{y_i} + m)} + \sum_{j=1, j \neq y_i}^{N} e^{s \cdot cos\theta_j}}, \quad (1)$$

where $x_i \in \mathbb{R}^d$ is the deep feature of the *i*-th sample and $W \in \mathbb{R}^{d \times N}$ indicates the weights, with d the embedding feature dimension and N the number of classes. θ_j stands for the angle between the feature x_i and the weight W_j. We set the value of the scaling variable s to 1, where s is the radius of the hypersphere in which the learned embedding features are distributed. The parameter m, set to 0.5, represents the margin penalty that will enhance the aforementioned intra-class and inter-class relations.

3.3 Per-Frame Possessor Identification

With possession likelihoods from isolated video tubes, we move to a more genuine situation with all players in the frame trying to point out which one is in control of the ball, if any. At this stage, we add a head to the network that chooses between the only two feasible options, a single player is in possession of the ball or none of them. For a given frame t, the new head takes as input the likelihoods of all the players in a window frame from $t - T_p$ to $t + T_p$, i.e., the dimensionality of the input is $(2T_p + 1) \times N$ dimensional, where T_p is the collar of frames set to provide temporal context and N the number of players. Checking the slope that possession likelihoods show when changing from possession to not possession or vice versa, we observe that it takes five frames to completely toggle, so we adopt two frames as the value for our parameter T_p. This is directly related to the previously specified T_f value (see Section 3.1), and varying it would also vary the slope in possession likelihoods and, therefore, the preferable value for T_p. When a player is not visible in any of the frames, we add zero padding for their likelihood in that frame. We smoothed these inputs using a Gaussian filter to reduce the effect of noisy spurious detections.

After that, the possessor identification head uses Conv-TasNet (Luo and Mesgarani, 2019), a convolutional solution created for speech separation. This
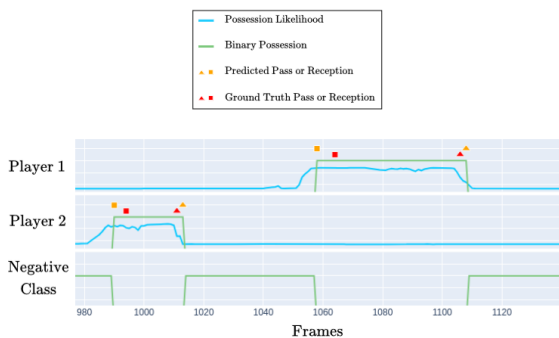
Figure 3: **Events estimation.** The figure shows our model output at the three stages and the corresponding ground truth for a 160-frame chunk. As it can be seen, our estimation (orange symbols) is very close to the reference one (red symbols).

network uses 1D convolutions in the time domain to separate the speakers from an audio input, a task that uncovers clear similitude with ours at this stage, where we want to find who is the possessor of the ball among all players in the pitch. Still taking advantage of the resemblance to speech recognition tasks, we add a Time-Delay Neural Network (TDNN) (Waibel et al., 2013) that will let our model scan the past and future of possession scenarios in a time-shift invariant way. At this point, we do an average pooling to end up with only a value per player. Lastly, there are two fully connected layers that get us to the output of this stage. The first expands the embedding size to capture the relations between all players. The second shrinks back to a single hot vector of size $N+1$, that is, each player that could be the possessor plus the negative class if none of them are. Thanks to that, we have a guess for who the possessor is in every frame, if any. At this stage, we use a cross-entropy loss between the prediction and the ground truth class when training.

3.4 Touch Events Detection

Touch events get their name from the requirement that a player is in contact with the ball when performing them. Knowing the possessor of the ball in every frame, if any, we designed a logic that tags a reception whenever a player gains possession of the ball and a pass when they are no longer the possessor. In this logic, there are two parameters, T_s and T_e, that help filtering out some false positives that may appear. When a player touches the ball just once, annotators tag a single-pass event in the ground truth instead of annotating both a reception and a pass. To match this, we make use of T_s, set to seven frames, which skips receptions for first-touch passes. The value of T_e determines the minimum required frames a possession

sequence must last to be considered. The majority of false positives appear when the ball passes in front of a player who is behind it some meters away. This is one of the main challenges for models that use visual information only, but setting the value of T_e to three we discard a large part of them. In Figure 3 one can observe the output of our approach at each of the three stages for a chunk of 160 frames and how the final predicted events result close in time to the ground truth.

4 EXPERIMENTAL RESULTS

Our data comes from 25Hz tactical camera videos of 36 matches (28 training, 4 validation, and 4 test) from Spanish LaLiga (1st and 2nd division) and Italian SerieA season 2022-2023 with player bounding boxes that we use as our input, and the touch events in frames precision that become our ground truth.

As our full pipeline is composed of 3 blocks, we propose to evaluate everyone of them, incorporating some comparisons w.r.t. competing approaches when possible. Particularly, we consider three tasks: (1) obtaining possession likelihoods of isolated video tubes, (2) identifying the possessor, if any, from all players present in a frame and (3) spotting the touch events with a super-tight temporal precision.

4.1 Possession Likelihood from Video Tubes

To create a first dataset of isolated video tubes, we select every frame in which a touch event occurs and crop a tube for each player. Non-visible players in all the context frames or overlapping with the positive box are discarded. For every positive tube we have approximately 20 negatives, so the dataset is very imbalanced, with 75,683 positives and 1,516,586 negatives.

In this first task, we evaluate the model ability to discern whether an isolated video tube follows on a player in possession of the ball. To overcome the imbalance, during training we force a 50% ratio of positives in every batch, with the other half randomly selected negatives. We drop the remaining negative samples at the end of every epoch. To prove that using video tubes instead of still images makes sense, we train a Timeless model that only uses features from the middle frame of the tubes. Next, we set a Baseline without shift modules to corroborate the convenience of TSM (Lin et al., 2019) or GSM (Gate-Shift Module) (Sudhakaran et al., 2020). As shown in Table 2, not considering the temporal

Table 2: **Possession likelihood from video tubes analysis.** The table includes precision, recall and Area Under Receiver Operating Characteristic (AUROC) curve metrics. In all cases, two possibilities by using ResNet18 and ResNet50 are provided.

Model	ResNet	↑ Precision	↑ Recall	↑ AUROC
Timeless	18	73.12	92.80	99.10
	50	77.58	92.87	99.14
Baseline	18	82.84	97.90	99.78
	50	85.85	97.59	99.78
Baseline + TSM	18	80.73	97.67	99.76
	50	<u>86.29</u>	**98.01**	**99.81**
Baseline + GSM	18	84.41	97.77	99.79
	50	**88.12**	<u>97.85</u>	**99.81**

Table 3: **Possessor identification evaluation.** The table reports purity and coverage on possessor sequences as well as Accuracy per frame showing the use of Gaussian filter. As it can be observed, our solution provides better solution than the baseline.

Model	Gaussian filter	↑ Purity	↑ Coverage	↑ Accuracy
Max + Threshold	No	49.87	66.96	70.12
	Yes	52.65	<u>67.50</u>	70.60
Ours w/o CTN+TDNN	No	44.08	59.83	60.90
	Yes	44.19	59.74	60.86
Ours with CTN+TDNN	No	**57.03**	66.87	<u>70.95</u>
	Yes	<u>56.61</u>	**67.79**	**71.88**

context achieves a minor performance according to all metrics, while ResNet50 (He et al., 2016) always outperforms ResNet18 (He et al., 2016). Moreover, adding the shift modules to the feature extraction is advantageous, as the solution always beats that provided by the baseline. TSM (Lin et al., 2019) is chosen because, for our use case, a higher recall is preferred, as we will see later.

4.2 Per-Frame Possessor Identification

After the classification task, the model can retrieve the likelihood of an isolated video tube following a player in control of the ball. We now have to distinguish between the video tubes of all players in the frame and find who is the actual ball possessor, if any. To evaluate this task, we create a new dataset where the ground truth possessor is derived from the annotated touch events. A player is assigned as the possessor from the first touch event until the last one they perform consecutively. With this we have a background or negative class found between the last event performed by a player and the first event performed by the next one. We reduce the number of matches because now, unlike for the previous task, we can use all the frames when the ball is in play, that is, when the game is not stopped. The new dataset consists of 5 new matches: 3 for training, 1 for validation, and 1 for testing.

In the evaluation of this task, apart from the usual per-frame accuracy, we introduce purity and coverage, metrics usually employed for segment-wise comparison in audio speaker change detection. The use of standard metrics like precision and recall requires the definition of a tolerance parameter that every author fixes to set the maximum distance to be matched between boundaries and still will not show the dissimilarity between segments. For that reason, we adopt these metrics to suit this segment-wise comparison. According to (Bredin, 2017), given $\mathcal{R}$ the set of reference possessor segments for our task and $\mathcal{H}$ the set of hypothesized segments, coverage is defined as:

$$\text{coverage}(\mathcal{R}, \mathcal{H}) = \frac{\sum_{r \in \mathcal{R}} \max_{h \in \mathcal{H}} |r \cap h|}{\sum_{r \in \mathcal{R}} |r|}, \quad (2)$$

where $|r|$ is the duration of segment r and $r \cap h$ is the intersection of segments r and h. Purity is computed analogously by interchanging $\mathcal{R}$ and $\mathcal{H}$ in Equation (2). An over-segmented hypothesis with too many possession changes implies high purity but low coverage because possessor predictions cover a low percentage of the ground truth. In contrast, an under-segmented hypothesis implies a high coverage but low purity.

We test the fully connected network with and without ConvTasNet and TDNN (Waibel et al., 2013), while also toggling the use of a Gaussian filter. We ablate them against a straightforward 2-step solution. It first checks whether any player has a likelihood over a given threshold, set to 0.5. If not, it chooses the negative class; otherwise, it selects the player with the maximum likelihood. Table 3 shows how this straightforward solution is better than a simple neural network, but the incorporation of the audio separation strategies outperforms in this analogous task of finding the possessor. We also observe how the regularization from the Gaussian filter slightly upgrades the outcome.

4.3 Touch Events Detection

As it was discussed in Section 2, datasets and source code for event detection tasks are usually not available, as they are normally linked to private companies. A dataset (Deliège et al., 2023) for touch events was available, but it did not contain information about player bounding boxes in the image and, therefore, we were unable to use it. We compare to the top-performing model in that dataset, but contrast to ours in a new set of matches that none of the methods has already seen before. This new test set comprises 6 matches from LaLiga 2023-2024 with various stadium sizes and weather conditions. To do this com-

Table 4: **Events detection evaluation and comparison.** The table reports the results separated by pass and reception. Precision, Recall and F1 are shown for both 0.6- and 1-second windows. Mean Average Precision with 1 second acceptance windows (mAP@1) is the metric used in the state of the art.

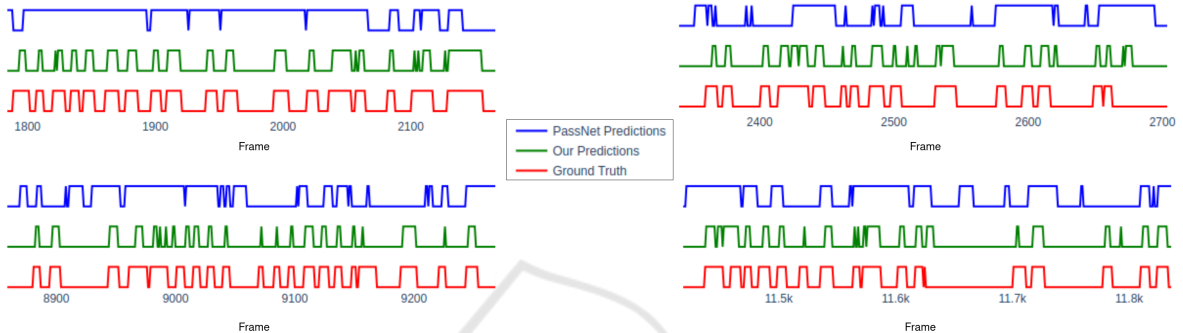| Model	Event	0.6s			1s			
		↑ Prec	↑ Rec	↑ F1	↑ Prec	↑ Rec	↑ F1	↑ mAP@1
Baikulov (Baikulov, 2023)	Recep	-	-	-	-	-	-	-
	Pass	**77.15**	52.36	62.38	**81.22**	56.44	66.60	29.09
Ours	Recep	54.23	57.53	55.83	64.96	67.12	66.02	38.36
	Pass	70.12	**64.67**	**67.28**	77.62	**71.33**	**74.34**	**44.15**

Figure 4: **Four chunks from a test set match** that show how our model fits the ground truth better than PassNet (Sorano et al., 2021), even when using their criteria of non-instantaneous passes.

parison, we separate detections of passes from receptions, because (Baikulov, 2023) is trained to spot passes and drives. We can contrast our pass predictions to theirs, but receptions do not map to drives as they are semantically different.

Table 4 shows how they obtain a better precision and we achieve a higher recall in the 0.6- and 1s-acceptance windows. According to soccer experts, when treating passes as instantaneous events that happen in a specific frame, a model with high recall that lets them filter out false positives is preferable. A model with high precision and lower recall would force them to go again through the match to find the false negatives missed by the model, achieving no reduction in the time they spent analyzing a match. Apart from that, we also reach a higher F1 and larger mean Average Precision for 1 second acceptance windows (mAP@1), the latter being the metric they use for their evaluation.

We also show a comparison with PassNet (Sorano et al., 2021). Apart from only looking for passes and not retrieving which is the player passing or receiving the ball, the main difference between their model and ours is that they consider passes to have a start frame and an end one. Not accounting for passes as instantaneous events makes their metrics focus on how many frames in the segments considered positives can the model predict as frames from a pass. But they will never be able to find how many instances of a real

pass occurred in a chunk of a game.

Their dataset is not available, but the code for their method is, so we train it in 9 of our dataset matches (225% times the data in their original train set) and compare using the same set of 6 test matches that we used in the previous comparison. We try to train their algorithm on instantaneous events to use our metrics for comparison, but PassNet (Sorano et al., 2021) only learns to put every frame to negative due to the imbalance in frames. Therefore, we train it by considering their criteria of each pass having a start and end frame for each pass, and with this we are able to replicate very similar results to the ones they show in their paper using their metrics. Having PassNet (Sorano et al., 2021) trained, we try to use our metrics by defining the start frame of each pass as the frame in which the event happens, but they barely get to a 19% precision and 21% recall, so comparison here is nonsense.

For being able to compare we have to redefine our network output considering the frames after a pass detection as positives and the ones after a reception as negatives, building this way predictions that follow their non-instantaneous definition of a pass. With this twist, we can compare their domain using their metrics. As seen in Table 5, we outperform PassNet (Sorano et al., 2021) in all their metrics except the number of positive frames detected as positive, that is, the recall of frames contained in passes. Figure 4 shows how their model has a high recall on passes because

Table 5: **Pass detection comparison.** The table exposes the Precision, Recall and F1 metrics for pass detection, by considering passes as segments with a start frame and an end one. Therefore each frame is part of a pass (Pass) or a negative frame (No Pass).

Model	Pass			No Pass		
	↑ **Prec**	↑ **Rec**	↑ **F1**	↑ **Prec**	↑ **Rec**	↑ **F1**
PassNet (Sorano et al., 2021)	43.13	**72.24**	54.01	70.57	41.22	52.04
Ours	**69.95**	52.74	**60.14**	**74.46**	**86.10**	**79.86**

it just learned to predict positive in high-action regions. For that reason, they have a low precision in passes and a low recall on the negative class. From their output, it would be impossible to detect when the passes happen or even just the number of instances of passes that appear. Our model, despite being trained on instantaneous passes, is capable of capturing the dynamics that real passes show in the ground truth.

4.4 Extended Model Output

Figure 3 shows a simple example of two players making two successive passes to clarify the model output at every stage. We extend the model outputs for a longer 550 frames (22s) chunk in Figure 5. The chunk starts with a pass (specifically, a throw-in) performed by the player $P3$ and follows a possession of the players in the same team. Some players are clearly not in possession of the ball during this chunk, so by removing those we focus only on relevant players. We specifically selected a chunk where some of our model errors are represented. We can observe in Figure 5(b) how we wrongly detected a reception before a pass event where the ground truth was a single first-touch pass. In Figure 5(a) there is a false positive that was not filtered out, but in Figure 5(c) a spurious detection for the same player was correctly discarded. A false negative appears in Figure 5(d), where a pass was missed because our model considered the detection too spiky.

5 CONCLUSIONS

In this paper, we have proposed a method that precisely detects touch events in soccer from videos. Unlike most of the recent work, our approach is capable of retrieving which player from all players in the pitch is the one performing each action while delivering at the needed temporal accuracy. Despite low data availability, we compare to two state-of-the-art models that partially fit our task requirements and prove our superior robustness in the detections even when using their metrics. In the future, we plan to follow the intricate duty of making an end-to-end version of our whole
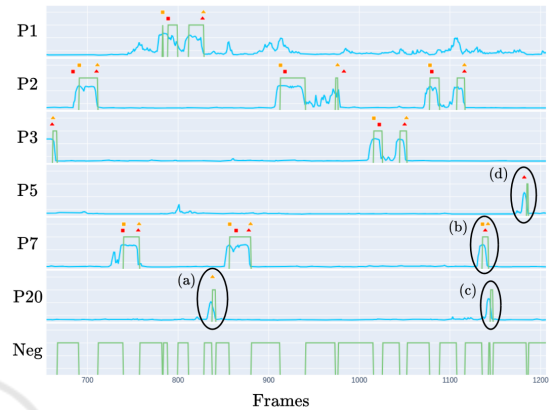

Figure 5: **Model outputs and limitations.** The graph shows the model outputs focused on relevant players for a chunk of 550 frames. A false positive can be observed in (a), a correctly filtered spurious detection in (c) and a false negative missed in (d). In (b) a first touch pass was detected as a multiple contact reception-pass. See legend in Figure 3.

model, as at the moment it is trained at every stage independently.

ACKNOWLEDGMENT

REFERENCES

Baikulov, R. (2023). Winning solution for soccernet ball action spotting challenge 2023. https://github.com/l Romul/ball-action-spotting.

Biermann, H., Komitova, R., Raabe, D., Müller-Budack, E., Ewerth, R., and Memmert, D. (2023). Synchronization of passes in event and spatiotemporal soccer data. *Scientific Reports*, 13(1):15878.

Bredin, H. (2017). Tristounet: triplet loss for speaker turn embedding. In *ICASSP*, pages 5430–5434.

Capellera, G., Ferraz, L., Rubio, A., Agudo, A., and Moreno-Noguer, F. (2024a). Footbots: A transformer-

based architecture for motion prediction in soccer. In *ICIP*, pages 2313–2319.

Capellera, G., Ferraz, L., Rubio, A., Agudo, A., and Moreno-Noguer, F. (2024b). Transportmer: A holistic approach to trajectory understanding in multi-agent sports. In *ACCV*.

Cioppa, A., Giancola, S., Somers, V., Magera, F., Zhou, X., Mkhallati, H., Deliège, A., Held, J., Hinojosa, C., Mansourian, A. M., et al. (2023). Soccernet 2023 challenges results. *arXiv preprint arXiv:2309.06006*.

Cossich, V. R., Carlgren, D., Holash, R. J., and Katz, L. (2023). Technological breakthroughs in sport: Current practice and future potential of artificial intelligence, virtual reality, augmented reality, and modern data visualization in performance analysis. *Applied Sciences*, 13(23):12965.

Deliege, A., Cioppa, A., Giancola, S., Seikavandi, M. J., Dueholm, J. V., Nasrollahi, K., Ghanem, B., Moeslund, T. B., and Van Droogenbroeck, M. (2021). Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In *CVPR*, pages 4508–4519.

Deliège, A., Cioppa, A., Giancola, S., Seikavandi, M. J., Dueholm, J. V., Nasrollahi, K., Ghanem, B., Moeslund, T. B., and Droogenbroeck, M. V. (2023). Ball action data and labels for soccernet ball action spotting challenge. https://www.soccer-net.org/data#h.ykgf675j127d.

Deloitte (2023). Annual review of football finance 2023.

Deng, J., Guo, J., Xue, N., and Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699.

Denize, J., Liashuha, M., Rabarisoa, J., Orcesi, A., and Hérault, R. (2024). Comedian: Self-supervised learning and knowledge distillation for action spotting using transformers. In *WACV*, pages 530–540.

Giancola, S., Amine, M., Dghaily, T., and Ghanem, B. (2018). Soccernet: A scalable dataset for action spotting in soccer videos. In *CVPRW*, pages 1711–1721.

Giancola, S. and Ghanem, B. (2021). Temporally-aware feature pooling for action spotting in soccer broadcasts. In *CVPR*, pages 4490–4499.

Goes, F., Meerhoff, L., Bueno, M., Rodrigues, D., Moura, F., Brink, M., Elferink-Gemser, M., Knobbe, A., Cunha, S., Torres, R., et al. (2021). Unlocking the potential of big data to support tactical performance analysis in professional soccer: A systematic review. *European Journal of Sport Science*, 21(4):481–496.

Gutiérrez-Pérez, M. and Agudo, A. (2024a). No bells just whistles: Sports field registration by leveraging geometric properties. In *CVPRW*.

Gutiérrez-Pérez, M. and Agudo, A. (2024b). Pnlcalib: Sports field registration via points and lines optimization. *SSRN*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*, pages 770–778.

Honda, Y., Kawakami, R., Yoshihashi, R., Kato, K., and Naemura, T. (2022). Pass receiver prediction in soccer

using video and players' trajectories. In *CVPR*, pages 3503–3512.

Kim, H., Choi, H.-J., Kim, C. J., Yoon, J., and Ko, S.-K. (2023). Ball trajectory inference from multi-agent sports contexts using set transformer and hierarchical bi-lstm. In *ACM SIGKDD*, pages 4296–4307.

Liga, D. F. (2020). Positional tracking takes a big leap forward as latest generation is installed at bundesliga and bundesliga 2 stadiums. https://www.dfl.de/en/innovation/positional-tracking-takes-a-big-leap-forward-as-latest-generation-is-installed-at-bundesliga-and-bundesliga-2-stadiums/.

Lin, J., Gan, C., and Han, S. (2019). Tsm: Temporal shift module for efficient video understanding. In *ICCV*, pages 7083–7093.

Luo, Y. and Mesgarani, N. (2019). Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *T AUDIO SPE*, 27(8):1256–1266.

Philipp Singer, Yauhen Babakhin, P. P. (2022). Winning solution for bundesliga data shootout. https://www.kaggle.com/competitions/dfl-bundesliga-data-shootout/discussion/359932.

Sanford, R., Gorji, S., Hafemann, L. G., Pourbabaee, B., and Javan, M. (2020). Group activity detection from trajectory and video data in soccer. In *CVPRW*, pages 898–899.

Simpson, I., Beal, R. J., Locke, D., and Norman, T. J. (2022). Seq2event: Learning the language of soccer using transformer-based match event prediction. In *ACM SIGKDD*, pages 3898–3908.

Sorano, D., Carrara, F., Cintia, P., Falchi, F., and Pappalardo, L. (2021). Automatic pass annotation from soccer video streams based on object detection and lstm. In *ECML*, pages 475–490.

Sudhakaran, S., Escalera, S., and Lanz, O. (2020). Gate-shift networks for video action recognition. In *CVPR*, pages 1102–1111.

Vidal-Codina, F., Evans, N., El Fakir, B., and Billingham, J. (2022). Automatic event detection in football using tracking data. *Sports Engineering*, 25(1):18.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. J. (2013). Phoneme recognition using time-delay neural networks. In *Backpropagation*, pages 35–61. Psychology Press.

Yamamoto, D. (2022). Third place solution for bundesliga data shootout. https://www.kaggle.com/competitions/dfl-bundesliga-data-shootout/discussion/360236.

Yeung, C. C., Sit, T., and Fujii, K. (2023). Transformer-based neural marked spatio temporal point process model for football match events analysis. *arXiv preprint arXiv:2302.09276*.

Yu, G. and Yuan, J. (2015). Fast action proposals for human action detection and search. In *CVPR*, pages 1302–1311.

Zhou, X., Kang, L., Cheng, Z., He, B., and Xin, J. (2021). Feature combination meets attention: Baidu soccer embeddings and transformer based temporal detection. *arXiv preprint arXiv:2106.14447*.
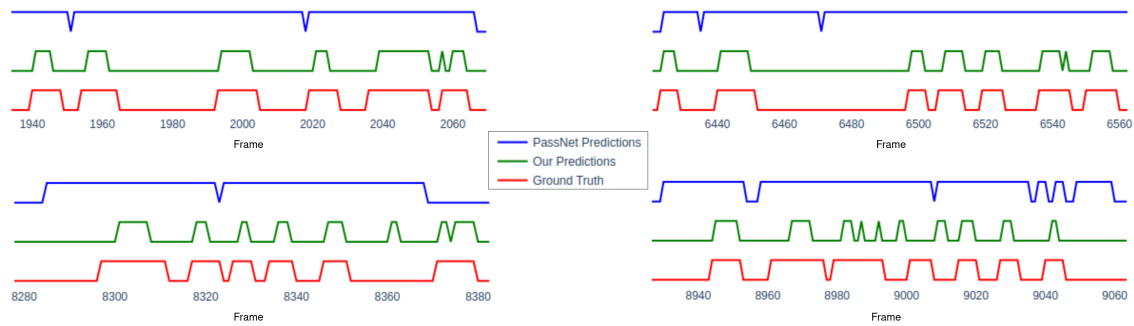
Figure 6: **Four extra chunks from a test set match** that show how our model fits the ground truth better than PassNet (Sorano et al., 2021), even when using their criteria of non-instantaneous passes.

Table 6: **Baikulov's method (Baikulov, 2023) folds.** Events detection results for each of Baikulov's method folds isolated and the arithmetic mean.

Fold	0.6s			1s			
	↑ Prec	↑ Rec	↑ F1	↑ Prec	↑ Rec	↑ F1	↑ mAP@1
0	71.40	64.71	67.89	76.50	70.01	73.11	33.37
1	86.32	37.37	52.16	90.08	39.98	55.38	20.53
2	79.65	47.66	59.64	83.74	51.62	63.86	27.06
3	78.15	54.70	64.36	82.16	58.98	68.67	31.66
4	79.01	54.42	64.45	82.03	57.88	67.87	29.54
5	74.19	53.93	62.46	78.32	58.30	66.84	30.38
6	71.36	53.74	61.31	75.73	58.34	65.91	31.06
Mean	77.15	52.36	62.38	81.22	56.44	66.60	29.09

APPENDIX

Video Tubes Crops Resizing: The size of bounding boxes in our dataset of matches recordings at 1920×1080px range from 15×20px for the furthest players that appear smaller in the image to 70×105px for the closest ones. We resize to 128×128 because it is the size of the regions to be cropped after adding the 20% extra margin to the biggest bounding boxes we find in our dataset, with this we avoid downsampling and therefore losing information in largest boxes.

Comparing with the State of the Art: To compare with (Baikulov, 2023), the winner of the SoccerNet Ball Action Spotting challenge (Cioppa et al., 2023), we just had to run their code in our matches footage downsampled from 1080p to 720p. Their approach employs a 7-fold cross-validation in the first pretraining step that leads to a set of 7 models. Therefore, we computed their best method performance as they did, computing the detection results (only for passes) for their 7 different-fold models and doing the arithmetic mean of their outputs. Table 6 exposes how each fold independently detects passes in our 6-match test set for event detection, and the aforementioned arithmetic mean that becomes the final output.

The comparison is done with the metrics we use for events detection (precision, recall and F1 at 0.6s

acceptance window) but we also consider their metric mean Average Precision with an acceptance window of 1 second (mAP@1). This metric was introduced by the SoccerNet first dataset (Giancola et al., 2018) and defines a prediction as a true positive if it lands inside an acceptance window of δ seconds. Then, varying the tolerance, they compute a Precision-Recall curve for each value of δ and finally average along all the classes. Unlike in the challenge they won where the acceptance windows range from 1 to 5 seconds, (Baikulov, 2023) exposes that actions are densely allocated and should be predicted more accurately using only 1 second windows, which goes along with our focus on temporally accurate detection.

In Figure 4 we also compare with PassNet (Sorano et al., 2021) and show how our model fits the ground truth better even when using their pass criteria. We add some zoomed in chunks in Figure 6 for extra comparison.