UNIVERSITY OF TWENTE
Faculty of Electrical Engineering, Mathematics &
Computer Science

INDIVIDUAL ACTION AND GROUP ACTIVITY RECOGNITION IN SOCCER VIDEOS

Master's thesis

B.G.A. Gerats

October 2020

STUDY PROGRAMMES
MSc. Interaction Technology
MSc. Computer Science

EXAMINATION COMMITTEE
dr.ing. G. Englebienne
dr.ir. L.J. Spreeuwers
prof.dr. D.K.J. Heylen
dr.ir. H. Bouma (TNO)
W. Uijens, MSc (TNO)

Individual Action and Group Activity Recognition in Soccer Videos

Beerend Gerats
Faculty of EEMCS
University of Twente

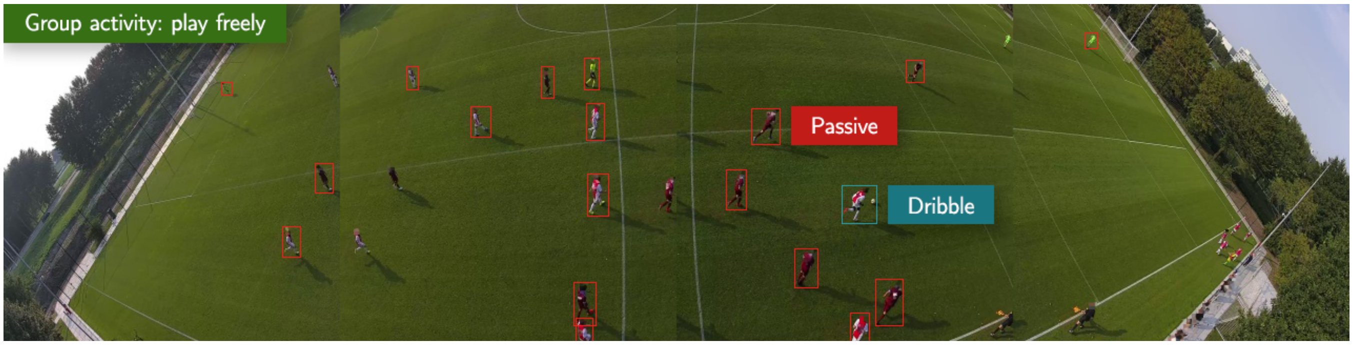


Figure 1: Action and group activity recognition in a soccer game.

ABSTRACT

Data and statistics are key to soccer analytics and have important roles in player evaluation and fan engagement. Automatic recognition of soccer events - such as passes and corners - would ease the data gathering process, potentially opening up the market for non-professional soccer analytics. We propose a novel method for the automatic recognition of soccer events from video. To the best of our knowledge, it is the first method that infers both individual actions and group activities simultaneously from soccer videos. Three key contributions in the proposed method are (1) the use of player-centric snippets as model input, (2) per-player feature extraction with an I3D CNN - based on RGB video and optical flow - and (3) the use of feature suppression and zero-padding in graph attention networks for feature contextualisation. The results show that the proposed method performs better than an alternative state-of-the-art method, designed for action and activity recognition in volleyball. Our method gains 98.7% accuracy for the recognition of eight actions and 75.2% for eleven activities.

1 INTRODUCTION

Professional sports and data have become inseparable, ranging from simple monitor devices, such as heart rate sensors, to Formula 1 racing cars that are fully packed with cameras and other sensors. The data streams are analysed by professionals to gain insight in the performance of athletes and to figure out how to do better than the opponents. Data analysis could play a major role in increasing the chances of winning, illustrated by the story of Billy Beane, general manager of baseball team *Oakland Athletics* from 1997 till 2015. Driven by data, he redefined how players are valued, contracting the most undervalued ones. By following this strategy, he was able to create low-cost teams that performed beyond all expectations [59]. Since then, the role of data gathering and statistical analysis only got bigger, in a wide variety of sports. For example, researchers at soccer team *Liverpool* created a model that automatically evaluates passes, shots and ball movements of more than 100.000 players [75]. The system was used for the acquisition of talented players and played a role in the appointment of their new manager in 2015. The wide availability of team and player evaluation tools [70] suggest that Liverpool is no exception in its dependence on data.

Data and statistics are commonly used in media as well, such as websites or broadcast shows that report on sport games. By the delivery of performance measures, these media increase fan engagement [1]. Often, the data used is referred to as *match event logs*. In soccer, one can think about the number of goals in a season, the number of corners by one team during a game or the number of successful passes by a particular player.

Companies that gather the event logs, called Competition Information Providers (CIPs), capture the event measurements by human annotation. Recently, Pappalardo *et al.* [71] described the procedure of data collection during soccer matches at CIP Wyscout. The annotation of events during one game is performed by a team of three or four people. Although the process is optimised by using annotation software and automatic data quality analysis, human labour remains necessary for event detection and the final quality assessments. Generally, CIPs offer match event logs from hundreds of games in dozens of professional competitions, which means that the companies largely invest in data collection. With the recent developments in computer vision and high-resolution cameras, it is likely that more and more will be invested in automated gathering of match event logs. Automatic event detection does not only result in a cheaper way of obtaining the measurements. Eventually, match data could be generated for amateur and professional youth teams as well, creating a new pool of customers for services that rely on soccer events. It is therefore relevant to search for effective methods to obtain match event logs using state-of-the-art computer vision techniques.

1.1 Contributions

In this study, we present a novel method for the automatic recognition of soccer events from video. To the best of our knowledge, it is the first method that infers both individual actions and group activities simultaneously from soccer videos. Besides, we have not seen other studies on event recognition in videos captured by a *one-perspective camera setup*, that is from one stationary perspective along the soccer field.

Three key contributions in the proposed method are (1) the use of player-centric snippets as model input, (2) per-player feature extraction with the two-stream I3D network [10] and (3) the use of feature suppression and zero-padding in graph attention networks (GATs) [96] for feature contextualisation. The player snippets are obtained using an Aggregated Channel Features (ACF) person detector [25] and a virtual camera that zooms in on each detected player, creating a standardised video frame cut-out. Feature suppression is the ability of a GAT to diminish large activations in a player embedding that correlate to a wrongly identified action class. Zero-padding is an approach to fill a graph with N player embeddings when less than N players are detected, without affecting the self-attention mechanism.

The proposed method is designed upon the Actor Relation Graph (ARG) [99] as baseline, a state-of-the-art method for action and activity recognition in volleyball. We show that the baseline is not directly applicable to soccer videos and that the proposed method performs better. Furthermore, we created the Soccer Dataset to train and evaluate our model. The dataset is created for this study only, although we show how one could construct a similar dataset.

1.2 Research questions

The described contributions give answers to (sub-)research questions that were formulated at the start of this study. Our main research question is:

RQ: "How to automatically recognise individual actions and group activities in soccer videos that are captured from one stationary perspective?"

Five sub-research questions were formulated, of which the first four questions are preparatory towards experimentation with the proposed method in RQ5. These are:

RQ1: "How to create a dataset for action and activity recognition in the soccer domain?"

RQ2: "Which soccer match events are relevant to be automatically gathered?"

RQ3: "Is the Actor Relation Graph method, used for action and activity recognition in volleyball, applicable to soccer?"

RQ4: "What are limitations to the Actor Relation Graph method when applied to soccer videos?"

RQ5: "Does the proposed method, based on player-centric snippets, I3D feature extraction, and graph attention networks with feature suppression and zero-padding, result in more accurate action and activity predictions than the Actor Relation Graph?"

The upcoming section discusses related work in event detection, action recognition, group activity recognition and soccer datasets. In Section 3, the design of the ARG and the proposed method are explained. Section 4 introduces the Soccer Dataset and explains how it is constructed. Results of the ARG as baseline and the proposed method on the Soccer Dataset are discussed in Section 5. The last two sections include limitations, future work and final conclusions that answer the research questions.

2 BACKGROUND

The automatic generation of match events in soccer games has been researched for at least three decades, mainly under the name of *event detection* [27][32][53][86]. The aim of research in this field is to detect temporal boundaries of a match event and classifying the cut-out sample accordingly. Generally, the methods are designed for television broadcasts that include cinematic shots such as slow-motions or close-ups.

Human action detection is a separate research field that aims to detect or recognise individual actions and group activities based on human movement rather than cinematic features. Research in *person detection* and *action localisation* aim to isolate people spatially and their actions temporally. A subset extends their method to be able to track the detected people such that movement trajectories can be constructed [67]. Methods in *action recognition* and *group activity recognition* take player bounding boxes or action tubelets as input, and

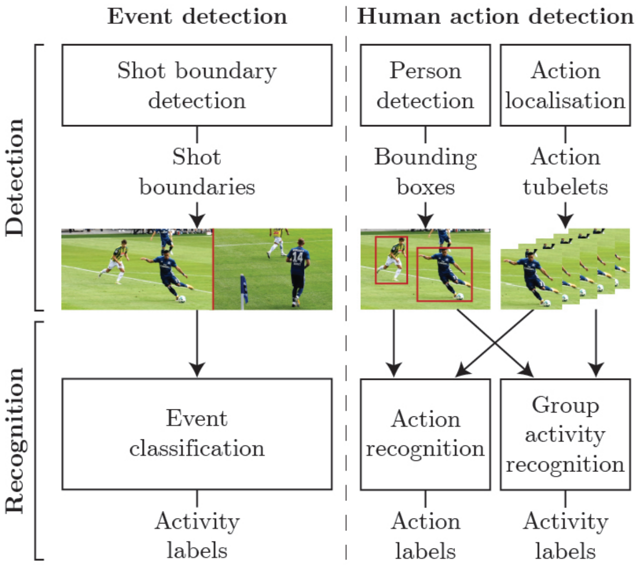


Figure 2: Two research areas event detection and human action detection aim to extract action and/or activity labels from sport videos.

classify the samples with action and activity labels. A subset is able to recognise actions and group activities simultaneously [8][9][72][99].

In Figure 2, it can be seen how the mentioned research fields relate to each other. Relevant work from these fields is discussed below, where we also clarify the decision to select the ARG as baseline. Finally, we consider several existing soccer datasets and describe their limitations.

2.1 Event detection

Early methods for the detection of highlight events in sport videos rely on the detection of low-level features by cinematic and object-based descriptors [27]. Cinematic features, such as dominant colours and camera motion, are based on general ways for television production teams to record soccer events on camera. For example, a goal attempt is often followed by a slow-motion shot of the event. Persons and the ball are detected using object-descriptors, based on object texture, shape and motion. For classification, the descriptors are combined with rule-based methods [80] or probabilistic models [40][100]. Note that for the use of hand-crafted features and rule-based classification, a priori knowledge about cinematic rules and game play is essential. We consider these dependencies undesirable as it limits the applicability of a model to broadcasts only. Also, it cannot be expected that such methods generalise well to new actions and activities, as they might be distinguishable from the other events only by applying whole new rules.

Methods that do not require a priori knowledge are often based on deep learning. Jiang *et al.* [46] use a combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to detect goals, goal attempts, corners and card events in soccer videos. Khan *et al.* [53] detect corners, shoots, goal-attempts and penalty kicks using a 3D-CNN. Not all methods try to classify events directly from video frames. Zhang *et al.* [106] propose to detect events from latent player embeddings and trajectories of the players and the ball. Their method creates a player embedding using a U-encoder on the pixels in the player's bounding box. As the embedding is in a reduced dimensionality, only characteristic information about the player is preserved, which often results in good features for classification. Similarly, our method creates latent player embeddings, but uses a CNN to do so.

2.2 Human action detection

When multiple persons and actions are present in a video, each person and action must be localised and isolated first, such that the resulting samples can be classified independently. The detection of people and the localisation of their actions in videos are two separate research domains.

2.2.1 *Person detection*.

Traditionally, rectangular bounding boxes form spatial boundaries to person detections in images and videos. More precise boundaries exist, in the form of segmentation masks [36], although their creation requires more computing power. Detections of the same person can be combined temporally to create trajectories. Even without action recognition, trajectories provide valuable insights for team analytics as they include information about player movements, running distances and positions. Applicability of methods for player detection and the creation of trajectories depends on the camera setup [87]. Manafifard *et al.* provide an extensive overview of available methods for person detection and tracking in soccer videos [67]. With Aggregated Channel Features (ACF) [25], persons can be detected in videos real-time and with only few computational resources. Our method requires person detections in large video frames of 8640 × 3840 pixels at 25 frames per second (fps). We use an ACF detector to be able to isolate the soccer players quickly.

2.2.2 *Action localisation*.

Where various methods exist that accurately detect persons in sport videos, it is not trivial how to isolate their actions temporally [81]. Kalogeiton *et al.* reported promising results for action tubelet generation on the UCF-Sports dataset [47], with 92.7 mean Average Precision (mAP) for detections with a minimal Intersection over Union (IoU) of 0.5 with ground truth tubelets. However, the videos in the set are temporally trimmed to the action and their method is only tested on small frames of 300 × 300 pixels. When considering state-of-the-art for datasets with untrimmed videos, such as ActivityNet-1.3 and THUMOS'14, the performance drops to mAP scores of 50.1 and 38.8 respectively [62].

The objective of this study is evaluate a model for action recognition rather than detection and it is expected that an off-the-self action detector would drastically lower its performance. Therefore, we train and evaluate our model with manually trimmed videos, meaning that a human annotator selects the precise moment that an event occurs. It is not uncommon to do so, as widely used datasets for action recognition in sports include trimmed videos exclusively, either by selecting a fixed time window [42] or by human annotation of the action duration [82].

2.3 Human action recognition

Spatio-temporally isolated actions are semantically categorised in the research area of human action recognition. The field can be separated into *action recognition*, which aims to classify action samples of individuals independently, and *group activity recognition*, which explores inter-human interactions to find shared activities or to analyse individual actions in context of other group members [92]. As the two domains share the same underlying techniques, early methods and the transition to deep learning are discussed first. Methods and benchmarks that are specific to group activity recognition are given afterwards.

2.3.1 ***Early methods.*** Early methods for action recognition are based on the detection of hand-crafted low-level features, extracted in mid-level action descriptors and resulting in high-level semantic interpretations [91]. For the extraction of low-level features, often local descriptors were used, such as Scale-Invariant Features Transform (SIFT), Histograms of Gradients (HOG) and Non-parametric Weighted Feature Extraction (NWFE) descriptors [50]. Such two-dimensional descriptors could be extended to three dimensions that describe motions over a short time period, or a tracker could be used to create long temporal trajectories [55]. Classification of the obtained features was typically performed by generative or discriminative models. Popular models were Hidden Markov Models (HMMs) [101], Support Vector Machines [76], Conditional Random Fields [98] and Neural Networks [44].

2.3.2 ***Deep learning.*** State-of-the-art methods for action recognition include deep learning architectures almost without exception. Such architectures, can combine low-level feature extraction, mid-level descriptors and action classification in one (end-to-end) trainable model. Three types of deep learning networks are most relevant to action recognition: spatio-temporal, multiple-stream and hybrid networks. Spatio-temporal networks, such as a 3D-CNN [45], search for volumetric patterns at different scales of the input videos. An extension thereof is the I3D CNN [10], which is a multiple-stream network that creates action predictions from RGB image frames as well as from optical flow. In a hybrid network for action recognition, spatial and temporal models, such as a CNN and an RNN, are combined subsequently [55]. Our method generates feature embeddings using the I3D.

2.3.3 ***Group activity recognition.*** Two domains are generally involved in group activity recognition: surveillance and sports. This is reflected by the two most popular benchmarks: the Collective Activity Dataset (CAD) [17] and the Volleyball Dataset [42]. An overview of publications on group activity recognition can be found in Table 22 (Appendix I).

The use of hybrid networks is especially popular for the recognition of group activities. In a first phase, a CNN extracts individual features and creates a latent embedding per group member. Often, the embedding has the form of a fixed-length vector. We will refer to this phase as *feature extraction*. Thereafter, a different network explores inter-human relations to update the embeddings accordingly and to predict the shared activities. We will refer to this phase as *feature contextualisation* as the independently created embeddings are put into the context of the embeddings from all other persons in the scene. RNNs with Long Short Term Memory (LSTM) [61][88][90][92] and Graph Convolutional Networks (GCNs) [41][99] are often used for the latter phase. Similar two-phase approaches, referred to as *bottom-up inference*, were explored before using HMMs [105], AND-OR graphs [4], hierarchical random fields [2] and hierarchical RNNs [42][78][97].

Different from action recognition is that many proposed methods make use of a graph as representation for the group members and their interactions [39][57][58][56][60][72][84][89]. In such models, a graph is constructed in which the nodes correspond to the detected persons and the edges represent inter-human relationships. Favourable for this approach is that attention is drawn to the important parts of the video, namely the individuals, and that their interactions are intuitively modelled.

2.3.4 ***Actor Relation Graph as baseline.*** The ARG [99] is a hybrid network that uses an Inception-V3 CNN [83] for feature extraction, uses GATs with self-attention [95] for feature contextualisation, and predicts actions and activities from videos simultaneously. From Table 1 can be observed that the method reaches near state-of-the-art performance on both benchmarks, which is an indication of good transferability to other domains, like soccer videos. Because the ARG uses the intuitiveness of a graphical model, it has an open-source implementation[1] available (where the top-3 methods on the Volleyball Dataset do not) and it gains a good performance

[1] https://github.com/wjchaoGit/Group-Activity-Recognition

on the recognition of individual actions, this method is selected as baseline for this study. The proposed approach for feature contextualisation will eventually be compared with the Actor-Transformer (AT) [31] method as well.

Method	Backbone	V-G	V-I	CAD
HDRM [42]	AlexNet	81.9%	-	81.5%
CERN [78]	VGG16	83.3%	-	87.2%
stagNet [72]	VGG16	89.3%	82.3%	89.1%
CCG-LSTM [84]	ImageNet	89.3%	-	93.0%
HRN [41]	VGG19	89.5%	-	-
SSU [8]	Inc-V3	90.6%	81.8%	-
SRG [39]	VGG16	91.4%	-	-
ST att. mech. [66]	-	91.7%	-	-
ARG [99]	**Inc-V3**	**92.5%**	**83.0%**	**91.0%**
CRM [6]	I3D	93.0%	-	85.8%
MLS-GAN [29]	ResNet-50	93.0%	-	91.7%
Actor-Transf. [31]	HRNet + I3D	94.4%	85.9%	91.2%

Table 1: State-of-the-art methods, in multi-class accuracy (MCA), on the Volleyball Dataset (V-G: group activities, V-I: individual actions) and the Collective Activity Dataset (CAD).

2.4 Soccer datasets

To train and evaluate a method for the recognition of soccer events, a dataset is required that contains one-perspective soccer video samples, player detections and labels of individual actions and group activities. However, no existing dataset could be found that satisfies these requirements. The dataset from D'Orazio *et al.* [26] and Tsunoda *et al.* [90] include only two and twenty minutes of soccer, respectively. Additionally, both use a multiple-perspective camera setup. The Soccer-8k dataset [30] contains video clips from broadcasts where the view is cropped to only the player that is in control of the ball. Last, although the dataset by Pappalardo *et al.* [71] includes a huge amount of match events from a wide variety of matches and competitions, the set does not include any video recordings. These findings motivated us to create a new dataset, which we present in Section 4

3 METHOD

In the first part of this section, we explain how the baseline ARG [99] operates. Thereafter, we discuss how the proposed method differs from the baseline architecture. Each approach is described along four phases in the data processing pipeline: data pre-processing, feature extraction per actor, feature contextualisation and model predictions.

3.1 Actor Relation Graph (ARG)

3.1.1 ***Pre-processing.*** The baseline ARG samples from five frames before and four frames after the annotated video frame at time t_0. As the videos are recorded with 25fps, the corresponding temporal window is from 0.2 seconds before to 0.16 seconds after t_0. We will denote this window as $[t_{-0.2s}, t_{0.16s}]$. When training the model, three of the ten frames are randomly selected and fed to the model to optimise its parameters. During inference, nine frames from $[t_{-0.16s}, t_{0.16s}]$ are fed to the model for evaluation. These are processed through the ARG in three batches of three subsequent frames and combined using mean-pooling right before the final predictions are made. The authors argue that sparse sampling during training yields better predictions due to increased input diversity. Moreover, the baseline does not pre-process the raw video input other than resizing the frames to 1280 × 720 pixels. We call these *full-field frames* as the images display the full soccer field.

3.1.2 ***Feature extraction.*** In the second phase, the baseline uses Inception-V3 [83] as backbone to extract a 157 × 87 × 1056 feature map per frame. Using Region of Interest (RoI) align, a 5 × 5 × 1056 feature map is cut out of the large feature map for every player. Features in this map relate to the player's area in the original frame, bounded by a detection bounding box. The feature maps are flattened and transformed to a d-dimensional vector using a fully connected layer and ReLU activation. Originally, d is set to 1024.

3.1.3 ***Feature contextualisation.*** To enable player embeddings to accumulate features from embeddings of nearby players, a set of H graph attention networks (GATs) [96] $G = (G^1, G^2, ..., G^H)$ is used. Each graph has the vertex set $E = (E_1, E_2, ..., E_{N\times T})$, including N player embeddings at all considered time frames T. Each vertex is adjacent to itself and all other vertices in E with a relation value R. This value is a function of an appearance and a position relation between two vertices, which are obtained using the embedded dot-product and a distance-mask respectively [99]. Each vertex can accumulate information from itself and others with a graph convolution as given in Equation 1. Right before the ReLU activation, layer normalisation [7] is applied over all $N\times T\times d$ embedding values. The authors do not report on this function in their paper, yet we found the layer normalisation in the code implementation. In the experiments, we show that layer normalisation is an important element for improved action and activity predictions in soccer.

$$\tilde{E}_i^{(h)} = \text{ReLU}\left(\text{LayerNorm}\left(\sum_{j=1}^{N\times T} R_{ij}^{(h)} E_j W^{(h)}\right)\right) \qquad (1)$$

with updated player embedding $\tilde{E}_i^{(h)} \in \mathbb{R}^d$ for vertex i in graph h, the relation value $R_{ij}^{(h)} \in [0,1]$ between vertices i and j, and a trainable weight set $W^{(h)} \in \mathbb{R}^{d\times d}$. The final player embedding E'_i is an element-wise addition of the H updated embeddings from each graph and the original embedding E_i.

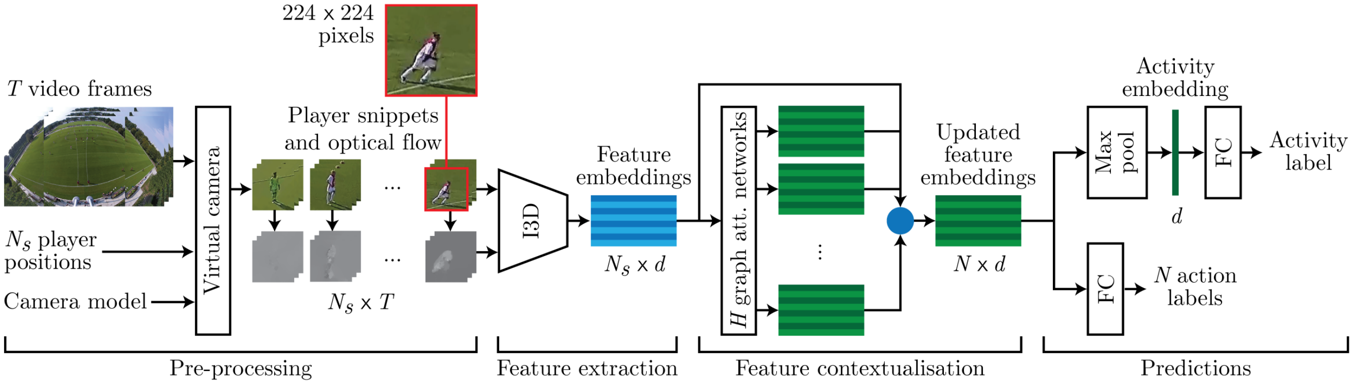


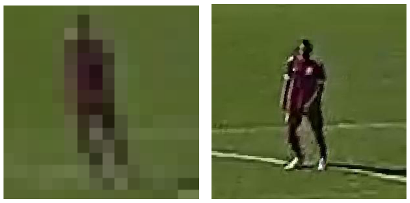
Figure 3: Architecture of the proposed method.

3.1.4 ***Predictions***. The refined player representations E' are grouped in a $N \times d$ feature matrix, for each time step in T. As previously mentioned, $T = 3$ while training and $T = 9$ during inference. The feature matrices are fused using average pooling. Thereafter, two output streams predict the action labels and activity label separately. Both classifications are performed through a fully-connected layer and using a *softmax* function. Cross entropy is used to calculate the action and activity prediction losses. Note that in the activity stream, a max pooling operation is applied to the feature matrix before the fully connected layer, to obtain one d-dimensional vector.

3.2 Proposed method

The data pipeline in the proposed method is displayed in Figure 3, in which the four phases can be found that we discussed previously. In short, the method pre-processes the data by generating player snippets using a virtual camera algorithm. Such an algorithm synthesises frames from a raw video stream where the camera virtually zooms and rotates, while normalising for lens distortion [68]. Features are extracted from the player snippets and optical flow images using an I3D CNN [10]. The resulting player embeddings are updated using feature contextualisation with GATs and self-attention [95]. The model outputs an action label per player and one shared activity label.

3.2.1 ***Pre-processing***. A soccer field is about forty times larger than a volleyball field, meaning that the distance between a player and the camera can become much larger, players become smaller and more pixels in the video capture irrelevant background. Therefore, the proposed method uses high resolution player representations as model input in the form of player-centric snippets. In Figure 4 can be seen that for a player standing in the centre circle the proposed method gains a lager number of pixels than when using full-field frames.



(a) **(b)**

Figure 4: Input pixels corresponding to a player standing on the centre circle in a cropped full-field frame (a) and in a player snippet (b).

To create player snippets, the position of each player must be known in field coordinates such that a virtual camera algorithm can zoom in on these positions. We denote these with (X, Y), where $(0, 0)$ is the centre spot of the field. The player coordinates are provided by the dataset and were obtained using an ACF person detector [25]. A camera model is essential, such that two-dimensional pixel coordinates can be transformed into three-dimensional world coordinates. The model is an approximation of the camera's internal (e.g. focal length, lens distortion) and external (e.g. translation, rotation) parameters, calibrated with field dimensions in world coordinates. With the camera model, a virtual camera zooms in on position (X_i, Y_i, Z) in a video frame. Then, the zoomed image is cut-out from the original frame and resized to 224×224 pixels, the standard resolution for I3D input, to create a snippet of player i. Generally, Dutch males have a body length around 1.8 meters. For the players to end up in the centre of the snippets, with a bit more space under the player to include the ball if present, we choose $Z = 0.8$m. An algorithm is used that virtually moves the camera with three degrees of freedom, increases the focal length to zoom in and normalises for a curved horizon. The pitch and yaw of the virtual camera are calculated using a player's position relative to the camera. The zoom parameter is a linear interpolation between two values recorded at the player closest and furthest from the camera, such that the pixel-height of all players is similar for each recording.

An input sample consists of $min(N_d, N) \times T$ player-centric images, where N_d is the number of detected persons, $N(= 23)$ is the number of persons that we strive to detect (22 players and one referee) and T is the number of temporal frames. we experiment with various values for T. It occurs that N_d is larger or smaller than N, either due to detection errors or game development. For example, a player could get a red card or get injured, leaving the field. Therefore, we sort the players on mean confidence of the person detector over all frames in a sample, in descending order. If the number of detected players N_d is larger than N, the first N players are selected to be processed. Players with a non-*passive* action label are put first in order, such that those players are always selected by the model (see Section 4 for an explanation of the class labels). For $N_d < N$, we simply create N_d player snippets. We denote $N_s = min(N_d, N)$.

3.2.2 ***Feature extraction.*** Since the players are already separated from each other in the pre-processing phase, RoI pooling is not used in our pipeline. The proposed method uses a two-stream I3D [10] for feature extraction, because it appears to give better results in group activity recognition than the Inception-V3 network [6][31] and the method gains state-of-the-art performance on the UCF-101 dataset [82]. We use an open source implementation[2] including model parameters pre-trained on ImageNet [21] and Kinetics [49]. I3D uses three-dimensional kernels to create spatio-temporal convolutions on RGB images and optical flow in separate streams. Both streams use the full I3D network and provide the d-dimensional player embeddings through d logits. We use $d = 256$ throughout this study. The outputs of the two streams are added element-wise, in a late fusion fashion. For the RGB stream, we input the player snippets as if they were a batch of N_s samples. Similarly, we input optical flow images that are obtained from the snippets using the TV-L1 algorithm [103]. The network returns one $N_s \times d$ feature matrix per sample, where Inception-V3 would return a $N_s \times d \times T$ matrix. This means that the temporal dimension is omitted from the feature contextualisation phase, reducing its complexity.

The I3D model is trained without feature contextualisation first. The model is trained in 20 epochs, with a batch size of 1, learning rate 1×10^{-5} (5×10^{-6} starting from epoch 15), dropout probability of 0.3, no weight decay and using an Adam optimiser [54]. For a fair comparison, the baseline is always trained with the same hyper-parameters when applied to soccer videos.

3.2.3 ***Feature contextualisation.*** State-of-the-art methods for group activity recognition have shown that attention is a useful mechanism for contextualisation [31][66][99]. We will follow this approach and use multi-head self-attention [95] in particular. Before the feature embeddings are put into context, we apply layer normalisation over each embedding independently and ReLU activation thereafter. Similar to the ARG, we construct H GATs where each graph contains N vertices representing the normalised feature embeddings. Per graph, features are exchanged between players depending on inter-player relationships using self-attention, as in Equation 2. Thereafter, the collection of features is non-linearly transformed via a graph convolution layer, as in Equation 3. The context features from all graphs are combined using Equation 4, similar to the combination function in the Transformer architecture [95]. As the ARG, the Transformer and our approach use self-attention at the core of the architecture, the methods display similarities. We specify when the original player embeddings E are transformed into *query*, *key* and *value* embeddings such that similarities can be identified more easily. Also, we found $H = 64$ to be optimal for our model.

$$E_A^{(h)} = \text{Softmax}\left(\frac{D\left(W_Q^{(h)}E + b_Q^{(h)}\right)\left(W_K^{(h)}E + b_K^{(h)}\right)^T}{\sqrt{d}}\right)E \quad (2)$$

with E the normalised player embeddings and $E_A^{(h)} \in \mathbb{R}^{N\times d}$ the collection of context features from graph h. Weight matrices $W_Q^{(h)}, W_K^{(h)} \in \mathbb{R}^{d\times d}$ and biases $b_Q^{(h)}, b_K^{(h)} \in \mathbb{R}^d$ linearly transform the player embeddings to *query* and *key* embeddings. Distance-mask $D \in [0,1]^{N\times N}$ prunes player-pairs from the graph when they are located too far from each other. $D_{i,j} = 0$ if the distance between player i and j is larger than μ, and $D_{i,j} = 1$ otherwise.

$$\tilde{E}^{(h)} = \text{ReLU}\left(\text{LayerNorm}\left(E_A^{(h)} W_V^{(h)}\right)\right) \quad (3)$$

with $\tilde{E}^{(h)}$ the updated context features from graph h and weight matrix $W_V^{(h)}$ that transforms the collection of features to *value* embeddings.

$$E' = E + \text{Concat}\left(\tilde{E}^{(1)}, \tilde{E}^{(2)}, \ldots, \tilde{E}^{(H)}\right)W_O \quad (4)$$

with E' the final feature embeddings before label prediction, a residual connection to the original embeddings E and weight matrix $W_O \in \mathbb{R}^{H\times d\times d}$. The embeddings from H graphs are concatenated and linearly transformed through W_O.

[2] https://github.com/piergiaj/pytorch-i3d

Where it was possible to process N_s players per sample in the previous phases, feature contextualisation requires precisely N feature embeddings when using a batch size (BS) larger than one. This is required, as the model processes feature matrices with dimensionality BS $\times N \times d$. Therefore, zero-padding is used to fill in for the $N - N_s$ missing players. The missing embeddings are d-dimensional vectors with only zeros, such that these are ignored by the self-attention mechanism and no features are exchanged with embeddings of other players. They are used by the baseline ARG, when applied to samples from the Collective Activity Dataset, and by the Transformer architecture [95] as well. Although we use BS=1 while training, the model benefits from larger batch sizes during inference, speeding up the process.

The GATs are trained separately from the feature extraction phase. When the models for feature contextualisation are trained, all layers in the I3D model are frozen, resulting in static player embeddings. Hyperparameters during training of feature contextualisation are unchanged, except for the number of iterations (40 instead of 20 epochs).

3.2.4 ***Predictions****.* The action and activity predictions are created from the updated feature embeddings using the approach of the ARG as described above.

4 THE SOCCER DATASET

A new dataset is constructed for this research that contains actions and activities from 280 minutes of soccer in four games. Videos in the dataset are captured by a one-perspective camera setup that records the full field. To be able to compare our results with work from others, and to increase applicability of the ARG as a baseline, we decided to make the new dataset similar to the Volleyball Dataset. Both datasets include video recordings from multiple games that are exclusively captured by cameras positioned at the long side of the field. However, where the videos in the Volleyball Dataset are recorded by a moving camera, the soccer videos are stationary. Four cameras that are located side-by-side, together capture the full soccer field. Each camera records vertically, at 25fps and with a 2160 × 3840 resolution. The output of the four cameras are combined in one video stream, as can be seen in Figure 5.

Similar to the Volleyball Dataset, samples are selected by human annotation. The time t_0 of a relevant action or group activity occurrence is noted and the associated frame f_t is recorded. To capture the temporal dimension, 25 subsequent frames before and after t_0 are stored. This means that every sample includes 51 frames, from f_{t-25} till f_{t+25}.

4.1 Player detections and tracks

It is common for group activity datasets to accompany the raw videos with person bounding boxes. Our dataset offers the locations of automatically detected players via bounding boxes and field coordinates. Players on the field are detected by an ACF detector [25] that outputs bounding boxes. The detection method is based on the aggregation of ten channels (normalised gradient magnitude, histogram of oriented gradients and LUV colours) applied to the original image, the computation of feature pyramids and AdaBoost [28]. We favoured the computationally cheap method over deep learning approaches, as the search space for persons is 2160 × 3840 pixels. In an initial experiment, we concluded that these detections are accurate enough to be used for action recognition (see Appendix II). Since the camera model is known and calibrated with the soccer field dimensions, pixel coordinates (x,y) can be converted to real-world coordinates (X,Y,Z). Note that Z is a pre-defined value as the camera has no access to depth information. To obtain the field coordinates that our method requires, the bottom-centre pixel in each player's bounding box is projected on the virtual plane with $Z = 0.0$m that corresponds to the soccer field.

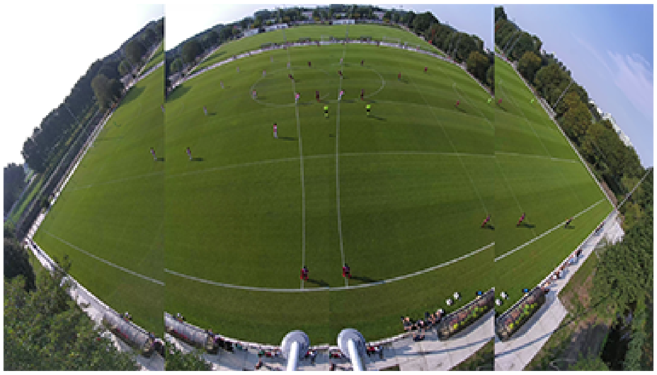


Figure 5: Example frame of the raw video stream.

Using tracking software, detections of the same player in multiple frames are linked in trajectories throughout the video. For each player in every sample, 51 bounding boxes and 51 field coordinates are stored. However, not all players have detections over the full 51 frames. Therefore, the 51 frame trajectories are linearly interpolated first and linearly extrapolated afterwards using the field coordinates of the players. To increase accuracy, the coordinates are smoothed before this step. Missing bounding boxes of a player get the average dimensions of the found bounding boxes and are placed above the inter- or extrapolated field coordinates.

In Section 3.2, we shortly addressed the presence of a curved horizon in the raw frames of our dataset. Due to the angle with which the most left and most right cameras are fixed and distortion in the middle cameras, the horizon is rotated for most field positions. This results in rotated players while the detections are placed upright and thus not capturing all body parts of the corresponding players. By projection of the world coordinates $(X, Y, 0)$ and $(X, Y, 1)$ to pixel values (x_0, y_0) and (x_1, y_1), the rotation ρ of the horizon can be calculated

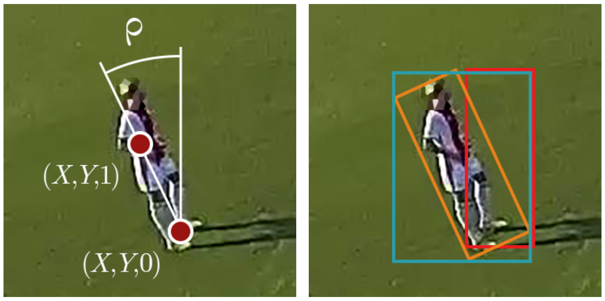


Figure 6: The rotation of the horizon is calculated first (left). The original bounding box (red) is rotated by ρ radians (orange) after which a new bounding box is drawn (blue) (right).

at each player position (X, Y) on the field (see Equation 5). A new bounding box is created by rotating the original bounding box by ρ radians and drawing a new rectangle around it. The transformation is illustrated in Figure 6. The renewed boxes form the input for the baseline model, where the field coordinates are used by the proposed method to generate player-centric snippets.

$$\rho = \text{atan2}((x_1, y_1) - (x_0, y_0)) \tag{5}$$

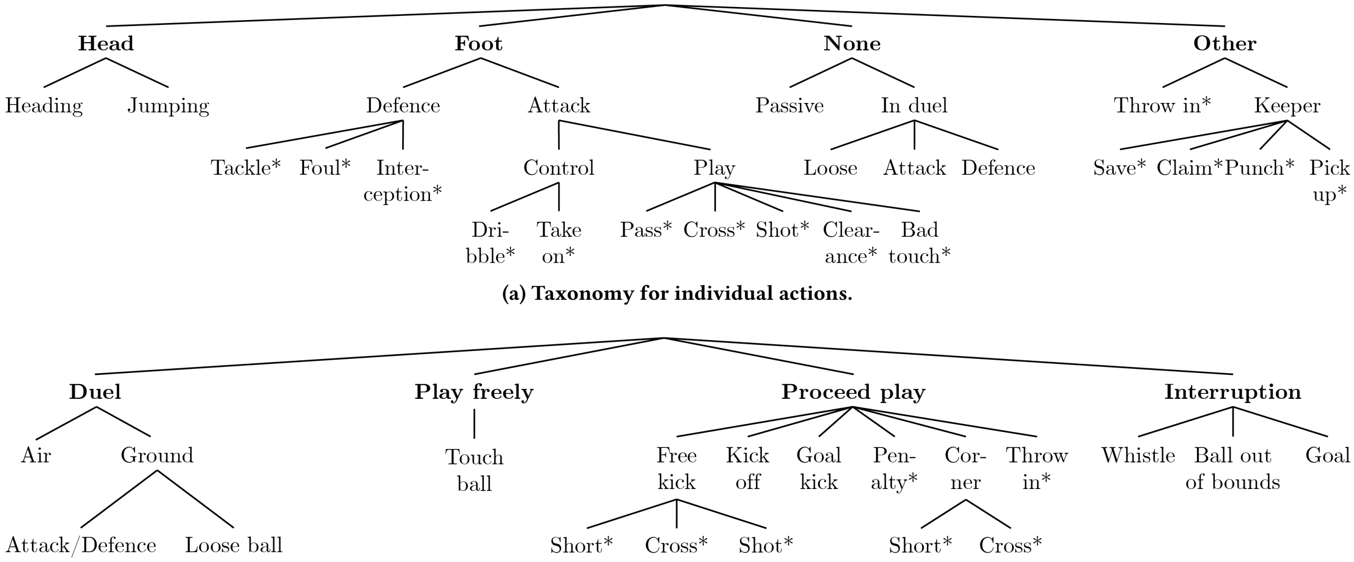
4.2 Action and activity labels

The data labels are placed in two taxonomies, with one for actions and another for activities (see Figure 7). The samples are annotated with labels from the leaves of the taxonomies. The tree-structure of the taxonomy guides the annotator to provide the right labels, resulting in exclusive labels and less ambiguous annotations. For example, to label a player's action, the annotator first observes with which body part the player reaches out to the ball. If the player reaches to the ball with its foot, it is observed whether the player's team was in possession of the ball (attack) or not (defence), and so on.

The set of action labels includes all action types in the Soccer Player Action Description Language (SPADL) [20], except for six actions that describe activities (like *Short corner*, *Penalty shot*). SPADL is constructed to unify player action types in match event logs from multiple CIPs into one framework. Labels are divided in groups, corresponding to the body part with which the ball is touched to perform an action. Six actions are added to the taxonomy to label players that are heading or jumping towards the ball, or that are not reaching out to the ball. Players in the latter category are part of a duel, or are labelled *passive* otherwise. The activity taxonomy includes the six left-out SPADL action types and is supplemented with activities from the list of match events provided by Wyscout (see Appendix IV). The definition of each label can be found in Tables 32 and 33 (Appendix V).

4.3 Annotation procedure

A Matlab tool is created to enable the author to annotate the soccer games efficiently. With this tool, the annotator could watch a game and pause it at every moment. To prevent the annotator from missing relevant events at the back of the field, a *virtual director* video was created per game. These videos were created by software that automatically operates a virtual



(a) Taxonomy for individual actions.

(b) Taxonomy for group activities.

Figure 7: The action and activity labels are represented in two separate taxonomies. All labels marked with an asterisk (*) can be found in the SPADL.

camera as if it were the camera operator at a broadcasted game. When an action or activity is observed, the annotator could go back and forth in the video frames to find the exact frame of the occurring event. When pressed a button, a second screen opened in which all player detections were shown, and the annotator was asked to input one group activity label and one action label per detection. By default, a player received the *passive* action label if he was positioned within the field boundaries. Otherwise, the player was labelled as an *incorrect* detection. The annotator could zoom in on the camera view, could create new bounding boxes for undetected players and was able to adapt the action and activity labels while the second screen was open. The tool was build on top of existing video viewer software, i.a. used for visual inspection of person detections in surveillance and sport videos.

4.4 The final label set

To obtain the final set of labels used in this study, the annotations were adapted in two ways. First, all players that have an *active* (non-*passive*) action label, in frames where the group activity label is a *duel*, get the action label *in-duel*. This means that whenever a player performs an action during a duel, the player is labelled with being in a duel rather than with the specific action. When multiple players are in a duel, it is likely that their bounding boxes overlap, e.g. as in Figure 8, which results in two almost identical data representations. Labelling the samples with different actions, e.g. *tackle* and *pass*, is likely to destabilise training. Therefore, we provide both samples with the same action label. As the labels *tackle*, *foul* and *take-on* can only occur in a duel, these are removed from the set of classes. It also means that we cannot distinguish between *in-duel-loose*, *in-duel-attack* and *in-duel-defence*. All these players get the action label *in-duel*.

Second, a number of action and activity labels occur only a few times, which is likely to result in overfitting models. We have re-arranged the set of labels by grouping action labels that occur less than a hundred times. We grouped *heading* and *jumping* under *heading*, grouped *pass*, *cross*, *shot*, *clearance* and *bad-touch* under *play-ball*, and grouped *save*, *claim*, *punch* and *pick-up* under *keeper*. The result is eight action classes. Note that grouped classes *heading* and *interception* contain less but close to a hundred samples each. Although the set only has 46 *keeper* samples, the class is not grouped further because the large semantic gap with throw ins is expected to be more problematic. Activities that occurred less than ten times are grouped as well. Here, we grouped *free-kick-short*, *free-kick-cross* and *free-kick-shot* under *free-kick*, grouped *corner-short* and *corner-cross* under *corner*, and grouped *ball out-of-bounds* and *goal* under *ball out-of-bounds*. The activity *penalty* is removed from the set as it occurred only three times and in one game. The final dataset includes eleven activity labels.

The resulting class labels and the number of instances are given in Table 2. Samples annotated from three games are included in the training set. The validation set consists of samples from the second half of the third game, which are excluded from the training set. The test set is formed by annotations from the first half of a fourth game. This game is played on a different soccer field and with another opponent team than the other three games.

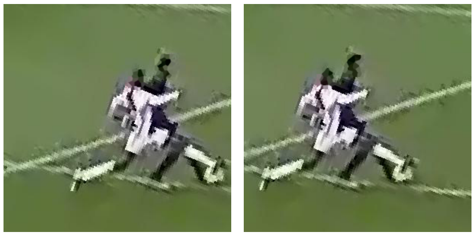


Figure 8: Player snippets for two players in a duel, one tackling (left) and another passing the ball (right). Problematic is that they are almost identical while the players have different action labels. Therefore, both are labelled as *in-duel*.

Action class	# Instances			Activity class	# Instances		
	Train	Validation	Test		Train	Validation	Test
Passive	59459	8710	11140	Duel (air)	59	10	11
Heading	70	6	15	Duel (loose ball)	221	26	32
Interception	67	11	17	Duel (ball possession)	468	53	85
Dribble	303	40	44	Play-freely	1541	231	255
Play-ball	1262	202	221	Free-kick	59	7	13
In-duel	1575	189	270	Kick-off	13	2	2
Throw-in	121	15	35	Goal-kick	41	9	9
Keeper	34	4	8	Corner	22	9	10
				Throw-in	121	15	35
				Whistle	74	9	12
				Ball out-of-bounds	182	32	49
Total	62891	9177	11750	**Total**	2801	403	513

Table 2: Action and activity label sets in the Soccer Dataset with number of instances in training, validation and test set.

5 EXPERIMENTS AND RESULTS

We start this section with an explanation of the metrics that were used to evaluate the baseline, intermediate optimisation steps and the proposed method. Thereafter, we show that the baseline ARG gives drastically worse predictions for soccer videos as opposed to volleyball. In the remaining experiments, we explain and evaluate the benefit of each optimisation step in the proposed method.

5.1 Evaluation metrics

When comparing the performance of the proposed method with those from others, Multi-Class Accuracy (MCA) is used, as this metric is generally reported in related work. However, as the Soccer Dataset is highly unbalanced, this metric only would give a too optimistic view. Therefore, we calculate a Matthew Correlation Coefficient (MCC) [69] per class label, using Equation 6, and use this metric to evaluate all intermediate optimisation steps. An MCC is independent of class imbalance and weights precision and recall rather than accuracy only [14][15]. The correlation value indicates the agreement between the predicted and the ground-truth labels. MCCs range from -1 (total disagreement) to 1 (perfect agreement), where a score of 0 equals random predictions. Our goal is to maximise the MCC for each class label, where we value the predictability for all classes equally important. To avoid reporting nineteen MCCs at every optimisation step, we average the scores over all actions (eight classes) and all activities (eleven classes). The result is two Mean Matthews Correlation Coefficient (MMCC) scores. Importantly, the averages are meta-scores meant to evaluate optimisations and are difficult to interpret. Therefore, we frequently report the individual MCCs as well.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad (6)$$

with the numbers of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) for each class calculated from the confusion matrix.

Where standard deviations (±σ) are mentioned, the model is trained and evaluated five times using different parameter initialisation. Standard deviations reported with † or ‡ are from three and ten runs, respectively. All results until Section 5.6 are on samples from the validation set.

5.2 Baseline experiments

As the baseline ARG has not been evaluated on soccer data before, it is no foregone conclusion that the method performs equally well with soccer videos as it does in volleyball. To measure the performance gap, the ARG is applied, in its original structure, to the Volleyball Dataset and the Soccer Dataset. When reproducing the results on the Volleyball Dataset, we were able to obtain similar performance scores as reported in the original paper [99]: 82.5% and 92.3% MCA for actions and activities respectively (see Table 3). When the baseline is applied to the Soccer Dataset, MCAs of 89.3% (actions) and 55.9% (activities) are obtained. The high score for the actions can be explained by the large imbalance in the dataset, where 94.9% of the player samples have the *passive* action label. The baseline results in MMCCs of 0.275 and 0.359 for action and activity predictions with soccer videos, which is drastically lower than its performance with volleyball videos.

Dataset	**Actions** (±σ)		**Activities** (±σ)	
	MCA	MMCC	MCA	MMCC
Volleyball [99]	83.0%	-	92.5%	-
Volleyball (repr.)	82.5%	0.641	92.3%	0.915
Soccer	89.3% (±3.3)	0.275 (±.030)	55.9% (±4.0)	0.359 (±.054)

Table 3: Performance of the baseline ARG, using Inception-V3 as backbone, on the Volleyball Dataset (as reported in the paper and reproduced) and on the Soccer Dataset.

The number of parameters of the baseline ARG is rather large (63.5M weights and biases), which is expected to result in unnecessary long training times. A way to reduce the the number of parameters is with lowering the dimensionality d of the player embeddings. We propose to lower d from the original size of 1024 to 256 dimensions, reducing the number of parameters with 69.4% (to 19.4M). In Table 4 can be seen that this decreases the average training time with 1.25 hours at the cost of a small performance reduction. To maximise the number of models we can train, we use $d = 256$ for further experiments. Also, we show that the proposed method looses performance when using the larger dimensionality in Section 5.5.7.

d	**Time** (±σ)	**Actions** (±σ)	**Activities** (±σ)
1024	7.23h (±0.33)	0.275 (±.030)	0.359 (±.054)
256	5.98h (±0.74)	0.197 (±.011)	0.278 (±.036)

Table 4: Performance differences (in MMCCs) and training times (on a GeForce GTX 1080 Ti GPU) when decreasing dimensionality d of player embeddings in the baseline ARG.

5.3 Data pre-processing

In soccer, players can be positioned far from the camera due to the large playing field. The baseline ARG uses full-field frames as model input, sub-sampling the high resolution images to a standard format of 720×1280 pixels. This causes soccer players that are farthest from the camera to be represented with very few pixels. In Figure 9 (a) it can be seen that for a soccer

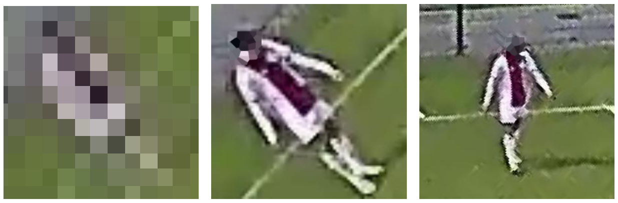

(a) Sub-sampled (b) High resolution (c) High res. + norm.

Figure 9: Sample of a player located at the left goal line.

player located at the goal line, its body pose is difficult to recognise. Therefore, we propose to use player-centric snippets that provide player representations in high resolution, Figure 9 (b), and with a normalised horizon, Figure 9 (c).

5.3.1 ***Predicting passive-active action classes.*** First, we explore the ability of an Inception-V3 network to distinguish players with an active label from the passive players, using full-field frames and the proposed player snippets. This is done in a binary setup, with only two action labels: *active* (including all actions but *passive*) and *passive*. It is expected that high-resolution player snippets are more informative than the player representations in the full-field frames. MCCs and confusions can be seen in Table 5. With player snippets as input, the model can better distinguish between active and passive players, with MCCs of 0.382 for full-field frames and 0.816 for player snippets. In Figure 10 it can be seen that a trade-off exists between precision and recall. The trade-off is less problematic when using player snippets, as the AUC is larger (99%) than the AUC when using full-field frames (88%). Also, for each precision value, player snippets provides a better recall than full-field frames, while for each recall it gives a better precision. Using player-centric snippets, an Inception-V3 can already recognise the players that are interacting with the ball confidently.

Input	TP	TN	FP	FN	MCC
Full-field	315	8064	646	244	**0.382**
Player snippets	529	8509	201	30	**0.816**

Table 5: Active-passive action recognition using Inception-V3 with full-field frames and player snippets as input. For the calculation of TP, TN, FP and FN the active class is considered the positive class.

5.3.2 ***Predicting all action and activity classes.*** Since the player snippets are good representations for active-passive recognition, it is expected that their use also increases model performance in the recognition of all actions and activities. However, it is not yet clear how much the snippets' increased resolution and the normalised horizon contribute to the improved predictions independently. To evaluate the effect of both steps, an Inception-V3 model is trained with the original input (full-field frames), with player snippets excluding horizon normalisation and with player snippets including the normalisation. In Table 6 it can be seen that using player snippets increases the MMCC in action recognition from 0.163 to 0.264 and in activity recognition from 0.364 to 0.437. Normalisation further increases these scores to 0.358 (actions) and 0.616 (activities). This means that both steps are essential for our model to generate good action and activity predictions.

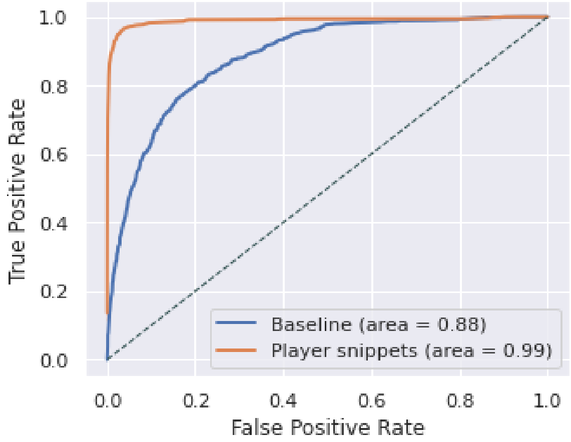


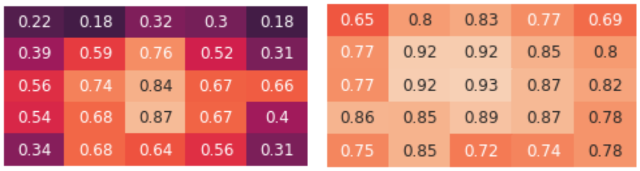
Figure 10: Active-passive action recognition using Inception-V3 with full-field frames (baseline) and player snippets as input.

Input	Norm.	Actions (±σ)	Activities (±σ)
Full-field	✕	0.163 (±.007)	0.364 (±.040)
Snippets	✕	0.264 (±.024)	0.437 (±.073)
Snippets	✓	0.358 (±.023)	0.616 (±.028)

Table 6: MMCC scores of an Inception-V3 using the full-field images (baseline) and player-centric snippets, with and without normalisation.

5.3.3 ***Predictions and camera distance.*** The previous experiments have shown that using high resolution normalised snippets increases the model's ability to recognise player actions and group activities more accurately. During the creation of the snippets, all players are scaled to a standard pixel-height. Since players that are farthest from the camera are represented with the fewest pixels in the full-field frames, the predictions of their actions should improve most. To test this hypothesis, action predictions from the Inception-V3 network, using player snippets and full-field frames, are separated into 25 field areas. We do not evaluate the activity predictions, as they are the result of multiple players that are located at several field areas. Also, not all actions occur at all areas, which makes it impossible to obtain an MCC for every action at every area. We could take the average score of the actions that occur in an area, but that would make a comparison between areas problematic. Therefore, MCA scores are reported per field region.

The results can be seen in Figure 11, where the top regions are farthest away from the camera. Although the baseline model obtains reasonable results for action recognition of players that are located in the centre row (mean accuracy of 69.4%), performance decreases quickly in other areas, especially the top row (mean accuracy of 24.0%). When using player snippets, action recognition performance is more uniform over the whole field, with mean accuracies of 86.2% (centre) and 74.8% (top). A similar effect can be observed for the most left and right field columns. The accuracy of action predictions with full-field frames depends largely on field location, with mean accuracies of 41.0% (left), 68.6% (centre) and 37.2% (right). These scores are more uniform when using player snippets: 76.0% (left), 85.8% (centre) and 77.4% (right).



(a) Full-field frames

(b) Player snippets

Figure 11: Accuracies of action predictions at different parts of the soccer field. The top regions are farthest away from the camera.

In this subsection, we have shown that the use of high resolution player snippets, with horizon normalisation, result in better action and activity recognition when using the Inception-V3 backbone. Therefore, these player snippets are used as model input in all further experiments (also for the baseline).

5.4 Per-player feature extraction

During the previous experiments, Inception-V3 was used to compress player representations into player embeddings. In this set of experiments, we first evaluate the benefit of using the I3D network for player-level feature extraction over Inception-V3, as is used by the baseline. Thereafter, it is explored whether it is beneficial to add the field location of a player to its embedding.

Backbone	BS	Actions (±σ)	Activities (±σ)
Inception-V3	4	0.358 (±.023)	0.616 (±.028)
Inception-V3	1	0.500 (±.016)	0.615 (±.045)
I3D RGB	1	0.555	0.576
I3D Flow	1	0.513	0.535
I3D RGB + Flow	1	0.646	0.619

Table 7: MMCC scores for action and activity recognition using player snippets sampled from temporal window $[t_{-0.16s}, t_{0.16s}]$ **(0.32 seconds).**

5.4.1 *I3D player embeddings.*

To evaluate the advantages and disadvantages of using I3D over Inception-V3, both backbones are trained for action and activity recognition simultaneously. Since the effectiveness of the feature extraction methods depends on the samples' temporal boundaries, multiple time windows are evaluated from which input frames are sampled. The window varies from a single frame $[t_0]$ (with t_0 the moment of annotation) to $[t_{-0.48s}, t_{0.48s}]$ (0.96 seconds). Where Inception-V3 can make predictions from one static frame, I3D requires a temporal dimensionality of at least nine subsequent frames due to spatio-temporal pooling. For time windows smaller than $[t_{-0.32s}, t_{0.32s}]$, the frames are sampled at 25fps. When using a larger time window, frames are sampled at 12.5fps because of GPU memory constraints when training the I3D models. From $[t_0]$, the window is enlarged with 0.08 seconds per step, at both ends of the annotated moment. One exception has been made: the window $[t_{-0.32s}, t_{0.32}]$ (0.64 seconds) has been replaced with $[t_{-0.32s}, t_{0.28s}]$ (0.60 seconds), such that frames could still be sampled at 25fps. The memory limit also forced the batch size (BS) to be one at most. Therefore, we trained both Inception-V3 and I3D with a BS of one instead of four. This already increased the MMCC scores from 0.358 to 0.519 for actions and from 0.616 to 0.669 for activities (see Table 7).

The I3D network is usually trained with one stream processing RGB images and another stream processing optical flow images. To assess their effectiveness independently, both streams are trained and evaluated separately at first. Thereafter, a two-stream network is trained, processing RGB and optical flow images simultaneously. Due to time restrictions, the two-stream network is trained on two temporal windows only. The model needs seven days of training on a GeForce GTX 1080 Ti GPU.

The results for all temporal windows can be seen in Figure 12. Inception-V3 is able to recognise *throw-in*, *passive* and *play-ball* in very strong agreement with the ground-truth

Backbone	BS	Actions (±σ)	Activities (±σ)
Inception-V3	1	0.425 (±.043)	0.661 (±.051)
I3D RGB	1	0.583	0.607
I3D Flow	1	0.545	0.598
I3D RGB + Flow	1	0.658 (±.033†)	0.641 (±.031†)

Table 8: MMCC scores for action and activity recognition using player snippets sampled from temporal window $[t_{-0.24s}, t_{0.24s}]$ **(0.48 seconds). (†) Standard deviation is from three runs instead of five.**

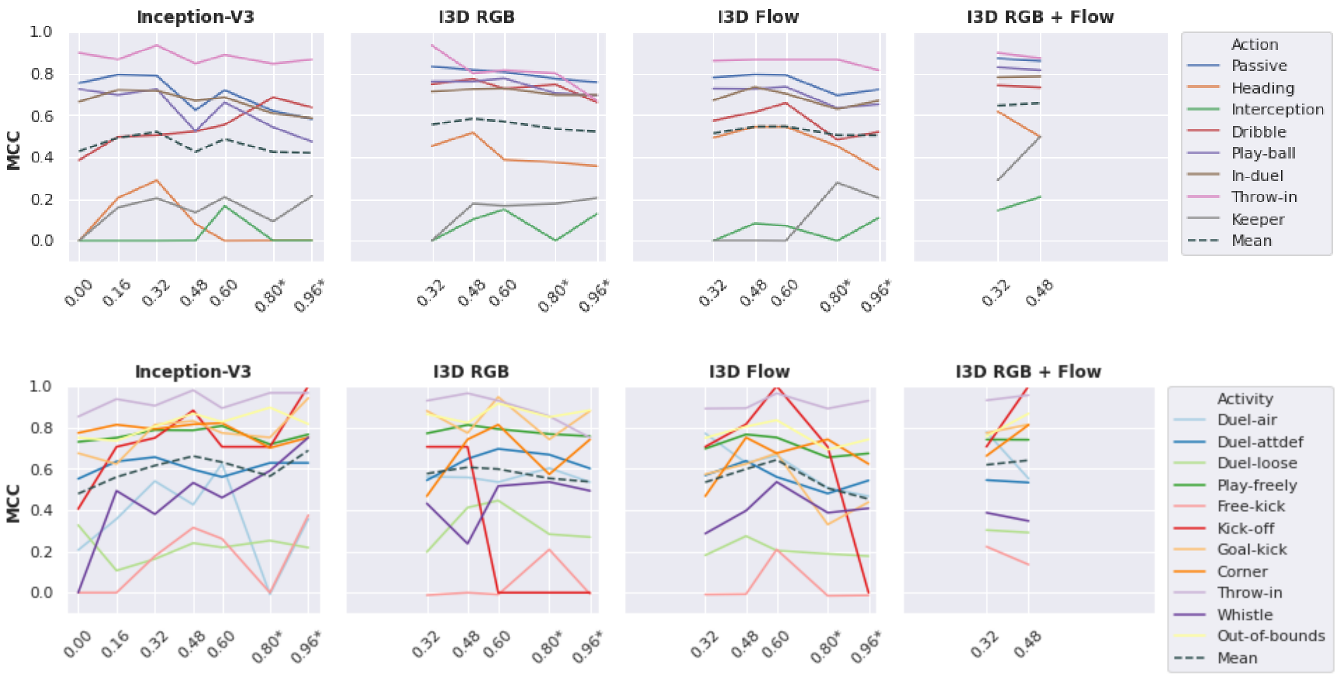


Figure 12: MCCs for action (top) and activity (bottom) recognition, using Inception-V3 and I3D as backbone with different temporal windows (in seconds). Snippets are sampled from these windows at 25 or 12.5 (*) frames per second.

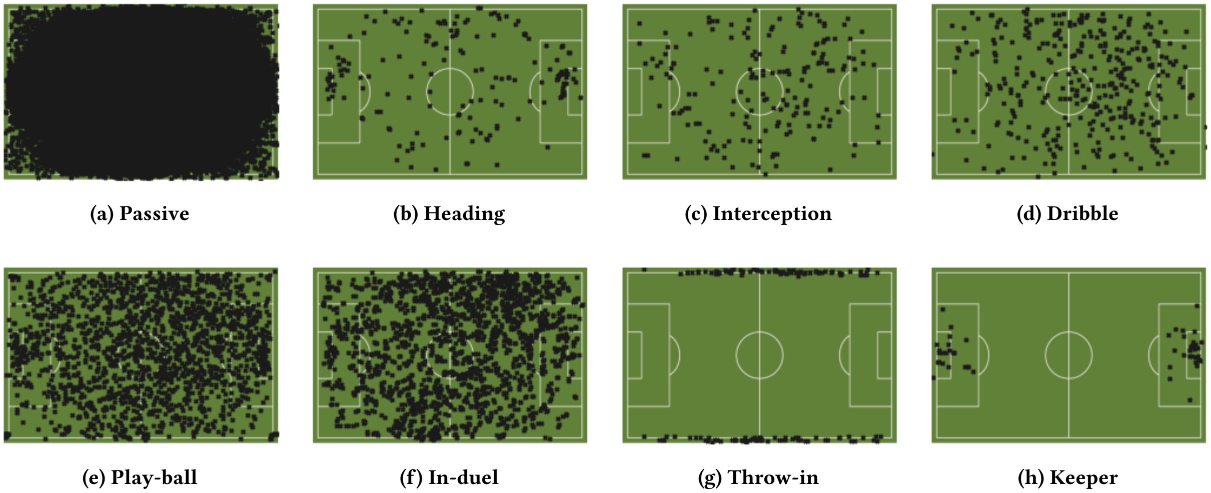
labels (MCC ≥ 0.7) from temporally static samples (0.00 seconds). Using samples from a 0.32 seconds window, Inception-V3 predicts *in-duel* in very strong agreement as well. At the same temporal window, I3D also recognises a *dribble* with an MCC above 0.7 and improves on the recognition of a *heading*. The MCCs for activity recognition are often unstable, most likely due to small sample sizes. For Inception-V3 and I3D, a *throw-in*, *corner*, *play-freely*, *ball out-of-bounds*, *goal-kick* and *kick-off* can generally be predicted with MCCs over 0.7. Interestingly, Inception-V3 shows a trend in which samples from larger temporal windows result in better predictions.

The optimal temporal window for action recognition seems to lay around 0.32 (Table 7) or 0.48 seconds (Table 8), where the optimal size is less clear for activity recognition. I3D (RGB + Flow) performs best in action recognition, with an MMCC of 0.658 at 0.48 seconds. Inception-V3 at 0.96 seconds gives the highest MMCC for activity recognition: 0.688. It would be possible to use two separate networks for action recognition (I3D) and activity recognition (Inception-V3). However, we argue that splitting the two does not trivially result in similar individual performance scores, since the two classification purposes might benefit from the multi-task learning approach. As the I3D (RGB + Flow) at 0.48 seconds obtains the highest cumulative MMCC, we continue to use this backbone in further experiments (also for the baseline).

5.4.2 *Position encodings*.

In soccer, some actions or activities require a player to be located at a particular area of the field, meaning that correlations exist between the absolute field position of the players and particular class labels. In Figure 13, the field coordinates of all players are plotted, grouped per action label. It can be seen that the actions *throw-in* and *keeper* only occur at particular areas of the field. Also, when splitting action label *play-ball* into its five sub-labels (Figure 14) it can be seen that *shot*, *cross* and *clearance* correlate with field position as well.

Therefore, it is evaluated whether enriching the player embeddings with position information results in better action and activity predictions. Note that the addition of absolute position information was not used in the baseline model. Three methods are evaluated to transform static positions of all players $P \in [0,1]^{N\times2}$ into position encodings $P_{enc} \in \mathbb{R}^{N\times d}$. These are: by linear transformation (PW_0), nonlinearly ($\mathrm{ReLU}(PW_1 + b_1)W_2 + b_2$) and using the sinusoidal encoding often used with self-attention [95]. For completeness, $W_0, W_1 \in \mathbb{R}^{2\times d}$ and $W_2 \in \mathbb{R}^{d\times d}$. Besides encoding a static field position, we hypothesise that it would be even more valuable to encode player trajectories. Apart from field location, trajectories include information about moving direction, speed and acceleration. Therefore, we evaluate the use of 1D-convolutional layers [38] to transform trajectories $T \in [0,1]^{N\times2\times51}$ into encodings $T_{enc} \in \mathbb{R}^{N\times d}$. Full trajectories


(a) Passive

(b) Heading

(c) Interception

(d) Dribble

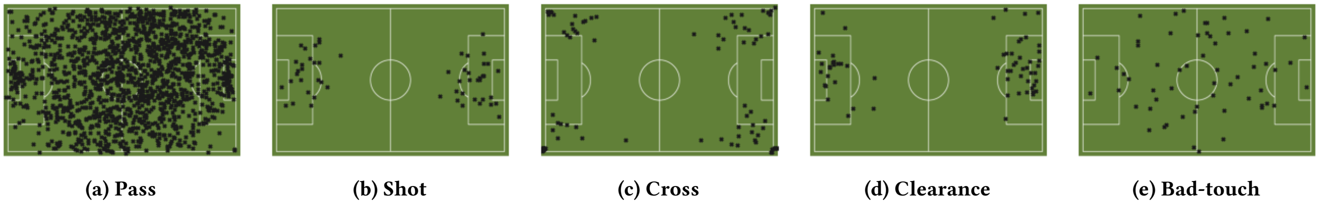
(e) Play-ball

(f) In-duel

(g) Throw-in

(h) Keeper

Figure 13: Field coordinates of action occurrences in the training and validation set. Actions clearly correlating to the field area of their occurrence are: *throw-in* and *keeper*.


(a) Pass

(b) Shot

(c) Cross

(d) Clearance

(e) Bad-touch

Figure 14: Field coordinates of sub-action occurrences from sub-categories of the *play-ball* action label. From these, *shot, cross* and *clearance* clearly correlate to the field area of their occurrence.

of 51 frames (2 seconds) are used to make the network robust to small errors in the position data. All encodings have a dimensionality equal to the player embeddings and the two are added element-wise. Thereafter, the enriched embeddings go through a fully connected layer with dropout ($p = 0.3$) and are classified into actions and activities as before.

The results, in Table 9, show that the addition of an absolute position encoding does not result in better action or activity predictions. The sinusoidal and convolutional encoding methods obtain better scores than the linear and non-linear transformations. The 1D-convolutional layers provide the best predictions, with MMCCs of 0.618 and 0.587 for actions and activities respectively. This is not enough to improve upon previous results, without a position encoding. In further experiments, we do not use an absolute position encoding.

Encoding method	Actions ($\pm\sigma$)	Activities ($\pm\sigma$)
None	0.658 (±.033†)	0.641 (±.031†)
Linear	0.435 (±.014)	0.356 (±.048)
Non-linear	0.539 (±.036)	0.486 (±.024)
Sinusoids	0.600 (±.028)	0.575 (±.045)
1D-conv. layers	0.618 (±.012)	0.587 (±.042)

Table 9: MMCCs for action and activity recognition with absolute position encodings added to the player embeddings.

5.5 Feature contextualisation

The action classifications made in the previous experiments are based on player embeddings independently. However, a player's action often depends on the actions of other players, e.g. if one player starts accelerating with the ball, other players are likely to start running as well. The previously used models are limited in that they cannot discover inter-player relations and adapt action and activity predictions accordingly. It is expected that when player embeddings are put into the context of other players, the model obtains better results in action and activity recognition.

In five experiments we explore how feature contextualisation can improve the model's action and activity predictions. First, the baseline ARG is evaluated with player embeddings from the I3D network. We experiment with different numbers of graphs and convolutional layers and select the best performing baseline model. Second, we gain an insight in the semantic value of features that are exchanged between players with self-attention. We show that some situations require features to be accumulated where others require features to be suppressed. We argue that the baseline ARG prevents feature suppression and is therefore limited in optimising the player embeddings. Third, we strengthen the previous argument by experimenting with two padding strategies when not all twenty-three players are detected. Here, we show that the proposed method for feature contextualisation improves upon the baseline ARG. Fourth, we show that the role of relative position information and the exchange of features via self-attention is small, while layer normalisation is more important. Finally, we compare the proposed method with the baseline ARG and the AT.

5.5.1 ***Baseline ARG.*** The original ARG architecture has sixteen graphs and one graph convolutional layer. To obtain the best performing baseline model in soccer videos, we experimented with different numbers of layers and graphs. In Table 10 can be seen that the baseline ARG does not improve upon the original I3D predictions when using one or two layers and graphs. Nevertheless, the results indicate a slight performance increase when using multiple graphs in one layer. Performance of the baseline ARG, with the number of graphs ranging from one to 128, can be observed in Figure 15. Especially the action predictions benefit from multiple graphs, while the optimal amount lays around 32 graphs. The model with 64 graphs is selected as best baseline model, as it gains the highest cumulative MMCC. Note that, with MMCCs of 0.651 (actions) and 0.628 (activities), the model still performs worse than I3D without feature contextualisation. Further experiments with the ARG are executed with 64 graphs and one convolutional layer. During the remainder of this section, the baseline ARG refers to the optimal model found here, using player snippets, I3D feature extraction and 64 graphs.

Model	#G	#L	Actions (±σ)	Activities (±σ)
I3D	-	-	0.658 (±.033†)	0.641 (±.031†)
ARG	1	1	0.584 (±.017)	0.586 (±.051)
ARG	1	2	0.570 (±.021)	0.581 (±.072)
ARG	2	1	0.595 (±.018)	0.608 (±.024)
ARG	2	2	0.610 (±.018)	0.599 (±.036)
ARG	64	1	0.651 (±.008)	0.628 (±.039)

Table 10: Processing I3D feature embeddings with the ARG. Different numbers of graphs (#G) and graph convolutional layers (#L) are evaluated.

5.5.2 ***Feature accumulation and suppression.*** A goal of feature contextualisation is the enforcement of implicit biases that can be found in soccer games, improving the model's predictions. One of the most obvious biases in soccer is that only one ball is present at all times. This means that if someone is interacting with the ball while not being involved in a duel, only that person can obtain an *active* (non-*passive*) action label. All other players are very likely to be *passive*. With the next experiment, we show that the baseline's approach to feature contextualisation is not able to effectively enforce this implicit bias and that the model can do so when enabling *feature suppression*.

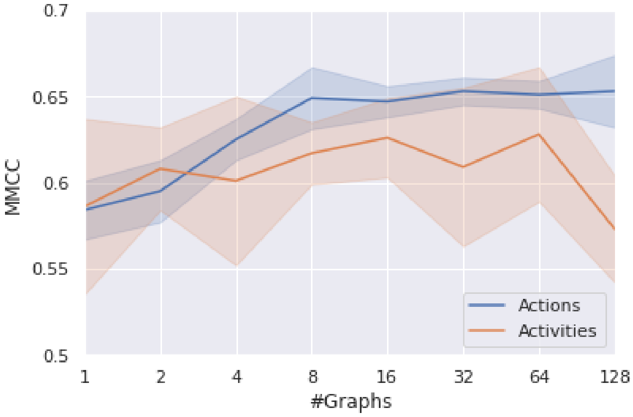


Figure 15: Performance of the baseline ARG with different numbers of graphs. The transparent area relates to the standard deviation (±σ).

A model that is not able to exploit the discussed bias is expected to predict a large portion of *passive* players as *active*, even though another player in the scene is also predicted to be *active*. We call *passive* players that are wrongly classified *false positives* (FPs) and we consider only on those FPs that are involved in activities where only one player can interact with the ball (e.g. *play-freely*, *throw-in*, *goal-kick*). While I3D causes 71 FPs without feature contextualisation, the baseline ARG does not improve on this, with 72 FPs on average. In scenes where another player is recognised as *active* as well, I3D gives 50 FPs where the baseline ARG gives 53 FPs. It shows that the baseline is not better at enforcing the implicit bias than I3D, which makes the action predictions player independently.

We hypothesised that the baseline ARG is not able to enforce the bias due to the ReLU activation function in the update function of the graph convolutional layer (see Equation 7). As ReLU activation clips negative values to zero, only positive values can be collected. This means that when a player embedding includes large positive values correlating to an *active* label, the baseline cannot reduce these values. Values in the player embeddings can only remain stable or become higher. We refer to this effect as *feature accumulation*. Equation 8 is an adapted update function in which the collected features from other players are subtracted from the

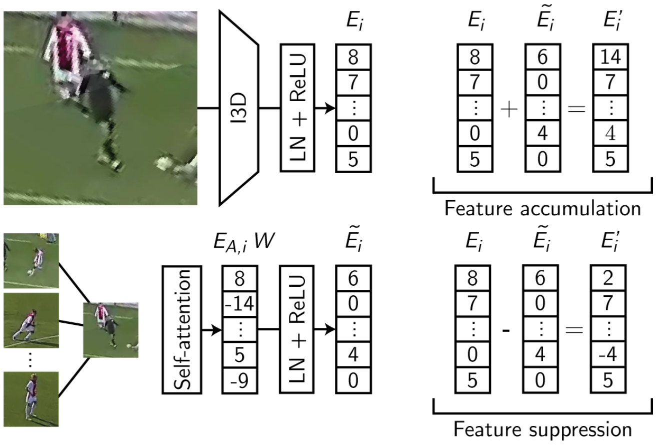


Figure 16: I3D creates player embedding E_i for player i. Embedding $E_{A,i}W$ is gained via attention and includes fractions of features relating to other players' embeddings. Only positive values remain in $\tilde{E}_i$ after layer normalisation (LN) and ReLU activation. Adding $\tilde{E}_i$ to E_i causes feature accumulation, while subtraction causes feature suppression.

original embedding. This causes the opposite effect wherein features can only be reduced. We describe this effect as *feature suppression*. The difference between the two can be observed in Figure 16. The player in the centre of the snippet is wrongly classified as *play-ball* by the baseline ARG. Feature suppression can help to reduce embedding activation correlating to the *play-ball* action label.

$$E'_i = E_i + \sum_{h=1}^{H} \text{ReLU}\Big(\text{LayerNorm}\Big(E_{A,i}^{(h)} W_V^{(h)}\Big)\Big) \tag{7}$$

$$E'_i = E_i - \sum_{h=1}^{H} \text{ReLU}\Big(\text{LayerNorm}\Big(E_{A,i}^{(h)} W_V^{(h)}\Big)\Big) \tag{8}$$

with E'_i the updated embedding and E_i the original embedding of player i, with $E_{A,i}^{(h)}$ a collection of features from other players gained via self-attention and weight matrix $W_V^{(h)} \in \mathbb{R}^{d \times d}$ in graph h.

It is expected that enabling feature suppression reduces the number of FPs and increases model performance above that of the model without feature contextualisation. To test this hypothesis, three versions of the ARG are evaluated: the baseline that enables feature accumulation only, a model with Equation 8 as update function that enables feature suppression only, and a model in which collected features from 32 graphs are added to the original embedding while features from the other 32 graphs are subtracted. The latter model thus enables feature accumulation and suppression explicitly per graph.

The results, in Table 11, show that the performance for action recognition increases (with 0.040 MMCC on average) when feature accumulation and suppression are enabled. No increase is found in activity recognition performance. Moreover, a model that only enables suppression provides similar or slightly better action predictions than the baseline with only feature accumulation. Enabling suppression only disrupts the model's ability to reach consensus over the right activity label.

Update function	Actions ($\pm\sigma$)	Activities ($\pm\sigma$)
Accumulation	0.651 (±.008)	0.628 (±.039)
Suppression	0.664 (±.015)	0.004 (±.039)
Acc. and supp.	0.691 (±.021)	0.612 (±.042)

Table 11: Action and activity recognition with the ARG facilitating feature accumulation, feature suppression or both.

When we focus on the number of FPs, we see that the model enabling accumulation and suppression, ARG (a&s), seems able to discover the discussed implicit bias (see Table 12). The model provides 46 FPs on average, a reduction of 35% in comparison with I3D and the baseline ARG. Importantly, ARG (a&s) only improves in situations where at least two players on the field are classified as *active*. This indicates that feature suppression indeed helps the model to exploit the implicit bias.

# players on the field classified as *active*	**I3D**	**ARG** ($\pm\sigma$)	**ARG (a&s)** ($\pm\sigma$)
≥2	50	53 (±8)	21 (±4)
1	21	19 (±2)	24 (±3)
Total	71	72 (±6)	46 (±6)

Table 12: Number of misclassifications of *passive* players that are recognised as *active* (false positives), in all activities except *duels*.

5.5.3 *Update function and padding strategy.*

Where the ARG (a&s) enables feature accumulation or suppression explicitly per graph, it might be beneficial to use an approach where accumulation and suppression is enabled implicitly. Therefore, two other models are evaluated with an adapted update function. The ReLU activation in all graphs is removed in the first. With this approach, the collection of features is not clipped at zero for negative values. In the second approach, all graphs use the original update function with ReLU activation. Instead of adding the feature collections from all graphs element-wise, all collections are concatenated. Thereafter, the long vector is reduced in dimensionality via a weight matrix (as in Equation 4).

Aside from the update function, another architectural element becomes relevant in relation to the discussed implicit bias: the padding strategy for missing player detections. As previously mentioned, the number of detected players N_d is not equal to N for all samples. To obtain N player embeddings, the baseline ARG uses the duplication strategy, where player embeddings of the top $N - N_d$ players are duplicated in the graph. Since we have put the *active* players on top of the list with detected players, their embeddings are duplicated most. It is expected that this padding strategy weakens the previously discussed implicit bias, as scenes occur in which two embeddings have an *active* ground truth action label. Therefore, we propose to use zero-padding, in which the missing player embeddings are filled with zeros only. We evaluated the discussed models with both padding strategies.

Update func.	Pad.	Actions ($\pm\sigma$)	Activities ($\pm\sigma$)
Accumulation	Dupl.	0.651 (±.008)	0.628 (±.039)
Accumulation	Zero	0.669 (±.018)	0.607 (±.050)
No ReLU	Dupl.	0.684 (±.022)	0.592 (±.039)
No ReLU	Zero	0.702 (±.034)	0.592 (±.065)
Acc. & supp.	Dupl.	0.691 (±.021)	0.612 (±.042)
Acc. & supp.	Zero	0.699 (±.030)	0.670 (±.031)
Concatenation	Dupl.	0.644 (±.020)	0.661 (±.023)
Concatenation	Zero	0.687 (±.020‡)	0.676 (±.020‡)

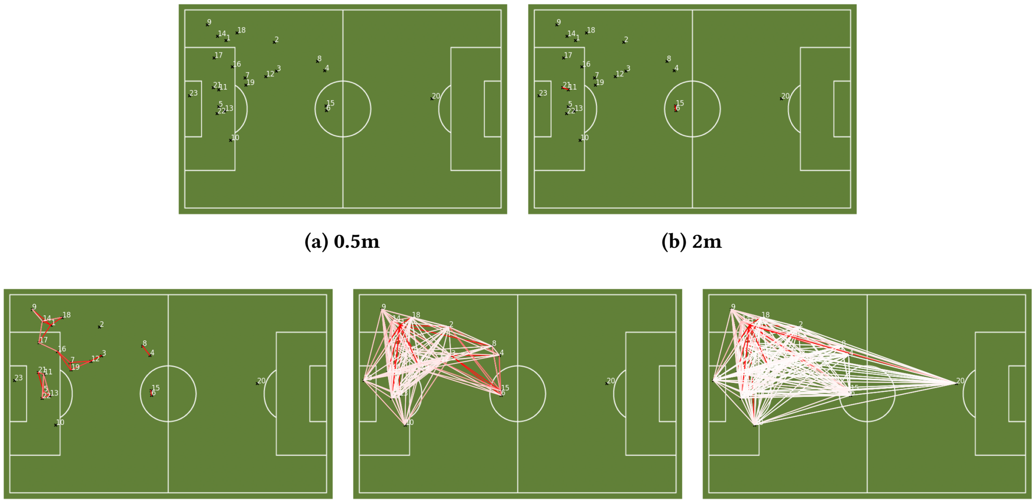
Table 13: MMCCs various update functions with two padding strategies. (‡) Standard deviation of the proposed method is from ten instead of five runs.

The results in Table 13 show that a ReLU activation in the graph convolutional layer is important for improved activity predictions. The two models that enable feature accumulation and suppression, while retaining ReLU activation and using zero-padding, obtain MMCCs around 0.69 and 0.67 for action and activity recognition respectively. We propose to use the concatenation layer, as this approach models feature accumulation and suppression implicitly. However, note that W_O gains more parameters quickly when more graphs are used, as the matrix includes $H \times d \times d$ weights.

5.5.4 ***Relative position encodings***. Both the authors of the ARG and the Actor-Transformer reported that activity predictions improved when they added relative position information to the model. The baseline ARG does so via the distance-mask, pruning graph edges between players that are physically too far from each other. Until now, we have used the distance-mask with a threshold $\mu = 20.8$ meters, which is equal to 0.2 times the width of a soccer court. This is similar to the baseline ARG that uses $\mu = 0.2$ times the width of the input frames. In Section 5.4.2, we have seen that the addition of a an absolute position embedding does not result in better predictions for the Soccer Dataset. Here, we evaluate the role of relative position information, i.e. the distance between players.

First, we assessed the effect of the distance-mask (DM) on action and activity predictions. We experimented with five distance thresholds, ranging from fully disconnected graphs (0.5m) to complete graphs (128m). Note that the diameter of a soccer court is 122 meters. The locality of the resulting graphs can be seen Figure 17, that displays the player-relations for one sample using the five thresholds.

In Figure 18, it can be seen that varying μ does not largely influence the prediction scores. The MMCCs when using the baseline DM and a loose DM (see Table 14) are very close, indicating that the DM is no essential architectural element for feature contextualisation in soccer. Interestingly, the results


(a) 0.5m

(b) 2m

(c) 8m

(d) 32m

(e) 128m

Figure 17: Locality of self-attention using five different numbers for distance threshold μ. Connected players in the graph have an edge (white to red). The larger the attention values between two players, to more red the connection is.

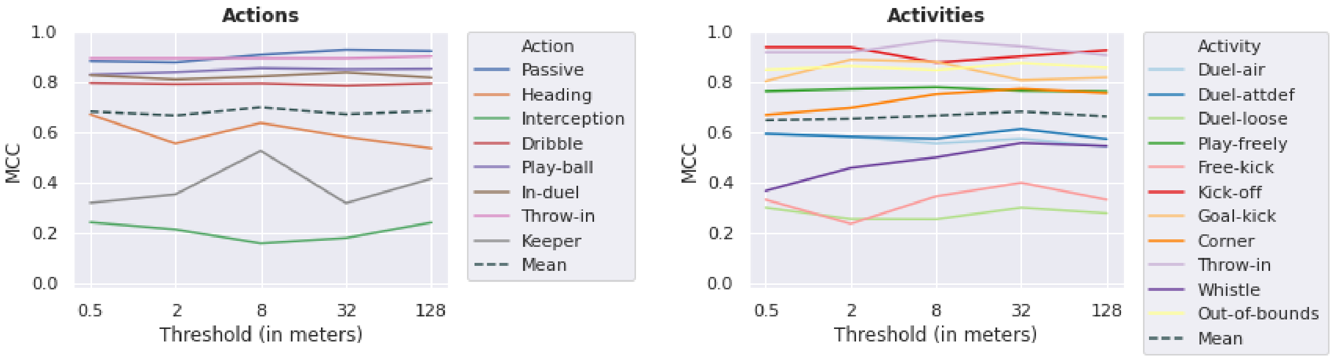


Figure 18: MCC scores of action and activity predictions with various values for threshold μ in the distance-mask. Only player-pairs with smaller distances can exchange features. Note that the diameter of a soccer field is 122m.

between the baseline DM and a strict DM are close as well. This means that an architecture with no feature exchange between different players can obtain comparable performance scores, indicating that the self-attention mechanism is not as influential as expected.

Second, we evaluate relation-aware self-attention [77] to inject relative position information into the attention mechanism. The approach is designed for natural language processing and thus for position distances in integers, where distances between players in soccer are continuous. Instead of the learnable weight matrices per position, we use sinusoids [95] to encode player distances in d-dimensions. Thereafter, we use weight matrices $W^K, W^V \in \mathbb{R}^{d \times d}$ to transform the encodings to key and value matrices that can be applied accordingly in the self-attention mechanism. The approach is thus deterministic: player-pairs that are equally far from each other obtain the same relative position encodings. This approach performs equally well as the GATs with a baseline DM (see Table 14). Even though it is not clear whether relative position information improves action and activity recognition in soccer, both approaches can be used without harming the results.

Position encoding	**Actions** ($\pm\sigma$)	**Activities** ($\pm\sigma$)
Loose DM (μ=128m)	0.685 (±.009)	0.663 (±.015)
Baseline DM (μ=20.8m)	0.687 (±.020‡)	0.676 (±.020‡)
Strict DM (μ=0.5m)	0.676 (±.036)	0.657 (±.014)
Relation-aware self-at.	0.686 (±.014)	0.679 (±.015)

Table 14: Effects of relative position encodings for feature contextualisation, with three distance thresholds μ in the distance-mask (DM) and using relation-aware self-attention.

That position information does not clearly boost model performance can be interpreted in three ways. First, player positions might not contain relevant information about the performed actions or activities. Second, the player-centric snippets might already include the relevant position information. For example, when a player is in duel with another, the players are generally visible in the snippet of each other. Third, we did not evaluate the right method for injecting position information into the system.

5.5.5 ***Layer normalisation.***

We have seen that a model without feature exchange between players (the strict DM) gains a performance comparable to the proposed model (using the baseline DM). This indicates that the role of feature exchange via self-attention is limited. Besides, it raises the question why a model without feature exchange still performs better than the I3D model without feature contextualisation. Aside from self-attention, there is one more mechanism in which feature embeddings are contextualised with the embeddings of other players: via layer normalisation.

In a layer normalisation block, deep embeddings are normalised per data sample by subtracting the mean of all features in the embedding and by dividing each feature by its standard deviation [7]. One bias and one gain parameter are learned by the network such that embedding values can be translated and scaled. Generally, the transformation is applied within one layer, right before activation. Importantly, the implementation of the ARG includes layer normalisation in the graph convolutional layer and is applied to all players at once. Thus, the mean and standard deviation are calculated over N feature embeddings. In this way, the updated player embedding of one is dependent on the embedding of another.

We assess the role of layer normalisation by the evaluation of three models: with normalisation over all N players, with normalisation per player independently, and with no layer normalisation at all. The results can be observed in Table 15. The model where embeddings are normalised per player obtains worse performance then the I3D model without feature contextualisation. When no normalisation is applied, the model scores are similarly low for action recognition. It shows that layer normalisation is valuable in the exploration of inter-player relationships and causes better action predictions in samples from the Soccer Dataset.

Normalisation	Actions (±σ)	Activities (±σ)
Per N players	0.687 (±.020‡)	0.676 (±.020‡)
Per individual player	0.602 (±.011)	0.647 (±.018)
No normalisation	0.611 (±.019)	0.666 (±.022)

Table 15: Effects of layer normalisation for feature contextualisation.

5.5.6 *Comparison with the Actor-Transformer*.

The current best performing model on the Volleyball Dataset is the Actor-Transformer (AT) [31] that uses the encoder part of the Transformer [95] for feature contextualisation. The method was published during the execution of this study, which is why we did not start with the method as our baseline and compare our results here. We reconstructed the AT for feature contextualisation, in order to compare its results with the proposed approach. In Figure 19, it can be seen that the model improves on the I3D player embeddings already from one attention head. Increasing the number of heads does only improve activity recognition slightly, where no effect can be observed for the action predictions. We select the AT with 64 heads as it provides representative MMCCs for the model and as a fair comparison with the proposed model using the same number of attention units. Similar to the ARG, the AT can be utilised with multiple layers. In Table 16 it can be seen that increasing the number of encoder layers does not result in better performance.

#Heads	#Layers	Actions (±σ)	Activities (±σ)
64	1	0.673 (±.017)	0.654 (±.016)
64	2	0.644 (±.026)	0.621 (±.026)
64	4	0.614 (±.020)	0.628 (±.038)

Table 16: Processing I3D feature embeddings with the Actor-Transformer for feature contextualisation. Different numbers of encoder layers are evaluated.

We compare the proposed method with the baseline ARG and the AT. All of these models are trained with 64 graphs or attention heads and one graph convolutional layer or encoder layer. Also, they use player-centric snippets as model input and use features extracted with I3D. In Table 17, it can be observed that both the AT and proposed method improve upon no feature contextualisation. Player feature embeddings can thus be enriched via inter-player relationships to gain better action and activity predictions. In contrast, the baseline ARG causes no improvement. This shows that feature contextualisation is not trivial and its approach should be selected carefully. Last, the proposed method performs better than the AT during the contextualisation phase. Yet the scores are close to each other, considering the standard deviations.

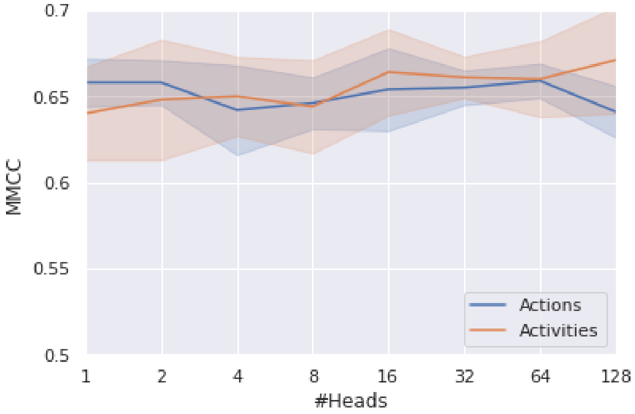


Figure 19: Performance of the Actor-Transformer with different numbers of attention heads. The transparent area relates to the standard deviation (±σ).

Model	Actions (±σ)	Activities (±σ)
No feat. context.	0.658 (±.033†)	0.641 (±.031†)
Baseline ARG	0.651 (±.008)	0.628 (±.039)
Actor-Transformer	0.673 (±.017)	0.654 (±.016)
Proposed	0.687 (±.020‡)	0.676 (±.020‡)

Table 17: Feature contextualisation using the Actor-Transformer or the proposed method increases performance in action and activity recognition.

5.5.7 *Player embedding dimensionality*.

In Section 5.2, we showed that decreasing the dimensionality d of the player embeddings from 1024 to 256 decreased model training time, but also lowered the performance. Here, we evaluate the effect of this decision on the performance of the proposed method. We retrained the I3D model for feature extraction and the proposed approach for feature contextualisation using $d = 1024$. Similar to the original ARG, the *query* and *key* embeddings are transformations to subspaces of 256 dimensions. In contrast to our previous results, in Table 18 it can be seen that the larger dimensionality causes worse predictions. Besides, the number of model parameters is huge when using $d = 1024$, partly due to weight matrix $W_O \in R^{H \times d \times d}$ in the concatenation layer. Such a large complex model is prone to overfitting on small datasets like the Soccer Dataset, which could explain the drop in performance.

d	Parameters	Actions (±σ)	Activities (±σ)
256	42.7M	0.687 (±.020‡)	0.676 (±.020‡)
1024	195.7M	0.603 (±.025)	0.530 (±.088)

Table 18: Performance differences for two sizes of dimensionality d of the player embeddings in the proposed method.

5.6 Evaluation with an unseen game

Finally, we evaluated the original ARG, the proposed method and two intermediate models on samples from a new game. This test data is hold out during the development of the proposed method. The two intermediate models get player snippets as input and have Inception-V3 and I3D as backbone, without feature contextualisation. Last, we evaluated the proposed method trained with training and validation data.

The MCCs are reported per class in Table 19. It can be concluded that the proposed elements for data pre-processing, feature extraction and contextualisation help to obtain better action and activity predictions overall. Compared to the baseline ARG, the proposed method gains equal or better performance for all classes in the Soccer Dataset. The proposed method predicts samples from ten out of nineteen class labels in very strong agreement with the ground-truth labels (MCC ≥ 0.7).

For the comparison with related work and for easy interpretation of the results, MCA and the accuracy per class are reported as well. These can be observed in Table 20. In terms of MCA, our method gains 98.7% and 75.2% accuracy in action and activity recognition respectively. It indicates that the method can successfully identify numerous soccer actions and activities, and is able to generalise patterns to an unseen soccer court and an unseen opponent team.

Last, we briefly investigated the wrongly classified players by the proposed method and assessed whether most errors would include players from the unseen opponent team in the fourth game. On average, the model (trained with training and validation data) misclassified 148 from the 11799 players. From these, 51.58 percent (±1.74) were players from the home team, present in all four games. Thus, the proposed model classified the unseen team just as well as the home team.

5.7 Comparison with related work

To the best of our knowledge, our method is the first to assign action labels to all detected players in soccer videos. Nevertheless, we can compare our result for individual action recognition with methods evaluated on the Volleyball Dataset. Our MCA is 12.8% larger than the best reported accuracy (85.9%) [31], partly due to class imbalance in our dataset. Since we have reproduced a state-of-the-art method on the Volleyball Dataset, we can make a more fair comparison using MMCC as evaluation metric. For action recognition, our method (0.623) gains an MMCC close to the ARG (0.641). The methods are thus proportional to each other in performance, while operating in different sport domains.

A comparison with related work for group activity recognition in soccer videos, can be observed in Table 21. It can be seen that our method is trained with a relatively large number of activity classes. Also, our method cannot rely on cinematic features in television broadcasts and does not combine video streams from multiple camera positions around the soccer field. Nevertheless, the maximum reduction in MCA is 20.3%. Moreover, if we group the three duel labels under one class label, the proposed system predicts with 81.2% MCA, narrowing the gap with state-of-the-art in soccer event recognition from broadcast videos.

6 DISCUSSION

In this work, we have proposed a data pipeline for simultaneous action and activity recognition in soccer videos. With the approach, match event logs can automatically be gathered from video recordings on individual and group level. The work could impact the full landscape of soccer data analysis with match event logs, from player evaluation to the generation of sports visualisations. Currently, most work in

Method	Passive	Heading	Interception	Dribble	Play-ball	In-duel	Throw-in	Keeper	**Mean MCC**	Duel (air)	Duel (ball possession)	Duel (loose ball)	Play-freely	Free-kick	Kick-off	Goal-kick	Corner	Throw-in	Whistle	Ball out-of-bounds	**Mean MCC**
Baseline ARG	.02	.00	.00	.01	-.01	.01	.02	.00	.005	.00	.00	-.01	.01	.00	.00	.00	.00	.00	.00	-.02	.000
Snippets	.67	.25	.00	.34	.57	.61	.73	.28	.431	.20	.55	.23	.65	.17	.82	.55	.86	.75	.15	.65	.506
I3D	.81	.34	.04	.52	.79	.69	.92	.33	.553	.46	.62	.27	.72	.42	.71	.55	.81	.82	.20	.68	.569
Proposed	.91	.46	.03	.55	.83	.80	.94	.54	.632	.53	.58	.31	.72	.44	.68	.78	.80	.90	.23	.69	.605
Proposed*	.96	.47	.00	.56	.84	.85	.85	.47	.623	.58	.59	.33	.74	.43	.76	.86	.81	.84	.28	.73	.632
# Instances	11140	15	17	44	221	27	35	8		11	32	85	255	13	2	9	10	35	12	49	

Table 19: MCCs for the baseline ARG, player-centric snippets with Inception-V3, player-centric snippets with I3D, and the proposed method, on samples from an unseen game. (*) Model is trained with the training and the validation set. In the last row, the number of test instances per class is repeated.

Method	Passive	Heading	Interception	Dribble	Play-ball	In-duel	Throw-in	Keeper	**MCA**	Duel (air)	Duel (ball possession)	Duel (loose ball)	Play-freely	Free-kick	Kick-off	Goal-kick	Corner	Throw-in	Whistle	Ball out-of-bounds	**MCA**
Baseline ARG	22	06	16	14	14	17	08	00	21.6	07	00	01	05	00	00	00	00	20	40	25	7.2
Snippets	95	28	00	31	82	88	67	28	93.4	13	62	28	91	08	70	31	88	67	13	60	70.6
I3D	99	31	03	47	76	64	93	21	97.3	30	68	22	93	39	50	33	73	97	14	62	75.6
Proposed	100	52	04	56	80	72	98	35	98.4	35	62	35	83	48	90	69	74	94	27	88	74.0
Proposed*	100	40	00	50	88	79	100	35	98.7	36	64	39	82	43	100	78	90	99	32	87	75.2

Table 20: Accuracies (in percentages) per class for the baseline ARG, player-centric snippets with Inception-V3, player-centric snippets with I3D, and the proposed method. (*) Model is trained with the training and the validation set.

Reference	**Camera setup**	**# Classes**	**MCA**
[86] (2014)	Television broadcasts	7	82.0%
[46] (2016)	Television broadcasts	4	89.1%
[53] (2018)	Television broadcasts	4	94.5%
[94] (2020)	Television broadcasts	4	95.5%
[90] (2017)	Multiple-perspective	3	70.2%
Ours	One-perspective	11	75.2%

Table 21: Comparison with other methods for activity recognition in soccer videos.

sports analysis is based on manually annotated event logs that are generally only available for professional games. Automating the obtainment of match events could open up data analysis solutions for a broad public, from amateur clubs to youth teams at professional clubs. Unique is that the proposed method is designed for stationary cameras that capture a full-view of the soccer field from a single location. This means that the system is applicable to affordable camera setups.

The current system, however, is not yet fully deployable for practical use. The designed method is focused on recognition rather than temporal detection of match events. All samples in the Soccer Dataset are manually selected in time and it is unclear how the method performs on arbitrary samples in which the action or activity is not performed at the middle frame. Moreover, the system requires a camera model, calibrated with the soccer field dimensions, for the virtual camera algorithm producing the player-centric snippets. A camera model is not always available and re-calibration is necessary when the cameras are moved (e.g. due to a ball hitting one of the cameras).

Another limiting factor is the small variation in games that are included in the dataset. Although the proposed model seems able to generalise to an unseen game, testing on one game is not enough to verify that good performance can be expected with many other unseen games. Importantly, the game in the test set involves one team that was also present in the three games used for training, while the skill level (young professionals) was similar as well. Also, the test game did not display deviating weather or lighting conditions and its videos were recorded at the same training complex, using the same camera setup, as those in the training set. The method's applicability to games with conditions deviating from these remain uncertain. It is expected that generalisation would increase when training on more games that involve stretching variation in the mentioned aspects. Positively, the test game was played on a field with artificial grass where the other games were played on regular grass, and the opponent team was not involved in a training game (see Appendix VII for a sample frame from each game).

Although the proposed method is able to recognise six activity labels in very strong agreement with the ground-truth labels (MCC ≥ 0.7), the dataset is very sparse for three of these. The validation and test set each contain ten or less examples for corners, goal kicks and kick offs. Therefore, it is uncertain how reliable these MCCs are. Evaluation with larger sample sizes is required to see if the results hold.

The experiments were performed on a GPU with 11GB of RAM. Due to the limited memory, the feature extraction and feature contextualisation phases are trained separately. This is a limitation for two reasons. First, a model that is trained end-to-end might result in better predictions, as weights in the I3D network could be adapted for the exploration of inter-player relations. Second, it is uncertain whether I3D would adapt the player embeddings based on the implicit bias discussed in Section 5.5.2, possibly removing the effectiveness of feature suppression. Evaluation with an end-to-end trained model would provide these insights.

Last, the Actor-Transformer [31] data pipeline is based on I3D feature extraction and pose estimations. The authors report to obtain slight action and activity recognition improvements when using pose information of volleyball players. Future work would include the experimentation with a pose estimator in the soccer domain.

7 CONCLUSION

We conclude this study by answering the sub-research questions. These are repeated below, accompanied with conclusions drawn from our experiments.

RQ1: "How to create a dataset for action and activity recognition in the soccer domain?"

We have shown how one could construct a soccer dataset that is useful for action and activity recognition. Our approach is unique in that it includes videos that capture the soccer field from one stationary perspective. Multiple-perspective camera setups, moving cameras or cinematic features, as often used in related work, are not essential for good action and activity predictions.

RQ2: "Which soccer match events are relevant to be automatically gathered?"

The set of class labels consists of eight individual actions and eleven group activities. Except for the *passive* class, only match events that are generally published by Competition Information Providers [20] are included, meaning that all labels are relevant to the soccer community. Besides, we noticed that two players in a duel can have different action labels, but almost identical player representations. We proposed to give these players the same *in-duel* action label for training stability.

RQ3: "Is the Actor Relation Graph method, used for action and activity recognition in volleyball, applicable to soccer?"

When the Actor Relation Graph (ARG) is applied to soccer videos instead of volleyball, performance is diminished drastically in action and activity recognition. The baseline predicts 21.6% of the actions and 7.2% of the activities accurately on samples from an unseen game. A different approach is necessary to gain good model predictions.

RQ4: "What are limitations to the Actor Relation Graph method when applied to soccer videos?"

Limitations of the baseline ARG were found in three phases of the data pipeline: data pre-processing, per-player feature extraction and feature contextualisation. In the pre-processing phase, the baseline sub-samples the raw frames, resulting in player representations that are too small for proper action and activity recognition. Also, a rotated horizon present at most field locations is worsening the baseline performance. In the feature extraction phase, the ARG uses Inception-V3 as backbone. The resulting feature embeddings are informative enough for very strong recognition (MCC ≥ 0.7) of four actions and six activities from samples in the validation set. However, the backbone does not generalise well to an unseen game, where only one action and three activities gain MCCs above 0.7. In the feature contextualisation phase, we found that the ARG lacks the ability to identify an implicit bias present in soccer games. Last, we saw that the distance-mask and the exchange of features via self-attention only play marginal roles in feature contextualisation for soccer videos.

RQ5: "Does the proposed method, based on player-centric snippets, I3D feature extraction, and graph attention networks with feature suppression and zero-padding, result in more accurate action and activity predictions than the Actor Relation Graph?"

We compared the proposed method with the baseline ARG and saw that our model performs better on the Soccer Dataset. The proposed approach recognises eight actions and eleven activities with respectively 98.7% and 75.2% accuracy in samples from an unseen game. Moving to player-centric snippets as model input resulted in the largest performance increase, with overall accuracies of 93.4% for action and 70.6% for activity recognition. In the feature extraction phase, we saw that an I3D backbone generalises better to an unseen game than Inception-V3. Using player snippets and I3D, three actions and four activities were recognised with MCCs above 0.7 on the test set. Furthermore, we have shown that feature suppression in graph attention networks can help to identify an implicit bias present in soccer games. With our approach, the model was able to reduce the number of wrongly classified *passive* players with 58%, when at least two players were recognised as interacting with the ball. Together with utilising zero-padding for missing player detections, this resulted in better action and activity predictions.

ACKNOWLEDGMENTS

The thesis, as part of the Master's programmes Computer Science and Interaction Technology at the University of Twente, is performed at the Netherlands Organisation for Applied Scientific Research TNO. The soccer video recordings, player detections, player tracks and the virtual director software were provided by this organisation.

REFERENCES

[1] Bart Aalbers and Jan Van Haaren. 2018. Distinguishing between roles of football players in play-by-play match event data. In *International Workshop on Machine Learning and Data Mining for Sports Analytics*. Springer, 31–41.

[2] Mohamed Rabie Amer, Peng Lei, and Sinisa Todorovic. 2014. Hirf: Hierarchical random field for collective activity recognition in videos. In *European Conference on Computer Vision*. Springer, 572–585.

[3] Mohamed R Amer, Sinisa Todorovic, Alan Fern, and Song-Chun Zhu. 2013. Monte carlo tree search for scheduling activity recognition. In *Proceedings of the IEEE International Conference on Computer Vision*. 1353–1360.

[4] Mohamed R Amer, Dan Xie, Mingtian Zhao, Sinisa Todorovic, and Song-Chun Zhu. 2012. Cost-sensitive top-down/bottom-up inference for multiscale activity recognition. In *European Conference on Computer Vision*. Springer, 187–200.

[5] Borislav Antic and Björn Ommer. 2014. Learning latent constituents for recognition of group activities in video. In *European Conference on Computer Vision*. Springer, 33–47.

[6] Sina Mokhtarzadeh Azar, Mina Ghadimi Atigh, Ahmad Nickabadi, and Alexandre Alahi. 2019. Convolutional Relational Machine for Group Activity Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7892–7901.

[7] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[8] Timur Bagautdinov, Alexandre Alahi, François Fleuret, Pascal Fua, and Silvio Savarese. 2017. Social scene understanding: End-to-end multi-person action localization and collective activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4315–4324.

[9] Sovan Biswas and Juergen Gall. 2018. Structural recurrent neural network (srnn) for group activity analysis. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1625–1632.

[10] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6299–6308.

[11] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.

[12] Ming-Ching Chang, Nils Krahnstoever, Sernam Lim, and Ting Yu. 2010. Group level activity recognition in crowded environments across multiple cameras. In *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, 56–63.

[13] Sheng Chen, Zhongyuan Feng, Qingkai Lu, Behrooz Mahasseni, Trevor Fiez, Alan Fern, and Sinisa Todorovic. 2014. Play type recognition in real-world football video. In *IEEE Winter Conference on Applications of Computer Vision*. IEEE, 652–659.

[14] Davide Chicco. 2017. Ten quick tips for machine learning in computational biology. *BioData mining* 10, 1 (2017), 35.

[15] Davide Chicco and Giuseppe Jurman. 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics* 21, 1 (2020), 6.

[16] Wongun Choi and Silvio Savarese. 2012. A unified framework for multi-target tracking and collective activity recognition. In *European Conference on Computer Vision*. Springer, 215–230.

[17] Wongun Choi, Khuram Shahid, and Silvio Savarese. 2009. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE, 1282–1289.

[18] Wongun Choi, Khuram Shahid, and Silvio Savarese. 2011. Learning context for collective activity recognition. In *CVPR 2011*. IEEE, 3273–3280.

[19] Frédéric Cupillard, François Brémond, and Monique Thonnat. 2002. Group behavior recognition with multiple cameras. In *Sixth IEEE Workshop on Applications of Computer Vision, 2002.(WACV 2002). Proceedings*. IEEE, 177–183.

[20] Tom Decroos, Lotte Bransen, Jan Van Haaren, and Jesse Davis. 2019. Actions speak louder than goals: Valuing player actions in soccer. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1851–1861.

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[22] Zhiwei Deng, Arash Vahdat, Hexiang Hu, and Greg Mori. 2016. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4772–4781.

[23] Zhiwei Deng, Mengyao Zhai, Lei Chen, Yuhao Liu, Srikanth Muralidharan, Mehrsan Javan Roshtkhari, and Greg Mori. 2015. Deep structured models for group activity recognition. *arXiv preprint arXiv:1506.04191* (2015).

[24] Tijmen van Dijk. 2019. *Ball-I3D: Localizing Footballs from Player Coordinates*. Master's thesis. University of Amsterdam, the Netherlands.

[25] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. 2014. Fast feature pyramids for object detection. *IEEE transactions on pattern analysis and machine intelligence* 36, 8 (2014), 1532–1545.

[26] Tiziana D'Orazio, Marco Leo, Nicola Mosca, Paolo Spagnolo, and Pier Luigi Mazzeo. 2009. A semi-automatic system for ground truth generation of soccer video sequences. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, 559–564.

[27] Ahmet Ekin, A Murat Tekalp, and Rajiv Mehrotra. 2003. Automatic soccer video analysis and summarization. *IEEE Transactions on Image processing* 12, 7 (2003), 796–807.

[28] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. 2000. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics* 28, 2 (2000), 337–407.

[29] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. 2018. Multi-Level Sequence GAN for Group Activity Recognition. In *Asian Conference on Computer Vision*. Springer, 331–346.

[30] Yaparla Ganesh, Allaparthi Sri Teja, Sai Krishna Munnangi, and Garimella Rama Murthy. 2019. A Novel Framework for Fine Grained Action Recognition in Soccer. In *International Work-Conference on Artificial Neural Networks*. Springer, 137–150.

[31] Kirill Gavrilyuk, Ryan Sanford, Mehrsan Javan, and Cees GM Snoek. 2020. Actor-transformers for group activity recognition. *arXiv preprint arXiv:2003.12737* (2020).

[32] Yihong Gong, Lim Teck Sin, Chua Hock Chuan, Hongjiang Zhang, and Masao Sakauchi. 1995. Automatic parsing of TV soccer programs. In *Proceedings of the International Conference on Multimedia Computing and Systems*. IEEE, 167–174.

[33] Ping Guo, Zhenjiang Miao, Xiao-Ping Zhang, Yuan Shen, and Shu Wang. 2012. Coupled observation decomposed hidden Markov model for multiperson activity recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 9 (2012), 1306–1320.

[34] Hossein Hajimirsadeghi and Greg Mori. 2015. Learning ensembles of potential functions for structured prediction with latent variables. In *Proceedings of the IEEE International Conference on Computer Vision*. 4059–4067.

[35] Hossein Hajimirsadeghi, Wang Yan, Arash Vahdat, and Greg Mori. 2015. Visual recognition by counting instances: A multi-instance cardinality potential kernel. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2596–2605.

[36] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[38] Michael Horton. 2020. Learning feature representations from football tracking. MIT Sloan Sports Analytics Conference.

[39] Guyue Hu, Bo Cui, Yuan He, and Shan Yu. 2019. Progressive Relation Learning for Group Activity Recognition. *arXiv preprint arXiv:1908.02948* (2019).

[40] Chung-Lin Huang, Huang-Chia Shih, and Chung-Yuan Chao. 2006. Semantic analysis of soccer video using dynamic Bayesian network. *IEEE Transactions on Multimedia* 8, 4 (2006), 749–760.

[41] Mostafa S Ibrahim and Greg Mori. 2018. Hierarchical relational networks for group activity recognition and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 721–736.

[42] Mostafa S Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. 2016. A hierarchical deep temporal model for

group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1971–1980.

[43] Stephen S Intille and Aaron F Bobick. 2001. Recognizing planned, multiperson action. *Computer Vision and Image Understanding* 81, 3 (2001), 414–445.

[44] Alexandros Iosifidis, Anastasios Tefas, and Ioannis Pitas. 2012. View-invariant action recognition based on artificial neural networks. *IEEE transactions on neural networks and learning systems* 23, 3 (2012), 412–424.

[45] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2012. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence* 35, 1 (2012), 221–231.

[46] Haohao Jiang, Yao Lu, and Jing Xue. 2016. Automatic soccer video event detection based on a deep neural network combined cnn and rnn. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 490–494.

[47] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. 2017. Action tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*. 4405–4413.

[48] Takuhiro Kaneko, Masamichi Shimosaka, Shigeyuki Odashima, Rui Fukui, and Tomomasa Sato. 2014. A fully connected model for consistent collective activity recognition in videos. *Pattern Recognition Letters* 43 (2014), 109–118.

[49] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950* (2017).

[50] Shian-Ru Ke, Hoang Le Uyen Thuc, Yong-Jin Lee, Jenq-Neng Hwang, Jang-Hee Yoo, and Kyoung-Ho Choi. 2013. A review on video-based human activity recognition. *computers* 2, 2 (2013), 88–131.

[51] Sameh Khamis, Vlad I Morariu, and Larry S Davis. 2012. Combining per-frame and per-track cues for multi-person action recognition. In *European Conference on Computer Vision*. Springer, 116–129.

[52] Abdullah Khan, Beatrice Lazzerini, Gaetano Calabrese, and Luciano Serafini. 2018. Soccer event detection. In *4th international conference on image processing and pattern recognition (IPPR 2018). AIRCC Publishing Corporation*. 119–129.

[53] Muhammad Zeeshan Khan, Summra Saleem, Muhammad A Hassan, and Muhammad Usman Ghanni Khan. 2018. Learning Deep C3D Features For Soccer Video Event Detection. In *2018 14th International Conference on Emerging Technologies (ICET)*. IEEE, 1–6.

[54] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[55] Yu Kong and Yun Fu. 2018. Human action recognition and prediction: A survey. *arXiv preprint arXiv:1806.11230* (2018).

[56] Tian Lan, Leonid Sigal, and Greg Mori. 2012. Social roles in hierarchical models for human activity recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1354–1361.

[57] Tian Lan, Yang Wang, Weilong Yang, and Greg Mori. 2010. Beyond actions: Discriminative models for contextual group activities. In *Advances in neural information processing systems*. 1216–1224.

[58] Tian Lan, Yang Wang, Weilong Yang, Stephen N Robinovitch, and Greg Mori. 2011. Discriminative latent models for recognizing contextual group activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 8 (2011), 1549–1562.

[59] Michael Lewis. 2004. *Moneyball: The art of winning an unfair game*. WW Norton & Company.

[60] Wenbo Li, Ming-Ching Chang, and Siwei Lyu. 2018. Who did What at Where and When: Simultaneous Multi-Person Tracking and Activity Recognition. *arXiv preprint arXiv:1807.01253* (2018).

[61] Xin Li and Mooi Choo Chuah. 2017. Sbgar: Semantics based group activity recognition. In *Proceedings of the IEEE International Conference on Computer Vision*. 2876–2885.

[62] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. 2019. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*. 3889–3898.

[63] Weiyao Lin, Hang Chu, Jianxin Wu, Bin Sheng, and Zhenzhong Chen. 2013. A heat-map-based algorithm for recognizing group activities in videos. *IEEE Transactions on Circuits and Systems for Video Technology* 23, 11 (2013), 1980–1992.

[64] Weiyao Lin, Ming-Ting Sun, Radha Poovendran, and Zhengyou Zhang. 2010. Group event detection with a varying number of group members for video surveillance. *IEEE Transactions on Circuits and Systems for Video Technology* 20, 8 (2010), 1057–1067.

[65] Daniel Link and Martin Hoernig. 2017. Individual ball possession in soccer. *PloS one* 12, 7 (2017), e0179953.

[66] Lihua Lu, Huijun Di, Yao Lu, Lin Zhang, and Shunzhou Wang. 2019. Spatio-temporal attention mechanisms based model for collective activity recognition. *Signal Processing: Image Communication* 74 (2019), 162–174.

[67] M Manafifard, Hamid Ebadi, and H Abrishami Moghaddam. 2017. A survey on player tracking in soccer videos. *Computer Vision and Image Understanding* 159 (2017), 19–46.

[68] Kentaro Matsui, Masaki Iwase, Masato Agata, Toshimi Tsu Tanaka, and Noboru Ohnishi. 1998. Soccer image sequence computed by a virtual camera. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*. IEEE, 860–865.

[69] Brian W Matthews. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405, 2 (1975), 442–451.

[70] Luca Pappalardo, Paolo Cintia, Paolo Ferragina, Emanuele Massucco, Dino Pedreschi, and Fosca Giannotti. 2019. PlayeRank: data-driven performance evaluation and player ranking in soccer via a machine learning approach. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 5 (2019), 59.

[71] Luca Pappalardo, Paolo Cintia, Alessio Rossi, Emanuele Massucco, Paolo Ferragina, Dino Pedreschi, and Fosca Giannotti. 2019. A public data set of spatio-temporal match events in soccer competitions. *Scientific data* 6, 1 (2019), 1–15.

[72] Mengshi Qi, Yunhong Wang, Jie Qin, Annan Li, Jiebo Luo, and Luc Van Gool. 2019. stagNet: An Attentive Semantic RNN for Group Activity and Individual Action Recognition. *IEEE Transactions on Circuits and Systems for Video Technology* (2019).

[73] Vignesh Ramanathan, Jonathan Huang, Sami Abu-El-Haija, Alexander Gorban, Kevin Murphy, and Li Fei-Fei. 2016. Detecting events and key actors in multi-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3043–3053.

[74] MS Ryoo and JK Aggarwal. 2011. Stochastic representation and recognition of high-level group activities. *International journal of computer Vision* 93, 2 (2011), 183–200.

[75] Bruce Schoenfeld. 2019. How Data (and Some Breathtaking Soccer) Brought Liverpool to the Cusp of Glory. *The New York Times Magazine* (May 2019). https://www.nytimes.com/2019/05/22/magazine/soccer-data-liverpool.html

[76] Christian Schuldt, Ivan Laptev, and Barbara Caputo. 2004. Recognizing human actions: a local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 3. IEEE, 32–36.

[77] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*

(2018).

[78] Tianmin Shu, Sinisa Todorovic, and Song-Chun Zhu. 2017. CERN: confidence-energy recurrent network for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5523–5531.

[79] Xiangbo Shu, Jinhui Tang, Guojun Qi, Wei Liu, and Jian Yang. 2019. Hierarchical long short-term concurrent memory for human interaction recognition. *IEEE transactions on pattern analysis and machine intelligence* (2019).

[80] Mei-Ling Shyu, Zongxing Xie, Min Chen, and Shu-Ching Chen. 2008. Video semantic event/concept detection using a subspace-based multimedia data mining framework. *IEEE Transactions on Multimedia* 10, 2 (2008), 252–259.

[81] Khurram Soomro and Amir R Zamir. 2014. Action recognition in realistic sports videos. In *Computer vision in sports*. Springer, 181–208.

[82] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).

[83] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

[84] Jinhui Tang, Xiangbo Shu, Rui Yan, and Liyan Zhang. 2019. Coherence Constrained Graph LSTM for Group Activity Recognition. *IEEE transactions on pattern analysis and machine intelligence* (2019).

[85] Yansong Tang, Zian Wang, Peiyang Li, Jiwen Lu, Ming Yang, and Jie Zhou. 2018. Mining Semantics-Preserving Attention for Group Activity Recognition. In *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 1283–1291.

[86] Mostafa Tavassolipour, Mahmood Karimian, and Shohreh Kasaei. 2013. Event detection and summarization in soccer videos using bayesian network and copula. *IEEE Transactions on circuits and systems for video technology* 24, 2 (2013), 291–304.

[87] Graham Thomas, Rikke Gade, Thomas B Moeslund, Peter Carr, and Adrian Hilton. 2017. Computer vision for sports: Current applications and research topics. *Computer Vision and Image Understanding* 159 (2017), 3–18.

[88] Moumita Roy Tora, Jianhui Chen, and James J Little. 2017. Classification of puck possession events in ice hockey. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 147–154.

[89] Khai N Tran, Apurva Gala, Ioannis A Kakadiaris, and Shishir K Shah. 2014. Activity analysis in crowded environments using social cues for group discovery and human interaction modeling. *Pattern Recognition Letters* 44 (2014), 49–57.

[90] Takamasa Tsunoda, Yasuhiro Komori, Masakazu Matsugu, and Tatsuya Harada. 2017. Football action recognition using hierarchical LSTM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 99–107.

[91] Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea. 2008. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video technology* 18, 11 (2008), 1473.

[92] SA Vahora and NC Chauhan. 2017. A comprehensive study of group activity recognition methods in video. *Indian Journal of Science and Technology* 10, 23 (2017), 1–11.

[93] SA Vahora and NC Chauhan. 2019. Deep neural network model for group activity recognition using contextual relationship. *Engineering Science and Technology, an International Journal* 22, 1 (2019), 47–54.

[94] Bastien Vanderplaetse and Stephane Dupont. 2020. Improved Soccer Action Spotting Using Both Audio and Video Streams. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 896–897.

[95] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[96] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[97] Minsi Wang, Bingbing Ni, and Xiaokang Yang. 2017. Recurrent modeling of interaction context for collective activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3048–3056.

[98] Yang Wang and Greg Mori. 2009. Max-margin hidden conditional random fields for human action recognition. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 872–879.

[99] Jianchao Wu, Limin Wang, Li Wang, Jie Guo, and Gangshan Wu. 2019. Learning Actor Relation Graphs for Group Activity Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9964–9974.

[100] Lexing Xie, Shih-Fu Chang, Ajay Divakaran, and Huifang Sun. 2002. Structure analysis of soccer video with hidden Markov models. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 4. IEEE, IV–4096.

[101] Junji Yamato, Jun Ohya, and Kenichiro Ishii. 1992. Recognizing human action in time-sequential images using hidden markov model.. In *CVPR*, Vol. 92. 379–385.

[102] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems*. 3320–3328.

[103] Christopher Zach, Thomas Pock, and Horst Bischof. 2007. A duality based approach for realtime TV-L 1 optical flow. In *Joint pattern recognition symposium*. Springer, 214–223.

[104] Sofia Zaidenberg, Bernard Boulay, and François Brémond. 2012. A generic framework for video understanding applied to group behavior recognition. In *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*. IEEE, 136–142.

[105] Dong Zhang, Daniel Gatica-Perez, Samy Bengio, and Iain McCowan. 2006. Modeling individual and group actions in meetings with layered HMMs. *IEEE Transactions on Multimedia* 8, 3 (2006), 509–520.

[106] Kailai Zhang, Ji Wu, Xiaofeng Tong, and Yumeng Wang. 2019. An automatic multi-camera-based event extraction system for real soccer videos. *Pattern Analysis and Applications* (2019), 1–13.

Appendix I. Overview of publications in group activity recognition (2001-2019)

		Aim							Performance					
Year	Ref.	D	T	I	G	S	K	Method	C-5	C-6	C-N	NH	V-I	V-G
2001	[43]				✓	A		Bayesian networks						
2002	[19]				✓			Rule-based finite state automaton						
2006	[105]				✓			Audio-visual features in a two-layer HMM framework						
2009	[17]				✓			Spatio-temporal local descriptor	65.9					
2010	[12]				✓			Divisive clustering combined with rule based recognition						
	[57]			✓	✓			Learnable latent graph structure, SVM for classification	79.1					
	[64]	✓			✓			Asynchronous HMM						
2011	[18]				✓			Random Forest classification, 3D Markov Rand. Field localization	70.9	82.0				
	[58]				✓			Learnable latent graph structure, AC descriptor, multiclass SVM	79.7			78.5		
	[74]				✓			Markov chain Monte Carlo based probability distribution sampling						
2012	[4]	✓		✓	✓			Three-layered AND-OR graph (bottom-up, top-down inference)	83.6					
	[16]	✓	✓		✓			Iterative belief propagation, multiclass SVM classifiers	79.1		83.0			
	[33]				✓			Coupled observation decomposed HMM, trained with EM						
	[51]			✓	✓			Action Context (AC) descriptor, SVM, harmony normalization	72.0	85.8				
	[56]			✓	✓	H		Learnable latent graph structure, HOG, structured SVM				80.5		
	[104]	✓	✓		✓			Rule based: event model tree						
2013	[3]			✓	✓			Spatio-temporal AND-OR graph, Monte Carlo Tree Search	88.9		84.2			
	[63]				✓			Heat-maps from trajectories, heat-map alignment, surface-fitting for classification						
2014	[2]				✓			Hierarchical Random Field, bottom-up, top-down inference, max-margin learning	**92.0**		87.2			
	[5]				✓			Learning latent constituents, multi-class SVM	75.1	90.1				
	[13]				✓	A		Noisy detector, HMM, logistic regression						
	[48]			✓	✓			AC descriptor, SVM, fully connected random fields	74.7		70.7			
	[89]	✓			✓			Graph-based clustering, local group activity descriptor, BoW features, SVM	78.8	80.8				
2015	[23]			✓	✓			CNN predictions (on scene, actions, poses), message passing neural network	80.6			84.7		
	[34]			✓	✓			Hidden conditional random fields-Boost	82.5			73.0		
	[35]			✓	✓			Cardinality kernels applied to bags of instances, kernel classifier	83.4					
2016	[22]				✓			CNN predictions (on scene, actions), RNN models relations, message passing	81.2	90.2		**85.5**		
	[42]			✓	✓			Two-stage LSTM	81.5					81.9
	[73]	✓			✓	B	✓	Inception7-CNN feat. vectors, BLSTM context feat., player attention						

		Aim							Performance					
Year	**Ref.**	**D**	**T**	**I**	**G**	**S**	**K**	**Method(s)**	**C-5**	**C-6**	**C-N**	**NH**	**V-I**	**V-G**
2017	[8]	✓		✓	✓	V		CNN detections, predictions and feat. map per frame, Markov Rand. Field, RNN					82.4	89.9
	[61]				✓	V		Optical Flow, CNN feat. vectors and LSTM caption per frame, LSTM classifier	86.1					66.9
	[78]			✓	✓	V		Two-stage LSTM (instead of softmax, an energy layer, maximizing confidence)	87.2					83.3
	[88]			✓	✓	I		CNN feat. vectors (scene, actions), LSTM						
	[90]			✓	✓	S		CNN feat. vectors (actions) added with meta-information (player, ball and camera location, team), hierarchical LSTM						
	[97]				✓			CNN feat. vectors (image, optical flow), three-level LSTM (person, group, scene)			89.4			
2018	[9]			✓	✓	V		Structural RNN, modelling inter-person relations					76.7	83.5
	[29]				✓	V		Multi-Level Sequence GAN (CNN feat. vectors, LSTM, GFUs)	**91.7**					**93.0**
	[41]			✓	✓	V		CNN feat. vectors, hierarchical relational network						89.5
	[60]	✓	✓	✓	✓			Hypergraph construction, modelling inter-person relations	**92.4**	**94.3**	89.3			
	[85]				✓	V	✓	Semantics-Preserving Teacher-Student model (CNN, BLSTM), using attention						90.7
2019	[6]			✓	✓	V		Convolutional Relational Machine: I3D feat. map per frame, pred. activity maps	85.8					**93.0**
	[39]				✓	V		Semantic Relation Graph, refined with feature-distilling and relation-gating agents						**91.4**
	[66]				✓	V		Pose and RGB feat. extraction, two-stage GRU network (pose and temp. attention)						**91.7**
	[72]			✓	✓	V		stagNet: semantic graph, structural RNNs, pose and spatial-temporal attention	89.1		**90.2**		82.3	89.3
	[79]			✓	✓	V		Hierarchical Long Short-Term Concurrent Memory	83.75					88.4
	[84]				✓	V		Coherence Constrained Graph LSTM, temporal and spatial conf. gates, attention	**93.0**					89.3
	[93]				✓			CNN feat. (scene, action, pose), RNN as ST-group descriptor, prob. inference	83.5					
	[99]			✓	✓	V	✓	Actor Relation Graph: CNN feat. (action), multi-head graph attention network	**91.0**				**83.1**	**92.6**

Table 22: An (incomplete) overview of papers published with the goal of group activity recognition between 2001 and 2019. Aim of their method is noted: detection (D) and tracking (T) of people, classification of individual actions (I) and group activities (G), (partly) designed for sport videos (S), and identification of the key actor. Sports found are: American football (A), hockey (H), basketball (B), volleyball (V), ice-hockey (I) and soccer (S). Model performances are given, when available, for the Collective Activity Dataset: the original set including 5 classes (C-5) [17], the extended set including 6 classes (C-6) [18] and the new set (C-N) [16]; the Nursing Home Dataset [58]; and the Volleyball Dataset [42], on individual actions (V-I) and group activities (V-G). The scores are given in multi-class accuracy (MCA). For evaluation sets C-5 and V-G, the top-5 methods are in bold. The one best performing method is in bold for the other sets.

Appendix II. Initial experiment

In order to gain insight in the suitability of a new dataset to the research objective, an initial experiment is executed. The video recordings in the new dataset are captured by a one-perspective camera setup, but have never been used for action or group activity recognition before. The dataset consists of two parts: raw video footage that captures the full soccer field and player detections. With this initial experiment, we want to confirm two hypotheses. First, that it is possible to accurately recognise a particular individual action by processing the direct visual surroundings of a soccer player. Second, that it is possible to do so using the automatic player detections. If both are confirmed, it means that it is reasonable to use this dataset for semantic labelling and that no manual annotations of player detections are necessary to do so.

The remainder starts with the definition of individual ball possession as this is not trivial. The experiment works with different definitions, such that the results are useful for different meanings of ball possession. It immediately demonstrates the issues of capturing semantics between class boundaries. Afterwards, the setup of the experiment is discussed, including how the dataset is used, which models to evaluate and how they are evaluated. Model design and results are discussed per model in separate sections. Last, the results are compared in an overview.

1. Definitions

Since no official definition of individual ball possession could be found, three states have been labelled per player as to cover most definitions: 'near distance' (N), 'control over the ball' (C) and 'future possession' (F). First, it is annotated whether the distance between the player and the ball is such that the player is directly able to influence the trajectory of the ball, with a maximum of one body movement, such as a jump, a step or a kick (N). Multiple players can be in this state simultaneously. Secondly annotated is whether the player is in control of the ball (C). This state starts when the player touches the ball. From then, the player is in control as long as the ball moves as a direct result of the contact between the player and the ball and as long as the ball moves according to the player's intentions. Note that this state does not depend on the distance to the ball. Only one player can be in this state at a time. Last is annotated if the player will definitely gain control soon (F). In this state, the trajectory of the ball and the player are such that it is clear that this player, and no other player, will become able to influence the trajectory of the ball soon. Also this state does not depend on player-ball distance. Only one player can be in this state simultaneously. These three states, together with four combinations are shown in Table 23. State N and C can directly be used as proper definitions for individual ball possession. However, state F is not enough to describe a full definition. Considering all possible combinations, it has been decided to consider N&C, C&F, (N&C)|F and N|C|F as proper definitions as well. For a definition to be proper it has to serve a particular use, e.g. that it depends on ball-player distance or ball control, or that it results in a maximum number of one player to be in ball possession at a time. For an extended overview of all possible definitions and and explanation where they are proper definitions, see Table 30 (Appendix III).

Condition	**Definition: A player is in ball possession when he/she...**	**>1 players**
N	...has a close distance to the ball.	Yes
C	...has control over the ball.	No
F	...will gain control over the ball undoubtedly and very soon.	No
N&C	...has a close distance to the ball and has control over it.	No
C|F	...has control over the ball, or will gain control very soon.	Yes
(N&C)|F	...has a close distance to the ball and has control over it, or will gain control very soon.	No
N|C|F	...can (very soon) influence the ball trajectory or has already done so.	Yes

Table 23: Definitions of individual ball possession, using three states 'near distance' (N), 'control over the ball' (C) and 'future possession' (F), and two logical operators AND (&) and OR (|). The most right column shows if it is possible that multiple players are in ball possession simultaneously under a definition. State F is considered not a proper definition of individual ball possession. The other six are.

2. Experiment setup

During the experiment, three different models will be evaluated on their performance in ball-possession detection from individual player samples. An overview of these models is given in Table 24. The first model is inspired by earlier work of Link and Hoernig [65] and uses a threshold for the distance between the player and the ball to detect individual ball possession. Input of the model is player and ball locations. During training, the model finds an optimal distance threshold to classify the players correctly, using their distances to the ball. The second and third model depend on visual data around the players and do not depend on ball locations. Both models classify players as in ball possession or not, from RGB-pixels in the bounding boxes around the players. The third model extends the approach of the second model with multitask learning. In this case, the model does not only output a classification of the bounding box on individual ball possession. It also outputs a prediction on presence of the ball in the image and pixel-coordinates of the location of the ball. So, the second model is performing classification only, where the third model performs detection and localisation as well. We know that the location of the ball is important for the detection of ball possession. Therefore, such a multi-task learning approach functions as an inductive bias from domain knowledge that is expected to increase model performance.

To get an insight on the effect of using automatic ACF detections instead of ground-truth player detections, all models are evaluated on both situations. Additionally, to increase the scientific usefulness of this experiment, the first model is evaluated on automatic and ground-truth ball detections. Other methods for individual ball possession detection use a player-ball distance, similar to model 1, and thus depends on ball detections [52][65]. Our hypothesis is that inaccurate player-ball distances, due to incorrect ball detections, quickly drops the accuracy of the model.

Model input	**Player locations**		**Ball location**		**RGB-pixels**
Detection type	**Manual**	**Automatic**	**Manual**	**Automatic**	
Model 1.A (player-ball distance)	X		X		
Model 1.B (player-ball distance)	X			X	
Model 1.C (player-ball distance)		X	X		
Model 1.D (player-ball distance)		X		X	
Model 2.A (bounding box)	X				X
Model 2.B (bounding box)		X			X
Model 3.A (multitask learning)	X		(X)		X
Model 3.B (multitask learning)		X	(X)		X

Table 24: Three models are examined for individual ball possession in a binary classification setting: based on player-ball distance (model 1), based on bounding boxes around players (model 2) and based on bounding boxes around players and using multitask learning (model 3). Each model is evaluated on two or four test sets, that include manual (ground-truth) or automatic detections of players and the ball. Note that model 2 depends on player detections only. Model 3 uses ground-truth ball locations, but exclusively during training. Hence, the (X) notation.

2.1 Dataset

Before execution of the experiments, videos of three soccer matches were annotated with player positions, ball locations and ball possession labels (N, C and F) for each annotated player. All videos have been captured from the same, static perspective at the long side of the soccer field such that the full field was visible at all times. The set of two cameras is used because manual ball detections were available for it, created by Van Dijk [24]. The ball locations were extended with annotations of player bounding boxes and ball possession labels for three soccer matches. In total, 653 frames were annotated with 1832 players. To reduce annotation time, while obtaining samples from all three matches, every fortieth frame from the original camera stream was annotated, resulting in 653 frames. A period of 1.6 seconds is between each pair of consecutive annotated frames. The set of players that was annotated per frame is not randomly picked, but is always a group closest to the ball. This group was chosen since the amount of players that could be annotated was limited and since it is considered most interesting to assess how the models perform on the most difficult samples. Per frame, two or more players were annotated. If a player was in possession of the ball, for any of the three labels, it was always included in this group. Since a camera model was calculated beforehand, all pixel coordinates could be ray casted to world coordinates. For each definition of individual ball possession, a balanced set is created such that the amount of players in ball possession is equal to the amount of players that are not. This resulted in a different set size for each definition: 964 (N), 670 (N&C), 806 (C), 980 (C|F), 844 ((N&C)|F) and 1060 (N|C|F) samples.

The automatic detections of the players are obtained using an ACF detector on the video footage. The method outputs player locations as well as bounding boxes around them. An ACF detector is used since player detections by this method were already available for the dataset. Additionally, the method is reasonable to use in practice due to its quick detections, such that video can be processed in real-time, and the ability to detect most players on the soccer field. Since the ACF detector makes errors, the automatic bounding boxes do not always overlap with the ground-truth bounding boxes around the players of interest. In Figure 20 it can be seen that for a minimal IoU of 0.5, the ratio of manual annotations that overlap with an automatic detection drops to 58.36%. For players in ball possession, following any definition, this ratio is 49.13%. The labels

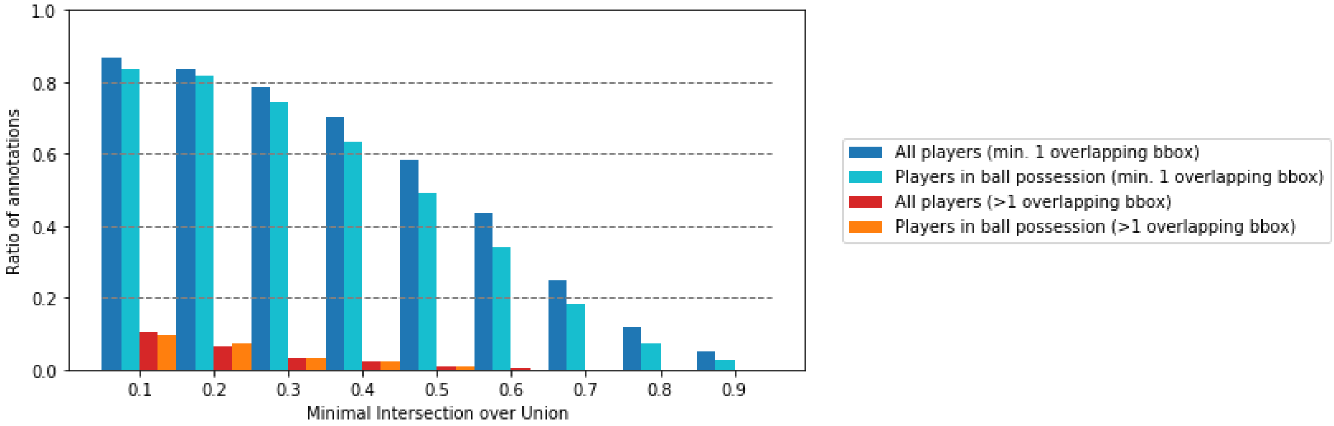
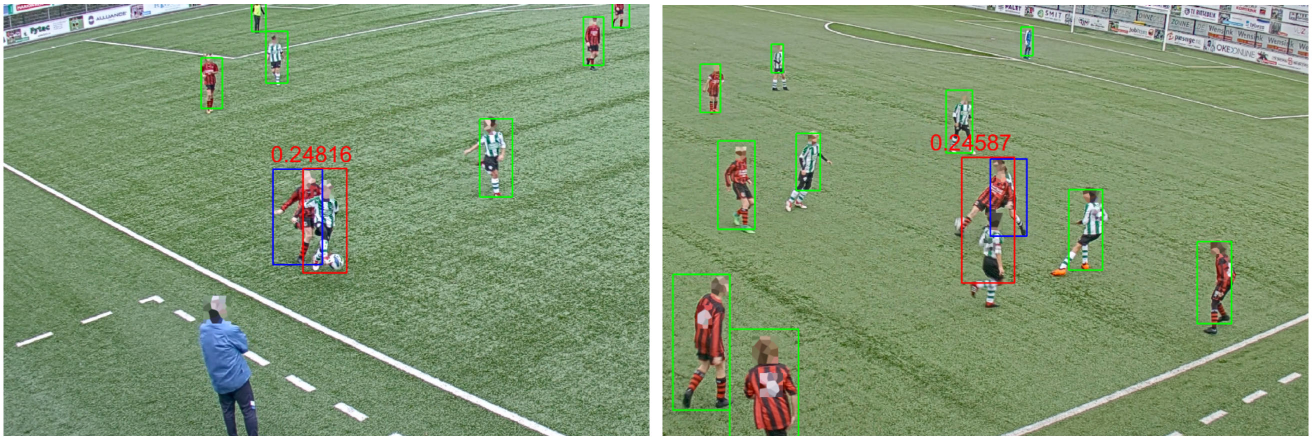


Figure 20: Performance of the ACF detector on soccer players, at different restrictions on the minimal relative overlap (IoU) between the manually annotated and the automatically detected bounding boxes. Performances are shown for two subsets: all players and players in ball possession. For both subsets, the ratio of manual annotations that overlap with multiple automatic detections is shown as well.



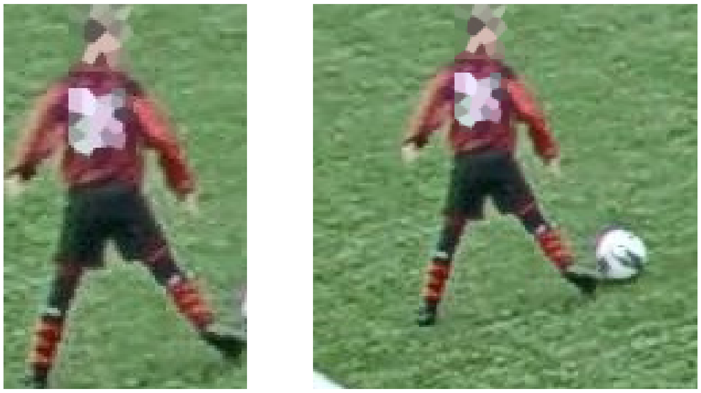
(a) Another player is detected

(b) Multiple players are detected

Figure 21: Errors when matching ground-truth bounding boxes (blue) with ACF detected players (red).

of manually annotated players that have no overlaps cannot be assigned to an automatic detection. As a result, the system will never be able to classify these players correctly and considered as not detected at all. Additionally, a subset of the player annotations does overlap with incorrect ACF detections. As can be seen in Figure 21, these overlapping detections can be detections of other players or an ambiguous detection including multiple players. In order to remove most incorrect detections, only ACF detections with an IoU of 0.25 or higher are considered correct detections. This number is chosen, since for lower values of IoU the ratio of incorrect detections is larger than that of correct detections. For each player, its labels are assigned to an ACF detection if the overlap ratio between the manual and automatic bounding boxes is larger than 0.25. If there are multiple overlaps, the labels are only assigned to the detection with the largest IoU. Eventually, this resulted in a ratio of 77.07% of the data being matched to automatic detections. The other 22.93% is excluded from the dataset for the models using the automatic player detections, since their labels cannot reliably be matched with the detections. Note that this causes the performance scores to reflect only the situation in which a player is indeed detected by the ACF detector.

Besides player detections, automatic ball detections are required as well. These are obtained using Ball-I3D [24]. This model creates 2D-histograms from the field coordinates of all players and predicts the ball location accordingly using an I3D-CNN



(a) Original bounding box

(b) Enlarged bounding box

Figure 22: Bounding boxes around players are enlarged as to include direct surroundings as well.

architecture. In order to create the 2D-histograms, the model uses the same ACF detections of players as before to obtain their locations. Ball-I3D is chosen to provide the ball-location estimations since the model was designed using the same dataset. Since there is a lack of benchmarks for ball detection in soccer videos, it is difficult to compare its performance to other methods.

Both manually and automatically annotated bounding boxes are used by model 2 and 3. As to include the direct surrounding of each individual player, the bounding boxes around the players are widened (see Figure 22). Twenty percent of the original height is added to the bottom of the bounding box, which is generally the area around the feet. When a player is in ball possession, it is likely that a ball is visible in this area. The bounding boxes are widened as well, such that the images become square and the player is in the centre of the image. The images are squared since this is desired by the architecture of model 2 and 3.

For model 3, two additional labels are generated per player: presence of the ball in the image and pixel-location of the ball. In order to generate these, manual (ground-truth) annotations of the ball are used. Whenever the ball pixel-coordinates are within the boundaries of the widened player bounding box, the sample is positively labelled. Note that this does not mean that the ball is visible; it can be occluded. Also, the pixel-coordinates of the ball are saved, relative to the bounding box coordinate system. If the ball is present in the bounding box, its relative pixel-coordinates are between zero and one for both axes. If the ball is outside the bounding box, these coordinates are negative or higher than one for at least one axis.

2.2 Performance metrics

The main performance metrics are accuracy, precision and recall. Players that are in ball possession are included in the positive class, where all other players are in the negative class. For model 3, errors in the two other output variables are considered as well: accuracy is used as the metric for ball detection where mean square error is the metric for localisation of the ball in the bounding boxes.

3. Model 1: player-ball distance

Experiments with the first model have two aims. First, the model is used to assess the importance of the distance between a player and the ball when detecting individual ball possession. The second aim is to assess the influence of automatic player and ball detections on the performance of a model using a distance threshold for classification. This is important, since in most situations, the locations of the ball and players are not known beforehand and manual annotation per frame is too labour intensive.

Similar to the method of Link and Hoernig [65], an optimal threshold T_b is found to define whether a player is in ball possession. When a player is within this range to the ball, the player is classified as in possession by the model. Players further away from the ball are negatively classified. Euclidean distance (in meters) is used to obtain the distance per player-ball pair. For each definition of individual ball possession, an optimal threshold is found using a grid search. Thresholds between zero and ten meters, in steps of one centimetres, are evaluated. The optimal threshold is the distance measure that results in the highest accuracy when classifying all players in a training set. This set includes 80% of the balanced data set including ground-truth player and ball locations. The optimal thresholds are evaluated using four test sets, each with a different combination of ground-truth (manual) and automatic detections of the players and the ball. Model 1.A is tested on the remaining 20% of the set, since the first 80% is used for finding the optimal thresholds. Model 1.B-D are evaluated using 100% of the samples.

From the test sets, it is expected that model 1.A provides the best performance in classification. Since the mean Euclidean error of the Ball-I3D estimations are 10.34 meters [24], it is expected that this error is too large to accurately use a player-ball distance threshold for classification using this data. A smaller error is present in the ACF-detection locations, which differ on average 1.62 meters from their ground-truth location. Although this is likely to cause a drop in performance, it is expected to be less problematic than the ball estimation errors. Model 1.D is expected to perform worst, since the input is the furthest away from reality compared to the other evaluation sets.

3.1 Results

The optimal values found for the distance threshold are given in Table 25, per definition. It can be seen that when the location data is close to reality (model 1.A), performance scores are high, especially for definitions that depend on player-ball distance. For example, model 1.A classifies 95.83% and 90.30% correctly using definitions N and N&C respectively. The definition that does not depend directly on player-ball distance, C|F, result in the lowest classification accuracy of 78.57%. Optimal values for T_b are found ranging from 1.96 to 2.13 meters. This makes sense since that is approximately the ratio in which a player is directly able to influence the ball's trajectory.

(a) Model 1.A: manual player and ball detections

Condition	T_b **(m)**	**Acc.**	**Prec.**	**Recall**
N	2.13	95.83	94.90	96.88
C	2.04	87.65	89.61	85.19
N&C	1.99	90.30	86.49	95.52
C|F	2.04	78.57	83.33	71.43
(N&C)|F	1.96	86.90	93.06	79.76
N|C|F	2.15	87.80	95.28	79.53

(b) Model 1.B: manual player and autom. ball detections

Condition	T_b **(m)**	**Acc.**	**Prec.**	**Recall**
N	2.13	51.04	60.00	6.22
C	2.04	50.37	54.29	4.71
N&C	1.99	51.34	66.67	5.37
C|F	2.04	50.92	63.64	4.29
(N&C)|F	1.96	50.83	60.61	4.74
N|C|F	2.15	51.65	72.34	5.35

(c) Model 1.C: autom. player and manual ball detections

Condition	T_b **(m)**	**Acc.**	**Prec.**	**Recall**
N	2.13	84.34	88.67	78.74
C	2.04	76.57	83.40	66.34
N&C	1.99	84.76	88.69	79.67
C|F	2.04	71.83	80.92	57.14
(N&C)|F	1.96	76.43	84.02	65.29
N|C|F	2.15	77.38	90.34	61.31

(d) Model 1.D: autom. player and ball detections

Condition	T_b **(m)**	**Acc.**	**Prec.**	**Recall**
N	2.13	51.44	69.23	5.17
C	2.04	49.34	41.67	3.30
N&C	1.99	50.61	58.82	4.07
C|F	2.04	50.13	52.17	3.23
(N&C)|F	1.96	50.32	55.00	3.50
N|C|F	2.15	50.53	56.41	4.65

Table 25: The optimal values for threshold T_b, for each definition and test set A-D, together with their performances in terms of accuracy (%), precision (%) and recall (%).

It can be seen that errors in the automatic detections have a large impact on the performance of the model. Even when all players are correctly located using the ground-truth, and the ball location is estimated using Ball-I3D (model 1.B), the classification accuracies drop to 50.37-51.65% for any definition. Note that a random guess in this setting would saturate to an accuracy of 50.00% for large numbers of predictions. Especially the recall rates are very low for this model (3.23-5.17%), meaning that most players in ball possession are classified as players that are not. Higher values for T_b could increase the recall scores. However, this will only slightly increase the accuracy of the model (up to 54.98% for definition N and a threshold of 6.40). Errors in player detections have less impact to the performance of the model. Model 1.C obtains classification accuracies of 71.83-84.76%. Note that the 22.93% of the ground-truth bounding boxes that did not have an overlap ratio with any ACF detection larger than 0.25 are not classified by the model and therefore not reflected upon by the performance scores. When one wants to incorporate the not detected players and treat all of them as misclassifications, the accuracy scores of model 1.B and 1.D can be multiplied with 0.7707. Model 1.D obtains a similar performance to model 1.B with classification accuracies of 49.34-51.44%. The experiment show that a player-ball distance threshold does perform well when ground-truth player and ball locations are known to the system. However, in most situations this will not be the case since manual annotation is expensive. The error in ball-location estimations of the Ball-I3D are too large to retain good performance. The impact of the ACF player detector is smaller, such that the model remains usable. However, these results do not reflect players that are not detected, which happens for 22.93% of the players that are near the ball.

4. Model 2: bounding boxes

The second model uses visual inspection of player surroundings, instead of location data, to classify the players. Aim of the experiment is to show the advantage of an RGB-based approach over the use of a distance-threshold, when player or ball locations could not be obtained accurately.

Similar to model 1.B and 1.D, the location of the ball is not known to the system. In fact, this method does not depend on its location. To assess the influence of an ACF player detector over ground-truth player locations, two sample sets are evaluated: one using manual detections (model 2.A) and one with automatic detections (model 2.B) as input. Both versions are trained using manual detections. The bounding boxes of the detected players are enlarged such that they include direct surroundings, and are cut out of the full view images. Afterwards, the enlarged bounding boxes are classified by a CNN. A ResNet-10 architecture is chosen to classify the images, since the model has a small amount of trainable parameters (~5 million). This reduces the rate of overfitting when using a small dataset, which is the case for the 964 to 1060 available samples. The ResNet architecture [37] is a standard CNN architecture for image classification.

Besides the decision for a small model, four other measures are used to be able to train on the small dataset. First a dropout-layer (fraction of 0.2) is added right before the output-layer, to prevent the model from overfitting. Second, the size of each dataset is doubled by horizontally flipping all images. These images are added after splitting the datasets in training and test samples such that an original and flipped image pair is always in the same set. Third, it is observed that several runs with the same data and hyper-parameters give variations in model performance. To get more stable results, the experiment is executed using five-fold cross validation. Last, the model is pre-trained on ImageNet-1K [21] such that relevant feature maps that are general to many image datasets are already learned. Especially the first layers include such feature maps, including Gabor filters and color blobs [102]. Since the pre-trained model is trained to classify between 1000 classes, the model has 1000 output nodes. These are replaced with two output nodes with *softmax* activation and *binary cross entropy* loss (see Equation 9). The paragraph below demonstrates the benefit of pre-training in more detail. Furthermore, the models are trained for 80 epochs with a batch size of 40 using stochastic gradient descent, with a learning rate of $1e^{-4}$, and 0.7 momentum.

$$BCE(Y,P) = -\sum_{n=1}^{N}(Y^{(n)}\log(P^{(n)}) + (1 - Y^{(n)})\log(1 - P^{(n)})) \tag{9}$$

with $BCE(Y,P)$ the binary cross entropy for ground-truth set Y and model predictions P. The loss is calculated per batch, including N samples. $Y^{(n)}$ is the binary label for sample n, where $P^{(n)}$ is the corresponding probability output prediction from the model.

In Figure 23 can be seen what the effect is of pre-training and, additionally, 'freezing' different amounts of pre-trained layers when fine-tuning the model on player bounding boxes. It is clear that initializing the weights with a pre-trained net prevents the model from both slow learning and a lack to decrease the training loss drastically. When freezing four or five layers the model is underfitting, resulting in slowly decreasing loss functions on the train and test set. The bottom-left subplot of Figure 23 is zoomed in on the three best performing models, with zero to two frozen layers. It can be seen that freezing the first two layers prevents the model from overfitting just enough to obtain proper results on the test set. That this model performs best can also be seen in Table 26, with an accuracy of 73.23% after epoch 80. Therefore, it has been decided to execute further experiments with this model. Note that the loss on the test set is lower than the loss on the training set during the first epochs, which is likely to be caused by the dropout-layer: this layer is disabled during the test phase. This experiment is executed using the previously mentioned measures and hyper-parameters, uses definition N for individual ball possession and receives the ground-truth bounding boxes as input.

Number of frozen layers	No PT	0	1	2	3	4	5
Accuracy (%)	57.56	70.10	70.78	73.23	69.21	68.33	62.03

Table 26: Acc. scores of the models evaluated in Figure 23 after epoch 80. The most left model is not pre-trained at all (No PT).

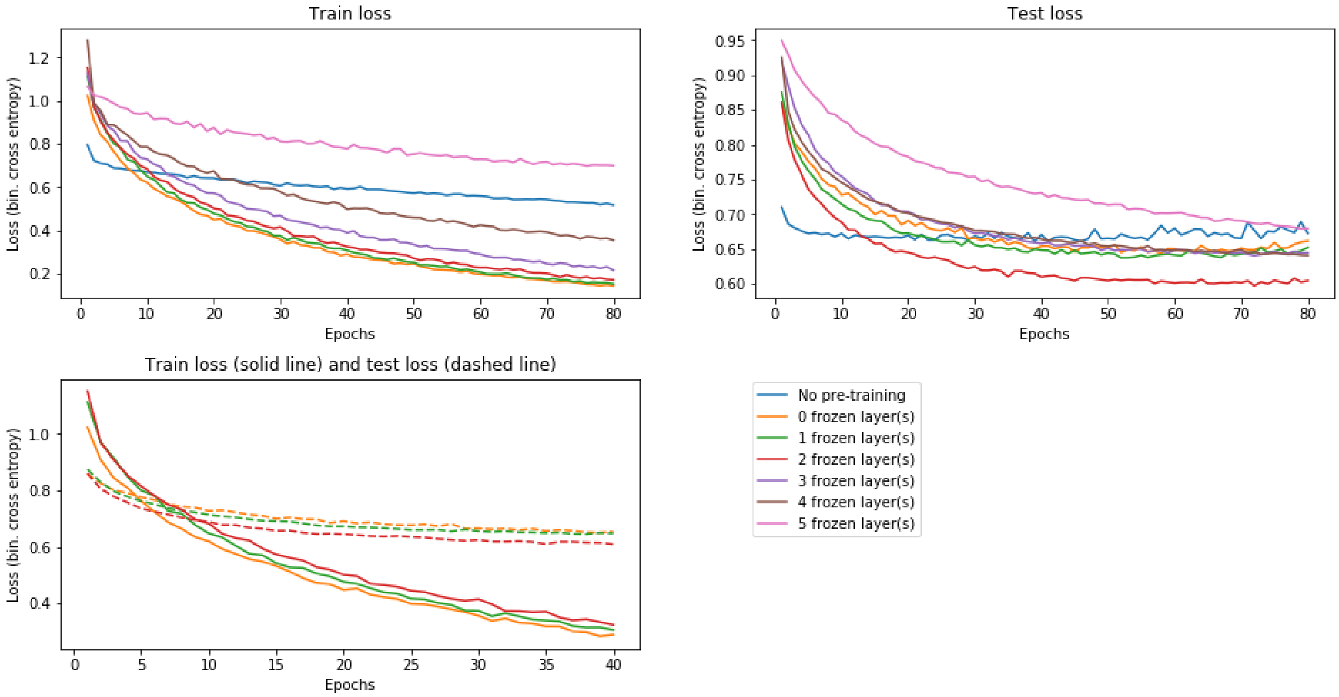


Figure 23: Loss functions on the train and test set for model 2.A, not pre-trained (blue) and pre-trained with fine-tuning using different numbers of 'frozen' layers.

As can be seen in Figure 23, it is likely that a CNN is trainable to classify players correctly as in ball possession or not. It is expected that visual cues, such as presence of the ball in the image, give the CNN relevant information such that it can classify a large portion of the players correctly. However, this would mean that considering definitions of individual ball possession in which near presence of the ball is not necessary, the model is likely to perform worse. Additionally, it is expected that the use of automatic ACF detections for the bounding boxes will lower the performance of the model.

(a) Model 2.A: manual player detections

Condition	Acc.	Prec.	Recall
N	73.23	73.35	73.02
C	65.43	66.47	62.83
N&C	67.43	66.97	68.87
C|F	66.00	66.24	65.78
(N&C)|F	66.55	67.02	65.48
N|C|F	69.27	69.48	68.97

(b) Model 2.B: autom. player detections

Condition	Acc.	Prec.	Recall
N	69.40	71.07	68.62
C	64.49	65.30	66.69
N&C	65.76	69.48	61.49
C|F	65.22	66.92	63.36
(N&C)|F	61.94	64.03	61.84
N|C|F	65.10	66.19	64.52

Table 27: Performance scores for the RGB-based method that classifies individual players as in ball possession or not.

4.1 Results

Performance scores of the proposed RGB-based method can be found in Table 27. As expected, the model is performing best for the near (N) definition, with an accuracy of 73.23%. Performances for the other definitions vary between 66.00% and 67.43%. Although the model does not perform as well as model 1.A and 1.C, which scored 95.83% and 84.34% respectively, model 2 does not depend on ball detections at all and still performs better than random guess. This indicates the presence of visual cues for individual ball possession in the direct environment of the players. Model 2.B, which receives ACF player detections as input during evaluation, performs less accurate with an average decrease of 2.67% in accuracy. These results can be visually inspected in Figure 24. It can be seen that for all definitions of model 2 the accuracy function on the test set saturates around epoch 80. Comparing the model to the first model, it obtains accuracies higher than model 1.B and 1.D, but lower than 1.C.

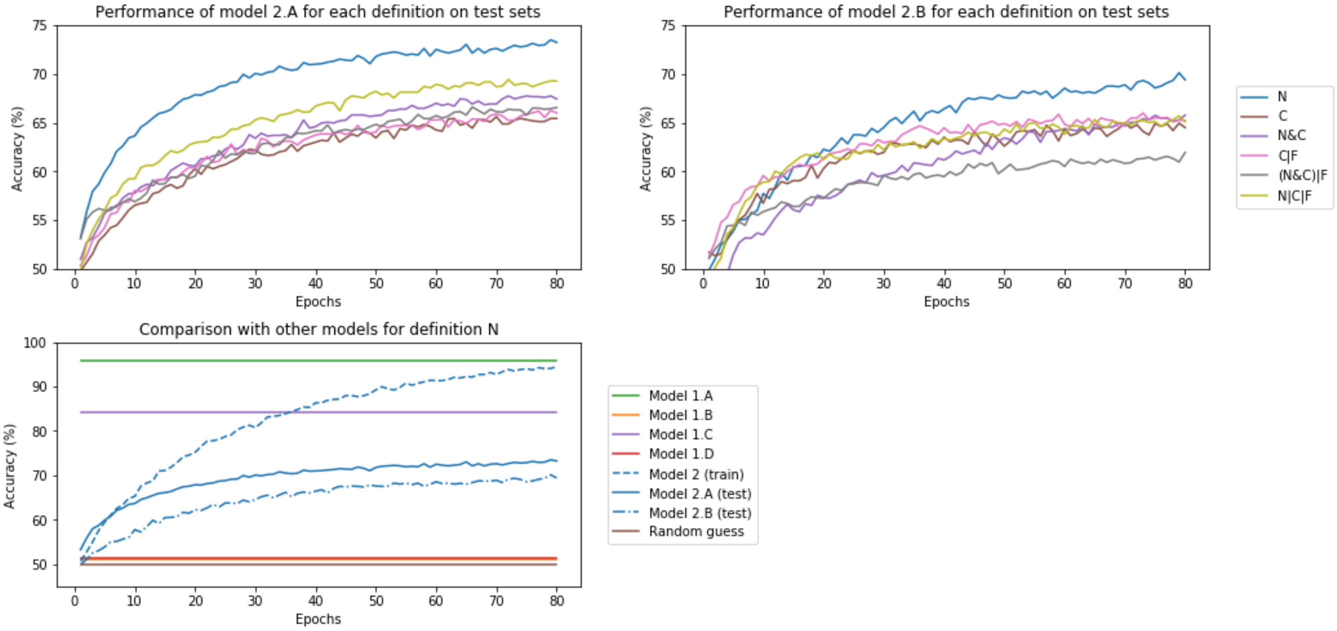


Figure 24: Accuracies of the model using manually annotated player bounding boxes (model 2.A) and ACF player detections (model 2.B) (up). Comparison between evaluation sets A and B, and the first model, using definition N (bottom).

5. Model 3: multitask learning

Multitask learning is widely used to boost performance of classifiers. By explicitly telling the model to predict aspects related to the original task, all aspects are predicted using the same representation. The model is then forced to use the implicit bias that is necessary to be used in predicting the labels of the new aspects [11]. In our case, it is desired to force an implicit bias on the presence of the ball near a player with individual ball possession. Therefore, an evolved version of the CNN, used for model 2, is created. In this version, the model does not only classify each player, but also predicts whether the ball is present in the image and predicts the ball location in the image.

In order to enable the multitask model to predict the additional aspects, four output nodes are added to the output-layer. Two of these are for classification with one node predicting the class ‘ball present in image’ and the other ‘no ball present in image’. The other two nodes output the predicted horizontal and vertical location of the ball in the image. Additionally, a combined loss function is created for optimisation of the model (see Equation 10). All terms in this loss function are weighted equally. It is expected that the model is able to detect the ball accurately within the enlarged player bounding boxes and that this increases the performance of the RGB-based method. Again, two evaluation sets are used: model 3.A is trained and evaluated using manually annotations of players, while model 3.B uses ACF player detections.

$$L = BCE_p(Y_p, P_p) + BCE_v(Y_v, P_v) + \sum_{n=1}^{N} \frac{Y_v^{(n)}}{N} \left\| Y_b^{(n)} - \hat{Y}_b^{(n)} \right\|_2^2 \tag{10}$$

with L as the total loss function, used by the multitask model. The first two terms define binary cross entropy loss for predicting individual ball possession (BCE_p) and presence of the ball in the images (BCE_v). The third term is the mean squared error (MSE) over a batch with $Y_b^{(n)}$ as the ground truth location of the ball in image n and $\hat{Y}_b^{(n)}$ the corresponding prediction by the model. Adapted to the regular MSE function is that if the ball is not present in an image, the squared error for that sample is set to zero.

5.1 Results

Performance scores of model 3 can be seen in Table 28. The overall accuracy performance has increased, with an average of 2.00%. For the models trained on ACF detections, the increase is larger, with an average of 3.37%. Overall, the model is able to increase the accuracy score for ten out of twelve sets, with a largest increase of 6.14%. For this model, the influence of using automatic player detections instead of ground-truth bounding boxes is only small (average of 1.30%). Besides, the model is able to detect whether the ball is present in an image with an average accuracy of 75.67%. In Figure 25, performance of model 3 can be inspected visually. Again, it is clear that the RGB-based method performs best for definition N. Model 3.A obtains the highest accuracy scores of all RBG-based models and is getting closer to the performance of model 1.C.

(a) Model 3.A: manual player detections

	Possession			Ball
Condition	**Acc.**	**Prec.**	**Recall**	**Acc.**
N	75.94	77.15	74.06	78.59
N (FR1)	77.60	77.78	77.40	80.21
C	67.67	67.87	67.42	75.63
N&C	71.85	73.42	68.84	73.12
C|F	68.18	68.99	66.19	75.15
(N&C)|F	65.65	66.95	31.90	75.83
N|C|F	70.65	71.49	68.96	77.81

(b) Model 3.B: autom. player detections

	Possession			Ball
Condition	**Acc.**	**Prec.**	**Recall**	**Acc.**
N	74.38	77.97	70.34	78.26
N (FR1)	73.36	74.75	73.26	73.06
C	70.63	72.92	69.46	72.54
N&C	68.62	71.69	66.67	73.54
C|F	64.46	67.44	60.07	75.24
(N&C)|F	65.45	68.69	60.65	70.58
N|C|F	68.61	71.26	64.26	78.35

Table 28: Accuracy, precision and recall scores for the RGB-based method for all definitions, using multitask learning. Additionally, the accuracy scores for predicting presence of the ball in the images are given in the most left columns.

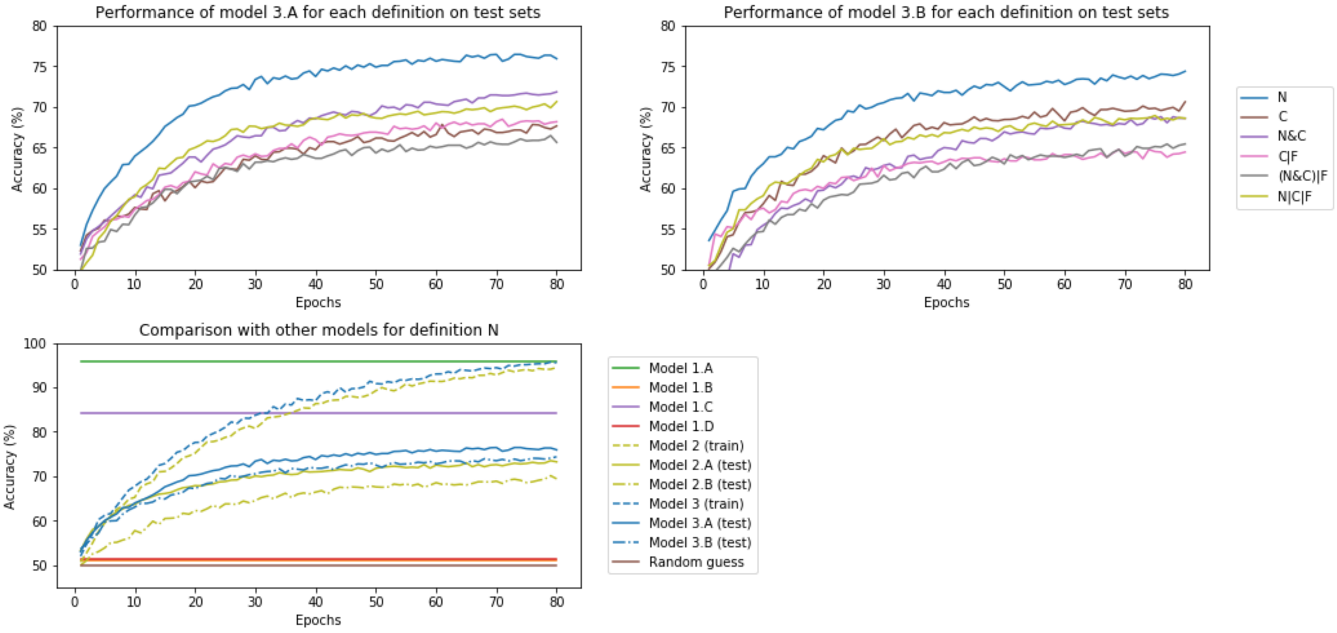


Figure 25: Accuracies of the model using manually annotated player bounding boxes (model 3.A) and ACF player detections (model 3.B) on the test set of each definition (up). Comparison between both models and all other models, using definition N (bottom).

6. Results overview and conclusion

Table 29 includes an overview of the accuracies of all three models when using different combinations of manual and automatic detections as input. For clarity, only the results when using the N definition for individual ball possession are given. The ratio between performance scores of the models is similar for the other definitions, although overall performance is generally lower. In cases where ground-truth ball locations are available, the model that puts a threshold on the player-ball distance (model 1) is dominant in accuracy. This is also the case when players are automatically detected, although performance decreases. However, in cases where ground-truth ball locations are not available, and estimated using Ball-I3D, the RGB-based methods (model 2 and 3) dominate. Since these models do not depend on ball locations, their performance does not change when ground-truth ball locations are available or not. Note that model 3 still requires ground-truth ball locations for training.

	Ball location	
Player locations	Manual	Automatic
Manual	Model 1: **95.83**	Model 1: 51.04
	Model 2: 73.23	Model 2: 73.23
	Model 3: 75.94	Model 3: **75.94**
Automatic	Model 1: **84.34**	Model 1: 51.44
	Model 2: 69.40	Model 2: 69.40
	Model 3: 74.38	Model 3: **74.38**

Table 29: Comparison between the accuracies (%) of all three models for the definition N, using the four combinations of input. Since model 2 and 3 do not depend on a ball location, their performances are the same for manual and automatic ball detections.

It can be concluded that it is reasonable to use the video recordings to train our novel method on for semantic labelling. By visual inspection of the player and its direct surroundings a deep network is able to extract semantic information, like ball possession. Although the baseline model provides high accuracies for ground truth player and ball locations, it is not robust against errors in the automatic detection of these. In contrast, the CNN based models are able to provide reasonable results for automatic detections. This means that the ACF detections can be used in further experiments such that players do not have to be manually annotated. However, it should be known that the detector is not perfect and is missing detections for 22.92% of the players that are near the ball. Last, an inductive bias based on domain-knowledge such as multitask-learning (model 3), is able to boost the performance of the classifier. This advocates for the use of more ways to force an inductive bias on the action recognition algorithm to improve its prediction accuracy.

Appendix III. List of possible definitions for individual ball possession

Condition	Valid	Explanation
N	**Yes**	If only player-ball distance is considered relevant.
C	**Yes**	If only ball control is considered relevant.
F	No	Future possession alone is not enough to cover a full definition.
N&C	**Yes**	If player-ball distance is considered relevant, but it is desired that only one player is in ball possession at a time.
N&F	No	Describes only the short moment where the ball is close to the player, but the player has not touched the ball yet. Not enough to cover a full definition.
C&F	No	Results in an empty set of positive samples.
N|C	No	Although such a definition would be possible, it does not follow a particular use. Definition N&C or N|C|F would better suit.
N|F	No	The inclusion of very soon ball control (F) without considering actual control (C) in the definition feels odd. Definition (N&C)|F or N|C|F would better suit.
C|F	**Yes**	If ball control is relevant, but possession starts already when it is clear that the player will gain control soon (e.g. a goal kick).
(N&C)|F	**Yes**	Similar to C|F, but ensures that at most one player is in possession at a time.
(N&F)|C	No	That future possession only counts as possession when the ball is close to the player feels odd. Definition C|F would better suit.
(C&F)|N	No	Is equivalent to N (C&F have no positive samples).
(N|C)&F	No	Is equivalent to N&F (C&F have no positive samples).
(N|F)&C	No	Is equivalent to N&C (C&F have no positive samples).
(C|F)&N	No	Extends N&C with N&F. That future possession only counts as possession when the ball is close to the player feels odd. Definition (N&C)|F would better suit.
N&C&F	No	Results in an empty set of positive samples.
N|C|F	**Yes**	Results in the most 'loose' definition, meant for one that is interested in players with any (potential) influence on the ball, now or very soon.

Table 30: List of all possible combinations of the three labels for individual ball possession: near distance to the ball (N), in ball control (C) and undoubted control in the near future (F). Variations are also considering the logical operators AND (&) and OR (|). The validity as a definition for individual ball possession is assessed for each combination.

Appendix IV. Overview of match event logs provided by CIP Wyscout

Metadata	Events	Subevents	Tags
Match ID	***Duel***	Air	Goal
Match period		Ground attacking	Own goal
Time past (sec.)		Ground defending	Assist
Team name		Ground loose ball	Key pass
Player name	***Foul***	Foul	Counter attack
Field position		Hand	Foot right/left
		Late card	Head/body
		Out of game	Direct/indirect
		Protest	Dangerous ball lost
		Simulation	Blocked
		Time lost	High/low
		Violent	Interception
	Free kick	Corner	Clearance
		Free kick	Opportunity
		Cross	Feint
		Shot	Missed ball
		Goal kick	Free space right/left
		Penalty	Take on right/left
		Throw in	Sliding tackle
	Goalkeeper leaving line	-	Anticipated/Anticipation
	Interruption	Ball out of the field	Red/(second) yellow card
		Whistle	Goal position (3x3 grid)
	Offside	-	Out position (8 options)
	Others on the ball	Acceleration	Post position (8 options)
		Clearance	Through
		Touch	Fairplay
	Pass	Cross	Lost/neutral/won
		Hand	Accurate/Inaccurate
		Head	
		High	
		Launch	
		Simple	
		Smart	
	Save attempt	Reflexes	
		Save attempt	
	Shot	-	

Table 31: List of annotated match event logs at Wyscout, corresponding to the dataset by Pappalardo *et al.* [71]. See also: https://apidocs.wyscout.com/matches-wyid-events.

Appendix V. List of definitions from labels in the Soccer Action and Activity Dataset

Body part	Sub-category	Leaves	Description
Head	-	Heading	Touching the ball with the head
		Jumping	Reaching with the head towards the ball without touching it
Foot	Defence	Tackle	Intercepting the ball from an opponent without gaining control
		Foul	Tackling without touching the ball, resulting in the opponent falling
		Interception	Gain ball control by intercepting the ball from an opponent
	Attack	Dribble	Control the ball freely for at least 3 meters
	(control ball)	Take on	Attempt to dribble past an opponent
	Attack	Pass	Pass the ball over the ground
	(play ball)	Cross	Pass the ball through the air in the penalty area of the opponent
		Shot	Shoot the ball towards the goal of the opponent (the goal may be missed)
		Clearance	Get the ball out of the defensive third in a hasty manner
		Bad touch	Touch the ball, but immediately loose control over it
None	In duel*	Loose	Involved in a duel, while no team is in ball possession
		Attack	Involved in a duel, while you are in ball possession
		Defence	Involved in a duel, while the opponent is in ball possession
	Non-duel	Passive	No interaction with the ball, not involved in a duel
Other	Field player	Throw-in	Throw-in the ball with the hands
	Goalkeeper	Save	Attempt to save a shot on goal
		Claim	Catch a cross
		Punch	Punch the ball to get it out of the defensive third
		Pick-up	Take the ball in the hands

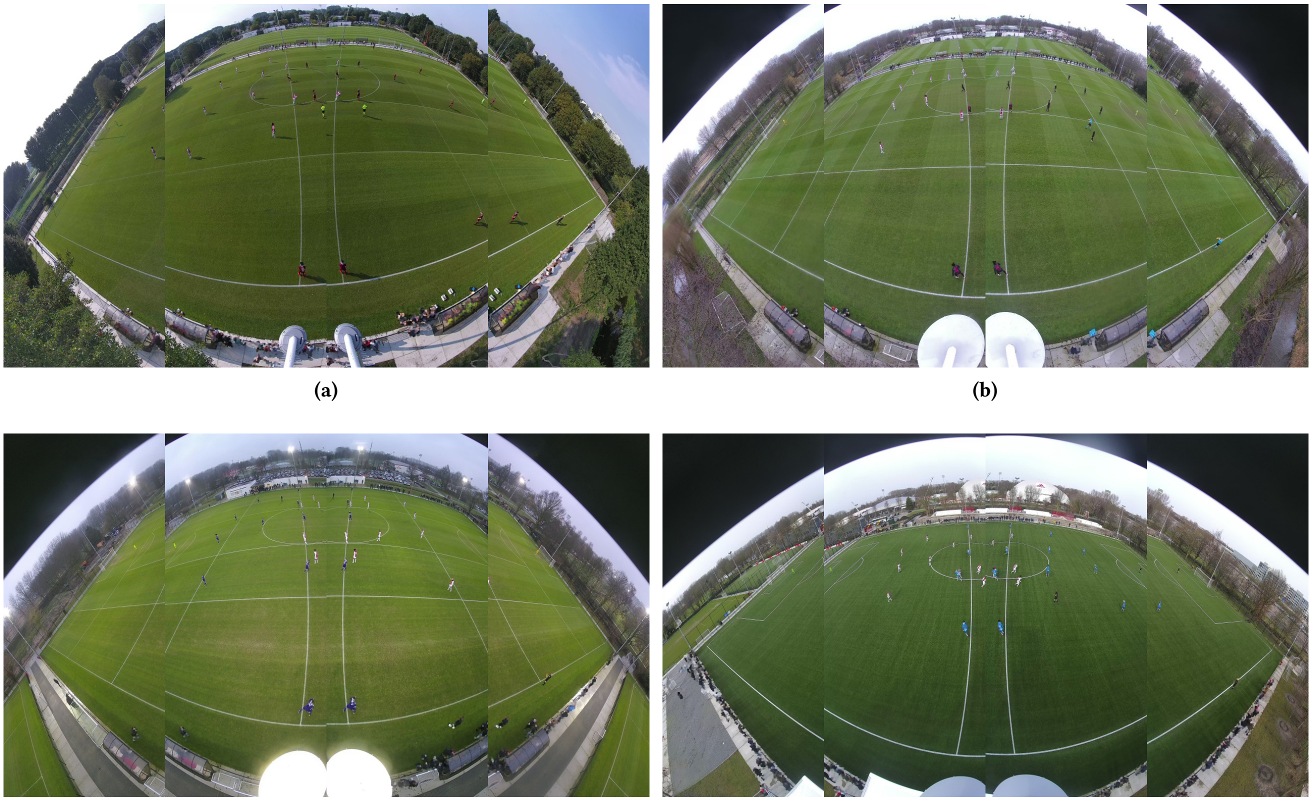
Table 32: The taxonomy of the action labels, based on the SPADL [20].

Category	Leaves	Description
Duel*	Air	At least two players try to influence the trajectory of the ball with their head
	Ground (Att/Def)	At least two players are in a duel, where one player is in ball possession
	Ground (Loose)	At least two players are in a duel, where no player is in ball possession
Play freely	Touch ball	A player freely controls the ball
Proceed play	Free kick (short)	A player takes a free kick by passing the ball over the ground
	Free kick (cross)	A player takes a free kick by passing the ball through the air
	Free kick (shot)	A player takes a free kick by shooting the ball towards the goal area of the opponent
	Kick off	A player takes a kick off
	Goal kick	A player takes a goal kick
	Penalty	A player takes a penalty
	Corner (short)	A player takes a corner by not passing the ball towards the penalty area
	Corner (cross)	A player takes a corner by passing the ball towards the penalty area
	Throw-in	A player performing a throw-in
Interruption	Whistle	Inactive play, not because the ball is out of the field boundaries
	Ball out of bounds	Inactive play, because the ball is out of the field boundaries
	Goal	A goal is scored

Table 33: The taxonomy of the activity labels, based on the SPADL [20] and the match (sub)-events recorded by Wyscout.

* A duel is where at least two players from different teams are trying to influence the trajectory of the ball, or are trying to gain/keep possession over the ball, or where one player is trying to slow-down its opponent with physicial contact between the two.

Appendix VI. Example frame from each game



(a)

(b)

(c)

(d)

Figure 26: The kick-off from all games in the training set (a-c) and the test set (d).